CSE251 – " Project Report "

1. Project Title

Smart Reverse Vending Machine with Gamification System

2. Project Description

The proposed project aims to design and implement a smart reverse vending machine (RVM) that encourages university students to recycle plastic waste through gamification and rewards. Students scan their unique QR codes, deposit plastic recyclables, and earn points based on weight. The system creates multi-level competition (student vs. student, college vs. college) with leaderboards to drive engagement.

Points earned can be redeemed at the university cafeteria for food and beverages, creating a closed-loop incentive system. The RVM is powered by Raspberry Pi with integrated sensors (QR scanner, weight sensor, material classifier, liquid detector) and operates with offline capability, automatically synchronizing data when connectivity is restored.

Material Classification: AI-powered sensors automatically identify whether deposited items are plastic or non-plastic waste. Non-plastic materials are diverted to a separate waste compartment, preventing contamination of recyclable materials.

Liquid Detection: Advanced weight and tilt sensors detect liquid content inside plastic bottles. Bottles containing liquid are automatically rejected with clear instructions displayed to users to empty them first, ensuring recycling quality and preventing machine damage.

The RVM includes a "Guest" button allowing unregistered campus members (doctors, TAs, faculty, staff, visitors) to contribute to recycling efforts without registration. Guest deposits contribute to overall campus sustainability metrics, promoting inclusive environmental responsibility across the entire university community.

The system consists of a mobile application for students to view their QR codes, track points, check leaderboards, and redeem rewards. Administrators monitor system performance, manage inventory, view analytics through a web dashboard, and manage multiple RVM machines deployed across the university campus. The system automatically alerts administrators when machine capacity reaches 80% for timely maintenance and emptying. Administrators can also manage cafeteria rewards and view detailed reports on transactions, rankings, and environmental impact.

By combining IoT hardware, AI-based material validation, gamification, renewable energy awareness, and smart campus technologies, this project transforms waste management into an engaging, competitive, and eco-friendly campus service available to ALL campus members. **The system architecture follows SOLID principles to ensure maintainability, scalability, and extensibility.**

"This project demonstrates the integration of IoT, AI validation, gamification, and sustainable practices into a scalable and eco-friendly campus recycling solution, designed with SOLID principles for robust software architecture and inclusive campus-wide participation."

3. Team Members

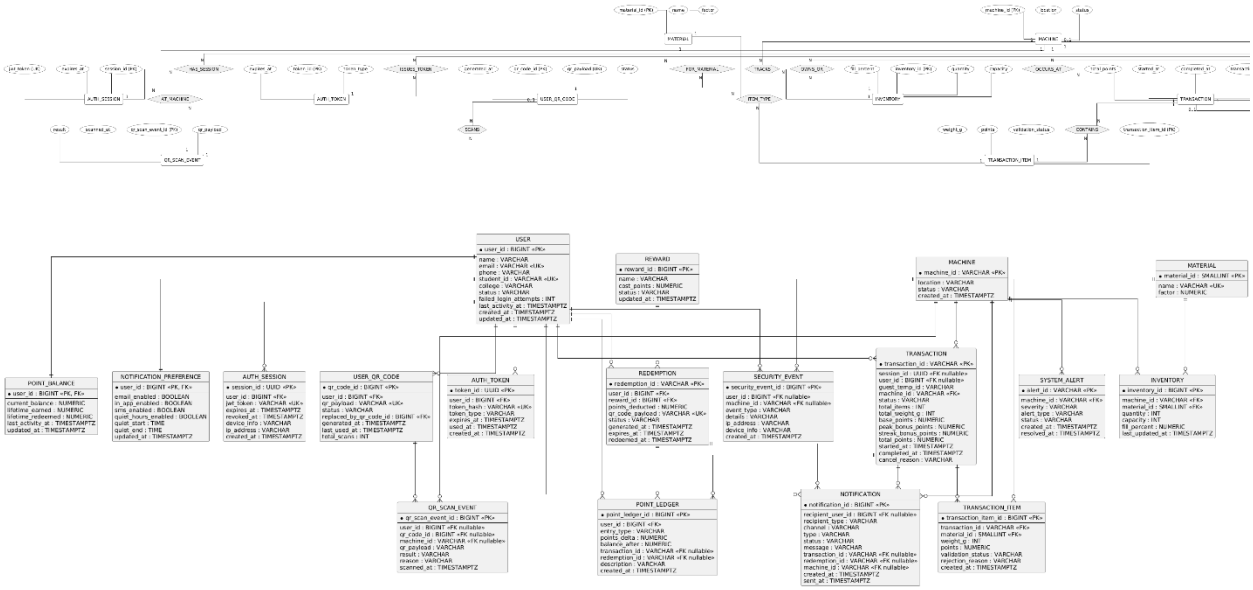
ID	Name
23101608	Yasseen Ahmed El-Sayed
23101594	Mohamed Mustafa Awad
23101545	Ahmed Khaled Said
23101599	Marawan Khaled Ahmed
23101899	Mohamed Adel Abdulrahman

4. Database design:

Table + Relation Summary

USER is the parent for almost everything.
AUTH_SESSION = active login sessions/JWTs (1 user → many sessions).
AUTH_TOKEN = one-time tokens (email verification + password reset) (1 user → many).
USER_QR_CODE = user QR lifecycle (active/compromised/replaced). Self-reference via replaced_by_qr_code_id.
QR_SCAN_EVENT logs every scan (supports invalid/fake scans; user_id nullable).

MACHINE + MATERIAL + INVENTORY = per-machine inventory by material (Machine 1→many Inventory; Material 1→many Inventory).
TRANSACTION + TRANSACTION_ITEM = recycling workflow and items. user_id nullable for guests with guest_temp_id.
POINT_BALANCE (1:1 with USER) + POINT_LEDGER (1:many) = accounting-grade points tracking. Ledger rows can reference a transaction_id or redemption_id.
REWARD + REDEMPTION = reward catalog and redemption QR with expiry.
NOTIFICATION_PREFERENCE + NOTIFICATION = user settings and notification history (notifications may reference transaction/redemption/machine).
SYSTEM_ALERT = machine alerts (full, sensor failure, etc).
SECURITY_EVENT = login failures, QR abuse, account lock, password changes, etc.



Relationships (Diamonds) + cardinalities (write 1 / N)

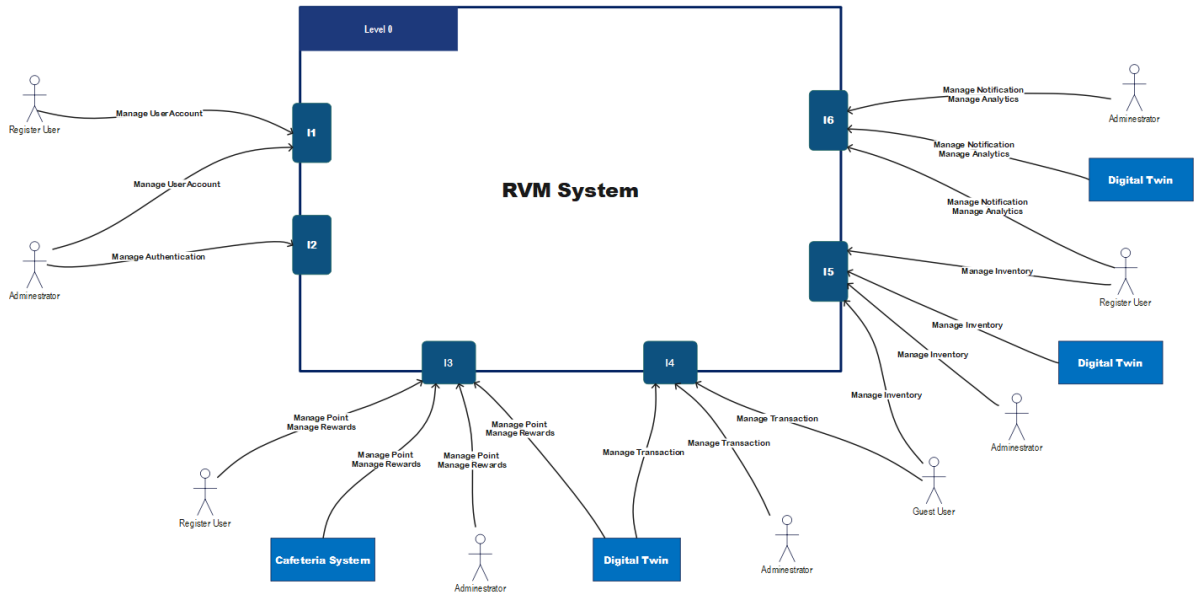
Use these diamonds between rectangles:

USER — “has” — AUTH_SESSION
USER (1) —— (N) AUTH_SESSION
USER — “receives” — AUTH_TOKEN
USER (1) —— (N) AUTH_TOKEN
USER — “owns” — USER_QR_CODE
USER (1) —— (N) USER_QR_CODE
USER_QR_CODE — “scanned_in” — QR_SCAN_EVENT
USER_QR_CODE (1) —— (N) QR_SCAN_EVENT
(QR_SCAN_EVENT may also exist without a valid QR in “fake QR” scenario — you can mark USER_QR_CODE optional)
MACHINE — “records” — QR_SCAN_EVENT
MACHINE (1) —— (N) QR_SCAN_EVENT
MACHINE — “has” — INVENTORY

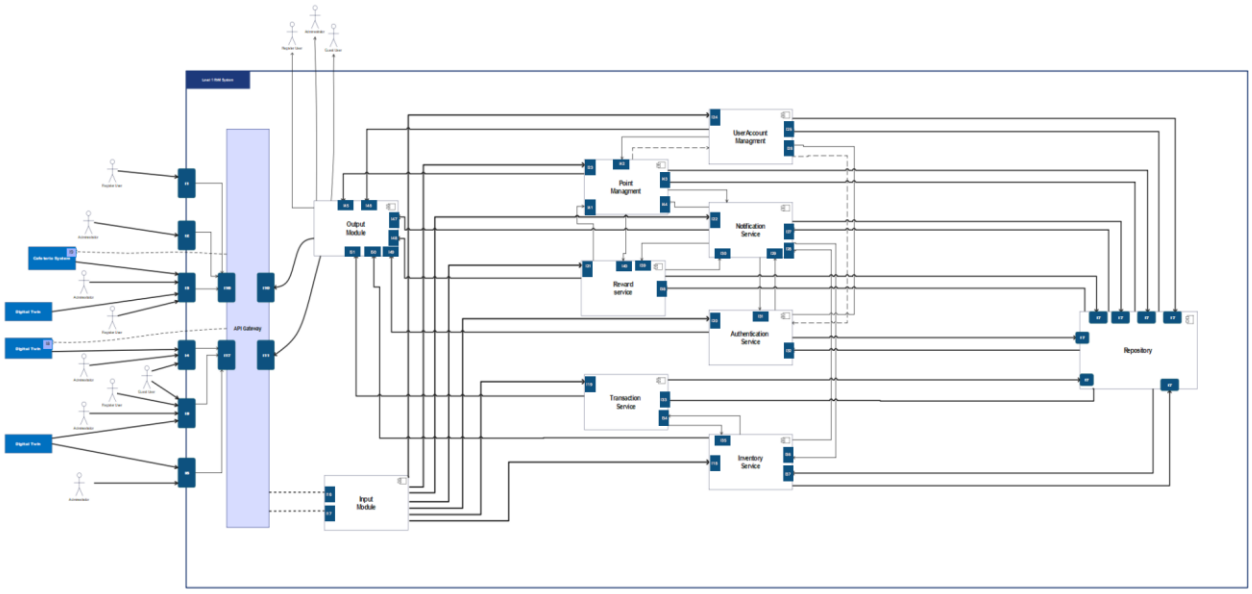
MACHINE (1) —— (N) INVENTORY
MATERIAL — “categorized_as” — INVENTORY
MATERIAL (1) —— (N) INVENTORY
USER — “performs” — TRANSACTION
USER (1) —— (N) TRANSACTION
(For guest: USER is optional; transaction has guest_temp_id)
MACHINE — “occurs_at” — TRANSACTION
MACHINE (1) —— (N) TRANSACTION
TRANSACTION — “contains” — TRANSACTION_ITEM
TRANSACTION (1) —— (N) TRANSACTION_ITEM
MATERIAL — “type_of” — TRANSACTION_ITEM
MATERIAL (1) —— (N) TRANSACTION_ITEM
USER — “has” — POINT_BALANCE
USER (1) —— (1) POINT_BALANCE
USER — “has” — POINT_LEDGER
USER (1) —— (N) POINT_LEDGER
TRANSACTION — “generates” — POINT_LEDGER
TRANSACTION (1) —— (N) POINT_LEDGER (earn entries)
USER — “makes” — REDEMPTION
USER (1) —— (N) REDEMPTION
REWARD — “is_redeemed_in” — REDEMPTION
REWARD (1) —— (N) REDEMPTION
REDEMPTION — “creates” — POINT_LEDGER
REDEMPTION (1) —— (N) POINT_LEDGER (redeem entries)
USER — “receives” — NOTIFICATION
USER (1) —— (N) NOTIFICATION
USER — “configures” — NOTIFICATION_PREFERENCE
USER (1) —— (1) NOTIFICATION_PREFERENCE
MACHINE — “triggers” — SYSTEM_ALERT
MACHINE (1) —— (N) SYSTEM_ALERT
USER/MACHINE — “logs” — SECURITY_EVENT
USER (0/1) —— (N) SECURITY_EVENT
MACHINE (0/1) —— (N) SECURITY_EVENT

5- The architecture models :

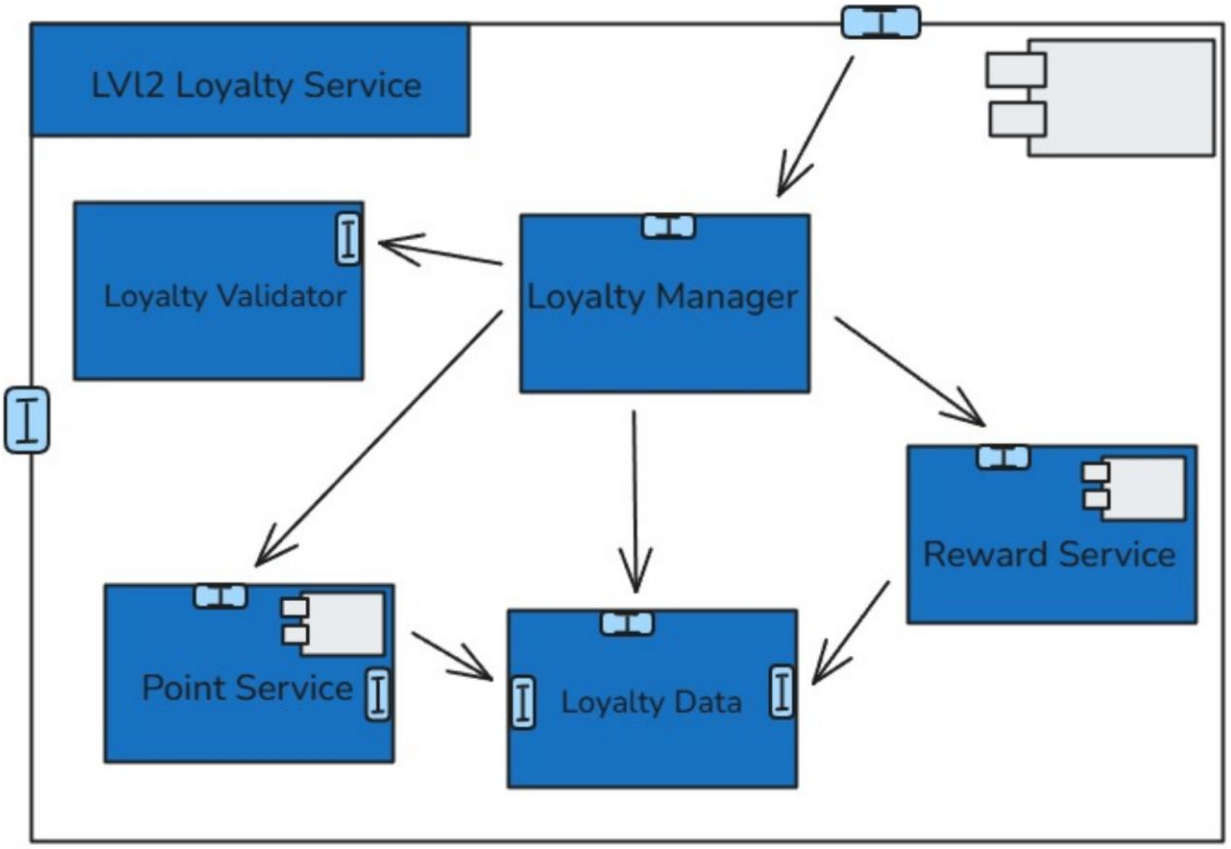
Level 0 :

