CS5700 Project 1: Object Oriented Programming - PackageManager

Yuxin Liang

1. Business Requirements – Package Managing System

The PackageManager is a Package Tracking System that helps users keep track of all their packages in one place. When users receive shipping confirmation emails, the system automatically reads these emails to extract important information like tracking numbers, carrier details (DHL, UPS, FedEx, or CHEMLOG), and expected delivery dates. The system organizes packages by delivery date and lets users know when packages are arriving soon. Users can add tags to their packages (like "Gift" or "Electronics") to help organize them and can easily search or filter their packages by status (like "Delivered" or "In Transit"). The system is especially helpful for international shipments through DHL and hazardous materials through CHEMLOG, providing special handling instructions when needed. Every time a package's status changes or is about to be delivered the next day, the system sends a notification to keep users updated. Users can also look back at the complete delivery history of each package, showing how it moved from shipment to delivery.

- **Nouns-verbs**: Noun Verbs

- **Target Audience**
  - Individual Consumers: Users who receive multiple packages and want to track their shipments efficiently.
  - Online Shoppers: Users frequently purchasing from e-commerce platforms like Amazon, eBay, or other online stores.
  - Business Professionals: People managing office shipments and deliveries.

- Rules
  - The system should automatically parse emails to detect tracking numbers.
  - The system should identify the delivery company (FedEx, UPS, DHL) from the email.
  - The system should extract the estimated delivery date from the email.
  - The system should store package details, including sender information, item description (if available), and tracking links.
  - The system should sort packages by estimated delivery date in ascending order.
  - Users should be able to filter packages based on delivery status: Delivered, In Transit, Arriving Soon.
  - The system should check delivery dates and simulate a console notification when a package is expected to arrive tomorrow.
  - Users should be able to view package contents.

- Users should be able to tag packages with categories such as Electronics, Clothing, or Gift.
- Users should be able to search for packages by tracking number, sender, or category.
- Users should be able to filter packages based on status (Delivered, In Transit).
- The system should extract delivery dates from emails and display upcoming deliveries in chronological order.

- **Challenge Questions (ask user) - questionnaire, no need to answer, but it is ok to make assumptions**
  - **Are 3-4 shipping companies enough?**
    - Assumption: We implemented 4 shipping companies, but we can always extend it in the future based on needs.
  - **Should package categories be predefined, or should users create their own custom labels?**
    - Assumption: No predefined categories, we simulate users adding string as a tag.
  - **Can users filter packages by multiple criteria at once?**
    - Assumption: Can only filter by one criterion at a time.
  - **What happens if an email contains multiple tracking numbers?**
    - Assumption: One tracking number per email

- **Summary of Classes, Attributes and Associations (from nouns and verbs)**
  - Package
    - Attributes:
      - trackingNumber: string
      - carrier: Carrier
      - sender: string
      - description: string
      - status: STATUS_TYPES (IN_TRANSIT, DELIVERED, etc.)
      - estimatedDeliveryDate: Date
      - tags: string[]
      - history: ShipmentHistory
    - Methods:
      - addTag(), removeTag()
      - updateStatus()
      - getContents()
    - Associations:
      - Has-One Carrier
      - Has-One ShipmentHistory

        - Belongs-To User
- Carrier
    - Attributes:
        - name: string
        - trackingPattern
        - trackingUrlTemplate: string
        - type: TYPES (DHL, UPS, FEDEX, CHEMLOG)
    - Methods:
        - generateTrackingLink()
        - validateTrackingNumber()
    - Specialized By:
        - DHLInternational (adds internationalZones)
        - HazmatCarrier (adds hazmatClasses, safetyProtocols)
- User
    - Attributes:
        - username: string
        - email: string
        - passwordHash: string
        - packages: Package[]
        - preferences: UserPreferences
    - Methods:
        - validatePassword()
        - addPackage(), removePackage()
        - getPackages()
    - Associations:
        - Has-Many Package
        - Has-One UserPreferences
        - **2. Supporting Classes**
- UserPreferences
    - Attributes:
        - notificationEnabled: boolean
        - emailNotifications: boolean
        - defaultTags: string[]
        - theme: string
    - Methods:
        - toggleNotifications()
        - setTheme()
        - addDefaultTag(), removeDefaultTag()
    - Associations:
        - Belongs-To User
- PackageTrackingSystem (Controller)
    - Attributes:

