

**SUMMER INTERNSHIP - SOPHOMORES (SPJ2001)**

**First Review of**

**UNIPAY**

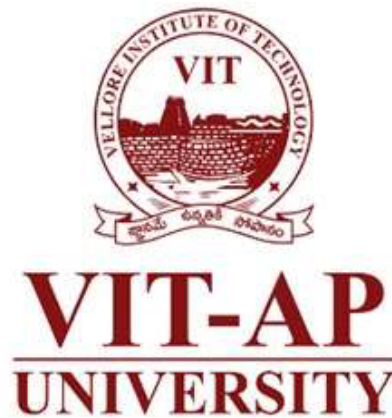
**Submitted by**

Gayatri Gohita Chilakapati - 22MIS7071

Eluri Sai Lakshmi Bhavya - 22MIS7265

**Under the guidance of**

Dr. Rajesh Duvvuru



**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING**

**(SCOPE)**

**July, 2025**

# **SOFTWARE REQUIREMENT SPECIFICATIONS**

## **PURPOSE OF THE SYSTEM**

UniPay is a digital wallet based system that targets the payments made within the campus environment by students and faculty to the merchants. Instead of generic cashback offers and other rewards, UniPay offers personalized discounts based on the user's spending analysis. UniPay also provides dashboards for the users to view their spending analysis based on categories and track wallet balances, promoting a cashless, efficient, and reward-driven banking system.

## **PROBLEMS IN THE EXISTING SYSTEM**

- Payments are made manually or through separate portals for different services (fees, canteen, hostel, etc.).
- No centralized system to manage all university-related financial transactions.
- Lack of personalized offers or discounts for frequent or loyal users.
- No AI-based recommendation system to suggest payment categories or manage spending.
- Time-consuming and prone to errors due to non-integrated platforms

## **SOLUTION OF THESE PROBLEMS**

- Single platform for all university payments including tuition, food, hostel, and events.
- AI/ML-powered recommendation system that suggests offers and payment reminders based on user behavior.
- Personalized discounts and rewards based on usage patterns and loyalty.
- Quick, secure, and user-friendly interface for easy transaction management.
- Centralized tracking and history of all payments for better financial planning.

## **SCOPE OF THE PROJECT**

UniPay can be implemented in any college or university campus where payments happen between students, faculty, and merchants. It simplifies on-campus transactions by offering a unified wallet-based system. The project has the potential to grow further with features like QR code-based payments, monthly spending limits, and budget planning tools for students. In the future, UniPay can also be expanded to include integrations with banks, loan request facilities, and even a full-fledged campus marketplace to support all types of financial interactions within the campus environment.

## **FUNCTIONAL COMPONENTS OF THE PROJECT**

There are registered users in the UniPay system. Some are admins, while others are students, faculty, or merchants. A user can have multiple roles depending on the access level.

In a college environment, this can be Students/Faculty/Merchants/Admin.

1. A user should be able to:

- **Login** securely using their credentials.
- **Change password** after logging in.
- **View wallet balance** and transaction stats on the dashboard.
- **Make payments** to canteens, events, library fines, or campus merchants.
- **Track payment history** with filters like date or category.
- **Receive personalized suggestions** based on spending habits using AI.
- **Claim rewards and discounts** automatically after eligible transactions.
- **Receive reminders** for due fees or scheduled payments.
- **Download reports** of their spending summary.
- **Contact support/admin** if help is needed.

2. A merchant should be able to:

- Login and view **transactions made by students/faculty** to their store.
- **Provide discounts or rewards** to users.
- **View reports** of sales and most frequent users.
- **Track reward claims** and redemptions.

3. An admin should be able to:

- Manage all registered users, merchants, and their access.
- **View and generate reports** for transaction summaries.
- Approve or manage **reward schemes**.

4. Automatic features:

- AI-based **categorization of spending** into areas like food, travel, stationery, etc.
- **Spending summary charts** and visualizations (like line graph, pie chart).
- Auto-generated **discount offers** based on frequent merchant usage.

## **STUDY OF THE SYSTEM**

UniPay is designed as a **campus-specific digital wallet system** that makes payments and rewards smarter and more personalized. The system aims to reduce the hassle of carrying cash or using generic payment apps within the college environment. With UniPay, **students and faculty** can make quick payments to campus merchants and receive **customized discounts** based on their transaction patterns. It also provides **dashboards** for users to track wallet balances, spending summaries, and reward history.

The backend handles user authentication, transaction recording, reward calculations, and AI-based categorization. The frontend offers a clean and responsive interface for all user types — students,

faculty, merchants, and admins.

## **INPUT/ OUTPUT**

### **Inputs:**

- User login credentials (email/password)
- Payment details (amount, merchant, category)
- User actions like:
  - Spending a certain amount
  - Claiming a reward
  - Viewing a report or downloading it
  - Adding merchant offers (by admin or merchant)

### **Outputs:**

- Spending summary charts (line, ring, bar)
- Wallet balance and total usage stats
- Transaction history with filters
- Reward notifications and discounts
- Downloadable reports (PDF/CSV)
- AI-generated suggestions based on user spending

## **PERFORMANCE REQUIREMENTS**

- Fast load time: Dashboards and transaction data should load within 2-3 seconds.
- Secure transactions: All payment actions must be done over HTTPS, with proper authentication.
- AI/ML processing: Transaction data should be analyzed in the background without affecting the UI responsiveness.