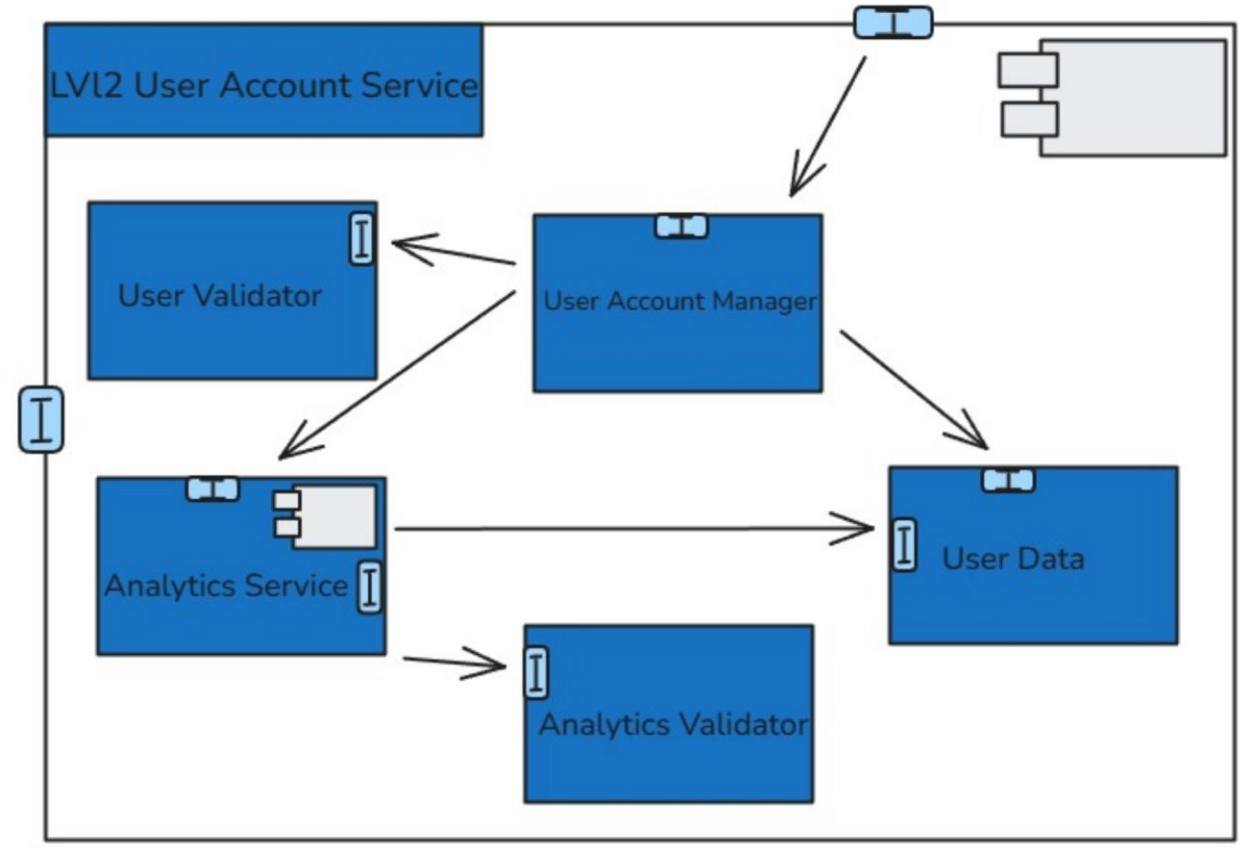


Level 1 :

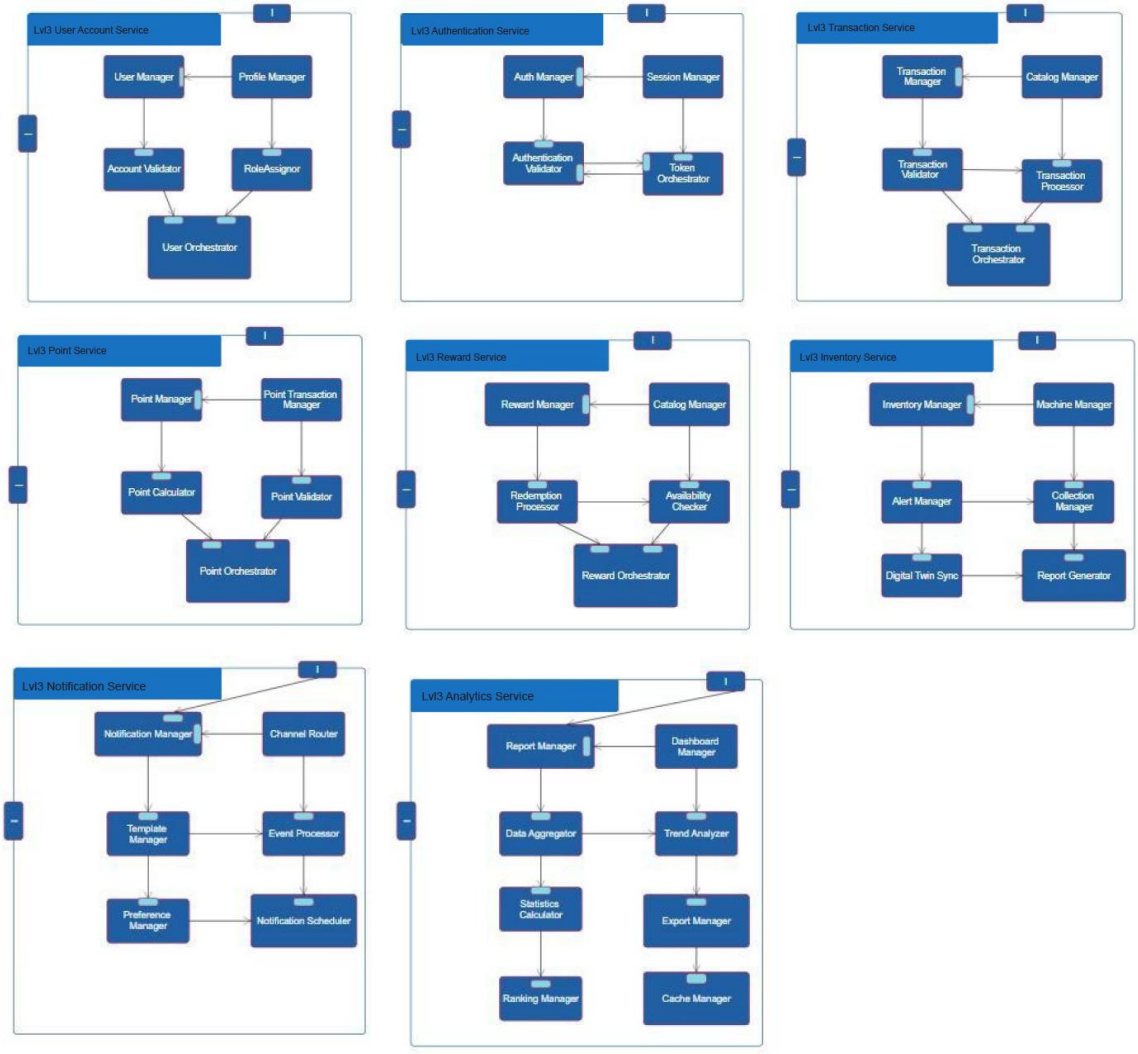


Level 2 :


LVl2 Loyalty Service
Loyalty Validator
Loyalty Manager
Reward Service
Point Service
Loyalty Data


LVl2 User Account Service
User Validator
User Account Manager
Analytics Service
User Data
Analytics Validator

Level 3:


Lvl3 User Account Service
User Manager
Profile Manager
Account Validator
RoleAssignor
User Orchestrator
Lvl3 Authentication Service
Auth Manager
Session Manager
Authentication Validator
Token Orchestrator
Lvl3 Transaction Service
Transaction Manager
Catalog Manager
Transaction Validator
Transaction Processor
Transaction Orchestrator
Lvl3 Point Service
Point Manager
Point Transaction Manager
Point Calculator
Point Validator
Point Orchestrator
Lvl3 Reward Service
Reward Manager
Catalog Manager
Redemption Processor
Availability Checker
Reward Orchestrator
Lvl3 Inventory Service
Inventory Manager
Machine Manager
Alert Manager
Collection Manager
Digital Twin Sync
Report Generator
Lvl3 Notification Service
Notification Manager
Channel Router
Template Manager
Event Processor
Preference Manager
Notification Scheduler
Lvl3 Analytics Service
Report Manager
Dashboard Manager
Data Aggregator
Trend Analyzer
Statistics Calculator
Export Manager
Ranking Manager
Cache Manager

6- List interfaces description :

List Description Diagram:

Interface 1 (I1): Manage User

① public boolean registerUser(String username, String password, String email);
② public boolean loginUser(String username, String password);
③ public boolean logoutUser(int userId);
④ public User getUserProfile(int userId);
⑤ public boolean updateUserProfile(int userId, String email, String phone);
⑥ public boolean deleteUserAccount(int userId);
⑦ public boolean resetPassword(String email);
⑧ public List<User> getAllUsers(String role);
⑨ public boolean assignUserRole(int userId, String role);
⑩ public boolean activateUserAccount(int userId);
⑪ public boolean deactivateUserAccount(int userId);

Interface 2 (I2): Manage Point, Rewards

① public int getUserPointBalance(int userId);
② public boolean addPoints(int userId, int points, String reason);
③ public boolean deductPoints(int userId, int points, String reason);
④ public List<PointTransaction> getPointHistory(int userId);
⑤ public List<Reward> getRewardCatalog();
⑥ public Reward getRewardDetails(int rewardId);
⑦ public boolean redeemReward(int userId, int rewardId);
⑧ public boolean addRewardToCatalog(String name, String description, int pointCost);
⑨ public boolean updateReward(int rewardId, String name, String description, int pointCost);
⑩ public boolean deleteReward(int rewardId);
⑪ public List<RedemptionHistory> getRedemptionHistory(int userId);
⑫ public boolean validateRedemption(int userId, int rewardId);

List Description Diagram:

Interface 3 (I3): Manage Authentication

① public AuthToken authenticateUser(String username, String password);
② public boolean validateToken(String token);
③ public boolean refreshToken(String token);
④ public boolean revokeToken(String token);
⑤ public boolean changePassword(int userId, String oldPassword, String newPassword);
⑥ public boolean verifyUserRole(int userId, String requiredRole);
⑦ public List<String> getUserPermissions(int userId);
⑧ public boolean enableTwoFactorAuth(int userId);
⑨ public boolean verifyTwoFactorCode(int userId, String code);
⑩ public boolean logSecurityEvent(int userId, String eventType, String description);
⑪ public List<SecurityLog> getSecurityLogs(int userId, Date startDate, Date endDate);

Interface 4 (I4): Manage Inventory

① public InventoryStatus getMachineInventoryLevel(int machineId);
② public boolean recordItemDeposit(int machineId, String itemType, int quantity);
③ public boolean updateInventoryLevel(int machineId, int currentLevel, int capacity);
④ public List<Machine> getAllMachines();
⑤ public Machine getMachineDetails(int machineId);
⑥ public boolean setInventoryAlert(int machineId, int thresholdPercentage);
⑦ public List<InventoryAlert> getInventoryAlerts();
⑧ public boolean recordCollection(int machineId, int collectedQuantity, Date collectionDate);
⑨ public List<CollectionHistory> getCollectionHistory(int machineId);
⑩ public boolean updateMachineStatus(int machineId, String status);
⑪ public InventoryReport generateInventoryReport(Date startDate, Date endDate);
⑫ public boolean syncWithDigitalTwin(int machineId, InventoryData data);

Interface 5 (I5): Manage Notification, Analytics

Notification Functions

① public boolean sendNotification(int userId, String message, String channel);
② public boolean sendTransactionNotification(int userId, Transaction transaction);
③ public boolean sendPointUpdateNotification(int userId, int points, String reason);
④ public boolean sendRewardRedemptionNotification(int userId, Reward reward);
⑤ public boolean sendSystemAnnouncement(String message, List<Integer> userIds);
⑥ public boolean sendInventoryAlert(int adminId, int machineId, String alertType);
⑦ public List<Notification> getNotificationHistory(int userId);
⑧ public boolean updateNotificationPreferences(int userId, NotificationPreferences prefs);

Analytics Functions

⑨ public AnalyticsReport generateUserActivityReport(Date startDate, Date endDate);
⑩ public AnalyticsReport generateTransactionReport(Date startDate, Date endDate);
⑪ public AnalyticsReport generateRecyclingTrendsReport(Date startDate, Date endDate);
⑫ public Dashboard getAdminDashboard();
⑬ public Map<String, Integer> getUserStatistics();
⑭ public Map<String, Double> getPointDistribution();
⑮ public List<TopUser> getTopRecyclers(int limit);
⑯ public FinancialReport generateFinancialReport(Date startDate, Date endDate);
⑰ public boolean exportReportToFile(Report report, String format);

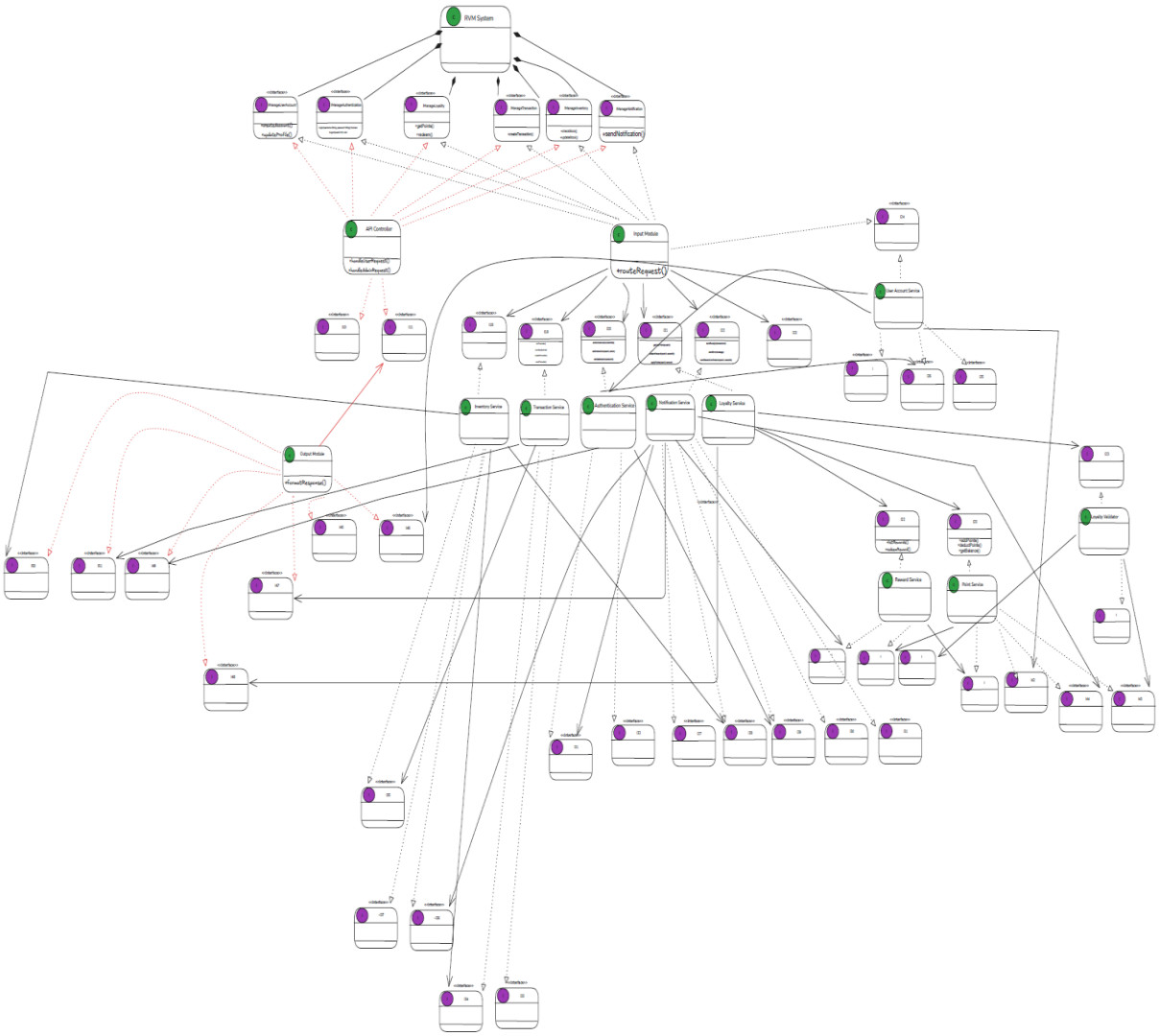
Interface 6 (I6): Manage Transaction

① public Transaction createGuestTransaction(int machineId, String itemType, int quantity);
② public Transaction createUserTransaction(int userId, int machineId, String itemType, int quantity);
③ public boolean processTransaction(Transaction transaction);
④ public Transaction getTransactionDetails(int transactionId);
⑤ public List<Transaction> getTransactionHistory(int userId);
⑥ public List<Transaction> getAllTransactions(Date startDate, Date endDate);
⑦ public boolean validateTransaction(Transaction transaction);
⑧ public boolean cancelTransaction(int transactionId, String reason);
⑨ public boolean recordRedemptionTransaction(int userId, int rewardId, int pointsUsed);
⑩ public TransactionSummary getTransactionSummary(int userId);

Internal Interface:

