**PERSONAL BUDGET**

**Hệ thống quản lý tài chính cá nhân**

Họ và tên: Lê Thành Vinh

Lớp: C2309G

Mã sinh viên: C2309G5975

**Mục lục**

[**A. Mục đích** 2](#_Toc186613411)

[**B. Ứng dụng trong thực tế** 2](#_Toc186613412)

[**1. Quản lý tài chính cá nhân hàng ngày** 2](#_Toc186613413)

[**2. Lập kế hoạch tiết kiệm** 2](#_Toc186613414)

[**3. Theo dõi và kiểm soát thói quen chi tiêu** 3](#_Toc186613415)

[**4. Tạo báo cáo tài chính cá nhân** 3](#_Toc186613416)

[**5. Tiết kiệm thời gian và tăng hiệu quả quản lý tài chính** 3](#_Toc186613417)

[**C. Công việc đã hoàn thành** 3](#_Toc186613418)

[**1. Thiết kế giao diện người dùng** 4](#_Toc186613419)

[**2. Thêm, sửa, xóa giao dịch** 4](#_Toc186613420)

[**3. Lọc giao dịch theo số tiền** 5](#_Toc186613421)

[**4. Sắp xếp giao dịch** 5](#_Toc186613422)

[**5. Tìm kiếm giao dịch** 5](#_Toc186613423)

[**6. Hiển thị biểu đồ trực quan** 5](#_Toc186613424)

[**7. Quản lý dữ liệu bằng cây nhị phân (Binary Tree)** 6](#_Toc186613425)

[**D.Phát triển tương lai** 6](#_Toc186613426)

[**1. Thêm tính năng Mục tiêu tiết kiệm (Savings Goals)** 6](#_Toc186613427)

[**2. Tìm kiếm nâng cao** 6](#_Toc186613428)

[3.**Xuất báo cáo** 7](#_Toc186613429)

[**4. Tính năng Đăng nhập và Đăng xuất** 7](#_Toc186613430)

[**E. Implementation** 8](#_Toc186613431)

[**1. Công nghệ sử dụng** 8](#_Toc186613432)

[**2. Thuật toán sử dụng** 8](#_Toc186613433)

**Giới thiệu đề tài: Hệ thống quản lý ngân sách cá nhân**

Trong cuộc sống hiện đại, việc quản lý tài chính cá nhân đã trở thành một kỹ năng quan trọng đối với mọi người. Tuy nhiên, nhiều cá nhân vẫn gặp khó khăn trong việc theo dõi các khoản thu nhập và chi tiêu một cách hiệu quả.

Hệ thống quản lý ngân sách cá nhân là một công cụ hỗ trợ người dùng ghi lại, phân loại và phân tích các giao dịch tài chính. Hệ thống không chỉ giúp người dùng hiểu rõ hơn về thói quen chi tiêu mà còn hỗ trợ lập kế hoạch tài chính cá nhân, đảm bảo sử dụng nguồn tiền một cách hợp lý và khoa học.

Hệ thống được thiết kế với các chức năng cơ bản như:

* Ghi nhận thu nhập và chi tiêu.
* Lọc và sắp xếp các giao dịch theo ngày, số tiền hoặc danh mục.
* Hiển thị biểu đồ trực quan để phân tích tỷ lệ thu nhập - chi tiêu.

# **A. Mục đích**

Hệ thống quản lý ngân sách cá nhân là một giải pháp phần mềm giúp người dùng ghi nhận, quản lý và phân tích dòng tiền thu nhập và chi tiêu. Mục tiêu chính của bài tập là:

1. Cung cấp một công cụ đơn giản nhưng hiệu quả để hỗ trợ cá nhân quản lý tài chính hàng ngày.
2. Xây dựng các chức năng cơ bản như thêm, sửa, xóa, lọc và sắp xếp các giao dịch thu nhập và chi tiêu.
3. Sử dụng biểu đồ trực quan (PieChart, BarChart) để phân tích tỷ lệ thu nhập và chi tiêu.

# **B. Ứng dụng trong thực tế**

## **1. Quản lý tài chính cá nhân hàng ngày**

* Người dùng có thể ghi nhận các giao dịch thu nhập như lương, thưởng, hoặc tiền từ các nguồn khác.
* Hệ thống cho phép ghi lại các khoản chi tiêu chi tiết như ăn uống, mua sắm, hóa đơn điện nước, hoặc các chi phí không thường xuyên.

## **2. Lập kế hoạch tiết kiệm**

* Hệ thống giúp người dùng phân tích tỷ lệ thu nhập và chi tiêu thông qua biểu đồ PieChart.
* Từ dữ liệu hiển thị, người dùng có thể điều chỉnh thói quen chi tiêu để đạt được mục tiêu tài chính cá nhân, ví dụ như tiết kiệm 20% thu nhập mỗi tháng.

