TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI

KHOA CÔNG NGHỆ THÔNG TIN

---------o0o---------

A blue and yellow logo

Description automatically generated

BÁO CÁO ĐẶC TẢ PHẦN MỀM

Đề tài: Ứng dụng quản lý thời gian biểu

**Giảng viên : Trần Văn Dũng**

**Sinh viên thực hiện : Vũ Bảo Lâm**

**Mã sinh viên : 211241205**

**Lớp : Cử nhân công nghệ thông tin 1**

**Khóa : 62**

**Hà Nội -2024**

1. TỔNG QUAN ĐỀ TÀI
   1. Mô tả
      1. Mô tả hệ thống

Hiện nay với sự phát triển nhanh chóng của công nghệ thông tin ngày càng hiện đại, việc quản lý thời gian cũng trở nên dễ dàng hơn vơi ứng dụng thời gian biểu. Hệ thống ứng dụng thời gian biểu là một ứng dụng phần mềm được thiết kế để giúp người dùng tổ chức và quản lý thời gian học tập và công việc một cách hiệu quả hơn.

* + 1. Mô tả các nghiệp vụ cơ bản

Ứng dụng thời gian biểu bao gồm những nghiệp vụ chủ yếu sau:

* Tạo nhắc nhở công việc: Người dùng có thể thêm mới các công việc của họ, bao gồm các thông tin như tên công việc, ngày thực hiện, thời gian bắt đầu, thời gian kết thúc, thẻ phân loại công việc, …
* Chỉnh sửa và xóa công việc: Người dùng có thể chỉnh sửa hoặc xóa các công việc đã được tạo trước đó nếu cần thiết, để cập nhật thông tin hoặc loại bỏ những sự kiện không còn cần thiết nữa.
* Xem và quản lý công việc: Người dùng có thể xem công việc của họ theo các chế độ khác nhau như tuần, tháng, hoặc danh sách, giúp họ có cái nhìn tổng quan và chi tiết về công việc của mình. Họ cũng có thể quản lý sự kiện bằng cách di chuyển, sao chép hoặc thay đổi thời gian của chúng.
* Phân loại và lọc công việc: Hệ thống cho phép người dùng phân loại công việc thành các danh mục khác nhau và sử dụng tính năng lọc để tìm kiếm và hiển thị chỉ những công việc cụ thể mà họ quan tâm.