- Scalability: Should handle hundreds of simultaneous users, especially during events or fee deadlines.
- Mobile responsiveness: The UI should adapt smoothly to various screen sizes and devices.
- Error handling: System should provide clear feedback on failed payments or login errors.
- Real-time updates: Wallet balance and transaction logs should update instantly after any transaction.

## **FEASIBILITY REPORT**

Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All systems are feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

### **Technical Feasibility**

UniPay is technically feasible with the current tools and technologies available. The system is built using widely used technologies like **HTML/CSS/JS** for the frontend, **Flask (Python)** for the backend, and **MongoDB** for the database. These are lightweight, open-source, and easy to integrate. Since the project mainly focuses on intra-campus usage, the server load and data storage requirements are manageable. APIs for wallet transactions, dashboards, and AI-based suggestions can be developed and maintained efficiently.

### **Economical Feasibility**

From an economic standpoint, UniPay is cost-effective. Most of the tech stack (Flask, MongoDB, Chart.js, etc.) is free and open-source, reducing software licensing costs. Deployment can be done on low-cost cloud services (like Heroku, Render, or DigitalOcean). Since the system is built for campus use, hardware and network infrastructure already in place can be reused. Maintenance and updates can be handled by a small team, keeping operational costs low.

### **Operational Feasibility**

Operationally, UniPay fits well into a campus environment. Students and faculty are already used to online systems, so learning the interface will be easy. Merchants can also be trained quickly to use their dashboard for tracking payments and offering discounts. The admin panel allows the management to monitor usage, generate reports, and ensure smooth functioning.

Overall, the system is easy to operate, doesn't require much training, and improves day-to-day payment handling on campus.

### **ER DIAGRAM**

The relation upon the system is structure through a conceptual ER-Diagram, which not only specifies the existential entities but also the standard relations through which the system exists and the cardinalities that are necessary for the system state to continue.

The entity Relationship Diagram (ERD) depicts the relationship between the data objects. The ERD is the notation that is used to conduct the data modeling activity. The attributes of each data object noted in the ERD can be described using data object descriptions.