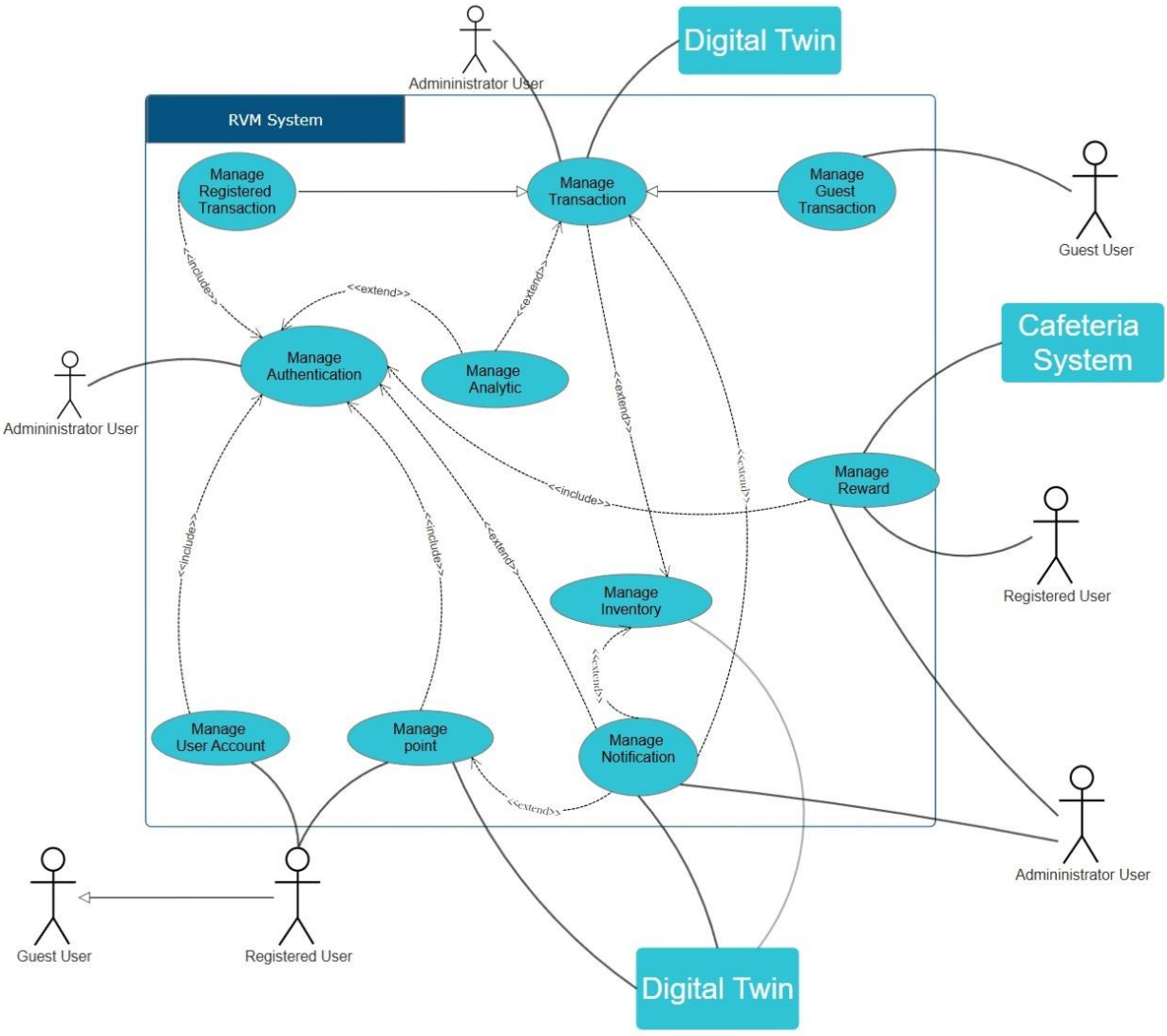
Interface Name	Type	Source Component	Target Component	Protocol/Method	Description	Data Format
IAuthManager	Service Interface	API Controller	Authentication Service	Method Calls	Manages user authentication and authorization	JSON Objects
ISessionManager	Service Interface	Authentication Service	DataLayer	Method Calls	Handles session creation, validation, and termination	Session Objects
ITokenOrchestrator	Service Interface	Authentication Service	API Controller	JWT/OAuth	Generates and validates authentication tokens	JWT Tokens
IUserManager	Service Interface	API Controller	User Account Service	Method Calls	Core user account CRUD operations	User DTOs
IProfileManager	Service Interface	User Account Service	DataLayer	Method Calls	Manages user profile information	Profile Objects
IRoleAssignor	Service Interface	User Account Service	Authentication Service	Method Calls	Assigns and manages user roles and permissions	Role Objects
IManagePoint	Service Interface	API Controller	Point Service	Method Calls	Handles point accumulation, balance, and transactions	Point Objects
IManageReward	Service Interface	API Controller	Reward Service	Method Calls	Manages reward catalog and redemption process	Reward Objects
IManageTransaction	Service Interface	API Controller	Transaction Service	Method Calls	Processes and records all system transactions	Transaction DTOs
IInventoryManager	Service Interface	API Controller	Inventory Service	Method Calls	Manages overall inventory tracking and updates	Inventory Objects
IMachineManager	Service Interface	Inventory Service	DataLayer	Method Calls	Manages RVM machine configurations and status	Machine Objects
IDigitalTwinSync	Service Interface	Inventory Service	External RVM Devices	IoT Protocol	Synchronizes virtual and physical RVM data	IoT Messages
IAlertManager	Service Interface	Inventory Service	Notification Service	Event-Driven	Triggers alerts for low inventory and issues	Alert Objects
IReportGenerator	Service Interface	Inventory Service	Analytics Service	Method Calls	Generates inventory and system reports	Report Objects
IManageNotification	Service Interface	Multiple Services	Notification Service	Event Queue	Sends push notifications and alerts to users	Notification Objects
IAnalyticsService	Service Interface	API Controller	Analytics Service	Method Calls	Processes and analyzes system data	Analytics DTOs
IUserData	Data Interface	Multiple Services	DataLayer	ORM/SQL	Provides data persistence for user information	Database Records
ILoyaltyData	Data Interface	Point/Reward Services	DataLayer	ORM/SQL	Provides data persistence for loyalty and points	Database Records

6- Class Diagram:


RVM System
ManageAuthentication
ManageLoyalty
ManageNotification
API Controller
Input Module
+routeRequest()
User Account Service
Inventory Service
Transaction Service
Authentication Service
Notification Service
Loyalty Service
Output Module
+formatResponse()
Reward Service
Point Service
Loyalty Validator

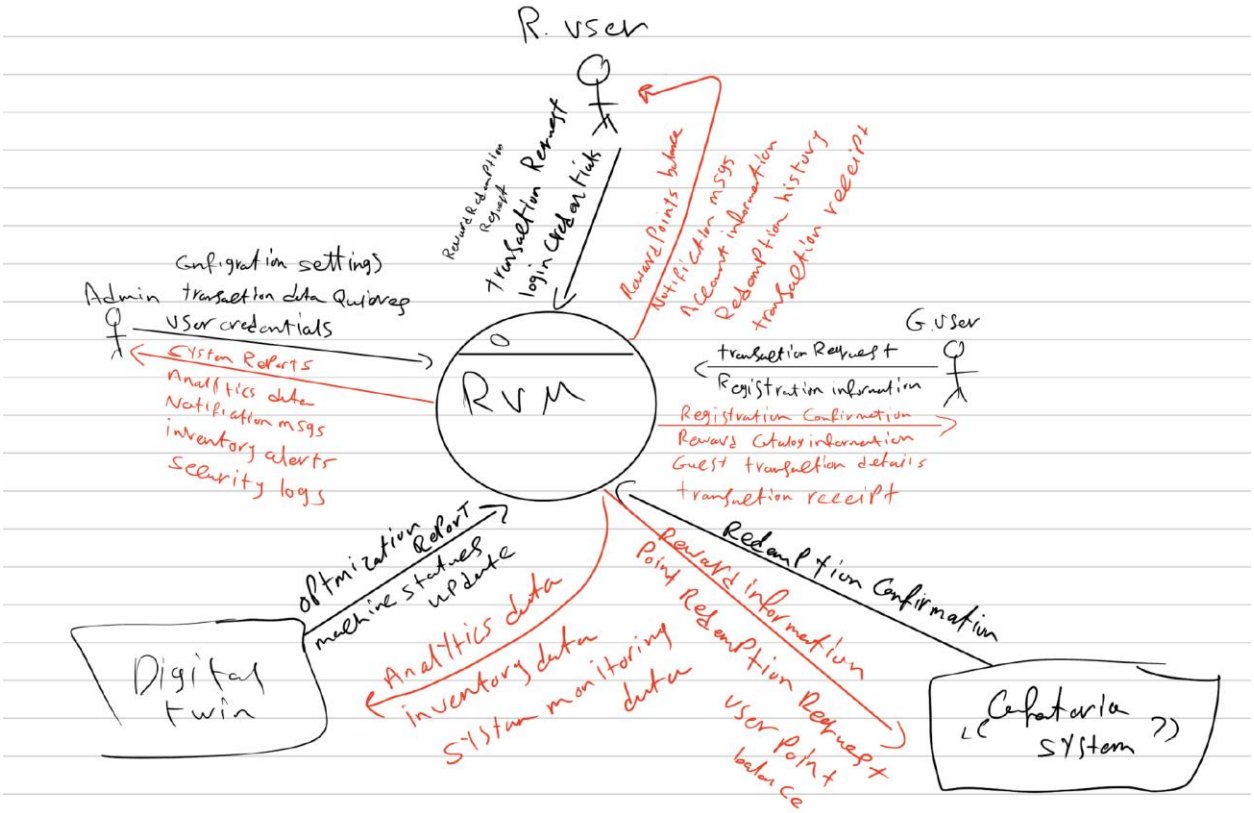
6- Diagrams :

Usecase Diagram:

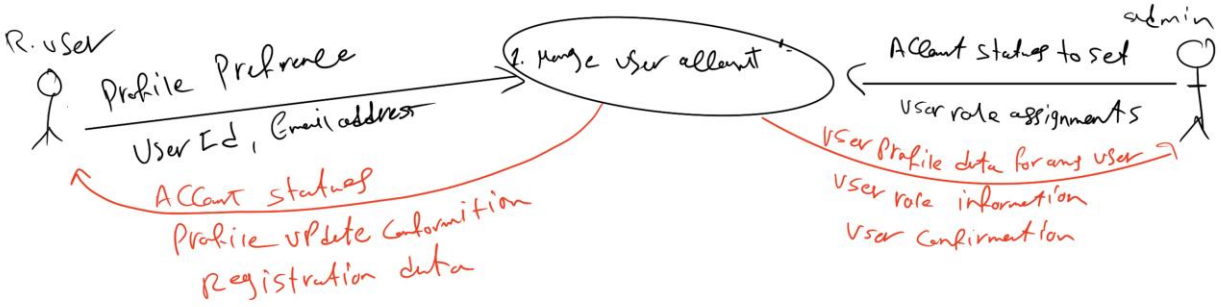


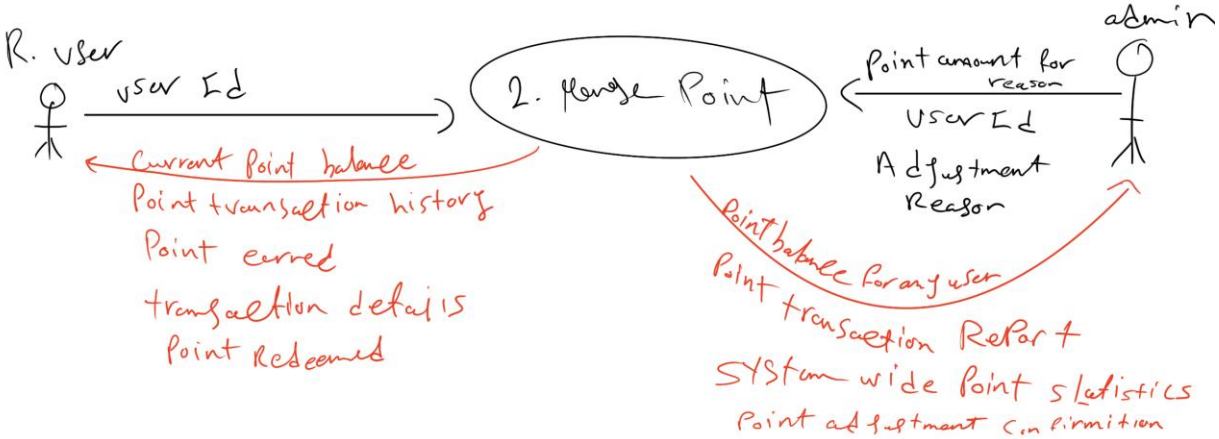
DFD Level 0:

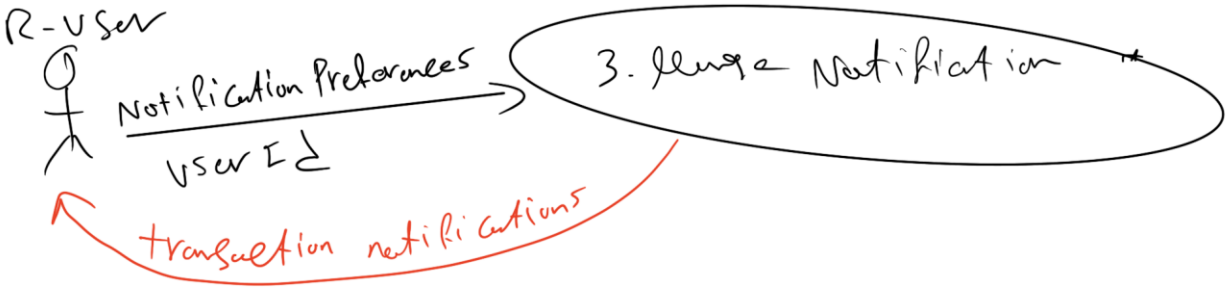

Guest User
transaction request
Registration info
transaction receipt
Registration Confirmation
Reward information
Configuration setting
transaction data
User credentials
admin
System Report
Analytics data
Notifications
RVM
Optmization Reports
Digital twin
Analytics data
System monitoring data
login credentials
transaction Request
Reward redemption
R. User
transaction receipt
Reward Points
Notifications
Account information
Redemption Confirmation
Reward information
Point Redemption
Cafeteria
<< System >>


R. User
Reward redemption Request
transaction Request
login credentials
Reward Points balance
Notification msgs
Account information
Redemption history
transaction receipt
Configuration setting
Admin
transaction data Queries
User credentials
System Reports
Analytics data
Notification msgs
inventory alerts
security logs
RVM
G User
transaction Request
Registration information
Registration Confirmation
Rewards Catalog information
Guest transaction details
transaction receipt
Optmization Report
machine status updates
Digital twin
Analytics data
inventory data
System monitoring data
Redemption Confirmation
Reward information
Point Redemption Request
User point balance
Cafeteria
<< System >>

