**NHẬP MÔN CÔNG NGHỆ PHẦN MỀM**

**TÀI LIỆU THIẾT KẾ**

*Yêu cầu nhóm sinh viên hoàn thành tài liệu thiết kế cho đồ án đã được giao theo biểu mẫu đính kèm.*

C:\Users\tdqua_000\Dropbox\SS-Slides\DeCuong-CDIO\Template CDIO v4.2\Templates\Hinh anh\LogoTruong.png

Bộ môn Công nghệ phần mềm

Khoa Công nghệ thông tin

Đại học Khoa học tự nhiên TP HCM

**MỤC LỤC**

[1 Thông tin nhóm 2](#_Toc383898123)

[2 Mô hình quan niệm 3](#_Toc383898124)

[3 Thiết kế kiến trúc 4](#_Toc383898125)

[3.1 Sơ đồ kiến trúc 4](#_Toc383898126)

[3.2 Sơ đồ lớp (Class Diagram) 4](#_Toc383898127)

[3.3 Đặc tả các lớp đối tượng 4](#_Toc383898128)

[3.3.1 Lớp C1 4](#_Toc383898129)

[3.3.2 Lớp C2 5](#_Toc383898130)

[4 Thiết kế dữ liệu 6](#_Toc383898131)

[4.1 Sơ đồ dữ liệu 6](#_Toc383898132)

[4.2 Đặc tả dữ liệu 6](#_Toc383898133)

[5 Thiết kế giao diện người dùng 7](#_Toc383898134)

[5.1 Sơ đồ và danh sách màn hình 7](#_Toc383898135)

[5.2 Đặc tả các màn hình giao diện 7](#_Toc383898136)

[5.2.1 Màn hình “A” 7](#_Toc383898137)

[5.2.2 Màn hình “B” 7](#_Toc383898138)

**TÀI LIỆU THIẾT KẾ**

Tài liệu tập trung vào các chủ đề:

* Tạo ra tài liệu thiết kế phần mềm.
* Hoàn chỉnh tài liệu thiết kế phần mềm với các nội dung:

Hiển thị dữ liệu phức tạp

* DataGridView
  + Mô hình quan niệm
  + Thiết kế kiến trúc
  + Thiết kế dữ liệu
  + Thiết kế giao diện người dùng
* Đọc hiểu tài liệu thiết kế phần mềm.

# Thông tin nhóm

**Website / Facebook nhóm:** <https://www.facebook.com/>Group

|  |  |  |  |
| --- | --- | --- | --- |
| **MSSV** | **Họ Tên** | **Email** | **Điện thoại** |
| *1612001* | Hoàng Dân An | anaizhu98@gmail.com | 0772108866 |
| *1612008* | Lê Trần Lâm An |  |  |
| *1612043* | Phan Ngọc Thanh Bình |  |  |
| *1612169* | Nguyễn Bá Thắng |  |  |