            - packages: Package[]
            - subscribers: Function[]
            - carriers: Map<string, Carrier>
            - emailParser: EmailParser
            - userManager: UserManager
        - Methods:
            - registerUser(), login(), logout()
            - extractFromEmail()
            - filterByStatus(), searchPackages()
            - subscribe(), notifySubscribers()
        - Associations:
            - Uses EmailParser
            - Uses UserManager
            - Manages Carriers
    - EmailParser
        - Attributes:
            - carriers: Map<string, Carrier>
        - Methods:
            - parse()
            - findTrackingInfo()
            - extractSender(), extractDescription()
            - extractDeliveryDate()
        - Associations:
            - Used-By PackageTrackingSystem
            - References Carriers
    - PackageTrackerCLI
        - Attributes:
            - trackingSystem: PackageTrackingSystem
        - Methods:
            - getNotificationHandler()
            - setupNotifications()
            - showMenu()
            - start()
        - Associations:
            - Uses PackageTrackingSystem

Key Relationships:

- Associations :
  User -Package (One-to-Many) A user can have multiple packages.
  User -UserPreferences (One-to-One) A user has preferences for

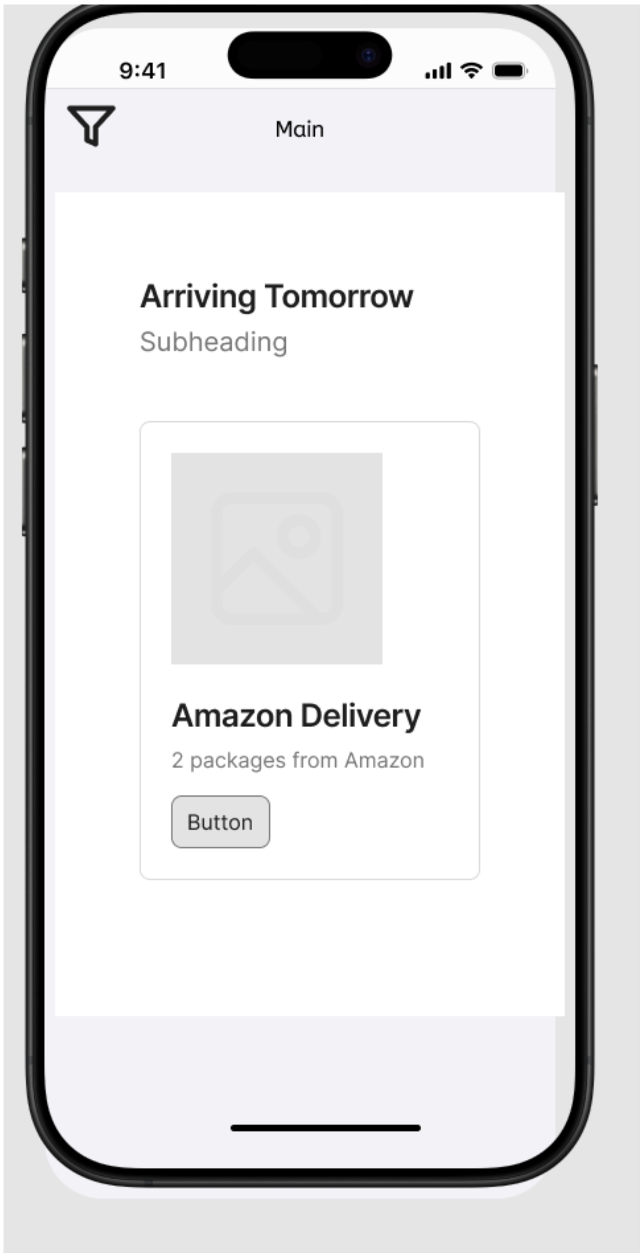
notifications and filters.
PackageTrackingSystem - UserManager (One-to-One) The package tracking system manages users via the UserManager.
Package - ShipmentHistory (One-to-One) Each package has a shipment history.