1. PHÂN TÍCH VÀ THIẾT KẾ
   1. Sơ đồ phân rã chức năng

A black and white screen with white rectangles

Description automatically generated

* 1. Đặc tả các chức năng
     1. Quản lý nhiệm vụ (Task Management)
        1. Thêm nhiệm vụ mới (Add Task):
* Người dùng có thể thêm tên công việc, ngày, giờ bắt đầu/kết thúc, tần suất lặp lại, và gắn thẻ (Tag).
* Hỗ trợ lưu công việc có thông tin đầy đủ vào cơ sở dữ liệu.
  + - 1. Hiển thị danh sách nhiệm vụ (Task List)
* Danh sách công việc được hiển thị theo ngày tháng cụ thể, tự động cập nhật khi người dùng chọn ngày trên **CalendarView**.
* Công việc có thể được tìm kiếm theo từ khóa trong tên nhiệm vụ.
  + - 1. Chỉnh sửa công việc (Edit Task)
* Người dùng có thể chỉnh sửa công việc đã tạo, bao gồm các thông tin chi tiết như thời gian, tên nhiệm vụ, và Tag.
* Dữ liệu được cập nhật trong cơ sở dữ liệu khi chỉnh sửa hoàn tất.
  + - 1. Xóa và phục hồi công việc (Delete and Restore Tasks)
* Công việc có thể được xóa thông qua thao tác vuốt ngang trên **RecyclerView**.
* Công việc bị xóa sẽ được đưa vào **Thùng rác (Bin)** và có thể phục hồi hoặc xóa vĩnh viễn tại đây.
  + 1. Quản lý thẻ (Tag Management)
       1. Thêm thẻ mới (Add Tag)
* Người dùng có thể tạo thẻ với tên, biểu tượng, và màu sắc tùy chỉnh.
* Thẻ được sử dụng để phân loại hoặc gắn nhãn nhiệm vụ.
  + - 1. Hiển thị danh sách thẻ (Tag List)
* Danh sách các thẻ hiển thị trên giao diện **TagFragment** với đầy đủ thông tin (tên, biểu tượng, màu sắc).
  + - 1. Chỉnh sửa thẻ (Edit Tag)
* Hỗ trợ chỉnh sửa tên, biểu tượng, và màu sắc của thẻ.
* Dữ liệu thẻ được cập nhật trong cơ sở dữ liệu và hiển thị lại danh sách.
  + - 1. Xóa thẻ (Delete Tag)
* Người dùng có thể xóa thẻ. Các công việc liên quan đến thẻ đó sẽ bị cập nhật trạng thái.
  + 1. Chức năng lịch (Calendar Integration)
       1. Chọn ngày hiển thị nhiệm vụ (Task Filtering by Date)
* Sử dụng **CalendarView**, người dùng có thể chọn ngày để xem danh sách công việc tương ứng.
* Ứng dụng tự động lọc công việc trong cơ sở dữ liệu dựa trên ngày được chọn.
  + - 1. Tự động hiển thị nhiệm vụ hôm nay (Today's Task Auto-load)
* Khi mở ứng dụng, danh sách công việc của ngày hiện tại sẽ tự động hiển thị.
  + 1. Tìm kiếm và lọc (Search and Filter)
       1. Tìm kiếm nhiệm vụ (Task Search)
* Hỗ trợ tìm kiếm công việc trong **Thùng rác (Bin)** bằng cách nhập từ khóa tên công việc.
* Danh sách công việc được lọc ngay lập tức dựa trên từ khóa.
  + - 1. Tìm kiếm nâng cao theo thẻ (Advanced Filtering by Tag)
* Công việc có thể được hiển thị theo từng thẻ cụ thể để hỗ trợ quản lý tốt hơn.
  + 1. Xử lý dữ liệu trong Thùng rác (Bin Management)
       1. Khôi phục nhiệm vụ (Restore Tasks)
* Công việc trong thùng rác có thể được khôi phục về danh sách nhiệm vụ chính.
  + - 1. Xóa vĩnh viễn công việc (Permanent Deletion)
* Người dùng có thể xóa vĩnh viễn từng công việc hoặc toàn bộ công việc trong thùng rác.
  1. Thiết kế cơ sở dữ liệu
     1. Sơ đồ quan hệ thực thể

A diagram of a task

Description automatically generated

* + 1. Đặc tả các thực thể
       1. Mô tả bảng Task

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu dữ liệu | Diễn giải |
| 1 | TASK\_ID | INTEGER | Là khóa chính để phân biệt các nhiệm vụ với nhau |
| 2 | TASK\_NAME | TEXT | Tên nhiệm vụ |
| 3 | TASK\_DATE | TEXT | Ngày bắt đầu nhiệm vụ |
| 4 | TASK\_START | TEXT | Thời gian bắt đầu nhiệm vụ trong ngày |
| 5 | TASK\_END | TEXT | Thời gian kết thúc nhiệm vụ trong ngày |
| 6 | TASK\_REPEAT | TEXT | Dùng để lựa chọn sự lặp lại cho một nhiệm vụ |
| 7 | TASK\_NOTIFICATION | TEXT | Dùng chọn có thông báo ra ngoài màn hình không |
| 8 | TASK\_CHECKED | INTEGER | Phân biệt với nhiệm vụ đã được cho vào thùng rác |
| 9 | TAG\_ID | INTEGER | Kết nối bảng TASK với bản TAG thông qua TAG\_ID |

* + - 1. Mô tả bảng TAG

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu dữ liệu | Diễn giải |
| 1 | TAG\_ID | INTEGER | Là khóa chính để phân biệt với các tag khác |
| 2 | TAG\_NAME | TEXT | Tên tag |
| 3 | TAG\_ICON | INTEGER | Nhãn dán đi kèm với tag |
| 4 | TAG\_COLOR | INTEGER | Màu sắc đi kèm với tag |

