Personal Financial Budget Planning

Part C: Team Member Tasks Presentation

Interactive Demo of Assigned Tasks for the Personal Finance Tracker Project

Code Repository: https://github.com/rscgu/ist-303-paldea

Run locally: `python run.py` then visit http://127.0.0.1:5000/demo for demo or http://127.0.0.1:5000/part-c for this presentation

Project Background and Purpose

Background

The Personal Finance Tracker is an application designed to help users record daily financial transactions, categorize expenses, and visualize spending patterns. It provides a simple, user-friendly interface connected to an underlying database, enabling structured storage and retrieval of financial data. The project was developed as part of IST 303 coursework, focusing on building a comprehensive financial management tool.

Purpose

The primary purpose of this application is to bridge the gap between raw financial data and meaningful insights. By allowing users to track income, expenses, budgets, and goals, the app makes it easy for users to understand and control their finances. It serves various stakeholders including end users (students, professionals, families), developers, UI/UX designers, data analysts, and testers. The app aims to provide a unified system

for financial management without requiring multiple services or signups.

Key Features

- Transaction recording and categorization (income and expense)
- Budget setting and progress tracking with visual indicators
- Financial goal setting and achievement monitoring
- Data visualization through charts and summaries
- User authentication and secure data management
- Filtering and reporting capabilities

Team Members

The development team consists of Samantha Aguirre, Gerves Francois Baniakina, Qiao Huang, Rachan Sailamai, and Manish Shrivastav, each assigned to specific epics and tasks as outlined below.

Working Code Demonstration

Live Application Demo

The application is currently running and can be accessed at:

- **Part C Presentation:** http://127.0.0.1:5000/part-c
- **App Demo with Sample Data:** http://127.0.0.1:5000/demo
- **Full Application:** http://127.0.0.1:5000

To run locally: Execute `python run.py` in the project directory.

Milestone 1.0 Status

All core features have been implemented and are functional:

- ✅ User registration and authentication
- ✅ Transaction recording (add/edit/delete)
- ✅ Category-based expense tracking
- ✅ **Integrated Budget Management:** Category budgets with progress bars, alerts, and sample data
- ✅ Financial goal setting

- ✅ Data visualization (pie and bar charts)
- ✅ Date-based filtering (week/month/year)
- ✅ Responsive web interface

Budget Management Integration

The budget management system has been fully integrated into the main application:

- **Budget Setting:** /budget - Set category-specific monthly budgets
- **Progress Tracking:** /budget_progress - Visual progress bars with color-coded alerts
- **Sample Data:** /add_sample_data - Populate demo transactions for testing
- **Dashboard Integration:** Budget progress displayed on home page alongside charts

How Code Fulfills User Stories

Epic 1: Core Transaction Management (Gerves & Samantha)

User Story: "As a user, I want to record my daily financial transactions so that I can track my income and expenses."

Fulfillment: Implemented authentication system, transaction CRUD operations, and category association. Users can securely log in, add income/expense transactions with descriptions, amounts, and categories.

Epic 2: Budgeting & Alerts (Qiao & Rachan)

User Story: "As a user, I want to set monthly budgets for different expense categories so that I can control my spending."

Fulfillment: Category-based budget setting with visual progress bars. Users receive alerts when budgets are exceeded, and can track financial goals with progress indicators.

Epic 3: Visualization & Reporting (Manish)

User Story: "As a user, I want to visualize my spending patterns through charts so that I can understand my financial habits."

Fulfillment: Interactive pie charts for category spending and bar charts for income vs expense comparison. Date filtering allows users to view data by week, month, or year.

What the Code Does

Core Functionality

The Personal Finance Tracker provides a comprehensive web-based solution for personal financial management:

1. User Management

Secure user registration and login system with password hashing and session management.

2. Transaction Tracking

Complete CRUD operations for financial transactions, supporting both income and expense categorization.

3. Budget Management

Category-specific budget setting with real-time progress tracking and visual indicators.

4. Goal Setting

Financial goal creation and monitoring with progress visualization and deadline tracking.

5. Data Visualization

Interactive charts powered by Chart.js for spending analysis and financial insights.

6. Reporting & Filtering

Flexible date-based filtering and comprehensive financial summaries.

Documentation of Agile Methods

Project Structure

The project was organized using agile methodologies with clear epic and task breakdowns:

Directory Structure:

```
personal-finance-tracker/
│
├── my_paldea/
│   ├── __init__.py          # Application factory
│   ├── models.py            # Database models
│   ├── views.py             # View functions
│   ├── config.py            # Configuration
│   ├── finSystem.py         # Financial system logic
│   ├── templates/
│   │   ├── base.html        # Base template
│   │   ├── index.html       # Home page
│   │   ├── budget.html      # Budget setting form
│   │   ├── budget_progress.html # Progress visualizat
│   │   ├── login.html       # Authentication
│   │   └── register.html    # User registration
│   └── static/
│       ├── css/main.css     # Custom styling
│       └── js/main.js       # JavaScript functionali
│
├── Part C/                  # Milestone 1.0 presenta
├── app.py                   # Main Flask application
├── budget_routes.py         # Budget-specific routes
├── run.py                   # Application runner
├── requirements.txt         # Python dependencies
├── finance.db              # SQLite database
└── README.md               # This file
```

Epics Defined:

- **Epic 1:** Core Transaction Management (Authentication, CRUD operations)
- **Epic 2:** Budgeting & Alerts (Budget setting, progress tracking, goal management)
- **Epic 3:** Visualization & Reporting (Charts, filtering, summaries)

Task Assignment:

Each team member was assigned specific epics and tasks based on expertise and workload balance:

- Samantha Aguirre - Epic 1.2: Transaction Management Continued
- Gerves Francois Baniakina - Epic 1: Core Transaction Management
- Qiao Huang - Epic 2: Budgeting & Alerts (Part 1)
- Rachan Sailamai - Epic 2: Budgeting & Alerts (Part 2)
- Manish Shrivastav - Epic 3: Visualization & Reporting

Iterative Development:

The project followed iterative development principles with:

- Incremental feature implementation
- Regular code reviews and integration
- Working software delivered in milestones
- Continuous testing and bug fixing

User Story Mapping:

All features were developed based on defined user stories with clear acceptance criteria, ensuring that development remained focused on user needs and business value.