- Inheritance
  Carrier (Superclass) → DHLInternational, HazmatCarrier (Subclasses) Different types of carriers inherit from the base Carrier class.
- Aggregation
  PackageTrackingSystem & Carrier: The PackageTrackingSystem references carriers but does not own them exclusively.
  Package & Tag: A package can have multiple tags, and tags can exist independently of packages.
- Composition
  User ◆ UserPreferences: The User owns UserPreferences, and they cannot exist independently.
  Package ◆ ShipmentHistory: A package contains a ShipmentHistory, which does not exist without a package.
  ShipmentHistory ◆ StatusUpdate: A shipment history contains multiple status updates, meaning they are tightly coupled.

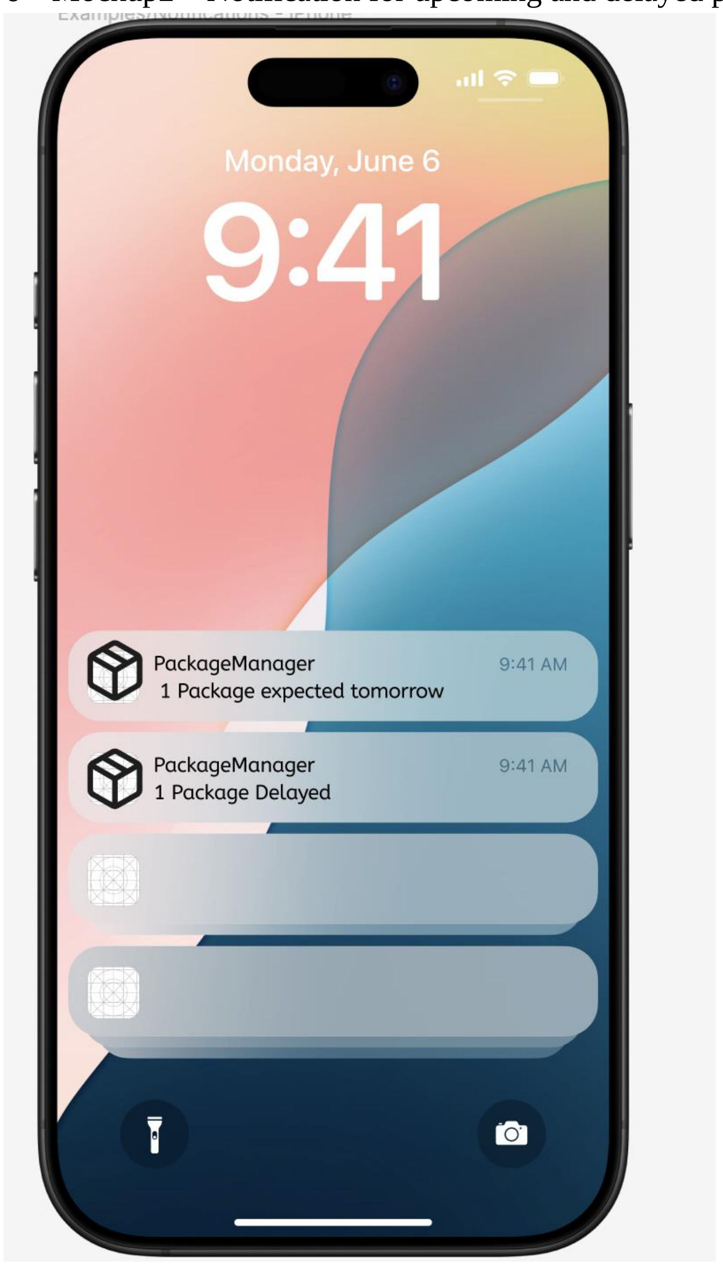
- **User personas (at least 2) and user stories (at least 3 per persona)**
  - **Two key Dimensions:**
    - Package Organization & Tracking Dimension
      - Primary Focus: How users interact with and organize their packages
    - Notification & Alert Dimension
      - Primary Focus: How users receive updates about their packages
  - **Persona 1: Sarah (Frequent Online Shopper)**
    - Dimension: Chronological package sorting & Next-day delivery notifications
    - Age: 28
    - Background: A busy marketing manager who frequently shops online and needs a convenient way to track deliveries.