DFD Level 1







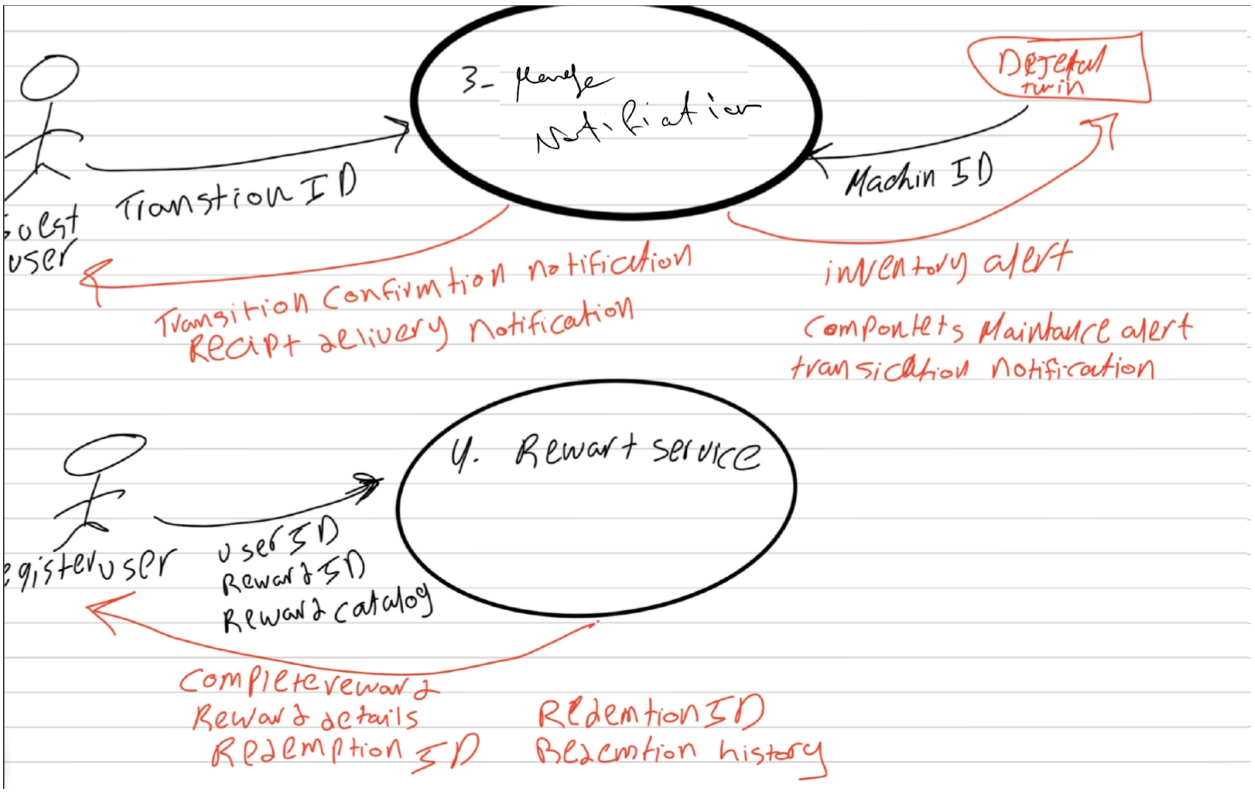
Notification history

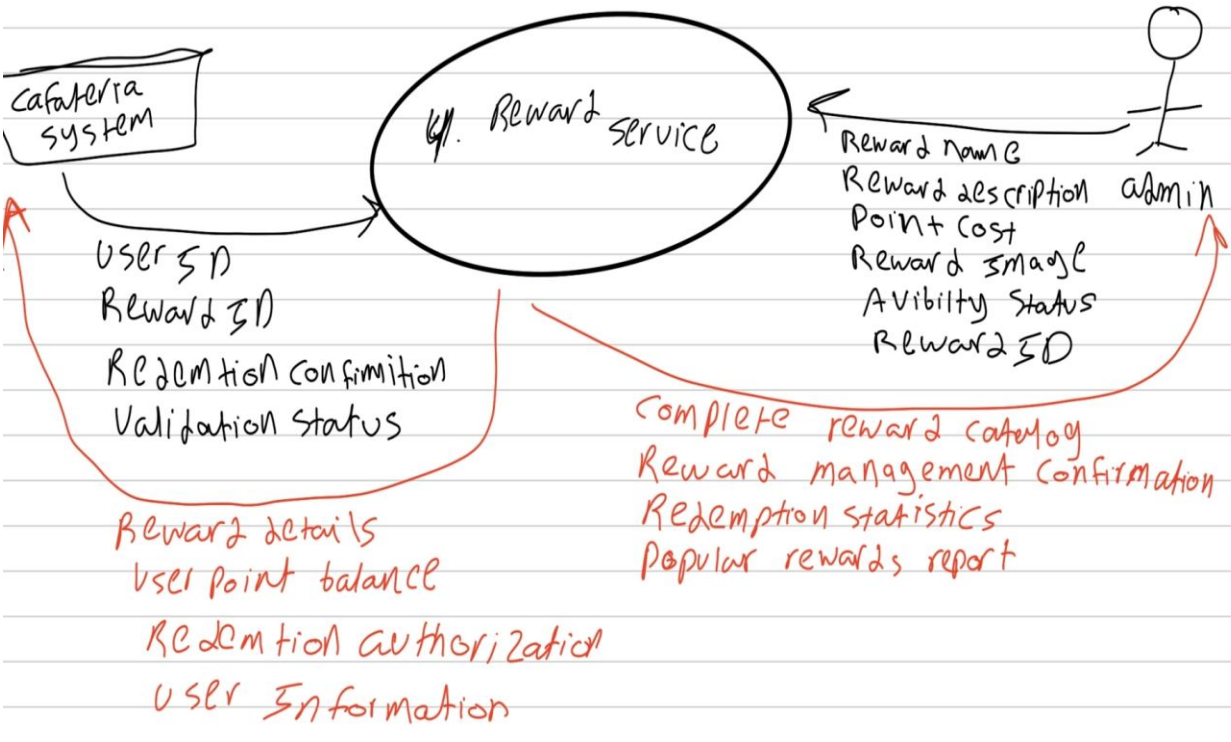
Promotional messages

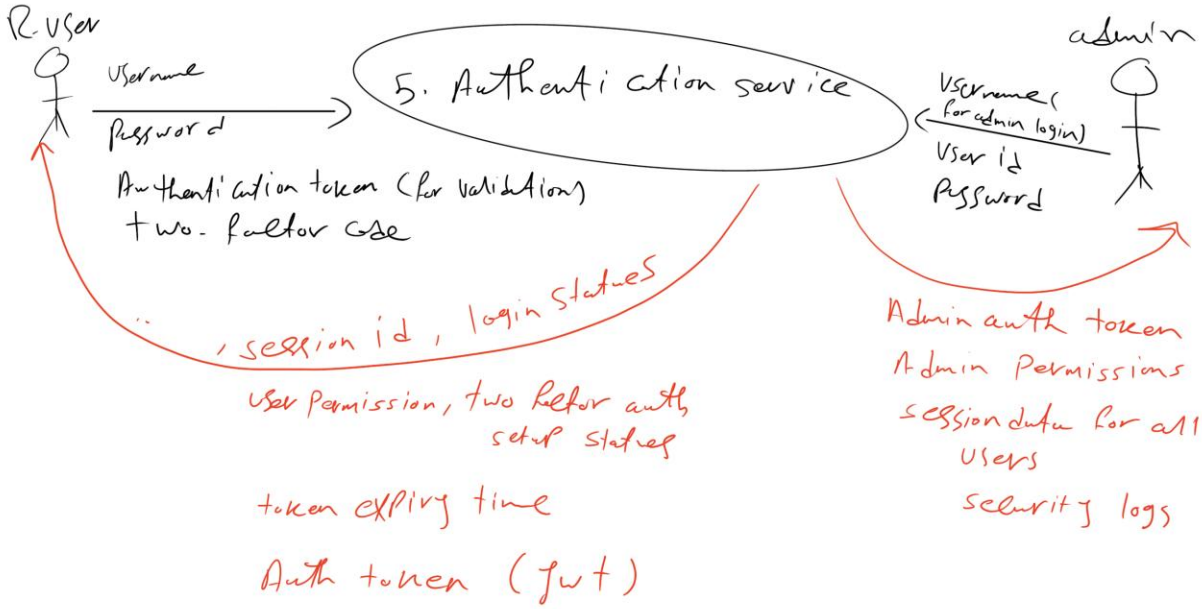
Point update Notifications

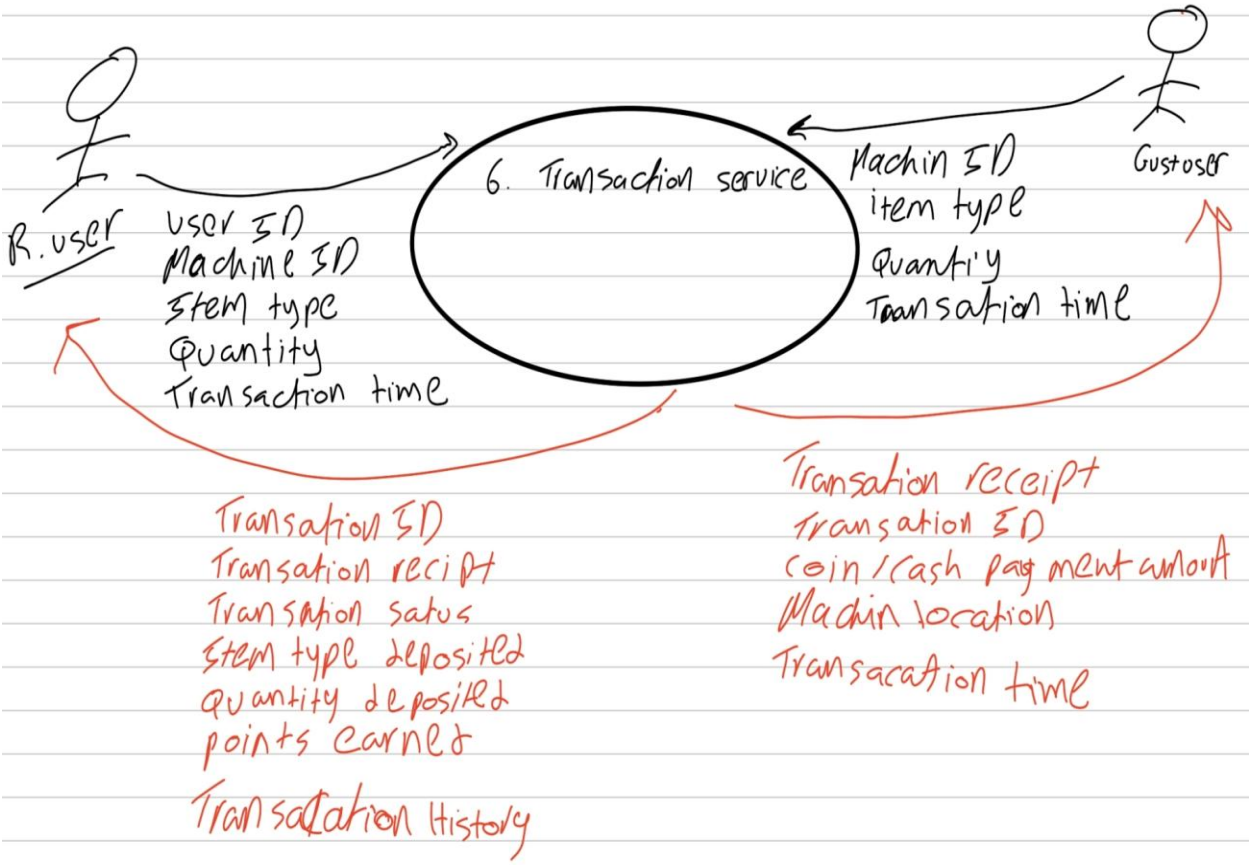
Reward Redemption Notifications

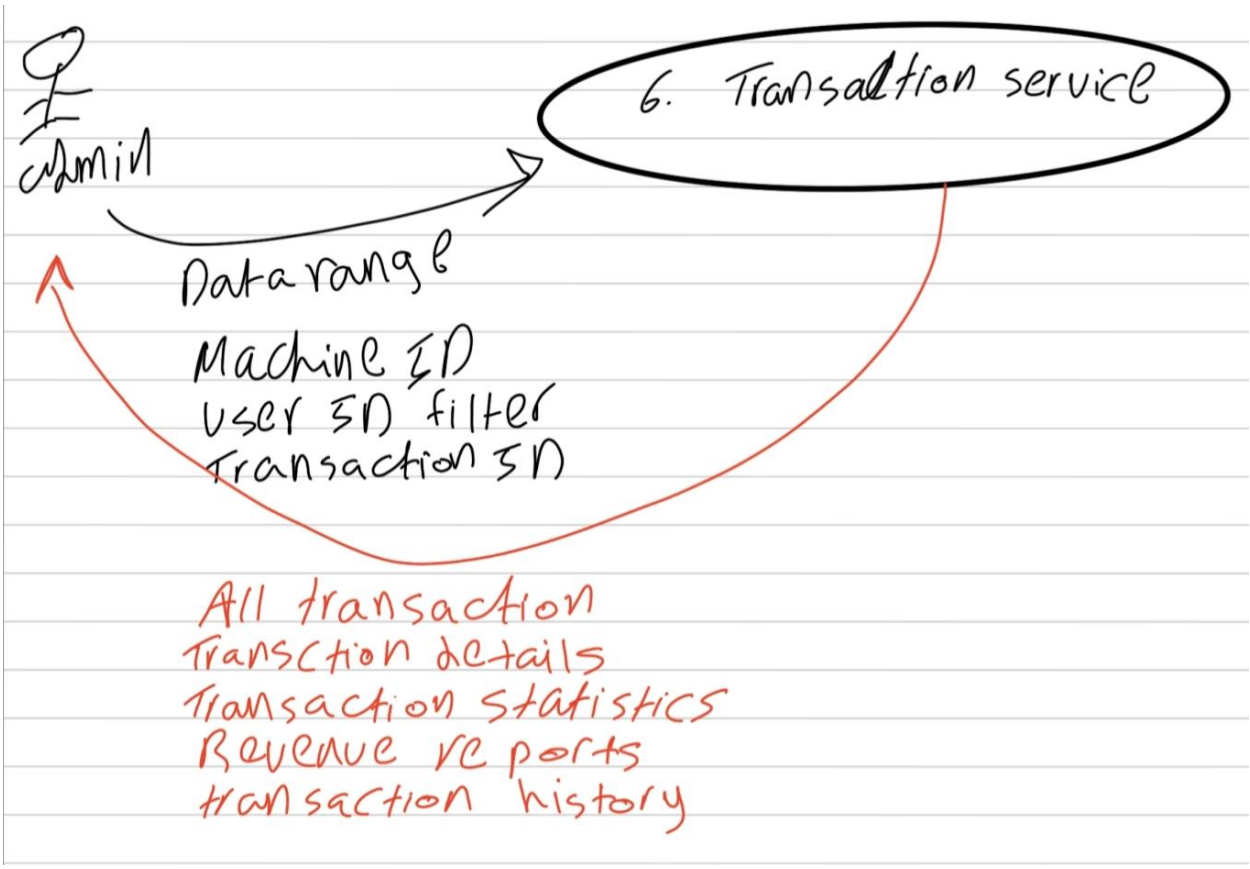
Notification Read status

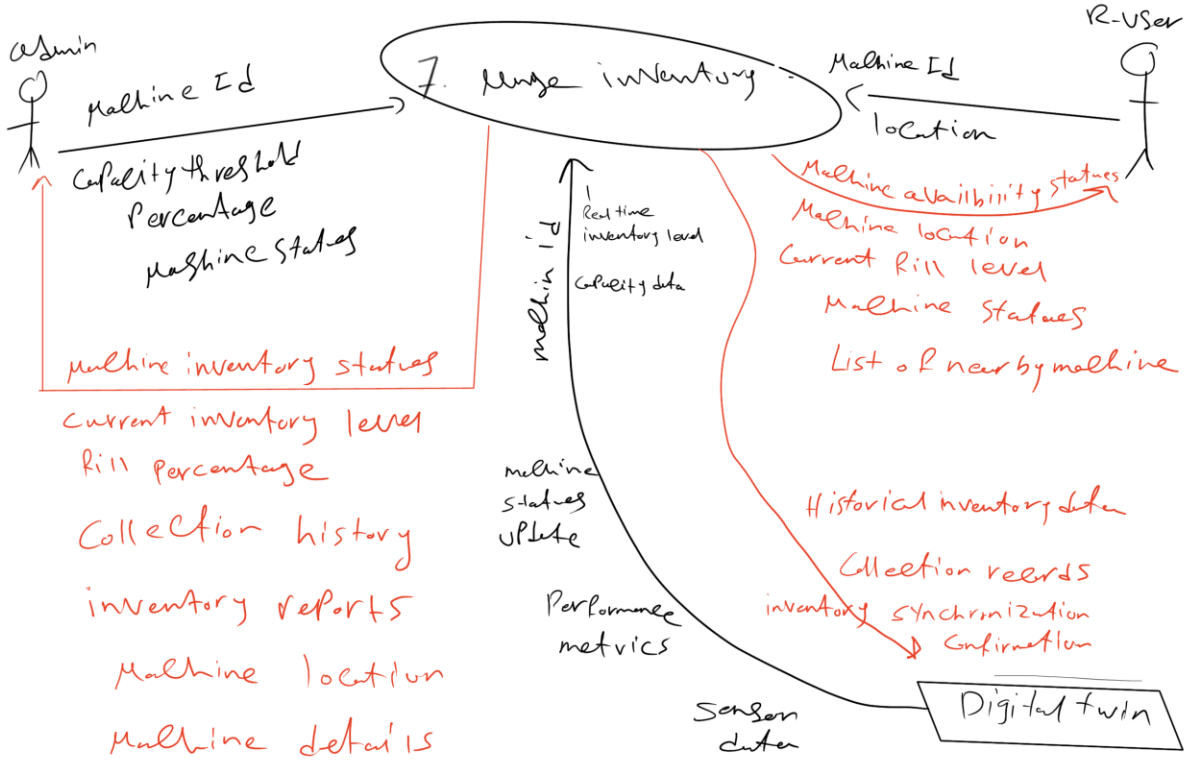

3- Manage Notification
Digital twin
Transtion ID
Machin ID
Guest user
Transition confirmation notification
Recipt delivery notification
inventory alert
Componets Maintaurce alert
transication notification
4. Rewart service
egisteruser
user ID
Reward ID
Reward catalog
Completereward
Reward details
Redemption ID
Redemption ID
Redemption history

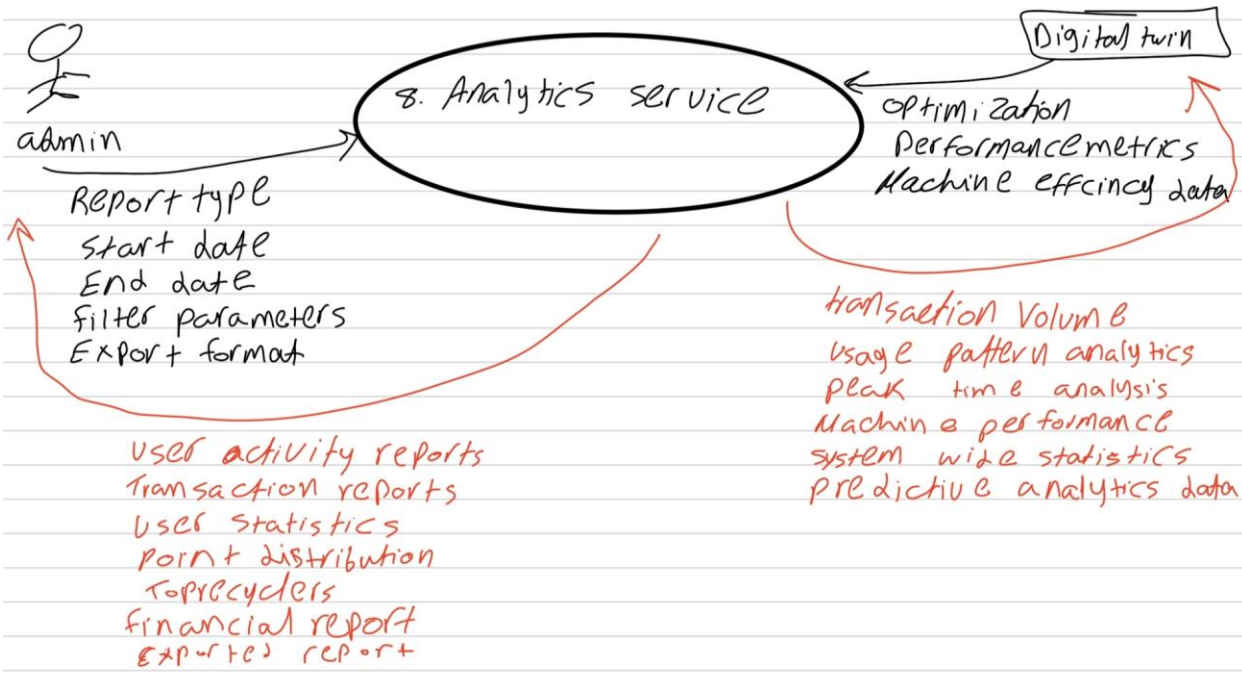

cafateria system
4. Reward service
admin
Reward name
Reward description
Point cost
Reward image
Avibilty Status
Reward ID
user ID
Reward ID
Redemtion confimition
Validation status
Complete reward catelog
Reward management confirmation
Redemption statistics
Popular rewards report
Reward details
User point balance
Redemtion authorization
User Information