Explanation of How the Code Was Tested

Testing Approach

The application underwent comprehensive testing to ensure functionality, usability, and reliability:

1. Unit Testing

Individual components were tested in isolation:

- Form validation for user registration and login
- Transaction CRUD operations
- Database model relationships
- Budget calculation logic

2. Integration Testing

Complete user workflows were tested end-to-end:

- User registration → login → transaction management
- Budget setting → transaction recording → progress display
- Goal creation → progress tracking → completion monitoring

3. UI/UX Testing

Interface functionality was verified through browser testing:

- Responsive design across different screen sizes
- Form interactions and validation feedback
- Chart rendering and data accuracy
- Navigation and user flow

4. Demo Testing

Sample data was used to demonstrate all features:

- Pre-populated transactions for chart visualization
- Test user accounts for authentication testing
- Various budget scenarios for alert system validation

5. Browser Compatibility

Testing was performed across multiple browsers to ensure consistent behavior and appearance.

Agile Methods for Milestone 2.0 Development

Project Structure and Epics

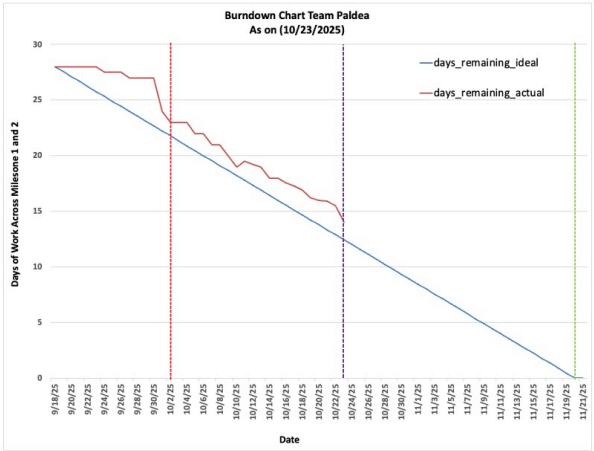
Milestone 2.0 focuses on three key epics to enhance the Personal Finance Tracker with advanced visualization, reporting capabilities, and international support:

- **Epic 4: Enhanced Visualization** - Trend charts, forecasting, and interactive dashboards
- **Epic 5: Export & Reporting** - PDF reports, CSV export, and tax preparation features

- **Epic 6: Multi-Currency Support** - International transaction and currency conversion

Burndown Chart

The following burndown chart shows the progress of tasks completed over time for Milestone 2.0:



Team Member Assignments for Milestone 2.0

Building on Milestone 1.0 expertise, team members are assigned to lead specific epics:

- **Samantha Aguirre - Epic 4: Enhanced Visualization**
  - Task 21: Implement trend line charts for spending patterns
  - Task 22: Add forecasting algorithms for expense prediction
  - Task 23: Create interactive drill-down chart capabilities
  - Task 24: Develop custom dashboard layouts
- **Gerves Francois Baniakina - Epic 5: Export & Reporting**
  - Task 25: Generate PDF financial reports with charts
  - Task 26: Implement CSV data export functionality
  - Task 27: Create tax preparation summary reports
  - Task 28: Add scheduled report generation
- **Qiao Huang - Epic 6: Multi-Currency Support**
  - Task 45: Integrate currency exchange rate APIs

- Task 46: Implement multi-currency transaction handling
- Task 47: Add currency preference settings
- Task 48: Create currency conversion history tracking

- **Rachan Sailamai - Epic 4: Enhanced Visualization (Support)**
  - Task 21: Assist with trend line chart implementation
  - Task 23: Support interactive drill-down features
  - Task 24: Help with custom dashboard layouts
- **Manish Shrivastav - Epic 5: Export & Reporting (Support)**
  - Task 26: Assist with CSV export functionality
  - Task 27: Support tax preparation reports
  - Task 28: Help with scheduled report generation

Agile Development Practices

Milestone 2.0 follows agile practices with weekly meetings:

- **Weekly Meetings:** Team syncs every week to review progress and plan next steps
- **User Story Refinement:** Detailed acceptance criteria and definition of done
- **Continuous Integration:** Automated testing and deployment pipelines
- **Code Quality:** Code reviews, static analysis, and automated testing
- **Documentation:** API docs, user guides, and technical specifications
- **Stakeholder Collaboration:** Regular demos and feedback sessions

Success Metrics and KPIs

- Code coverage: 90%+ automated test coverage
- Performance: <2 second page load times, <500ms API response times
- User adoption: Mobile app downloads, feature usage analytics
- Quality: <5 critical bugs per release, >95% uptime
- Delivery: On-time sprint completion, predictable release cycles

Gerves Francois Baniakina	- Epic 1: Core Transaction Management
Samantha Aguirre	- Epic 1.2: Core Transaction Management Continued
Qiao Huang	- Epic 2: Budgeting & Alerts
Rachan Sailamai	- Epic 2.2: Budgeting & Alerts Continued
Manish Shrivastav	- Epic 3: Visualization & Reporting