* 1. Cài đặt cơ sở dữ liệu
     1. Khai báo các trường dữ liệu

private Context context;  
 private static final String *DATABASE\_NAME* = "Demo001.db";  
 private static final int *DATABASE\_VERSION* =1;  
 private static final String *TB\_TASK* = "TASK";  
 private static final String *TB\_TAG* = "TAG";  
  
 private static final String *TB\_TASK\_NAME* = "TASK\_NAME";  
 private static final String *TB\_TASK\_DATE* = "TASK\_DATE";  
 private static final String *TB\_TASK\_ID* = "TASK\_ID";  
 private static final String *TB\_TASK\_REPEAT*="TASK\_REPEAT";  
 private static final String *TB\_TASK\_START* = "TASK\_START";  
 private static final String *TB\_TASK\_END* = "TASK\_END";  
 private static final String *TB\_TASK\_TAG* = "TAG\_ID";  
 private static final String *TB\_TASK\_NOTIFICATION* = "TASK\_NOTIFICATION";  
 private static final String *TB\_TASK\_CHECK*= "TASK\_CHECKED";

public static final String *TB\_TAG\_NAME* = "TAG\_NAME";  
 public static final String *TB\_TAG\_ID* = "TAG\_ID";  
 public static final String *TB\_TAG\_ICON* = "TAG\_ICON";  
 public static final String *TB\_TAG\_COLOR* = "TAG\_COLOR";

public CreateDatabase(@Nullable Context context) {  
  
 super(context, *DATABASE\_NAME*, null, *DATABASE\_VERSION*);  
 this.context = context;  
 }

* + 1. Hàm tạo bảng dữ liệu Task và Tag

@Override  
public void onCreate(SQLiteDatabase db) {  
  
 String tbTask = "CREATE TABLE IF NOT EXISTS " + *TB\_TASK* + " ("  
 + *TB\_TASK\_ID* + " INTEGER PRIMARY KEY AUTOINCREMENT, "  
 + *TB\_TASK\_NAME* + " TEXT, "  
 + *TB\_TASK\_DATE* + " TEXT, "  
 + *TB\_TASK\_START* + " TEXT, "  
 + *TB\_TASK\_END* + " TEXT, "  
 + *TB\_TASK\_REPEAT* + " TEXT, "  
 + *TB\_TASK\_TAG* + " INTEGER, "  
 + *TB\_TASK\_NOTIFICATION* + " TEXT, "  
 + *TB\_TASK\_CHECK* + " INTEGER, "  
 + "FOREIGN KEY (" + *TB\_TASK\_TAG* + ") REFERENCES " + *TB\_TAG* + "(" + *TB\_TAG\_ID* + ") ON DELETE CASCADE)";  
 db.execSQL(tbTask);  
  
 String tbTag = "CREATE TABLE IF NOT EXISTS " + *TB\_TAG* + " ("  
 + *TB\_TAG\_ID* + " INTEGER PRIMARY KEY AUTOINCREMENT, "  
 + *TB\_TAG\_NAME* + " TEXT, "  
 + *TB\_TAG\_ICON* + " INTEGER, "  
 + *TB\_TAG\_COLOR* + " INTEGER)";  
 db.execSQL(tbTag);  
}