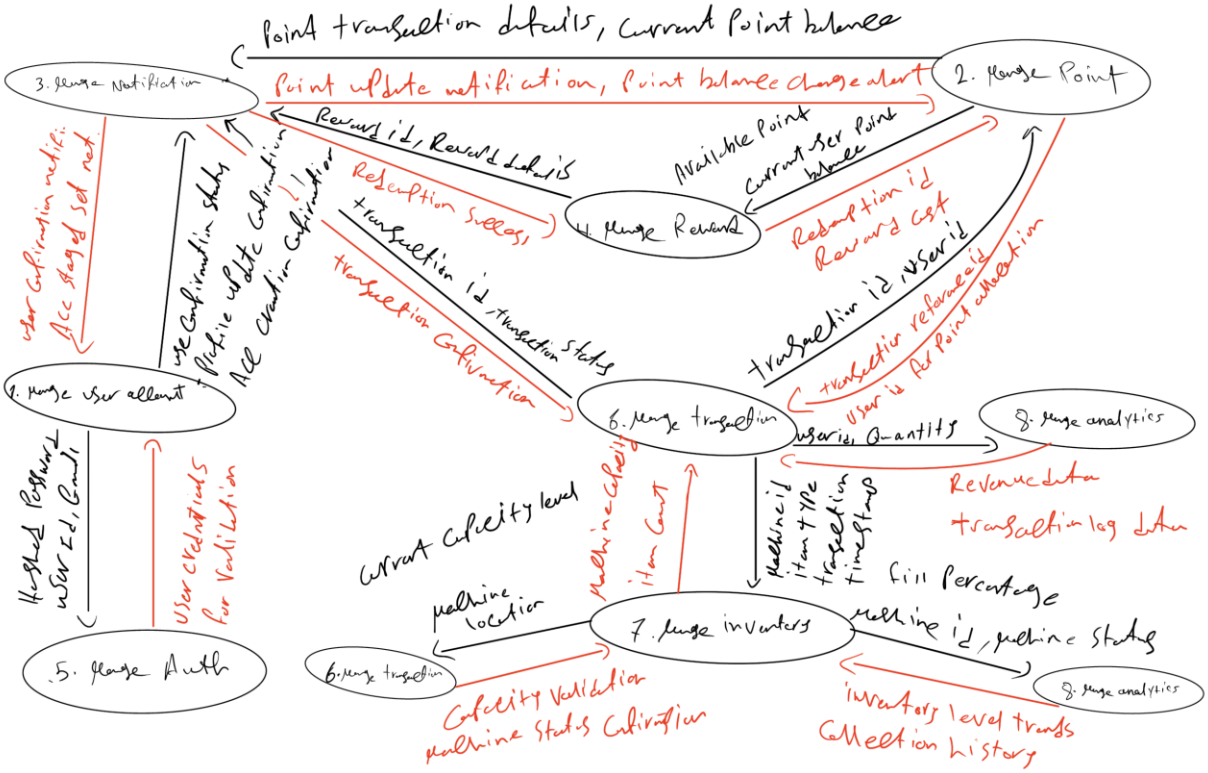

R.user
Username
Password
5. Authentication service
admin
Username (for admin login)
User id
Password
Authentication token (for validation)
two-factor code
, session id, login statues
User Permission, two factor auth, setup status
token expiry time
Auth token (jwt)
Admin auth token
Admin Permissions
session data for all users
security logs

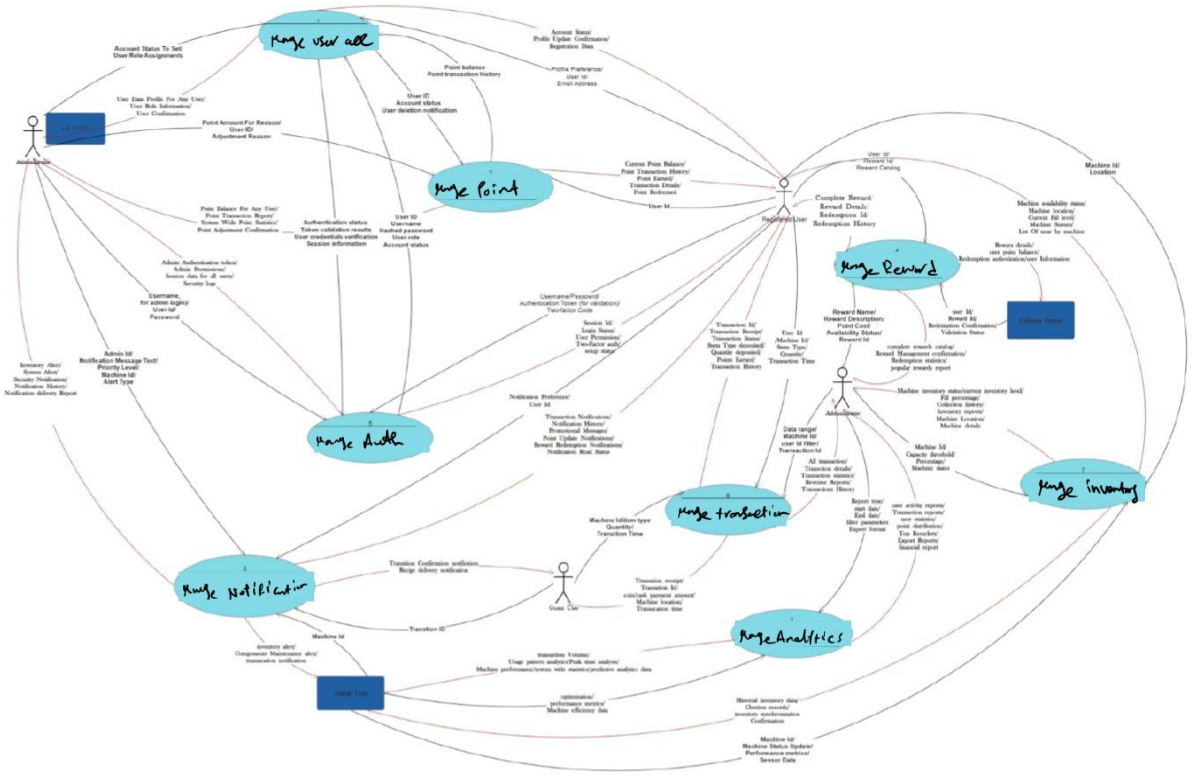

R.user
User ID
Machine ID
Item type
Quantity
Transaction time
6. Transaction service
Machin ID
item type
Quantity
Transaction time
Custoser
Transation ID
Transation reciept
Transation satus
Item type deposited
quantity deposited
points earned
Transaction History
Transation receipt
transation ID
coin/cash payment amount
Machin location
Transacation time


admin
6. Transaction service
Data range
Machine ID
user ID filter
Transaction ID
All transaction
Transction details
Transaction statistics
Revenue reports
transaction history


admin
Machine Id
7. Merge inventory
Machine Id
location
R-User
Capacity threshold
Percentage
Machine status
Machine availability statuses
Machine location
Current fill level
Machine Statuses
List of near by machine
Machine inventory status
Current inventory level
fill percentage
Collection history
inventory reports
Machine location
Machine details
Machine Id
Real time inventory level
Capacity data
Machine status update
Performance metrics
Historical inventory data
Collection records
inventory synchronization
Confirmation
Sensor data
Digital twin

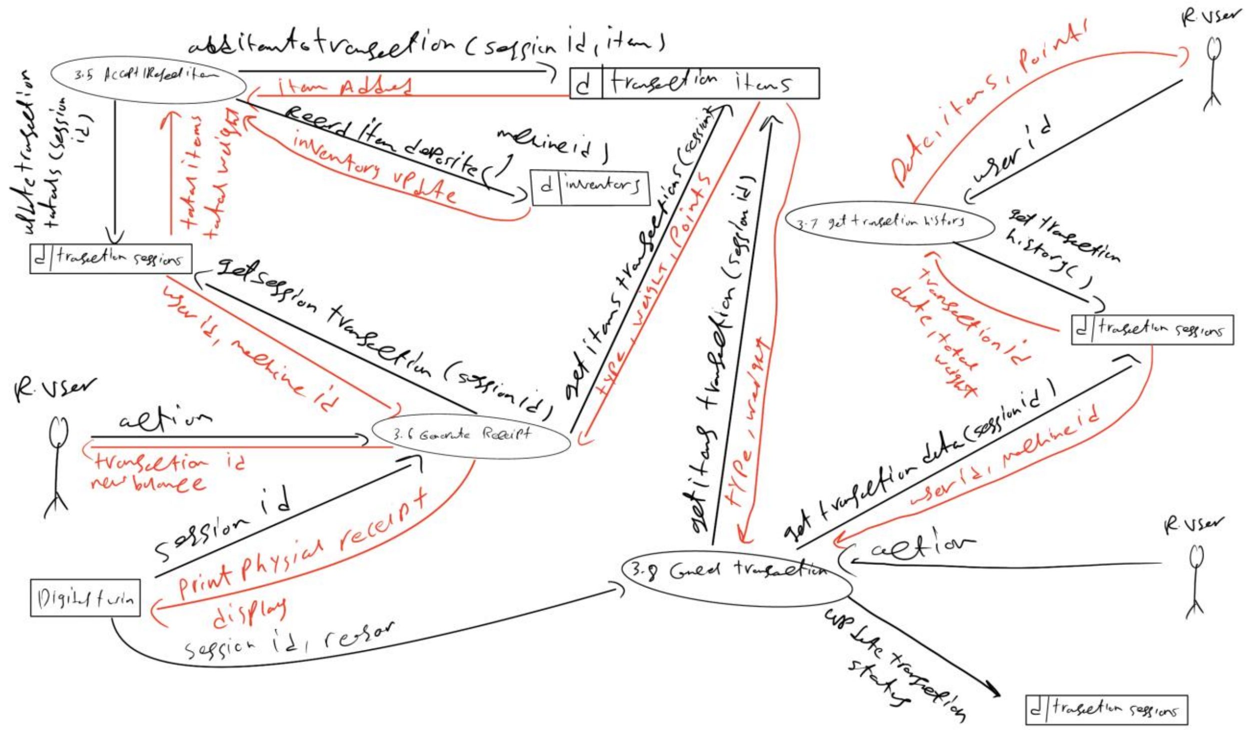

admin
8. Analytics service
Digital twin
Report type
start date
End date
filter parameters
Export format
optimization
Performance metrics
Machine effcincy data
transaction Volume
Usage pattern analytics
peak time analysis
Machine performance
system wide statistics
predictive analytics data
User activity reports
Transaction reports
User statistics
point distribution
Top recyclers
financial report
exported report

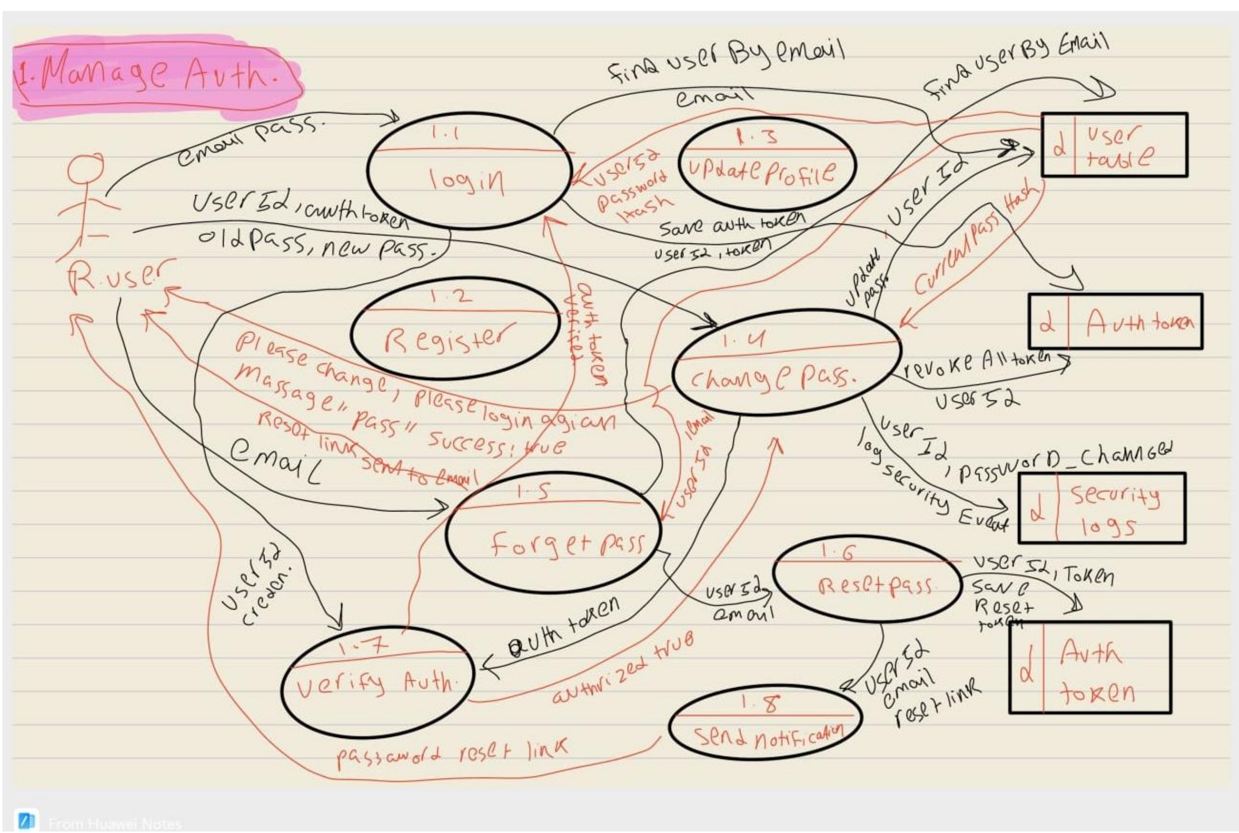

Point transaction details, current Point balance
3. Manage Notification
Point update notification, Point balance change alert
2. Manage Point
Reward id, Reward details
Available Point
Current User Point balance
4. Manage Rewards
Redemption id
Reward cost
Redemption Success,
transaction id, transaction status
transaction id, user id
transaction reference id
User is for Point allocation
transaction Confirmation
User confirmation status
Profile update Confirmation
Acc creation Confirmation
User confirmation notifi: Acc staged set not
1. Manage User account
6. Manage transaction
8. Manage analytics
user id, Quantity
Revenue data
transaction log data
current Capacity level
machine Capacity
item Count
machine id item type transaction timestamp
fill Percentage
machine location
7. Manage inventory
machine id, machine status
Highest Reward user id, Email
User credentials for Verification
5. Manage Auth
6. Manage transaction
8. Manage analytics
Capacity Validation
machine status Confirmation
inventory level trends
Collection history

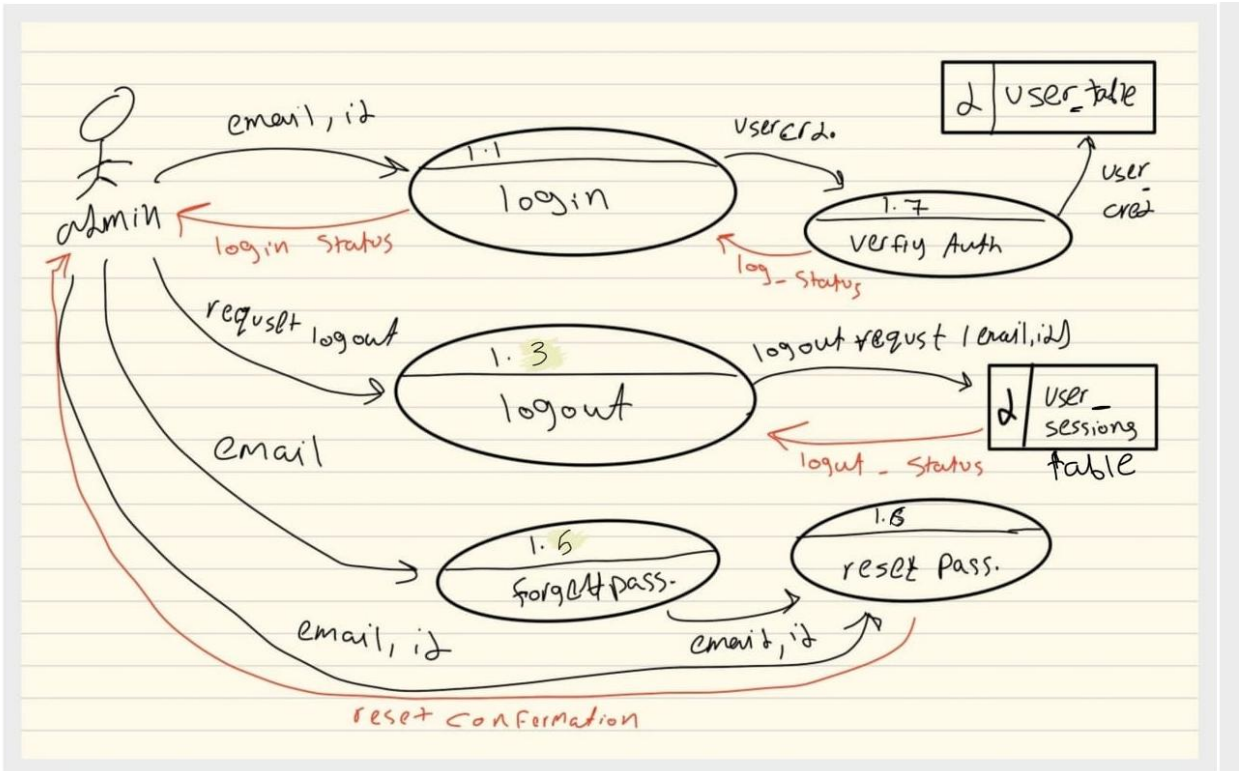

Manage user acc
Manage Point
Manage Reward
Manage Auth
Manage Notification
Manage transaction
Manage inventory
Manage Analytics

