# Software Engineering Project1a1

## Stakeholders:

1. **Restaurant Operator / Staff**
   – Accepts, prepares, and fulfills orders; manages menus and availability.
2. **Delivery Rider / Driver**
   – Picks up and delivers food, interacts with customers and restaurants.
3. **Support Agent / Customer Service Team**
   – Resolves complaints, disputes, refunds, escalations.
4. **System Administrators / Platform Owners**
   – Maintain system integrity, uptime, and compliance.
5. **Payment Provider / Gateway**
   – Handles payment authorization, refunds, fraud checks.
6. **Authentication / Identity Provider (Google/Apple, OTP service, etc.)**
   – Provides secure login and verification.
7. **Logistics / Dispatching Service**
   – Assigns drivers, optimizes routing.
8. **Recommendation & Analytics Engine**
   – Provides personalized restaurant and dish recommendations.

## Stakeholder Biases (conflicts in needs/priorities):

1. **Customer vs. Restaurant Bias**
   - Customers want *low prices, generous refunds, and fast delivery*.
   - Restaurants want *to minimize losses from cancellations, substitutions, and refund claims*.
2. **Customer vs. Rider Bias**
   - Customers want *on-time deliveries regardless of weather or traffic*.
   - Riders want *fair compensation, safety, and flexibility*, sometimes conflicting with customer expectations.
3. **Restaurant vs. Rider Bias**
   - Restaurants want *riders to wait until food is ready*.
   - Riders want *food ready on arrival*, to avoid idle time and low earnings.
4. **Marketing vs. Customers Bias**
   - Marketing pushes *promotions and upsells* to increase order size.
   - Customers may perceive *spamming or irrelevant offers*, lowering trust.
5. **Platform (Admin) vs. All Stakeholders Bias**
   - Platform enforces *policies for cancellations, refunds, and fees* to maintain profitability.
   - Customers, riders, and restaurants may feel these rules are *unfair or too rigid* when exceptions occur.

## Comment on prompt crafting:

**Prompt crafting** is the practice of designing clear and structured instructions for a language model so that its output is accurate, relevant, and useful. Like asking a person for help, the quality of the question strongly shapes the quality of the answer. Prompt crafting may involve giving context, examples, or constraints to guide the model. **Zero-shot prompting** means asking the model to do a task without examples—just direct instructions. For instance, “Translate the following sentence into French: The cat is on the table.” This approach is simple and often effective for straightforward tasks, but it can lead to vague or inconsistent responses when the task is complex or subjective. **Careful prompting** goes further by refining the instructions with context, roles, or constraints, sometimes even breaking the task into steps. For example: “You are an expert French translator. Translate the following sentence into formal French and provide one alternative phrasing.” This approach produces more reliable, nuanced, and controllable results but requires more effort to design. **In short, zero-shot prompting is quick and low-effort but less predictable, while careful prompting is structured and deliberate, yielding higher-quality responses for complex or nuanced tasks.**

**<u>Use Cases:</u>**

### 1. User Registration & Login, Account & Address Management

**Primary Actor:** Customer
**Supporting Actors:** Authentication Service, Social Login Provider, Notification Service
**Preconditions:** Customer has the app or website open with internet access.
**Postconditions:**

- Success: Customer authenticated; profile created/updated; addresses saved.
- Failure: No account created; customer not logged in.

**Main Flow:**

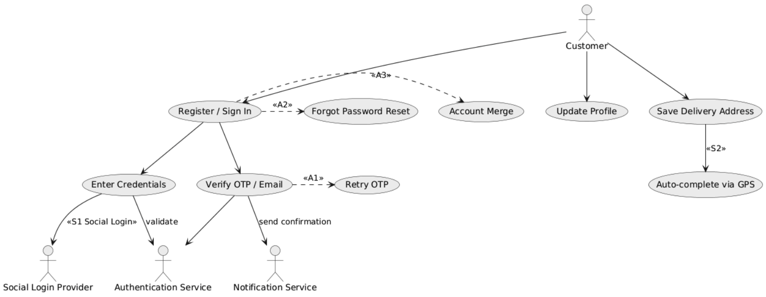
1. Customer selects “Register / Sign In.”
2. System prompts for credentials (email/phone/social login).
3. Customer enters credentials.
4. System sends OTP/email verification.
5. Customer verifies → Auth Service validates → account created.
6. Customer updates profile (name, contact info).
7. Customer saves one or more delivery addresses.

**Subflows:**

- S1: Social Login (Google/Apple).
- S2: Auto-complete address via GPS lookup.