## **3. Theo dõi và kiểm soát thói quen chi tiêu**

* Hệ thống cho phép người dùng lọc giao dịch theo giá trị số tiền, giúp họ xác định các khoản chi tiêu lớn nhất.
* Chức năng sắp xếp theo ngày tháng giúp người dùng kiểm tra chi tiêu theo tuần hoặc tháng, từ đó kiểm soát thói quen mua sắm.

## **4. Tạo báo cáo tài chính cá nhân**

* Người dùng có thể xem biểu đồ BarChart thống kê thu nhập và chi tiêu theo thời gian (hàng năm hoặc hàng tháng).
* Dữ liệu báo cáo giúp người dùng so sánh tình hình tài chính qua các giai đoạn và điều chỉnh kế hoạch tài chính cá nhân.

## **5. Tiết kiệm thời gian và tăng hiệu quả quản lý tài chính**

* Thay vì ghi chép thủ công, người dùng chỉ cần nhập thông tin giao dịch vào hệ thống.
* Chức năng sắp xếp, lọc và biểu đồ trực quan giúp người dùng nhanh chóng nắm bắt tình hình tài chính mà không cần tính toán phức tạp.

# **C. Công việc đã hoàn thành**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nhiệm vụ** | |  | | --- | |  |   **Trạng thái** | **Ghi chú** |
| Thiết kế giao diện | Hoàn thành | JavaFX TableView, PieChart, BarChart. |
| Thêm, sửa, xóa giao dịch | Hoàn thành | Quản lý giao dịch thu nhập và chi tiêu. |
| Lọc giao dịch theo số tiền | Hoàn thành | Lọc theo giá trị số tiền nhập vào. |
| Sắp xếp giao dịch | Hoàn thành | Bubble Sort, Insertion Sort, Selection Sort. |
| Tìm kiếm giao dịch | Hoàn thành | Tìm theo từ khóa trong phần mô tả. |
| |  | | --- | | Hiển thị biểu đồ thống kê |  |  | | --- | |  | | Hoàn thành | Biểu đồ trực quan hỗ trợ phân tích. |
| Quản lý dữ liệu | Hoàn thành | Lưu trữ và duyệt qua Binary Tree. |

## **1. Thiết kế giao diện người dùng**

- Mô tả:  
+ Giao diện được thiết kế với JavaFX để đảm bảo tính thân thiện và dễ sử dụng.

- Các thành phần giao diện chính bao gồm:

+ Bảng hiển thị giao dịch (TableView): Hiển thị danh sách các giao dịch thu nhập và chi tiêu.

+ Biểu đồ trực quan: PieChart và BarChart giúp người dùng phân tích dữ liệu.

+ Trường nhập liệu: Dùng để thêm mới hoặc chỉnh sửa thông tin giao dịch.

+ Các nút thao tác: Hỗ trợ các chức năng như thêm, sửa, xóa, lọc và sắp xếp giao dịch.

## **2. Thêm, sửa, xóa giao dịch**

**- Mô tả:** Hệ thống cung cấp các chức năng thao tác với dữ liệu giao dịch như:

**+ Thêm giao dịch:** Người dùng nhập thông tin (ngày, mô tả, số tiền, loại giao dịch) để thêm vào danh sách.

**+ Sửa giao dịch:** Chỉnh sửa thông tin của các giao dịch đã thêm.

**+ Xóa giao dịch:** Loại bỏ các giao dịch không còn cần thiết

## **3. Lọc giao dịch theo số tiền**

**- Mô tả:**Cung cấp chức năng lọc giao dịch dựa trên giá trị số tiền:

+ Hiển thị các giao dịch có giá trị nhỏ hơn hoặc bằng số tiền người dùng nhập.

+ Áp dụng được cho cả bảng thu nhập và bảng chi tiêu.

## **4. Sắp xếp giao dịch**

**- Mô tả:**Sử dụng các thuật toán sắp xếp như Bubble Sort, Insertion Sort, và Selection Sort để sắp xếp giao dịch theo:

**+ Ngày tháng:** Hiển thị danh sách giao dịch theo thứ tự thời gian.

**+ Số tiền:** Sắp xếp từ thấp đến cao hoặc ngược lại.

**+ Mô tả:** Sắp xếp các giao dịch theo thứ tự bảng chữ cái.

## **5. Tìm kiếm giao dịch**

**-Mô tả:** Hỗ trợ tìm kiếm giao dịch theo từ khóa trong phần mô tả. Người dùng nhập từ khóa, hệ thống sẽ:

+ Duyệt qua các giao dịch trong cây nhị phân (Binary Tree) để tìm kiếm.