    - Scenario: Sarah orders multiple packages from different retailers and often forgets when they will arrive. She wants an easy way to track her orders and receive reminders.
    - Motivation: Wants a centralized system to track package arrivals without manually checking emails.
    - User Stories:
        - As a frequent online shopper, I want to see a list of all my incoming packages sorted by delivery date so that I can easily know when my packages will arrive. (Mockup1)
        - As a frequent online shopper, I want to receive a notification when a package is arriving tomorrow so that I don't forget to check for it. (Mockup 2)
        - As a frequent online shopper, I want to search for a package using the tracking number so that I can quickly check its status. (Mockup3)
- Persona 2: Michael (Small Business Owner)
    - Dimension: Package categorization (tags) & Status filtering
    - Age: 35
    - Background: Owns an online retail business and frequently receives shipments of inventory. Needs to track incoming stock efficiently.
    - Scenario: Michael manages multiple deliveries from different suppliers and needs a system to categorize and filter packages based on status and content.
    - Motivation: Wants an organized system that helps him manage inventory shipments without delays.
    - User Stories:
        - As a small business owner, I want to categorize packages with labels such as 'Electronics' or 'Clothing' so that I can easily track my inventory. (Mockup5)
        - As a small business owner, I want to filter my packages by status (Delivered, In Transit) so that I can focus on pending deliveries.(Mockup4)
        - As a small business owner, I want to receive a notification if a package is delayed so that I can plan accordingly.(Mockup2)

- Interface low level mockups
    - Mockup1



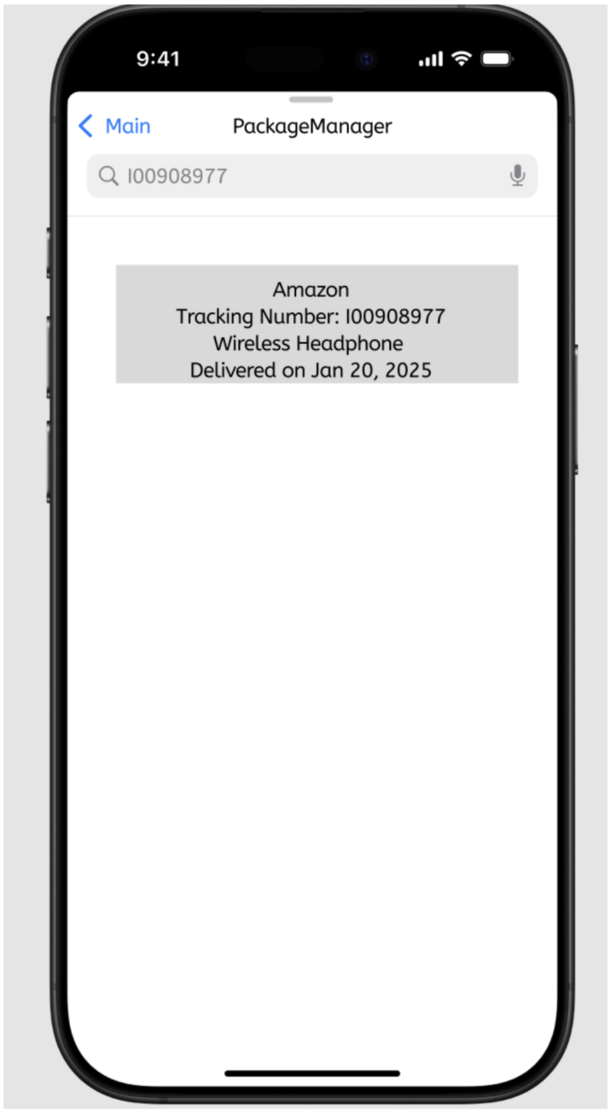
- Mockup2 – Notification for upcoming and delayed packages



