BỘ THÔNG TIN VÀ TRUYỀN THÔNG

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Second Report

Foundation Internship

ChippyCash

Smart Chatbot for Easier Expense Management

Instructor:	Kim Ngoc Bach
Student Name :	Dinh Manh Hung
Student ID :	B22DCCN359
Lớp :	E22CQCN05-B

Hà Nội - 2025

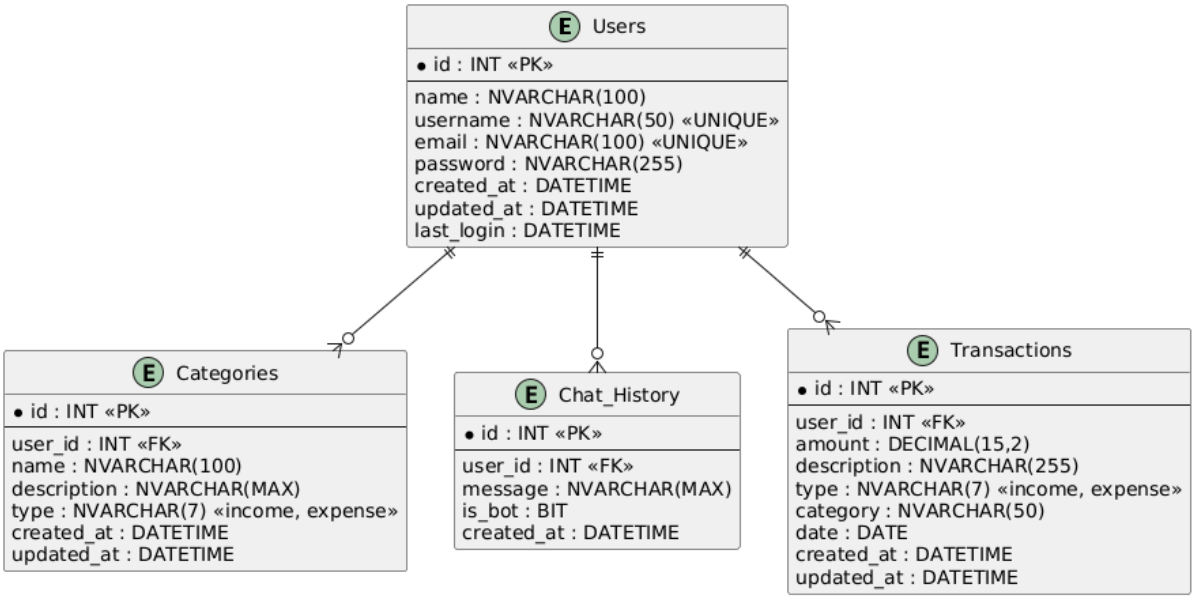
INTERNSHIP BASE REPORT - WEEK 2

As of March 21, 2025, I have completed the following tasks:

1. Requirement Analysis and System Design

- **System description**: The "Chippy" application provides features such as registration/login, income/expense category management, transaction management, financial analysis, and interaction with a financial chatbot. The system focuses on a user-friendly interface with full Vietnamese language support.
- **Database structure:**
  - `users` table: Manages user information (name, email, password, last login time, etc.).
  - `categories` table: Manages personalized income/expense categories.
  - `transactions` table: Stores transactions with fields like amount, description, type, and category.
  - `chat_history` table: Records chatbot conversation history.
  - Utilizes UTF-8 (`utf8mb4`) encoding and indexes for query optimization.
- **Main features:**
  - User management: Register, login, update profile, change password.
  - Transaction management: Add, edit, delete, filter transactions by time and category.
  - Category management: Create, edit, delete income/expense categories.
  - Financial analysis: Display total income, total expenses, balance, income-to-expense ratio.
  - Financial chatbot: Answer common questions and analyze personal financial data.

- **Class Diagram:**



2. Application Development

- **Backend (PHP):**
  - User management: The `functions.php` file includes functions like `registerUser()`, `updateUserProfile()`, `updateUserPassword()`, using PDO for secure queries and password hashing with `password_hash`.
  - Transaction management: Functions such as `addTransaction()`, `updateTransaction()`, `getFilteredTransactions()` support adding, editing, and filtering transactions by type, month, year, and category.
  - Category management: Functions like `addCategory()`, `updateCategory()`, `deleteCategory()` enable personalized category management, with logic to update transactions when category names change.
  - Financial chatbot: The `FinancialChatbot` class handles common questions (e.g., "How to budget effectively?") and analyzes financial data (overview, spending, savings).
  - Key files:
    - `index.php`: Dashboard and login/registration interface.
    - `categories.php`: Category management with card-based UI and modals.

        - `transactions.php`: Transaction management with advanced filtering.
        - `profile.php`: Personal profile updates.
        - `logout.php`: Secure logout functionality.
    -
- **Frontend (HTML/CSS):**
    - Login/registration interface (`auth.php`) with a responsive design using CSS Flexbox for centering.
    - Navigation bar (`header.php`) with a dropdown menu displaying the user's name.
- **Chatbot:**
    - Built the `FinancialChatbot` class with methods like `processMessage()`, `getBudgetAdvice()`, `getSpendingAnalysis()` to answer questions and analyze data.
    - Currently supports fixed questions and basic analysis based on transaction data.

3. Testing

- Tested registration/login with valid and invalid cases (e.g., duplicate email, short password).
- Verified adding/editing/deleting transactions and categories, ensuring data consistency.
- Tested the chatbot with common questions and responses based on mock data.
- Ensured the interface works well on Chrome, Firefox, and mobile devices.

The chatbot is now functional and adds significant value to the platform, though further tuning will enhance its intelligence and versatility.

4. Other Tasks

- Enhanced security with `htmlspecialchars` and PDO prepared statements to prevent SQL Injection.
- Researched integration of real AI APIs (e.g., Grok) to upgrade the chatbot.

5. Next steps

- Upgrade the chatbot to handle natural questions better, possibly integrating an AI API.
- Add visual charts (e.g., Chart.js) for financial analysis.
- Optimize performance and security, test across multiple devices.
- Complete user documentation and final report.