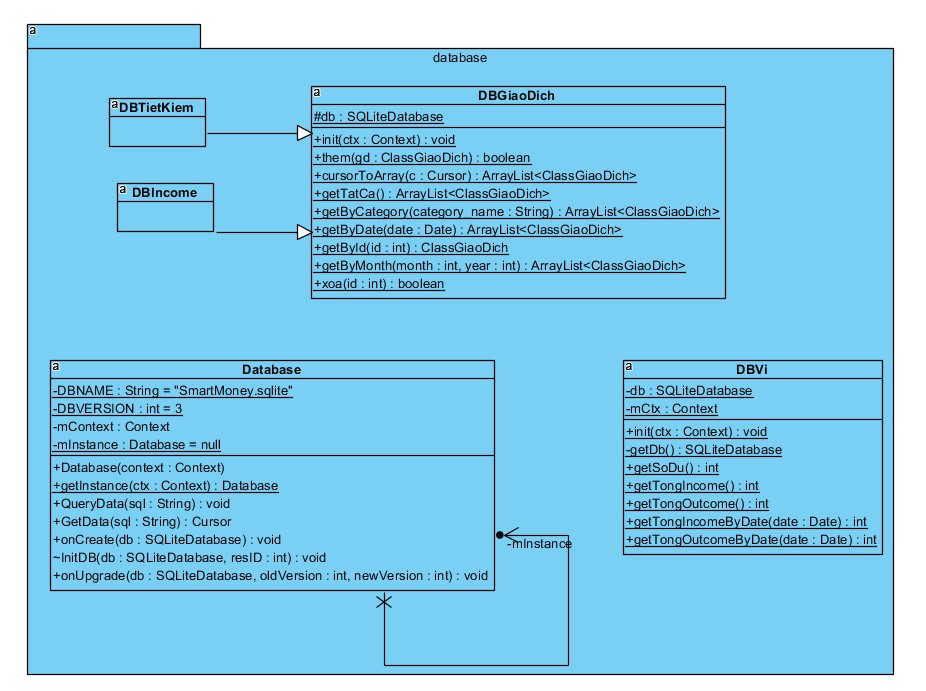
# Mô hình quan niệm

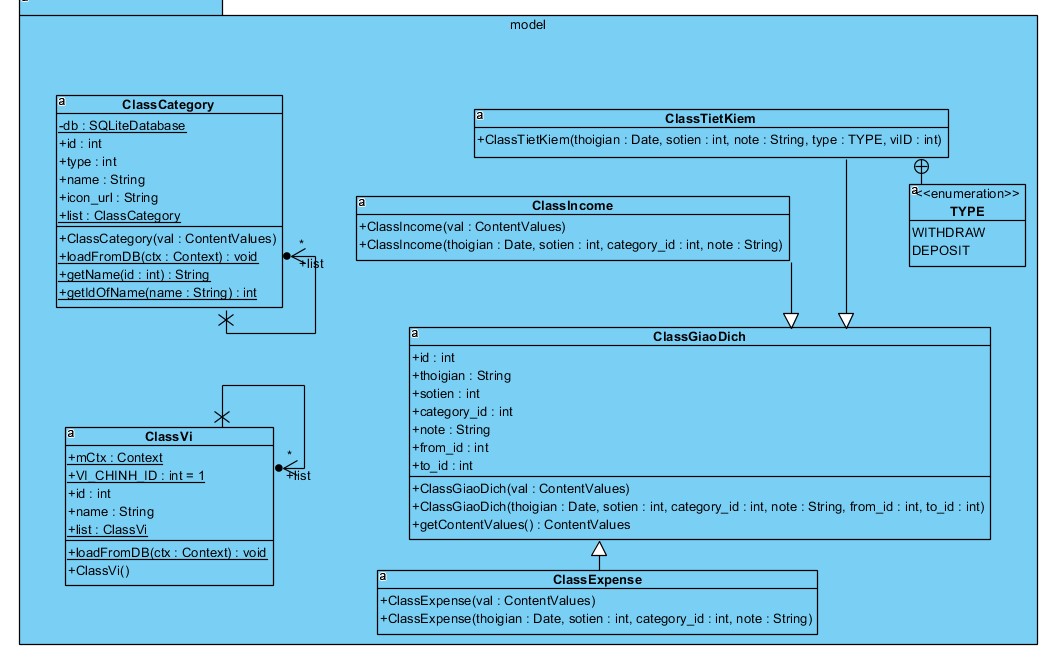
[Mô hình quan niệm dưới được tạo bỏi Visual Diagram]

# Thiết kế kiến trúc

## Sơ đồ kiến trúc

## Sơ đồ lớp (Class Diagram)





## Đặc tả các lớp đối tượng

*[Sinh viên chọn và trình bày đặc tả của vài (3-4) lớp đối tượng quan trọng nhất]*

### Lớp Database (kết thừa SQLiteOpenHelper)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Loại** | **Ràng buộc** | **Ý nghĩa** |
| 1 | DBNAME | Private static final |  | Tên database |
| 2 | DBVERSION | Private static final |  | Version của database, giúp ích cho việc update & replace các table hoặc data trong database cũ (qua hàm **onUpdate** của SQLOpenHelper) |
| 3 | mContext | Private |  | Lưu đối tượng context của application, dùng để yêu cầu connect với database |
| 4 | mInstance | Public static |  | Do SQLite chỉ cho phép open duy nhất 1 kết nối, để sử dụng ở nhiều class khác nhau ta phải tạo nhiều instance dạng static để các Đối tượng khác truy cập |