PackageManager 9:41 AM
1 Package expected tomorrow

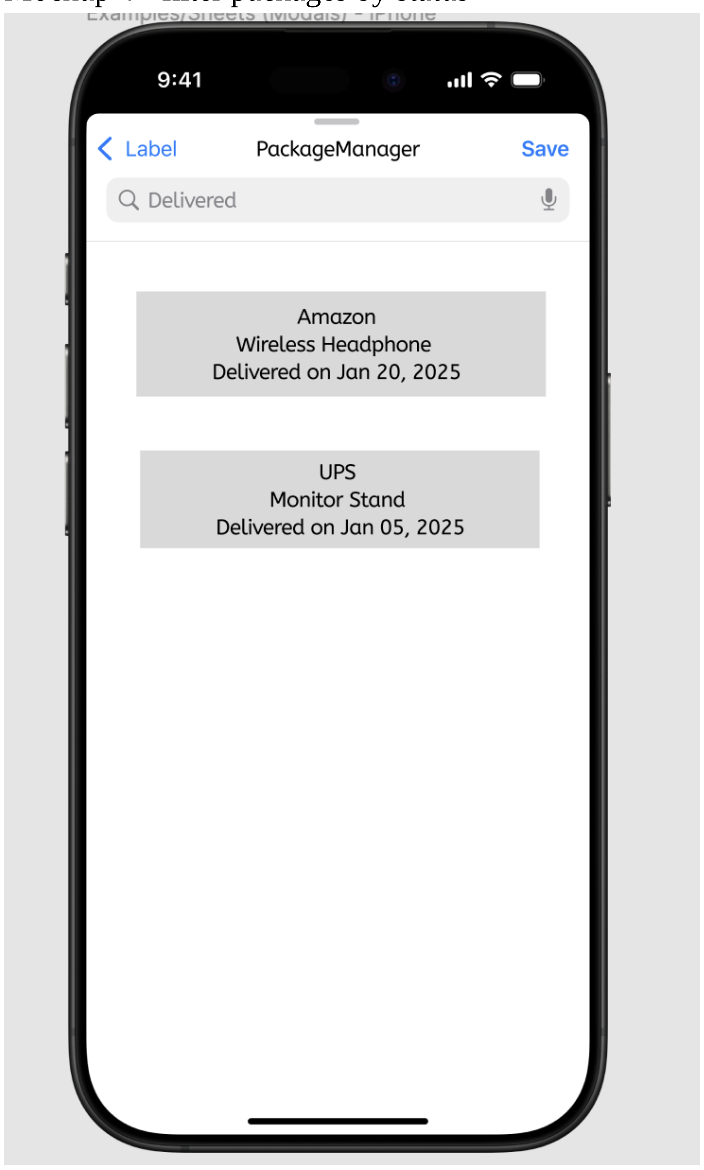
PackageManager 9:41 AM
1 Package Delayed