DFD Level 2 :

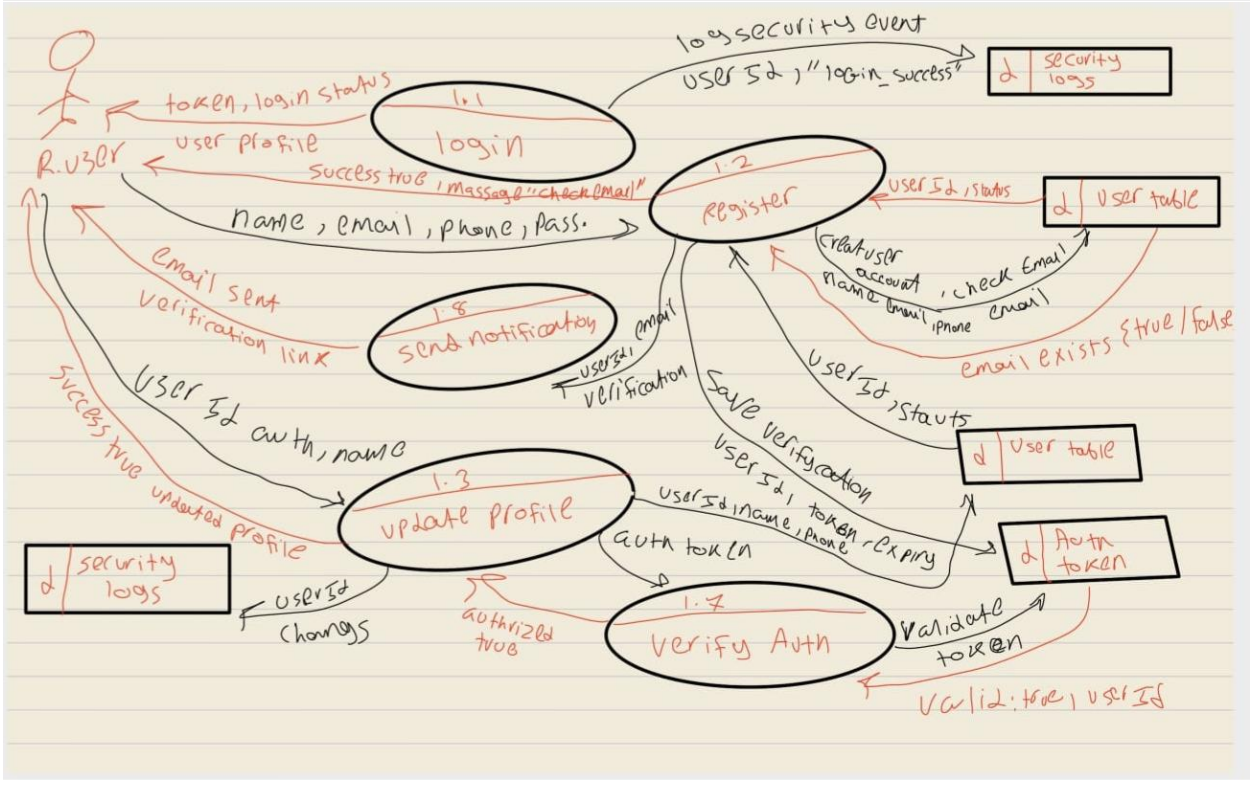

DFD level 2 : 3. Manage transaction
R.User
QR code
session id
username
machine id
Digital twin
Guest user
action
session id, user type
3.1 Start transaction
Find User By QR code (User code)
user id, username, status
d | User table
initialize transaction (user id, machine id)
session id, status
d | transaction sessions
3.5 Accept/Reject item
Status, total items
insert item on into RVM
3.2 Validate item
weight
material type
Point, Valid
d | material Factor
Get Material Factor
3.4 Calc item Points
Plastic, Glass
session id
weight
item type
session id, weight
Display, items


3.5 Accept Refeed item
d transaction items
d inventory
d transaction sessions
3.6 Generate Receipt
3.7 get transaction history
3.8 Cancel transaction
Digital twin
R.User
action
session id
print physical receipt
display
session id, reason
get session transaction (session id)
user id
get transaction history()
d transaction sessions
get items by transaction (session id)
type, weight
update transaction status
item Added
inventory update

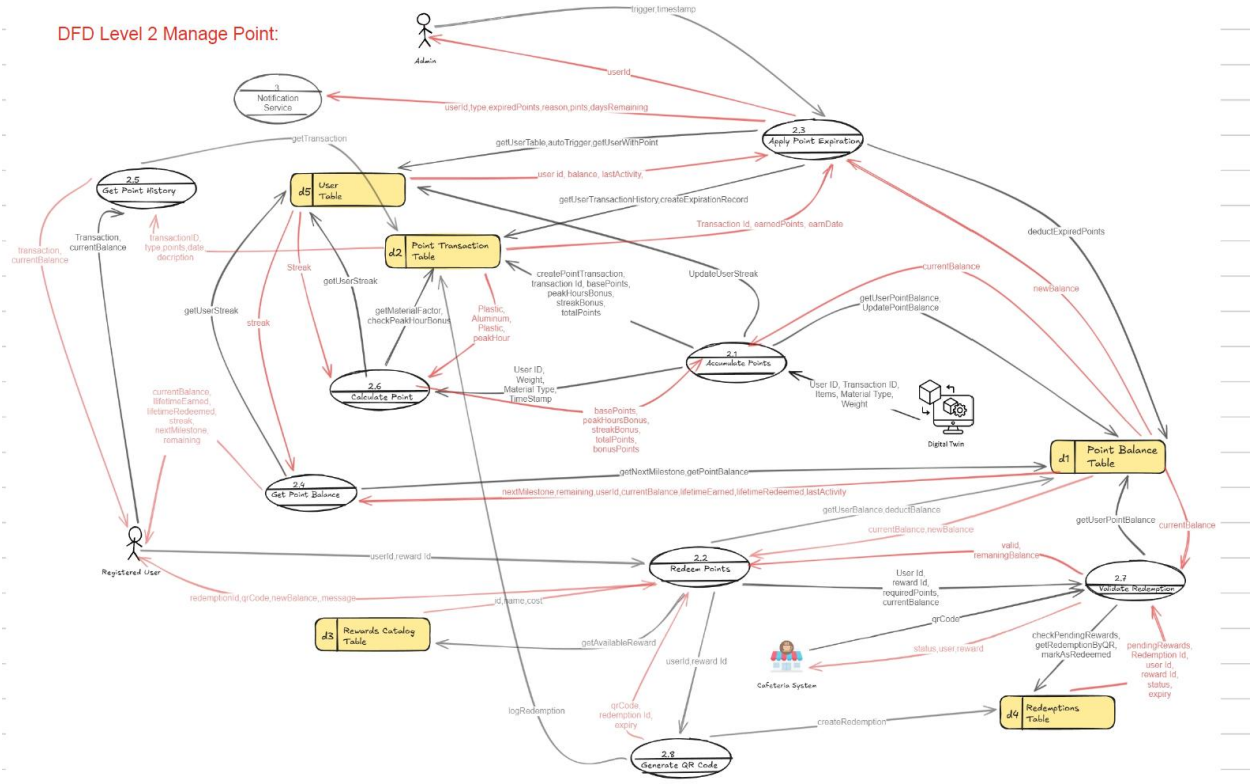

1. Manage Auth.
1.1 login
1.2 Register
1.3 Update Profile
1.4 change Pass.
1.5 Forget pass
1.6 Reset pass.
1.7 Verify Auth.
1.8 Send notification
R.user
d User table
d Auth token
d Security logs
email pass.
find user By email
email
save auth token
userId, token
oldpass, new pass.
userId, password_changed
log security Event
revoke All token
Please change, please login again
email
auth token
authrized true
password reset link
userId email reset link
userId, Token
Save Reset token
From Huawei Notes

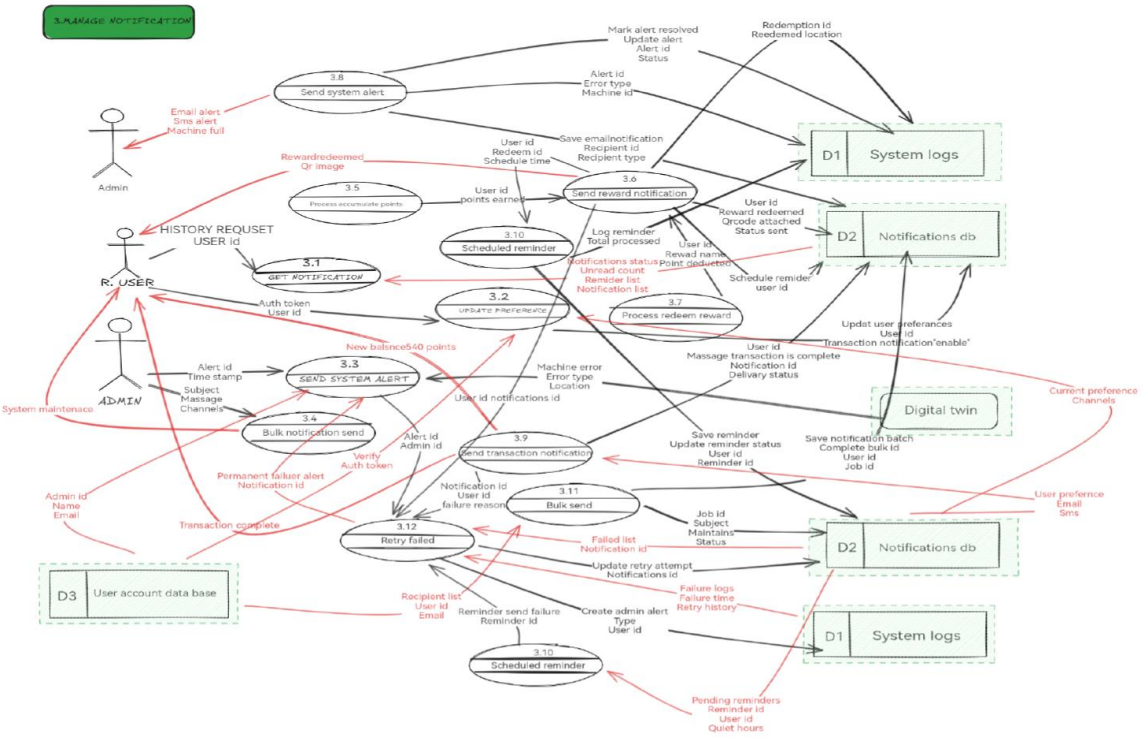


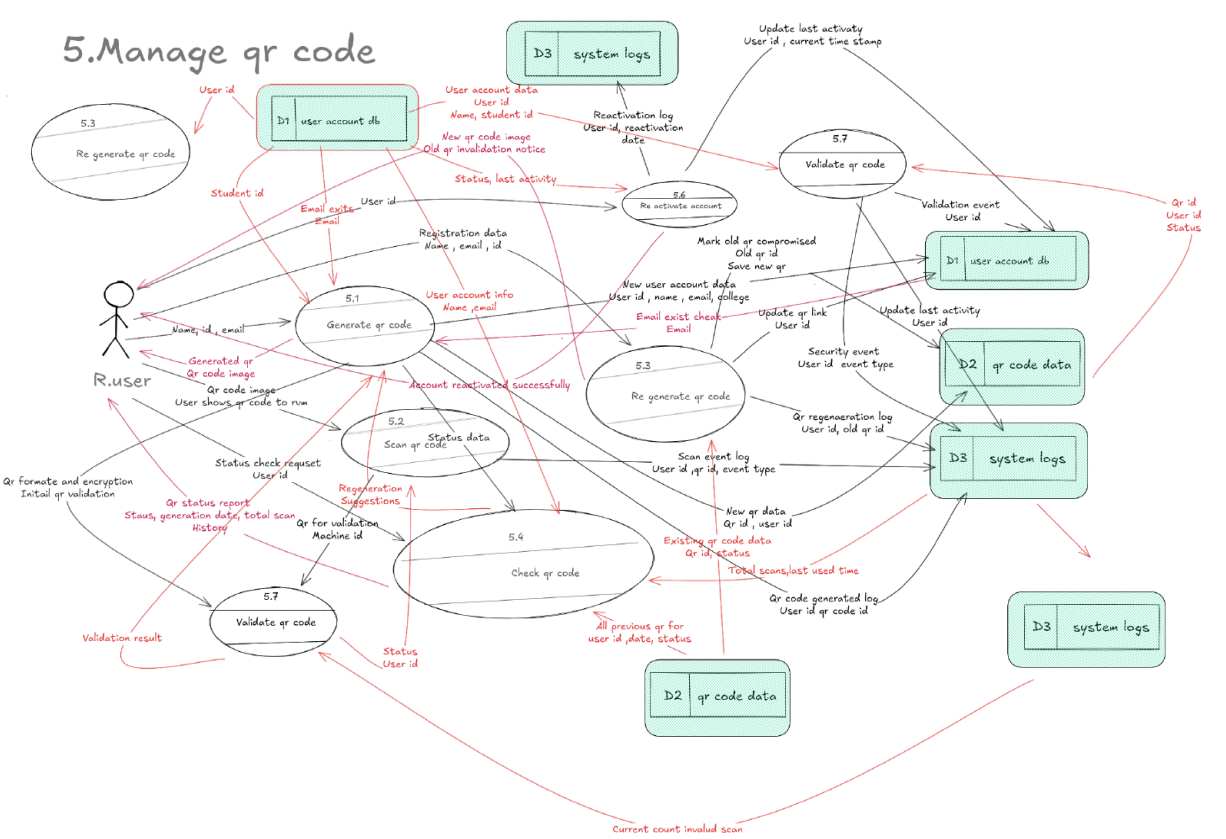
From Huawei Notes



DFD Level 2 Manage Point:






5.Manage qr code
D3 system logs
Update last activaty
User id , current time stamp
User id
D1 user account db
User account data
User id
Name, student id
5.3
Re generate qr code
Reactivation log
User id, reactivation date
5.7
Validate qr code
New qr code image
Old qr invalidation notice
Status, last activity
Student id
5.6
Re activate account
Validation event
User id
Qr id
User id
Status
Email exits
Email
User id
Registration data
Name , email , id
Mark old qr compromised
Old qr id
Save new qr
D1 user account db
New user account data
User id , name , email, college
User account info
Name ,email
5.1
Generate qr code
Email exist check
Email
Update qr link
User id
Update last activity
User id
Name, id , email
5.3
Re generate qr code
Security event
User id event type
D2 qr code data
Generated qr
Qr code image
Account reactivated successfully
R.user
Qr code image
User shows qr code to rvm
5.2
Scan qr code
Status data
Qr regenaeration log
User id, old qr id
Scan event log
User id ,qr id, event type
D3 system logs
Status check requset
User id
Qr formate and encryption
Initail qr validation
Regeneration
Suggestions
Qr status report
Staus, generation date, total scan
History
Qr for validation
Machine id
New qr data
Qr id , user id
Existing qr code data
Qr id, status
5.4
Check qr code
Total scans,last used time
Qr code generated log
User id qr code id
5.7
Validate qr code
All previous qr for
user id ,date, status
D3 system logs
Validation result
Status
User id
D2 qr code data
Current count invalud scan

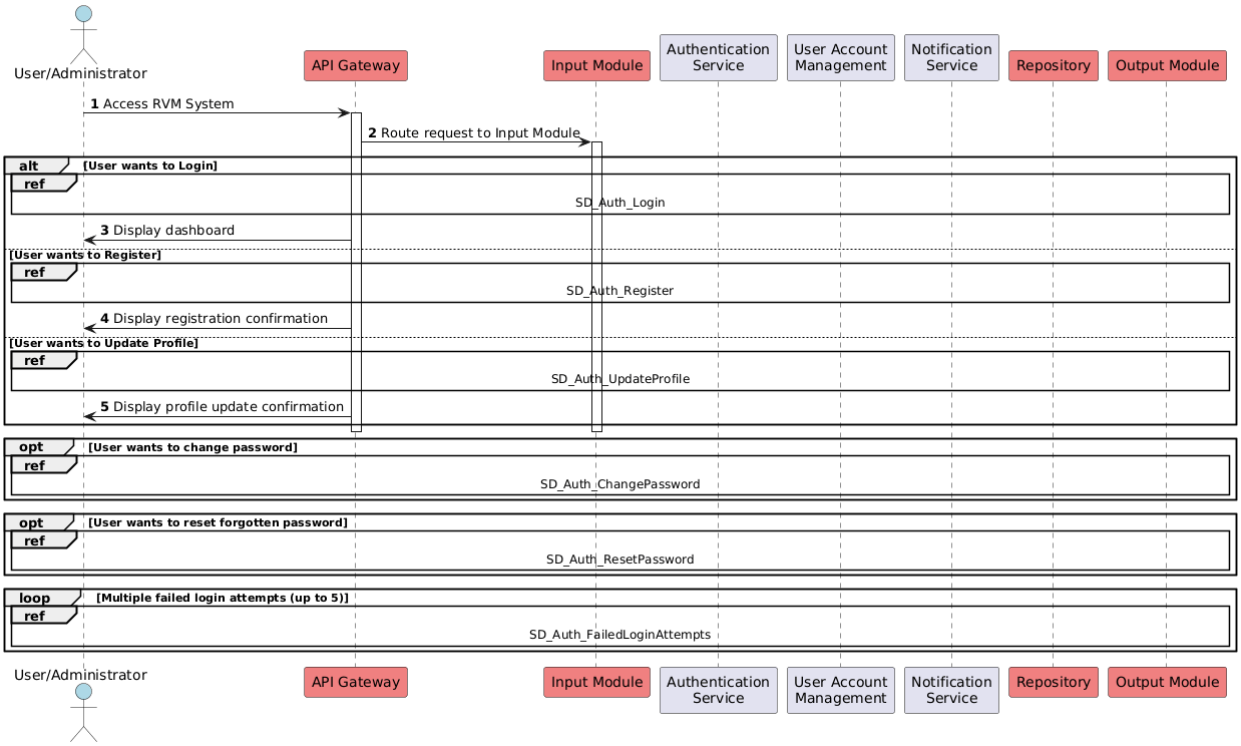
7- Tracibility :