*[Danh sách các các phương thức chính]*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên phương thức** | **Loại** | **Ràng buộc** | **Ý nghĩa** |
| 1 | getInstance | Public static |  | Cung cấp 1 instance cho class khác |
| 2 | onCreate (kế thừa) | Public |  | Khi database được tạo mới (lần đầu sử dụng) |
| 3 | onUpdate (kế thừa) | Public |  | Khi database đã có sẵn (từ lần chạy trước). Hàm này sẽ gọi lên để xử lý cập nhật (nếu có) |
| 4 | InitDB(SQLiteDatabase db, int resID) | Private |  | Tạo database & table được định nghĩ ở file .sql đã có (lấy theo resource resID) |

### Lớp DBGiaoDich

*Dùng để thao tác với Database theo đối tượng GiaoDich*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Loại** | **Ràng buộc** | **Ý nghĩa** |
| 1 | db | Protected static |  | Một databaseHelper instance được tạo ra từ class Database |

*[Danh sách các các phương thức chính]*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên phương thức** | **Loại** | **Ràng buộc** | **Ý nghĩa** |
| 1 | them(ClassGiaoDich) | Public static |  | Thực hiện thao tác thêm info giao dịch từ ClassGiaoDich vào database |
| 2 | getByDate(Date) | Public static |  | Lấy danh sách info các giao dịch theo ngày |
| 3 | getById(int) | Public static |  | Trả về ClassGiaoDich theo id của giao dịch |
| 4 | xoa(int) | Public static |  | Xóa giao dịch theo id |
| 5 | update(ClassGiaoDich) | Public static |  | cập nhật giao dịch theo thông tin classGiaoDich đã có |
|  |  |  |  |  |

### Lớp ClassGiaoDich

*Class Model để lưu thông tin 1 giao dịch*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Loại** | **Ràng buộc** | **Ý nghĩa** |
| 1 | id | public |  |  |
| 2 | thoigian | public |  | thời gian thực hiện giao dịch |
| 3 | sotien | public |  | số tiền của giao dịch |
| 4 | category\_id | public |  | id category của giao dich |
| 5 | note | public |  | ghi chú thêm của giao dich |
| 6 | from\_id, to\_id | public |  | 2 id của Class Ví mà số tiền từ đâu và đến đâu |
|  |  |  |  |  |

### Lớp ClassIncome

*Kế thừa ClassGiaoDich, income = một giao dịch đi từ chỗ khác vào ví tiền*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Loại** | **Ràng buộc** | **Ý nghĩa** |
| 1 | from\_id | private static | luôn = 0 | Chỗ nào đó |
| 2 | to\_id | private static | luôn = 1 | Ví chính |

### Lớp ClassExpense

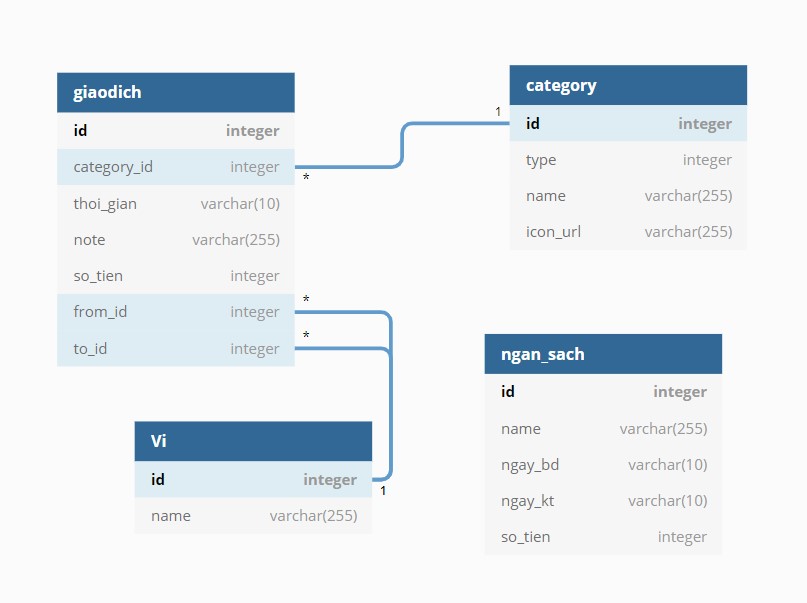
*Kế thừa ClassGiaoDich, expense = một giao dịch đi từ ví tiền đi ra chỗ khác*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Loại** | **Ràng buộc** | **Ý nghĩa** |
| 1 | from\_id | private static | luôn = 1 | Ví chính |
| 2 | to\_id | private static | luôn = 0 | Chỗ khác |