+ Hiển thị kết quả tìm kiếm ngay trong bảng giao dịch.

## **6. Hiển thị biểu đồ trực quan**

**- Mô tả:** Sử dụng PieChart và BarChart của JavaFX để trực quan hóa dữ liệu tài chính:

**+ PieChart:** Thống kê tỷ lệ giữa thu nhập và chi tiêu.

**+ BarChart:** Hiển thị thu nhập và chi tiêu theo từng năm, giúp người dùng so sánh xu hướng tài chính.

## **7. Quản lý dữ liệu bằng cây nhị phân (Binary Tree)**

**- Mô tả:**Dữ liệu giao dịch được lưu trữ trong cây nhị phân để tối ưu hóa việc tìm kiếm, duyệt, và sắp xếp:

**+ Lưu trữ:** Cây nhị phân giúp lưu trữ dữ liệu có cấu trúc rõ ràng.

# **D.Phát triển tương lai**

## **1. Thêm tính năng Mục tiêu tiết kiệm (Savings Goals)**

- **Mô tả:** Cung cấp tính năng cho phép người dùng đặt mục tiêu tiết kiệm, theo dõi tiến độ và nhận nhắc nhở để đạt được mục tiêu tài chính cá nhân.

- **Chức năng cụ thể:**

+ **Đặt mục tiêu:** Người dùng nhập số tiền muốn tiết kiệm và thời gian cần đạt được.

**+ Theo dõi tiến độ:** Hiển thị phần trăm tiến độ tiết kiệm dựa trên số tiền thực tế đã tiết kiệm.

**+ Thông báo:** Nhắc nhở khi tiến độ chậm hoặc chúc mừng khi hoàn thành mục tiêu.

- **Triển khai:**

+ Thêm một tab hoặc cửa sổ mới cho tính năng "Mục tiêu tiết kiệm".

+ Sử dụng ProgressBar để hiển thị tiến độ tiết kiệm.

+ Tạo lớp SavingsGoal để lưu trữ thông tin mục tiêu.

## **2. Tìm kiếm nâng cao**

- **Mô tả** : Hiện tại, hệ thống chỉ hỗ trợ tìm kiếm theo từ khóa trong mô tả. Trong tương lai, chức năng này sẽ được mở rộng:

+ Tìm kiếm theo ngày tháng, số tiền, hoặc loại giao dịch.

+ Kết hợp nhiều tiêu chí tìm kiếm.

**- Triển khai:**

+ Thêm các trường nhập liệu hoặc lựa chọn (ChoiceBox, DatePicker) cho các tiêu chí tìm kiếm.

+ Kết hợp các điều kiện tìm kiếm trong truy vấn dữ liệu

## 3.**Xuất báo cáo**

- **Mô tả**: Thêm chức năng xuất báo cáo tài chính dưới dạng PDF hoặc Excel.

**- Triển khai**

+ Sử dụng thư viện Apache POI để xuất dữ liệu sang Excel.

## **4. Tính năng Đăng nhập và Đăng xuất**

- **Mô tả:**

+ Thêm chức năng đăng nhập để bảo vệ dữ liệu cá nhân và hỗ trợ quản lý nhiều người.

+Sau khi đăng nhập, người dùng chỉ có thể xem và quản lý các giao dịch của riêng mình.

+ Tính năng đăng xuất sẽ đảm bảo người dùng khác không truy cập được dữ liệu cá nhân.

- **Triển khai:**

+ Thiết kế giao diện đăng nhập với trường nhập **Tên đăng nhập** và **Mật khẩu**.

+ Quản lý thông tin người dùng thông qua lớp User và danh sách UserDataStore.

+ Sử dụng JavaFX để chuyển đổi giữa màn hình đăng nhập và màn hình chính.

# **E. Implementation**

## **1. Công nghệ sử dụng**

Hệ thống **Quản lý ngân sách cá nhân** được xây dựng bằng các công nghệ sau:

**- Ngôn ngữ lập trình: Java**

+ Java được sử dụng để phát triển logic ứng dụng và quản lý giao dịch.

+ Hỗ trợ mạnh mẽ cho các cấu trúc dữ liệu và thuật toán cần thiết.

**- Giao diện người dùng: JavaFX**

+ JavaFX được sử dụng để thiết kế giao diện trực quan và thân thiện với người dùng.

+ Hỗ trợ các thành phần như TableView, PieChart, BarChart, ProgressBar để hiển thị dữ liệu và biểu đồ.

**- CSS**

+ CSS được sử dụng để tùy chỉnh giao diện người dùng của JavaFX.