**Alternative Flows:**

- A1: Invalid OTP → system prompts retry.
- A2: Forgot Password → reset flow.
- A3: Duplicate Account → system offers account merge.



---

### 2. Browse Restaurants & Menus

**Primary Actor:** Customer
**Supporting Actors:** Location Service, Catalog Service, Search/Filter Service
**Preconditions:** Customer logged in; catalog available.
**Postconditions:** Customer views restaurant and menu items; can add to cart.

**Main Flow:**

1. Customer sets or confirms delivery address.
2. System retrieves list of available restaurants with ETA and fees.
3. Customer browses menus, applies filters/sorts.

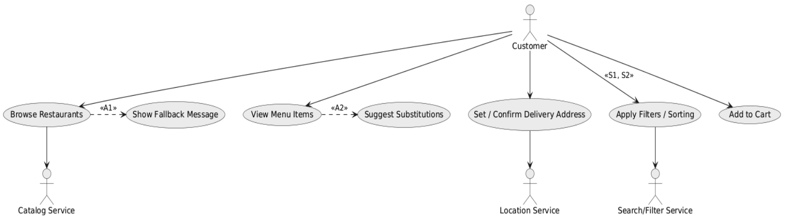
4. Customer views item details and adds to cart.

**Subflows:**

- S1: Dietary filters (vegan, gluten-free, etc.).
- S2: Price sorting.

**Alternative Flows:**

- A1: No restaurants available → fallback message.
- A2: Item unavailable → suggest substitutions.



---

### 3. Restaurant Recommendation (Past History & Location)

**Primary Actor:** Customer
**Supporting Actors:** Recommendation Engine, Analytics Service
**Preconditions:** Customer logged in.
**Postconditions:** Personalized restaurant suggestions displayed.

**Main Flow:**

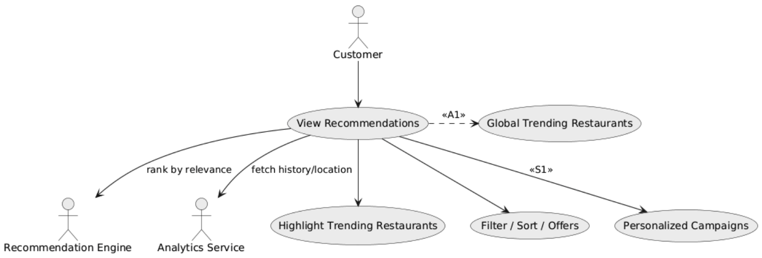
1. System fetches customer's past orders, ratings, and location.
2. Recommendation Engine ranks restaurants by relevance.
3. System highlights trending/popular ones if no history.
4. Customer can filter, sort, or view special offers.

**Subflows:**

- S1: Personalized campaign promotions.

**Alternative Flows:**

- A1: No history and no local data → show top global trending restaurants.



---

### 4. Place Order & Payment

**Primary Actor:** Customer
**Supporting Actors:** Payment Gateway, Promo Engine, Address Service
**Preconditions:** Customer logged in; cart contains items; valid delivery address.
**Postconditions:**

- Success: Order created; payment authorized; order ID assigned.
- Failure: No order created.

**Main Flow:**

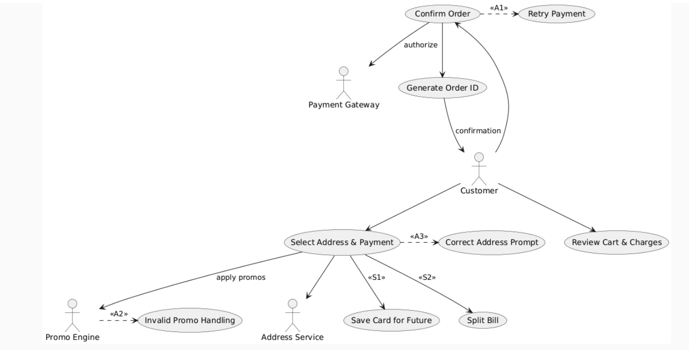
1. Customer reviews cart, charges, and promos.
2. Customer selects address and payment method.
3. System applies promo and computes taxes/fees.
4. Customer confirms → Payment Gateway authorizes.
5. System generates order ID and sends confirmation.

**Subflows:**

- S1: Save card for future.
- S2: Split bill between methods.

**Alternative Flows:**

- A1: Payment failure → retry flow.
- A2: Invalid promo → remove and recalc.
- A3: Address invalid → prompt correction.



---

**5. Restaurant Order Handling**

**Primary Actor:** Restaurant Operator
**Supporting Actors:** POS/KDS System, Order Management Service
**Preconditions:** Restaurant open; new order received.
**Postconditions:** Order accepted, updated to “ready for pickup” OR rejected.