The set of primary components that are identified by the ERD are

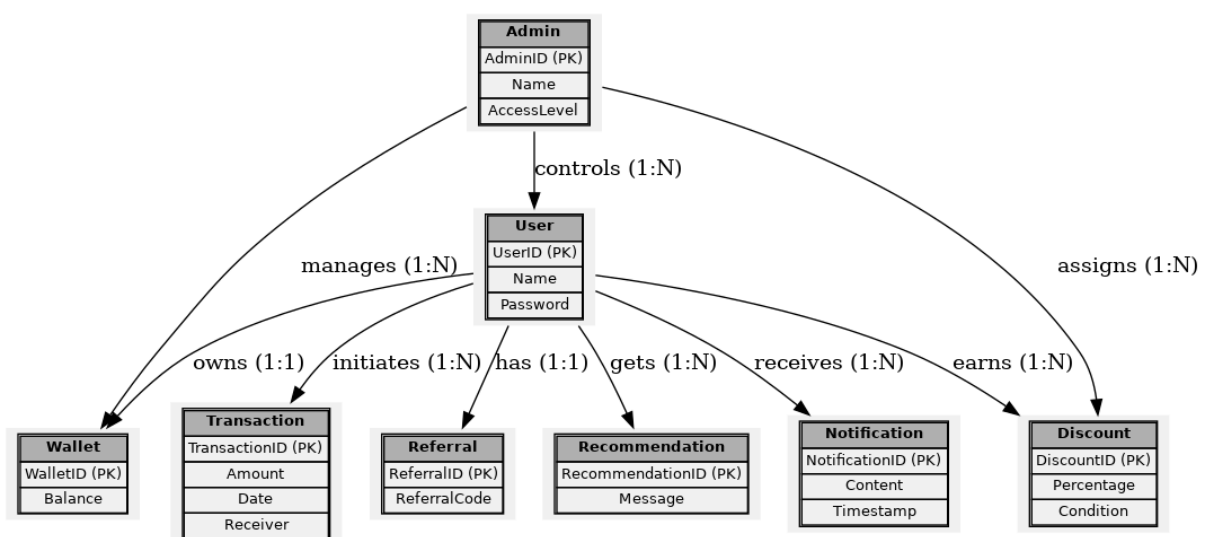
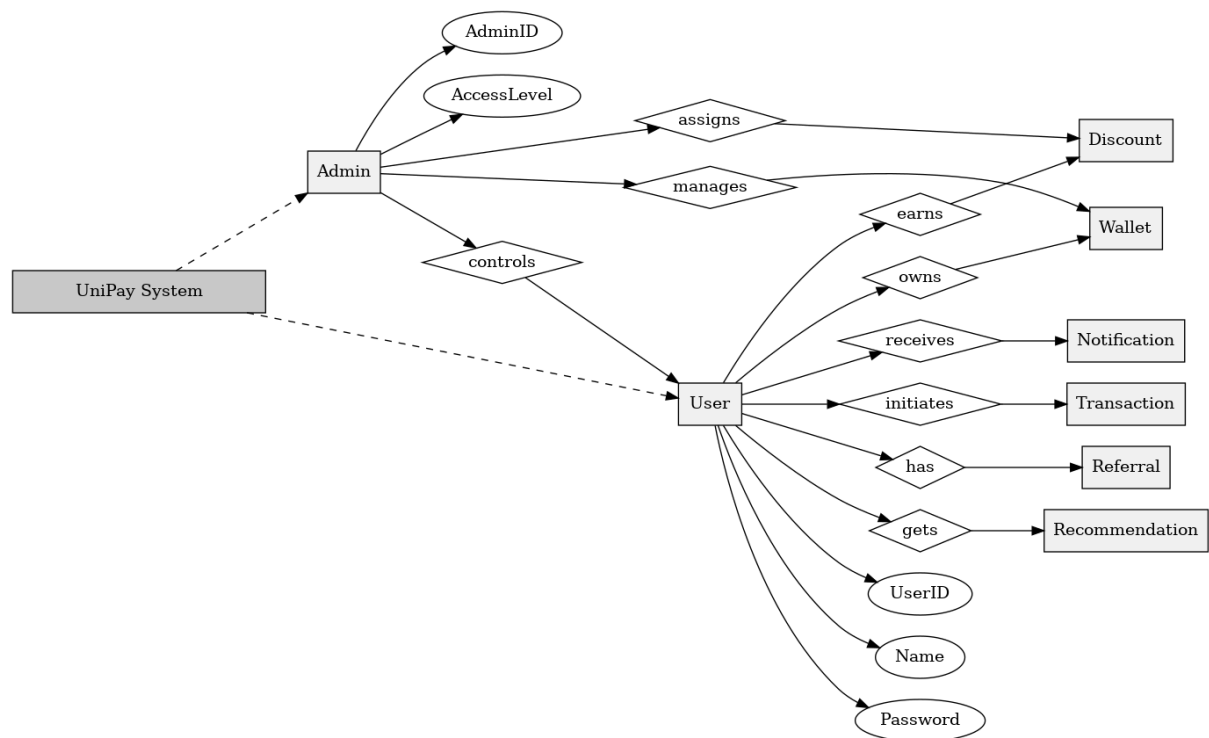
Data object

Relationships

Attributes

Various types of indicators.

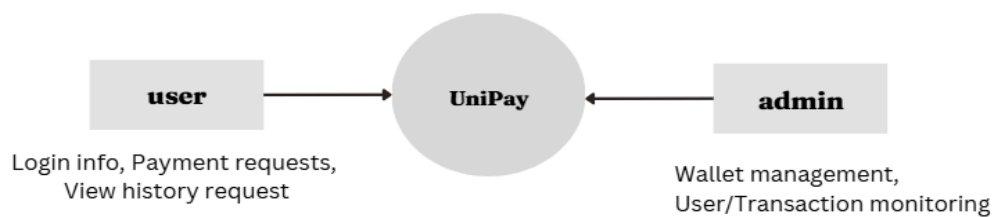
The primary purpose of the ERD is to represent data objects and their relationships.



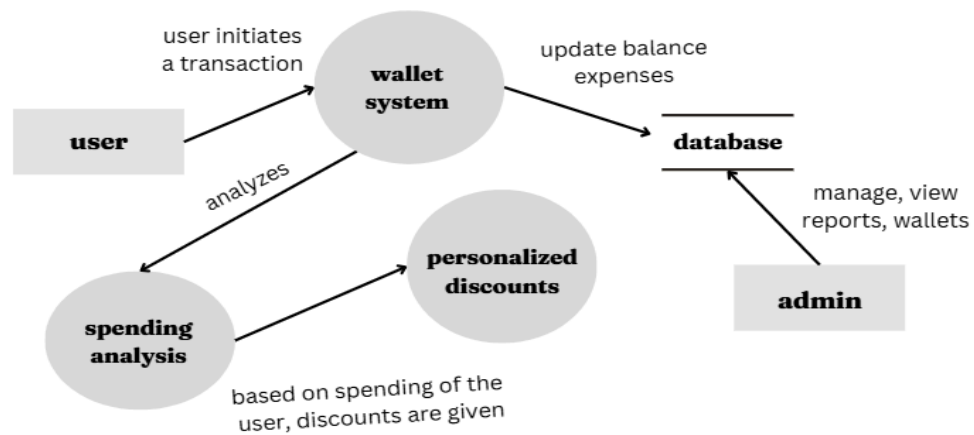


## DATA FLOW DIAGRAM

### level 0 DFD



### level 1 DFD



## SYSTEM DESIGN/ ARCHITECTURE