+ Hỗ trợ thiết kế màu sắc, kiểu chữ và bố cục, giúp giao diện trông chuyên nghiệp và dễ sử dụng hơn.

## **2. Thuật toán sử dụng**

Hệ thống sử dụng các thuật toán sau để xử lý dữ liệu :

**a) Thuật toán Binary Tree**

**- Mục đích:**

+ Lưu trữ và tổ chức các giao dịch.

+ Duyệt qua dữ liệu một cách nhanh chóng để sắp xếp hoặc hiển thị.

**- Cách hoạt động:**

+ Kiểm tra cây rỗng:

* Nếu cây rỗng (root == null), thêm giao dịch làm root (gốc).

+ Xác định loại giao dịch:

* **Income (Thu nhập):** Các giao dịch thu nhập được ưu tiên lưu ở **cây con bên trái**.
* **Expense (Chi tiêu):** Các giao dịch chi tiêu được ưu tiên lưu ở **cây con bên phải**.

+So sánh ngày giờ giây (LocalDateTime):

* Nếu giao dịch thuộc **Income**:
  + - Nếu thời gian của giao dịch mới nhỏ hơn thời gian của nút hiện tại, di chuyển đến **cây con bên trái**.
    - Nếu thời gian lớn hơn hoặc bằng, di chuyển đến **cây con bên phải**.
* Nếu giao dịch thuộc **Expense**:
  + - Nếu thời gian của giao dịch mới nhỏ hơn thời gian của nút hiện tại, di chuyển đến **cây con bên trái**.
    - Nếu thời gian lớn hơn hoặc bằng, di chuyển đến **cây con bên phải**.

+ Lặp lại đệ quy:

* Tiếp tục so sánh và di chuyển qua các nút trong cây cho đến khi tìm được vị trí trống thích hợp để thêm giao dịch.

+ Ứng dụng trong hệ thống:

* Khi người dùng thêm giao dịch mới, dữ liệu được thêm vào cây theo thứ tự.
* Hiển thị giao dịch trên TableView được thực hiện bằng cách duyệt cây.

**1. Bubble Sort (Sắp xếp nổi bọt)**

**- Ý tưởng:**

+ Duyệt qua danh sách nhiều lần.

+ Tại mỗi lần duyệt, so sánh từng cặp phần tử liền kề.

+ Nếu cặp phần tử không đúng thứ tự, hoán đổi chúng.

+ Quá trình này được lặp lại đến khi không còn phần tử nào cần hoán đổi.

**- Quy trình:**

+ Bắt đầu từ đầu danh sách.

+ So sánh phần tử hiện tại với phần tử tiếp theo.

+ Nếu phần tử hiện tại lớn hơn phần tử tiếp theo (theo tiêu chí sắp xếp), hoán đổi vị trí.

+ Tiếp tục duyệt đến cuối danh sách, sau mỗi vòng lặp, phần tử lớn nhất được đưa về đúng vị trí cuối cùng.

+ Giảm phạm vi duyệt sau mỗi vòng lặp, vì các phần tử cuối đã được sắp xếp.

**2. Insertion Sort (Sắp xếp chèn)**

**-Ý tưởng:**

+ Bắt đầu với một danh sách con đã được sắp xếp (ban đầu chỉ chứa một phần tử).

+ Lấy từng phần tử trong danh sách chính, chèn nó vào đúng vị trí trong danh sách con đã được sắp xếp.

+ Dịch chuyển các phần tử lớn hơn để tạo khoảng trống cho phần tử mới.

**- Quy trình:**

+ Duyệt qua danh sách từ phần tử thứ hai đến cuối.

+ So sánh phần tử hiện tại với các phần tử trong danh sách con đã sắp xếp.

+ Dịch chuyển các phần tử lớn hơn sang phải để tạo khoảng trống.

+ Chèn phần tử hiện tại vào đúng vị trí trong danh sách con.

**3. Selection Sort (Sắp xếp chọn)**

**- Ý tưởng:**

+ Tìm phần tử nhỏ nhất (hoặc lớn nhất) trong danh sách và đưa nó lên đầu.

+ Tiếp tục tìm phần tử nhỏ nhất trong danh sách còn lại và đặt nó vào vị trí kế tiếp.

+ Quá trình này được lặp lại cho đến khi toàn bộ danh sách được sắp xếp.

**- Quy trình:**

+ Duyệt qua toàn bộ danh sách để tìm phần tử nhỏ nhất.

+ Đưa phần tử nhỏ nhất lên đầu danh sách.

+ Tiếp tục tìm phần tử nhỏ nhất trong danh sách còn lại.

+ Lặp lại cho đến khi danh sách được sắp xếp.