Mapping Modeling Matrix: Use Cases to Design Entities

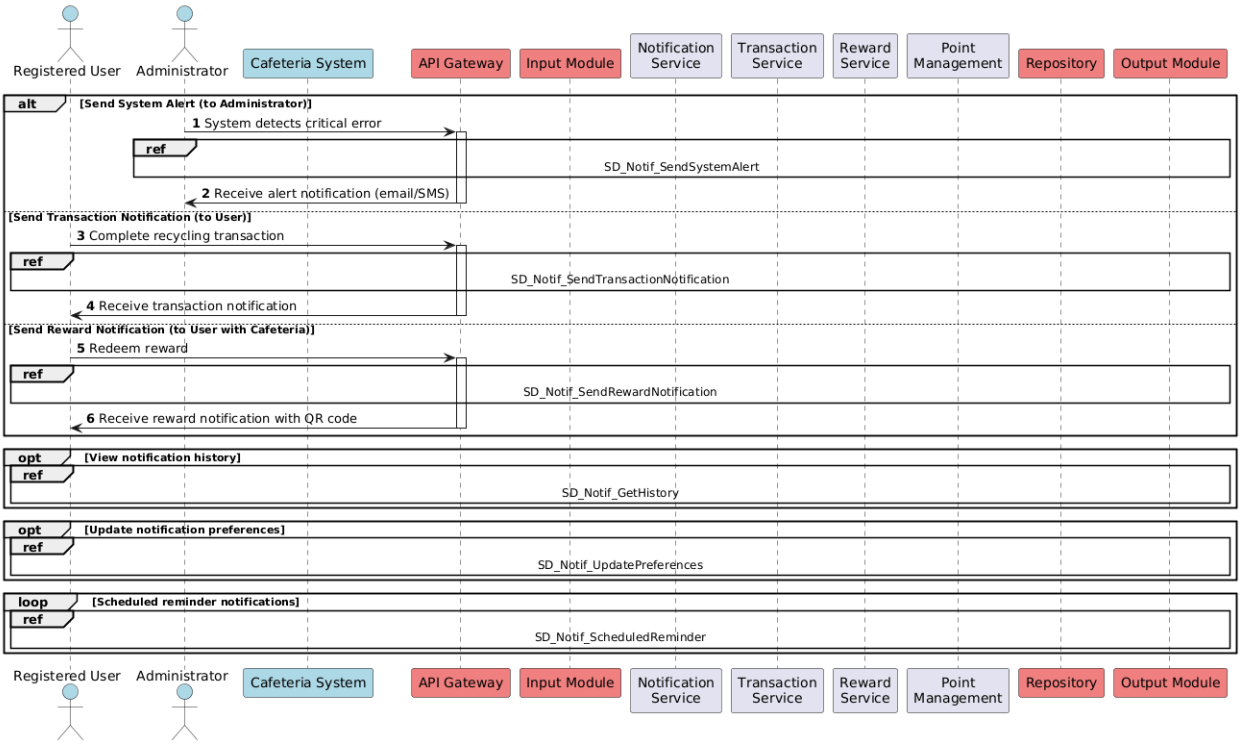
Use Case \ Design Entity	UserAccount	Authentication	Point	Reward	Transaction	Inventory	Machine	Notification	Analytics	Admin
UC_ManageAuth	✓	✓						✓		✓
UC_ManageAccount	✓	✓						✓		✓
UC_ManagePoint	✓		✓					✓	✓	
UC_ManageReward	✓		✓	✓	✓			✓	✓	
UC_ManageTransaction	✓		✓		✓	✓		✓	✓	
UC_ManageInventory						✓	✓	✓	✓	✓
UC_ManageMachine						✓	✓	✓		✓
UC_ManageNotification	✓							✓		✓
UC_ManageAnalytics	✓		✓	✓	✓	✓			✓	✓
UC_ManageReport					✓	✓			✓	✓
UC_ManageAlert						✓	✓	✓		✓
UC_DigitalTwinSync						✓	✓		✓	
UC_ManageRole	✓	✓								✓
UC_ManageSession	✓	✓								

8- Sequence Diagram :

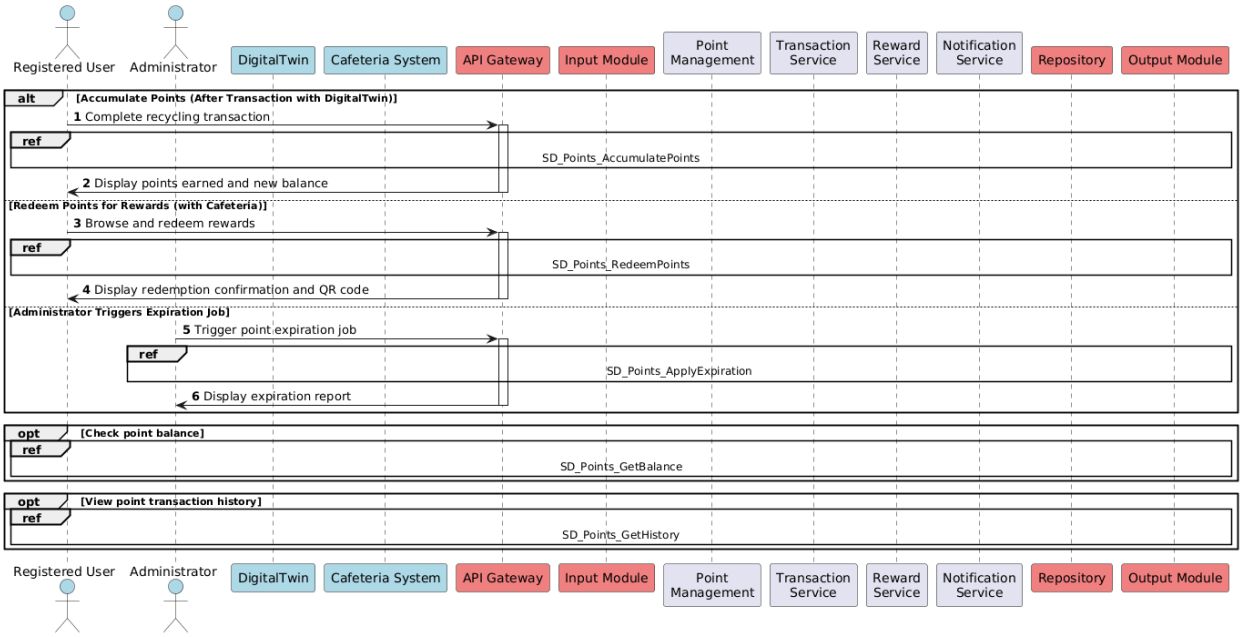
USE CASE: MANAGE AUTHENTICATION - Main Flow (with References)



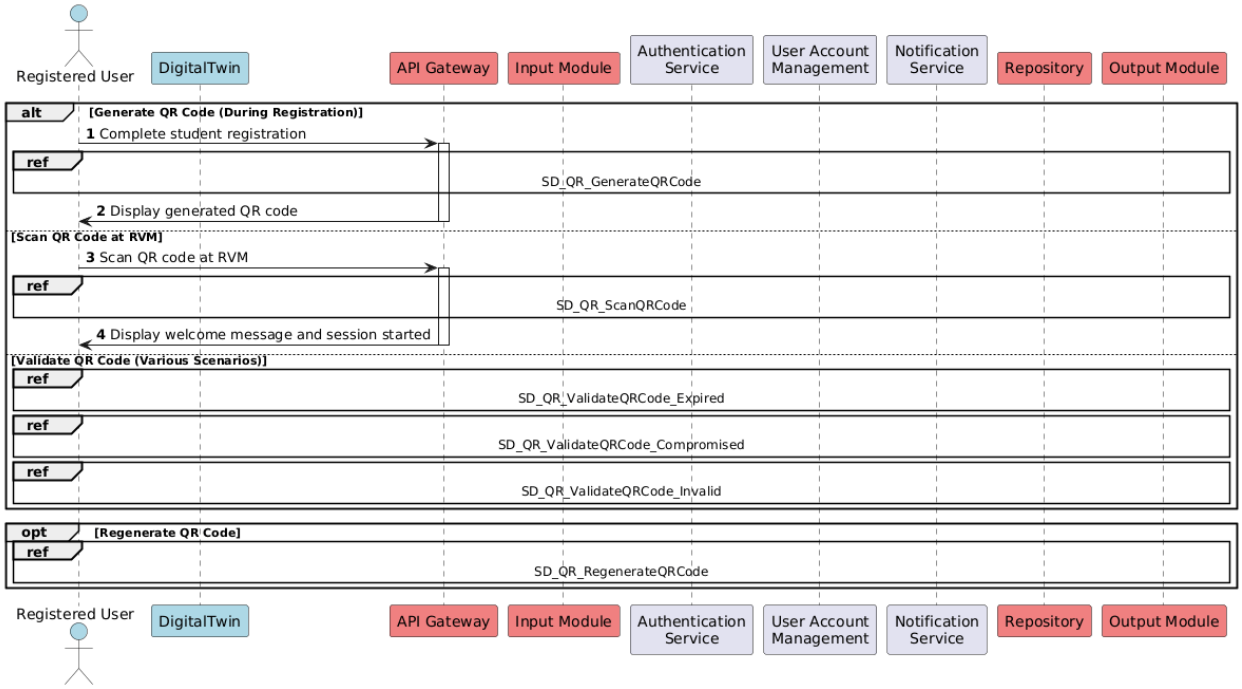
USE CASE: MANAGE NOTIFICATIONS - Main Flow (with References)

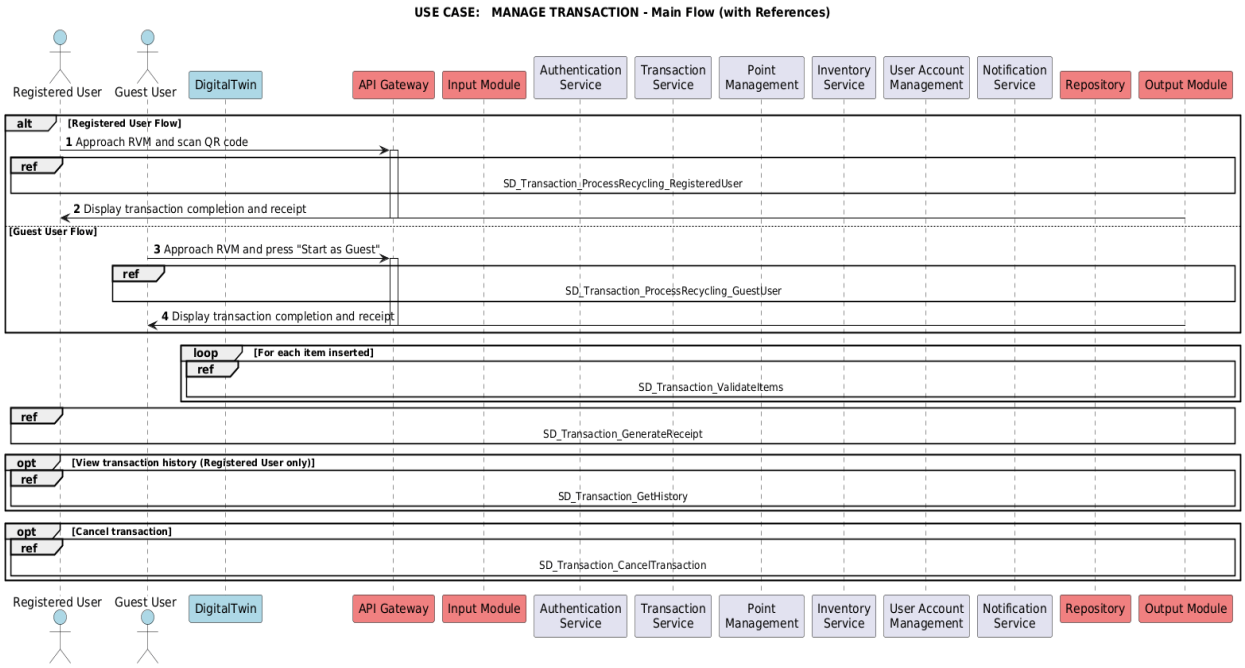


USE CASE: MANAGE POINTS - Main Flow (with References)



USE CASE: MANAGE QR CODE - Main Flow (with References)





RVM System — Use Case Estimation (Fibonacci / Planning Poker)

Team size:5• Project duration so far:~3 months• Estimation style:Fibonacci story points
This document provides a professional, report-ready use case estimate based on your RVM use case diagram (actors, modules, and relationships).

Story points are relative size; a conversion to calendar time is included using a stated assumption.

1. Scope & Assumptions

Scope. Reverse Vending Machine (RVM) System core modules: Authentication, User Account, Transactions (registered/guest), Points, Rewards, Inventory, Notifications, Analytics; plus integrations with Digital Twin and Cafeteria System.
Estimation method. Planning Poker / Fibonacci: {1, 2, 3, 5, 8, 13} story points (SP).
Time conversion assumption (editable). . With 5 members and expected focus factor, schedule is estimated with an effective capacity (see summary).
If your course uses a different conversion (e.g., 1 SP = 2 days), tell me and I'll regenerate the doc instantly.

2. Main Actors & External Systems

Actor / System	Role	Key Interactions
Administrator User	Manages users, configurations, monitoring, and reporting	Manage Authentication, Manage Inventory, Manage Analytics, Manage Rewards (admin ops)
Registered User	Uses account features; earns & redeems points	Manage User Account, Manage Registered Transaction, Manage Points, Manage Rewards, Notifications
Guest User	Performs recycling transaction without account	Manage Guest Transaction, Receipts/outputs
Digital Twin System	Telemetry, optimization, monitoring for machine/inventory	Inventory signals, analytics inputs, machine status synchronization
Cafeteria System	Partner redemption channel	Reward redemption confirmation, catalog sync (if applicable)

3. Complexity Scale (Fibonacci)

Story Points	Complexity	Typical Indicators
1	Very Small	Simple CRUD, minimal rules, no integrations
2	Small	Some validation, small branching, basic persistence
3	Medium	Multiple steps, moderate rules, error handling, audit logs
5	Large	Multi-entity workflows, role checks, reporting, retries
8	Complex	Integration-heavy, concurrency/consistency needs, complex state
13	Very Complex	Hard-to-estimate scope, many edge cases, non-trivial performance/security

4. Use Case Estimation Table

Derived from your RVM use case diagram. Use cases are phrased in “Manage …” form for consistency with your diagram.

UC ID & Name	Primary Actor(s)	Complexity Drivers	Story Points
UC-01 — Manage Authentication	Administrator, Registered User	Credential validation, session/token lifecycle, password reset,	8

		lockouts, audit/security events	
UC-02 — Manage User Account	Registered User, Administrator	Profile updates, account status changes, role management, linkage to notifications/security history	5
UC-03 — Manage Registered Transaction	Registered User	End-to-end transaction workflow, item validation loop, receipt generation, points calculation, persistence	13
UC-04 — Manage Guest Transaction	Guest User	Transaction without account, reduced identity context, receipt generation, inventory + analytics updates	8
UC-05 — Manage Points	Registered User, Administrator	Ledger integrity, earn/redeem/expire, reconciliation with transactions and rewards, reporting	8
UC-06 — Manage Rewards	Registered User, Administrator	Catalog management, redemption rules, integration with Cafeteria (optional), fraud checks	5
UC-07 — Manage Inventory	Administrator, Digital Twin	Machine status + capacity thresholds, telemetry ingestion, inventory synchronization, alerts	8
UC-08 — Manage Notifications	Registered User	Preference management, event triggers (tx/points/rewards), delivery retries, templates	5
UC-09 — Manage Analytics	Administrator, Digital Twin	Aggregation, filters, export formats, dashboards, KPI definitions, performance considerations	8

Note: If your instructor expects separate estimates for “includes/extends” (e.g., transaction includes authentication), we can split UC-03 into smaller stories.

5. Effort & Schedule Summary

How to interpret this section

- Story points arerelative sizeand require a conversion to estimate time.
- This report provides3 effort scenarios(different “days per point”) in the Effort Summary section below, matching your sample format.
- The Mermaid Gantt chart is avisual plan; the Word document includes a durations/dependencies table for grading.

If you want a schedule aligned with the “~3 months already worked”, we can present this as remaining work / completed work,
but I need your current completion percentage (e.g., 60% done).

6. User Stories Breakdown (Low-Level)

The 6 high-level features are decomposed into low-level user stories with acceptance criteria, similar to your example.

Feature 1: Infrastructure (Total: 5 Pts)

1.1 Database ConfigurationEst: 3 PtsAs aSystem Admin, I need to configure and save the database connection settings, so that the RVM system can persist users, transactions, and points reliably.Acceptance CriteriaSystem can connect to the database using the stored configuration.Core entities (User, Transaction, Points) can be created and retrieved successfully.Configuration persists after restarting the application.
1.2 External Integration SetupEst: 2 PtsAs aDeveloper, I need to configure integration endpoints/keys (Digital Twin and Cafeteria), so that the system can authenticate requests and exchange data securely.Acceptance CriteriaSystem can store and retrieve integration configuration (URLs/keys) securely.A connectivity check returns “Connected” or a clear error reason.Access to integration calls is restricted to authorized roles.