**Main Flow:**

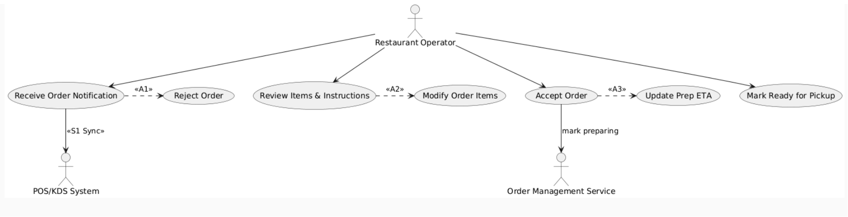
1. Restaurant receives order notification.
2. Restaurant reviews items and instructions.
3. Restaurant accepts order.
4. Marks status as “preparing.”
5. Marks “ready for pickup” once finished.

**Subflows:**

- S1: POS sync for automated order ingestion.

**Alternative Flows:**

- A1: Reject order (stock issues) → system cancels/refunds.
- A2: Modify order (remove unavailable items).
- A3: Update prep ETA.



---

**6. Driver Assignment & Pickup**

**Primary Actor:** Rider
**Supporting Actors:** Dispatching Service, Navigation, Restaurant
**Preconditions:** Order ready; driver pool available.
**Postconditions:** Rider picks up order.

**Main Flow:**

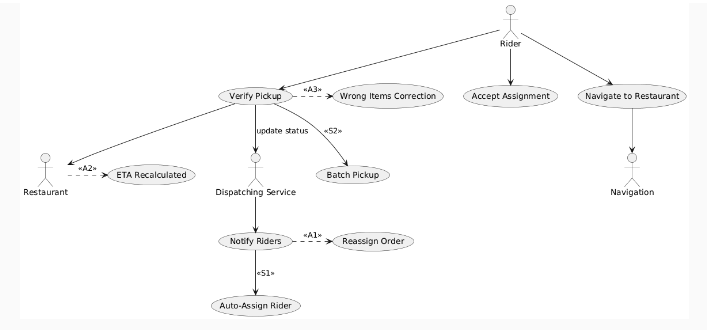
1. Dispatching Service notifies nearby riders.
2. Rider accepts.
3. Rider navigates to restaurant.
4. Rider verifies pickup (QR/photo).
5. Status updated to “picked up.”

**Subflows:**

- S1: Auto-assign driver if no manual acceptance.
- S2: Batch pickup for multiple orders.

**Alternative Flows:**

- A1: No driver accepts → escalate/reassign.
- A2: Restaurant delay → ETA recalculated.
- A3: Wrong items → correction or cancel.



---

### 7. Delivery and Tracking

**Primary Actors:** Customer and Rider
**Supporting Actors:** GPS Service, Map/Navigation Service, Notification Service
**Preconditions:** Order picked up by rider (or en route);  GPS/location services enabled.
**Postconditions:** Order delivered to customer OR returned.; Customer sees live tracking and ETA throughout delivery.

**Main Flow:**

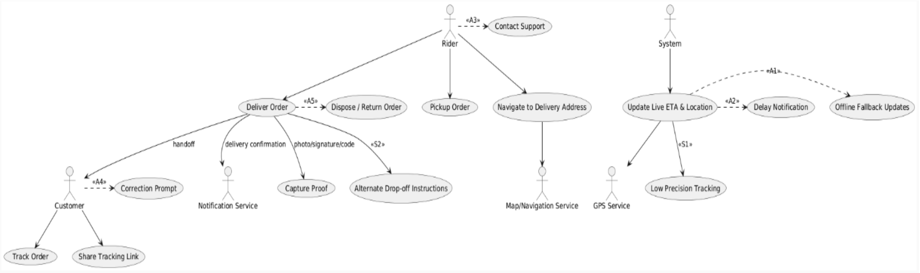
1. Rider picks up the order from the restaurant.
2. Rider navigates to delivery address using Navigation Service.
3. System updates order status and live location for customer (preparing → picked up → arriving).
4. Customer views live ETA and map; may share tracking link.
5. Rider delivers order to customer or leaves at designated drop-off location.
6. Rider captures delivery proof (photo/signature/code).
7. System marks order as “delivered” and notifies customer.

**Subflows:**

- S1: Low-precision tracking if GPS weak or unavailable.
- S2: Alternate drop-off instructions (neighbor, security desk, safe spot).