- Mockup 3 – Search package by tracking number



Amazon
Tracking Number: I00908977
Wireless Headphone
Delivered on Jan 20, 2025

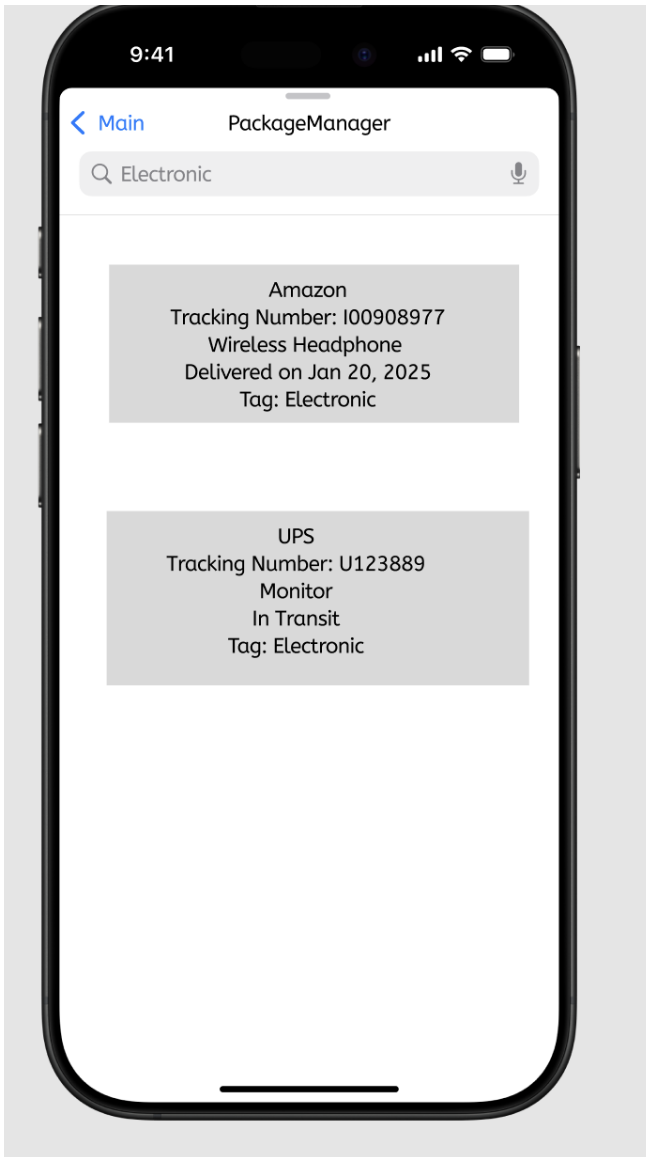
- Mockup 4 - filter packages by status



Amazon
Wireless Headphone
Delivered on Jan 20, 2025

UPS
Monitor Stand
Delivered on Jan 05, 2025

- Mockup5 – Filter Package by different Tags

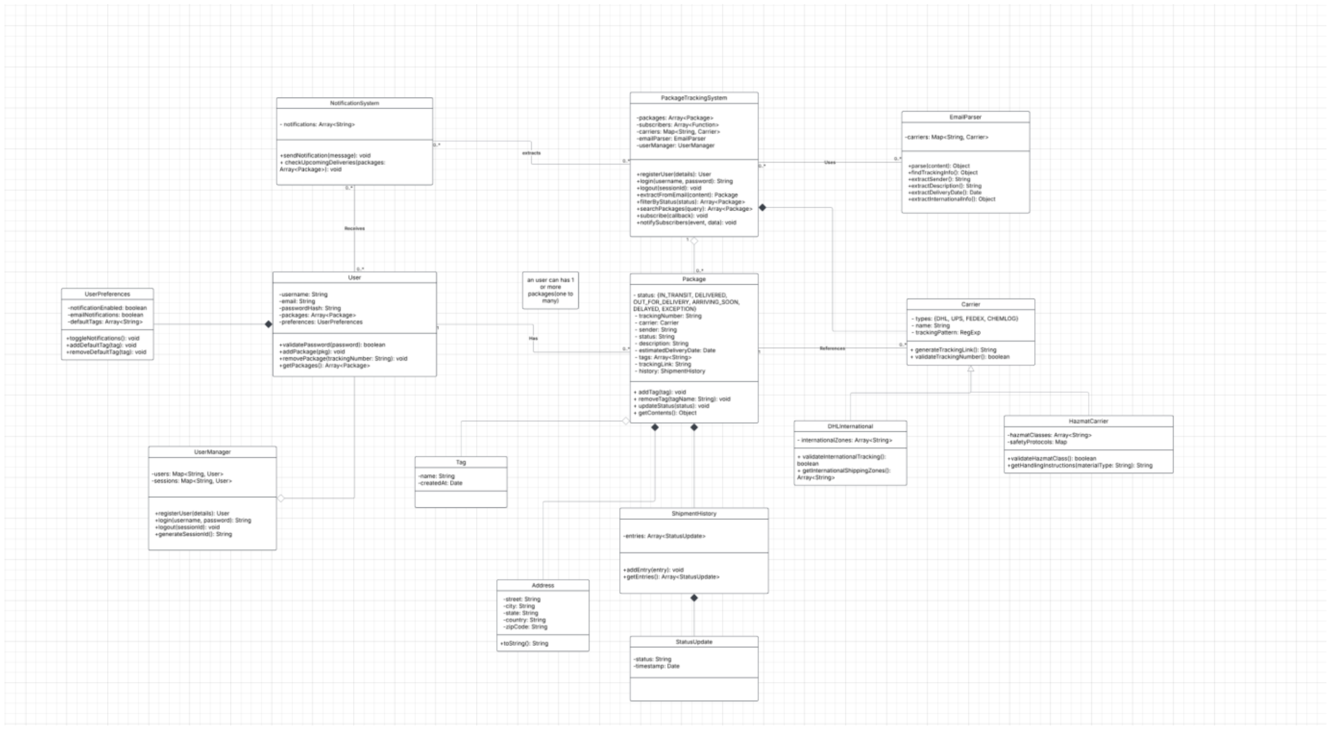


Amazon
Tracking Number: I00908977
Wireless Headphone
Delivered on Jan 20, 2025
Tag: Electronic