As a **System Admin**, I need to configure and save the database connection settings, so that the RVM system can persist users, transactions, and points reliably.

Acceptance Criteria

1. System can connect to the database using the stored configuration.
2. Core entities (User, Transaction, Points) can be created and retrieved successfully.
3. Configuration persists after restarting the application.

As a **Developer**, I need to configure integration endpoints/keys (Digital Twin and Cafeteria), so that the system can authenticate requests and exchange data securely.

Acceptance Criteria

4. System can store and retrieve integration configuration (URLs/keys) securely.
5. A connectivity check returns “Connected” or a clear error reason.
6. Access to integration calls is restricted to authorized roles.

Feature 2: Authentication & Accounts (Total: 13 Pts)

2.1 Login / Register / Token HandlingEst: 8 PtsAs aRegistered User, I need to register and log in, so that I can access my account and earn points during recycling transactions.Acceptance CriteriaUser can register with unique username/email and receive a successful confirmation response.User can log in with valid credentials and receives a session/token with expiry.Invalid credentials trigger a safe error response and a security event is recorded.
2.2 Manage Profile & RolesEst: 5 PtsAs anAdministrator, I need to manage user profiles and roles, so that access to administrative modules is controlled.Acceptance CriteriaAdmin can view/search user profiles and change status (active/disabled).Role changes take effect immediately for protected endpoints.All administrative changes are logged with timestamp and actor.

As a **Registered User**, I need to register and log in, so that I can access my account and earn points during recycling transactions.

Acceptance Criteria

7. User can register with unique username/email and receive a successful confirmation response.
8. User can log in with valid credentials and receives a session/token with expiry.
9. Invalid credentials trigger a safe error response and a security event is recorded.

As an **Administrator**, I need to manage user profiles and roles, so that access to administrative modules is controlled.

Acceptance Criteria

10. Admin can view/search user profiles and change status (active/disabled).
11. Role changes take effect immediately for protected endpoints.
12. All administrative changes are logged with timestamp and actor.

Feature 3: Transactions (Total: 13 Pts)

3.1 Registered Transaction WorkflowEst: 8 PtsAs aRegistered User, I need to start a transaction, validate deposited items, and complete the process, so that I receive points and a receipt.Acceptance CriteriaSystem creates a transaction with status lifecycle (started → completed/cancelled).Each scanned item is validated and recorded as a transaction item.Completing a transaction generates a receipt and stores the final totals.
3.2 Guest Transaction + ReceiptEst: 5 PtsAs aGuest User, I need to complete a transaction without an account, so that I can recycle items and receive a receipt.Acceptance CriteriaGuest transactions can be created and completed without a user profile.Receipt is generated and can be displayed/printed for the guest.Transaction still contributes to inventory counters and analytics metrics.

As a **Registered User**, I need to start a transaction, validate deposited items, and complete the process, so that I receive points and a receipt.

Acceptance Criteria

13. System creates a transaction with status lifecycle (started → completed/cancelled).
14. Each scanned item is validated and recorded as a transaction item.
15. Completing a transaction generates a receipt and stores the final totals.

As a **Guest User**, I need to complete a transaction without an account, so that I can recycle items and receive a receipt.

Acceptance Criteria

16. Guest transactions can be created and completed without a user profile.
17. Receipt is generated and can be displayed/printed for the guest.
18. Transaction still contributes to inventory counters and analytics metrics.

Feature 4: Points & Rewards (Total: 13 Pts)

4.1 Points Ledger (Earn / Expire)Est: 8 PtsAs aRegistered User, I need points to be calculated and recorded in a ledger, so that my balance and history are correct over time.Acceptance CriteriaPoints are added when a transaction completes and appear in balance/history.Each ledger entry references its source transaction and timestamp.Expired points are handled consistently and visible in history.
4.2 Reward Catalog + RedemptionEst: 5 PtsAs aRegistered User, I need to redeem rewards using points, so that I can exchange points for benefits (e.g., cafeteria redemption).Acceptance CriteriaUser can view the reward catalog with required points and availability status.Redemption decreases points balance and records a redemption transaction.Redemption triggers a notification and produces a redemption confirmation reference.

As a **Registered User**, I need points to be calculated and recorded in a ledger, so that my balance and history are correct over time.

Acceptance Criteria

19. Points are added when a transaction completes and appear in balance/history.
20. Each ledger entry references its source transaction and timestamp.
21. Expired points are handled consistently and visible in history.

As a **Registered User**, I need to redeem rewards using points, so that I can exchange points for benefits (e.g., cafeteria redemption).

Acceptance Criteria

22. User can view the reward catalog with required points and availability status.
23. Redemption decreases points balance and records a redemption transaction.
24. Redemption triggers a notification and produces a redemption confirmation reference.

Feature 5: Inventory & Digital Twin (Total: 8 Pts)

5.1 Machine Inventory SynchronizationEst: 5 PtsAs anAdministrator, I need the inventory state of each machine to be tracked and synchronized, so that collection/maintenance decisions are accurate.Acceptance CriteriaSystem stores per-machine inventory levels and status updates.Inventory updates are applied when transactions add items.Admin can retrieve a current inventory report per machine/location.
5.2 Telemetry Thresholds & AlertsEst: 3 PtsAs aSystem, I need to detect threshold breaches (capacity, errors) and generate alerts, so that issues are handled quickly.Acceptance CriteriaThreshold rules can be configured (e.g., 80% capacity).System creates a system alert when the threshold is exceeded.Alerts are visible to administrators with severity and timestamp.

As an **Administrator**, I need the inventory state of each machine to be tracked and synchronized, so that collection/maintenance decisions are accurate.

Acceptance Criteria

25. System stores per-machine inventory levels and status updates.
26. Inventory updates are applied when transactions add items.
27. Admin can retrieve a current inventory report per machine/location.

As a **System**, I need to detect threshold breaches (capacity, errors) and generate alerts, so that issues are handled quickly.

Acceptance Criteria

28. Threshold rules can be configured (e.g., 80% capacity).
29. System creates a system alert when the threshold is exceeded.
30. Alerts are visible to administrators with severity and timestamp.

Feature 6: Notifications & Analytics (Total: 16 Pts)

6.1 Notification Preferences & DeliveryEst: 5 PtsAs aRegistered User, I need to configure notification preferences, so that I receive relevant updates (transactions, points, rewards).Acceptance CriteriaUser can enable/disable notification channels (email/SMS/push) in preferences.System sends a notification on key events (transaction complete, redemption).Delivery failures are retried or recorded with a clear status.
6.2 Admin Analytics DashboardEst: 8 PtsAs anAdministrator, I need analytics dashboards (KPIs, trends), so that I can monitor usage, recycling volume, and system performance.Acceptance CriteriaAdmin can view key metrics over a date range (transactions, items, points).Filters apply correctly and results match stored transaction data.Dashboard load time is acceptable for typical date ranges.
6.3 Export Reports (CSV/PDF)Est: 3 PtsAs anAdministrator, I need to export analytics reports, so that I can share results and archive project evidence.Acceptance CriteriaAdmin can export a selected report type for a chosen date range.Export includes headers and matches the displayed metrics.Generated file is downloadable and named with timestamp.

As a **Registered User**, I need to configure notification preferences, so that I receive relevant updates (transactions, points, rewards).

Acceptance Criteria

31. User can enable/disable notification channels (email/SMS/push) in preferences.
32. System sends a notification on key events (transaction complete, redemption).
33. Delivery failures are retried or recorded with a clear status.

As an **Administrator**, I need analytics dashboards (KPIs, trends), so that I can monitor usage, recycling volume, and system performance.

Acceptance Criteria

34. Admin can view key metrics over a date range (transactions, items, points).
35. Filters apply correctly and results match stored transaction data.
36. Dashboard load time is acceptable for typical date ranges.

As an **Administrator**, I need to export analytics reports, so that I can share results and archive project evidence.

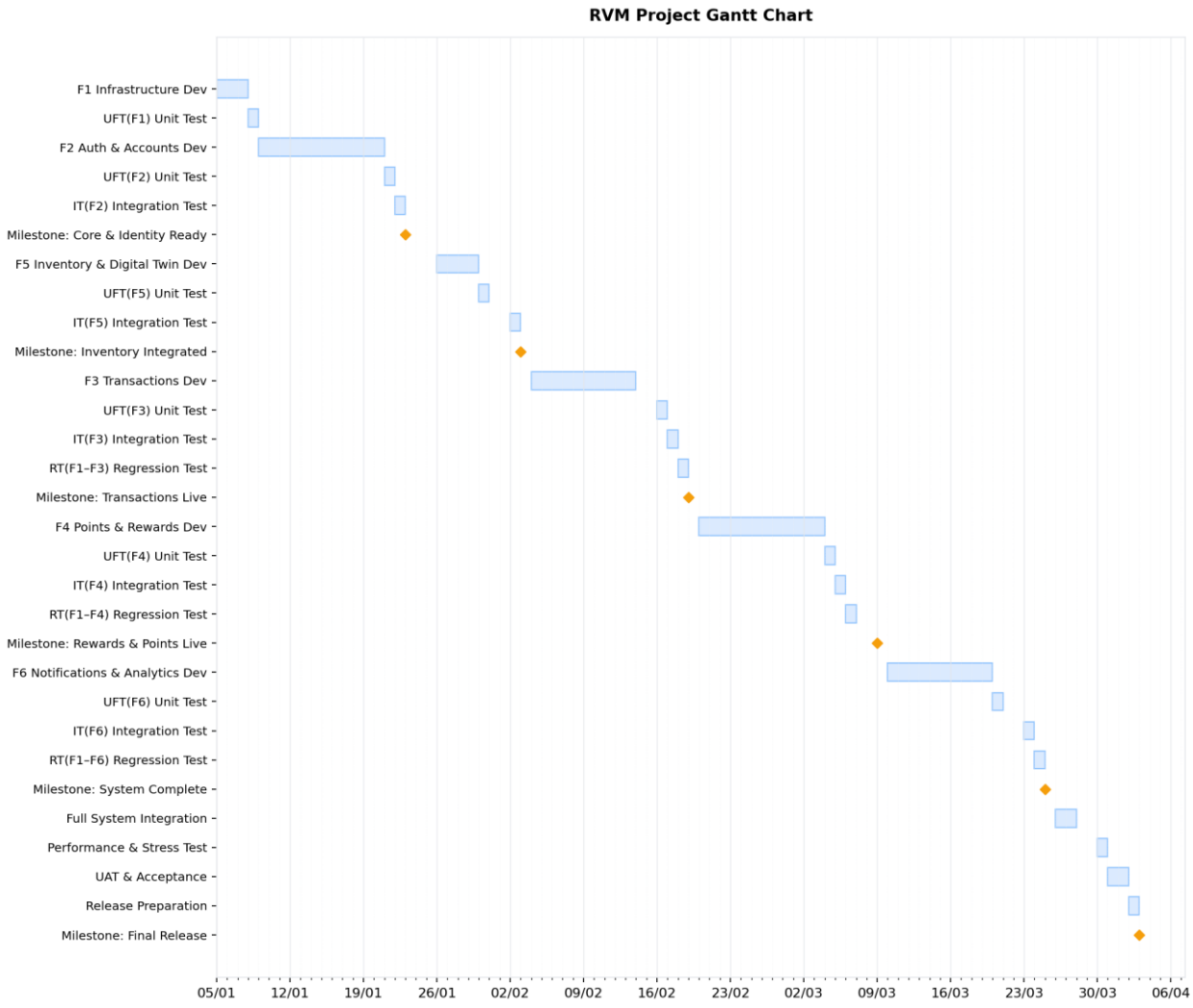
Acceptance Criteria

37. Admin can export a selected report type for a chosen date range.

38. Export includes headers and matches the displayed metrics.
39. Generated file is downloadable and named with timestamp.

7. Gantt Chart (Rough Schedule)

The chart below is a **detailed plan** in the same style as your example:
feature development + unit tests (UFT) + integration tests (IT) + regression tests (RT) + milestones and release.
It is sized to approximately **6.5 sprints** (≈ 65 working days). Start date is editable.



Task Durations & Dependencies (for Word/PDF readability)

Activity	Duration (days)	Dependencies
F1 Infrastructure Dev	3	—

UFT(F1) Unit Test	1	F1
F2 Auth & Accounts Dev	8	UFT(F1)
UFT(F2) Unit Test	1	F2
IT(F2) Integration Test	1	UFT(F2)
Milestone: Core & Identity Ready	1	IT(F2)
F5 Inventory & Digital Twin Dev	4	Milestone: Core & Identity Ready
UFT(F5) Unit Test	1	F5
IT(F5) Integration Test	1	UFT(F5)
Milestone: Inventory Integrated	1	IT(F5)
F3 Transactions Dev	8	Milestone: Inventory Integrated
UFT(F3) Unit Test	1	F3
IT(F3) Integration Test	1	UFT(F3)
RT(F1–F3) Regression Test	1	IT(F3)
Milestone: Transactions Live	1	RT(F1–F3)
F4 Points & Rewards Dev	8	Milestone: Transactions Live
UFT(F4) Unit Test	1	F4
IT(F4) Integration Test	1	UFT(F4)
RT(F1–F4) Regression Test	1	IT(F4)
Milestone: Rewards & Points Live	1	RT(F1–F4)
F6 Notifications & Analytics Dev	8	Milestone: Rewards & Points Live
UFT(F6) Unit Test	1	F6
IT(F6) Integration Test	1	UFT(F6)
RT(F1–F6) Regression Test	1	IT(F6)
Milestone: System Complete	1	RT(F1–F6)
Full System Integration	2	Milestone: System Complete
Performance & Stress Test	1	Full System Integration
UAT & Acceptance	2	Performance & Stress Test
Release Preparation	1	UAT & Acceptance
Milestone: Final Release	1	Release Preparation

If you want the Gantt to reflect “team size = 5 over ~3 months already”, tell me whether this schedule is yourremaining workor thefull project plan, and your completion %.

8. Effort Summary (3 Scenarios)

This section mirrors the style of your sample: total points → person-days → working days → calendar days → sprints.

Feature Summary

High Level Feature	Low Level Stories Included	Total Complexity

F1: Infrastructure	1.1 Database Configuration, 1.2 External Integration Setup	5 Pts
F2: Authentication & Accounts	2.1 Login/Register/Token Handling, 2.2 Manage Profile & Roles	13 Pts
F3: Transactions	3.1 Registered Transaction Workflow, 3.2 Guest Transaction + Receipt	13 Pts
F4: Points & Rewards	4.1 Points Ledger (Earn/Expire), 4.2 Reward Catalog + Redemption	13 Pts
F5: Inventory & Digital Twin	5.1 Machine Inventory Synchronization, 5.2 Telemetry Thresholds & Alerts	8 Pts
F6: Notifications & Analytics	6.1 Notification Preferences & Delivery, 6.2 Admin Analytics Dashboard, 6.3 Export Reports	16 Pts
Total Project	—	68 Points

Scenario A (Optimistic)

- Total Effort:68 pts × 5 days/pt =340 Person-Days
- Duration (Working Days):340 / 5 members =68 Working Days
- Duration (Calendar):68 working days ÷ 5 days/week × 7 ≈95 Calendar Days(≈ 3.2 months)
- Sprints:68 working days ÷ 10 days/sprint =6.8 Sprints

Scenario B (Most Likely — Matches Sample Style)

- Total Effort:68 pts × 7 days/pt =476 Person-Days
- Duration (Working Days):476 / 5 members =95.2 Working Days
- Duration (Calendar):95.2 ÷ 5 × 7 ≈133 Calendar Days(≈ 4.4 months)
- Sprints:95.2 ÷ 10 =9.5 Sprints

Scenario C (Conservative)

- Total Effort:68 pts × 9 days/pt =612 Person-Days
- Duration (Working Days):612 / 5 members =122.4 Working Days
- Duration (Calendar):122.4 ÷ 5 × 7 ≈171 Calendar Days(≈ 5.7 months)
- Sprints:122.4 ÷ 10 =12.2 Sprints

If your instructor mandates a specific conversion (e.g., 1 pt = 7 days), we'll keep only that scenario.

9. Risks & Contingency (Professional Add-on)

Key risks commonly associated with systems like RVM (matching your lecture slides):

- Requirements change: changes in reward/points rules can introduce redesign (mitigation: baseline rules + change control).
- Technology/integration risk: Digital Twin / Cafeteria integrations may require retries and reconciliation (mitigation: contract tests).
- Estimation risk: transaction edge cases and device behavior uncertainty (mitigation: split UC-03 into smaller stories + spike).
- People risk: team availability fluctuations (mitigation: buffer 15–25%).

Recommended contingency: add 15–25% schedule buffer for integration and testing.

10. Traceability (Use Cases → Modules)

Module	Mapped Use Cases
Authentication	UC-01
User Account	UC-02
Transactions	UC-03, UC-04
Points	UC-05
Rewards	UC-06
Inventory	UC-07
Notifications	UC-08
Analytics	UC-09

Appendix A — Notes to Customize

- Tell me your preferred conversion:1 SP = 1 dayor1 SP = 2 days.
- If you have a deadline date, I can compute required velocity and a Gantt schedule that matches.
- If you want “perfect”, send completion % of the project after 3 months (e.g., 70% done).

11. Validation & Verfication

11.1 Forward Traceability

- All functional requirements map to system components
- All components have defined interfaces
- All interfaces support specific use cases

11.2 Backward Traceability

- All interfaces trace back to requirements
- All components fulfill specific functional requirements
- No orphaned interfaces or components

11.3 Completeness Check

- Requirements Coverage: 100% (15/15)
- Component Coverage: 100% (10/10)
- Interface Documentation: 100% (25+/25+)

Test Case Planning: 0% (0/15 executed)