**Alternative Flows:**

- A1: Rider offline → system provides status updates via text or notifications.
- A2: Delay detected → ETA recalculated and customer notified.
- A3: Customer unreachable → rider contacts support.
- A4: Address invalid → system prompts rider/customer for correction.
- A5: Order cannot be delivered or must be returned/disposed → system updates status and notifies customer.



### 8. Ratings & Feedback

**Primary Actor:** Customer
**Supporting Actors:** Restaurant, Rider, Moderation Service
**Preconditions:** Order delivered.
**Postconditions:** Ratings stored, aggregated, visible to stakeholders.

**Main Flow:**

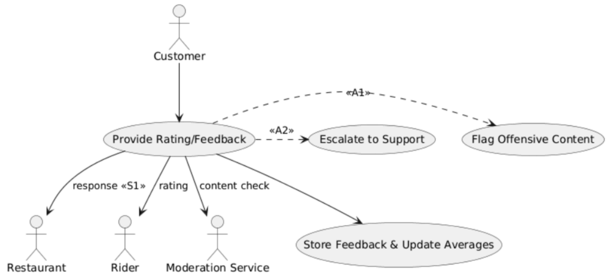
1. The system prompts customers after delivery.
2. Customer provides star rating, text, or private feedback.
3. System stores and updates averages.

**Subflows:**

- S1: Restaurant responds to review.

**Alternative Flows:**

- A1: Offensive content flagged.
- A2: Issue instead of review → escalate to support.



---

### 9. Customer Support & Dispute Resolution

**Primary Actor:** Customer / Support Agent
**Supporting Actors:** Restaurant, Rider, Payment Provider
**Preconditions:** Order completed or in-progress with reported issue.
**Postconditions:** Ticket resolved/closed or escalated.

**Main Flow:**

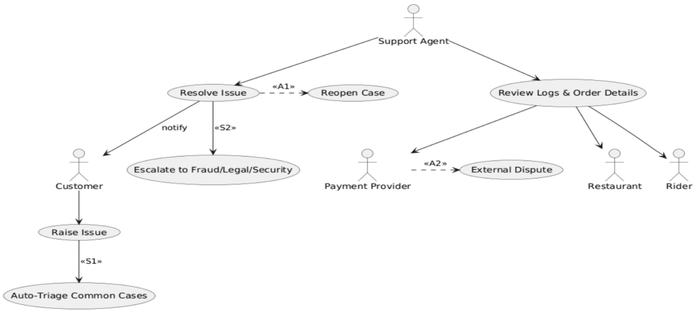
1. Customer raises issue (chat/call/ticket).
2. Support agent reviews logs and order details.
3. Contacts restaurant/rider/payment if needed.
4. Issues resolution (refund, redelivery, credit).
5. Ticket closed; customer notified.

**Subflows:**

- S1: Auto-triage resolves common cases like refund, replacement and order cancellation.
- S2: Escalation path to fraud/legal/security.

**Alternative Flows:**

- A1: Customer disputes resolution → reopen case.
- A2: Payment dispute filed externally.



---

### 10. Restaurant Menu & Availability Management

**Primary Actor:** Restaurant Operator
**Supporting Actors:** Catalog Service, Validation Service
**Preconditions:** Restaurant onboarded.
**Postconditions:** Menu updated and visible.

**Main Flow:**

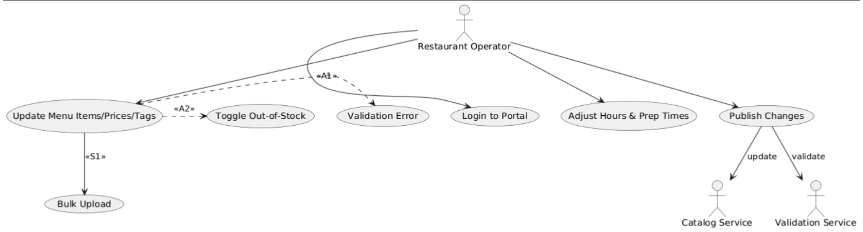
1. Restaurant logs in to portal.
2. Updates menu items, prices, tags.
3. Adjusts hours/prep times.
4. Publishes changes; system validates and updates catalog.

**Subflows:**

- S1: Bulk upload.

**Alternative Flows:**

- A1: Invalid entry → validation error.
- A2: Out-of-stock toggle.


Restaurant Operator
«A1»
«A2»
Update Menu Items/Prices/Tags
Toggle Out-of-Stock
Validation Error
Login to Portal
Adjust Hours & Prep Times
Publish Changes
«S1»
update
validate
Bulk Upload
Catalog Service
Validation Service