UPS
Tracking Number: U123889
Monitor
In Transit
Tag: Electronic

- UML Class Diagram
  https://lucid.app/lucidchart/f09e9faa-2289-42c8-aac8-

a565e59f32de/edit?invitationId=inv_32b86b0b-176c-4a36-99b2-6c56b5d748f3&page=0_0#

- 
NotificationSystem
- notifications: Array<String>
+sendNotification(message): void
+ checkUpcomingDeliveries(packages: Array<Package>): void
extracts
0..*
Receives
User
-username: String
-email: String
-passwordHash: String
-packages: Array<Package>
-preferences: UserPreferences
+validatePassword(password): boolean
+addPackage(pkg): void
+removePackage(trackingNumber: String): void
+getPackages(): Array<Package>
UserPreferences
-notificationEnabled: boolean
-emailNotifications: boolean
-defaultTags: Array<String>
+toggleNotifications(): void
+addDefaultTag(tag): void
+removeDefaultTag(tag): void
an user can has 1 or more packages(one to many)
Has
PackageTrackingSystem
-packages: Array<Package>
-subscribers: Array<Function>
-carriers: Map<String, Carrier>
-emailParser: EmailParser
-userManager: UserManager
+registerUser(details): User
+login(username, password): String
+logout(sessionId): void
+extractFromEmail(content): Package
+filterByStatus(status): Array<Package>
+searchPackages(query): Array<Package>
+subscribe(callback): void
+notifySubscribers(event, data): void
Uses
EmailParser
-carriers: Map<String, Carrier>
+parse(content): Object
+findTrackingInfo(): Object
+extractSender(): String
+extractDescription(): String
+extractDeliveryDate(): Date
+extractInternationalInfo(): Object
Package
- status: (IN_TRANSIT, DELIVERED, OUT_FOR_DELIVERY, ARRIVING_SOON, DELAYED, EXCEPTION)
- trackingNumber: String
- carrier: Carrier
- sender: String
- status: String
- description: String
- estimatedDeliveryDate: Date
- tags: Array<String>
- trackingLink: String
- history: ShipmentHistory
+ addTag(tag): void
+ removeTag(tagName: String): void
+ updateStatus(status): void
+ getContents(): Object
References
Carrier
- types: (DHL, UPS, FEDEX, CHEMLOG)
- name: String
- trackingPattern: RegExp
+ generateTrackingLink(): String
+ validateTrackingNumber(): boolean
DHLInternational
- internationalZones: Array<String>
+ validateInternationalTracking(): boolean
+ getInternationalShippingZones(): Array<String>
HazmatCarrier
-hazmatClasses: Array<String>
-safetyProtocols: Map
+validateHazmatClass(): boolean
+getHandlingInstructions(materialType: String): String
UserManager
-users: Map<String, User>
-sessions: Map<String, User>
+registerUser(details): User
+login(username, password): String
+logout(sessionId): void
+generateSessionId(): String
Tag
-name: String
-createdAt: Date
ShipmentHistory
-entries: Array<StatusUpdate>
+addEntry(entry): void
+getEntries(): Array<StatusUpdate>
Address
-street: String
-city: String
-state: String
-country: String
-zipCode: String
+toString(): String
StatusUpdate
-status: String
-timestamp: Date

- **Use of Generative AI in Development**
  - **Purpose**
    - Generative AI (**ChatGPT**) was consulted to enhance my understanding of **class structures and relationships** for the **Package Tracking System**, specifically focusing on the **PackageManager** class.
  - **Prompts and Responses**
    - **Prompt:** " What other classes should I have besides User and PackageManager?"
    - **Response:** ChatGPT suggested additional classes such as:EmailParser (to handle package extraction from emails),NotificationSystem (to manage package delivery alerts),FilterSystem (to allow users to search and filter packages). These additions helped refine the class diagram and improve the logical structure of the system.
    - **Prompt:** " What should be the associative relationships between User, Package, and PackageManager? "
    - **Response:** ChatGPT explained that:
      - User interacts with PackageManager to track and manage packages.
      - PackageManager aggregates multiple Package objects and provides sorting/filtering.
      - NotificationSystem queries PackageManager to check for upcoming deliveries and send notifications.

- **How it was Used**
  The AI-generated insights served as a starting point for refining my UML class diagram and system structure. While I ultimately designed my own PackageManager class and relationships, ChatGPT's responses helped me think through associations, class dependencies, and functionality gaps before finalizing the design.