# Thiết kế dữ liệu

## Sơ đồ dữ liệu

*[Mô hình dữ liệu dưới được tạo bằng* [*https://dbdiagram.io*](https://dbdiagram.io) *]*



## Đặc tả dữ liệu

Bảng **Vi**:

* Dùng để lưu thông tin Ví chính và các sổ tiết kiệm, gồm
  + **ID**: Nếu là Ví chính, ID = 1 (mặc định). Nếu là sổ tiết kiệm, ID lớn hơn 1.
  + **Name**: tên của ví (sổ tiết kiệm).

Bảng **Transaction**:

* Dùng để lưu thông tin một giao dịch gồm:
  + **Category\_id:** phân loại của transaction đó (Thu/Chi, Food/Drinks/…)
  + **Thoi\_gian:** Ngày (giờ) thực hiện giao dịch này
  + **Ghi\_chu:** ghi chú thêm về giao dịch này
  + **So\_tien:** số tiền của giao dịch
  + **From, To:** Nếu là thu (income), from = 0 và to = 1 (id số 1 của ví chính). Nếu là chi tiêu (expense), from = 1 (ví chính) và to = 0. Nếu là một giao dịch như là thêm tiết kiệm hoặc rút tiết kiệm, from và to sẽ là 2 id của số tiết kiệm hoặc Ví chính.

Bảng **Category**:

* Dùng để lưu thông tin các phân loại, có sẵn và cho người dung thêm vào, gồm:
  + **Type**: là 1 boolean (0 và 1), 0 ứng với thu (income), 1 ứng với chi (expense)
  + **Name**: Tên của phân loại đó. VD: “Ăn uống”, “Đi lại”, “Lương”, ….
  + **Icon\_url**: Lưu icon cho phân loại, dành cho UI. Để người có thể tùy chọn icon theo ý thích.

Bảng **Ngan\_Sach**:

* Dùng để lưu các Số ngân sách cho user, gồm:
  + **ID**
  + **Name**: tên phân biệt các sổ với nhau
  + **NgayBD**: ngày bắt đầu tính để trừ vào ngân sách
  + **NgayKT**: ngày kết thúc
  + **So\_tien**: ngân sách dự tính từ ngày bắt đầu đến ngày kết thúc

# Thiết kế giao diện người dùng

## Sơ đồ và danh sách màn hình

*[Vẽ sơ đồ màn hình, thể hiện mối quan hệ và sự chuyển tiếp giữa các màn hình]*

*[Liệt kê danh sách các màn hình]*

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên màn hình** | **Ý nghĩa/ Ghi chú** |
|  |  |  |

## Đặc tả các màn hình giao diện

*[Sinh viên chọn và trình bày đặc tả của vài (3-4) màn hình quan trọng nhất. Các màn hình khác chỉ cần vẽ thiết kế giao diện của màn hình]*

### Màn hình “A”

*[Mô tả rõ hình thức trình bày và các xử lý đối với từng biến cố trong màn hình. Kèm theo hình ảnh thiết kế từng màn hình]*

### Màn hình “B”

*[Mô tả rõ hình thức trình bày và các xử lý đối với từng biến cố trong màn hình. Kèm theo hình ảnh thiết kế từng màn hình]*