* + 1. Chèn 5 bản ghi dữ liệu mẫu
       1. Hàm Insert Task và Tag

*// Hàm kiểm tra bảng có rỗng hay không*private boolean isTableEmpty(SQLiteDatabase db, String tableName) {  
 Cursor cursor = db.rawQuery("SELECT COUNT(\*) FROM " + tableName, null);  
 boolean isEmpty = false;  
 if (cursor.moveToFirst()) {  
 isEmpty = cursor.getInt(0) == 0;  
 }  
 cursor.close();  
 return isEmpty;  
}  
private void insertTask(SQLiteDatabase db, String name, String date, String start, String end,  
 String repeat, int tagId, String check, String notification) {  
 ContentValues values = new ContentValues();  
 values.put(*TB\_TASK\_NAME*, name);  
 values.put(*TB\_TASK\_DATE*, date);  
 values.put(*TB\_TASK\_START*, start);  
 values.put(*TB\_TASK\_END*, end);  
 values.put(*TB\_TASK\_REPEAT*, repeat);  
 values.put(*TB\_TAG\_ID*, tagId);  
 values.put(*TB\_TASK\_CHECK*, check);  
 values.put(*TB\_TASK\_NOTIFICATION*, notification);  
 db.insert(*TB\_TASK*, null, values);  
}  
private void insertTag(SQLiteDatabase db, String name, int icon, int color) {  
 ContentValues values = new ContentValues();  
 values.put(*TB\_TAG\_NAME*, name);  
 values.put(*TB\_TAG\_ICON*, icon);  
 values.put(*TB\_TAG\_COLOR*, color);  
 db.insert(*TB\_TAG*, null, values);  
}

* + - 1. Insert dữ liệu vào bảng Task và Tag

public void insertSampleData() {  
 SQLiteDatabase db = this.getWritableDatabase();  
  
 *// Kiểm tra nếu bảng đã có dữ liệu thì không thêm nữa* if (isTableEmpty(db, *TB\_TASK*) && isTableEmpty(db, *TB\_TAG*)) {  
  
 *// Thêm dữ liệu mẫu cho bảng Tags* insertTag(db, "EAT", 2131230912, -354250);  
 insertTag(db, "LAUNDRY", 2131230914, -2455817);  
 insertTag(db, "GYM", 2131230921, -7944457);  
 insertTag(db, "READING", 2131230918, -555347);  
 insertTag(db, "STUDY", 2131230919, -7952905);  
  
 *// Thêm dữ liệu mẫu cho bảng Tasks* insertTask(db, "LUNCH", "10/12/2024", "12:00", "12:20",  
 "Do not repeat", 1, "1", "test");  
 insertTask(db, "Do laundry", "10/12/2024", "6:00", "6:30",  
 "Do not repeat", 2, "1", "test");  
 insertTask(db, "GYM", "10/12/2024", "05:30", "06:00",  
 "Do not repeat", 3, "1", "test");  
 insertTask(db, "READ BOOKS", "10/12/2024", "18:30", "19:00",  
 "Do not repeat", 4, "1", "test");  
 insertTask(db, "EXAM", "11/12/2024", "09:30", "10:30",  
 "Do not repeat", 5, "1", "test");  
 }  
 db.close();  
}

1. CÀI ĐẶT CHƯƠNG TRÌNH
   1. Đặc tả 2 form cài đặt và giao diện chương trình
      1. Thẻ phân loại Tag

|  |  |
| --- | --- |
| 1. Người dùng chọn TagFragment ở dưới BottomNavigationBar. Giao diện TagFragment sẽ hiện ra và hiển thị danh sách các loại thẻ phân loại hiện đang có trong ứng dụng.   A screenshot of a cell phone  Description automatically generated  Hình giao diện TagFragment | 1. Giao diện TagFragment có chức năng thêm, sửa, xoá những thẻ Tag đang có. Khi người dùng bấm vào nút thêm giao diện AddTag sẽ hiện ra cho phép tạo thêm thẻ Tag mới.   Người dùng có thể tạo thẻ Tag với tên, biểu tượng, và màu sắc tùy chỉnh.  A group of icons on a white background  Description automatically generated  Hính giao diện AddTag |

* + 1. Công việc Task

|  |  |
| --- | --- |
| 1. Người dùng chọn TaskFragment bên dưới thanh BottomNavigationBar. Giao diện thêm mới công việc (AddTask) hiện ra.   Người dùng có thể thêm tên công việc, ngày, giờ bắt đầu/kết thúc, tần suất lặp lại, và gắn thẻ (Tag). Khi bấm nút thêm, công việc mới sẽ được thêm vào dữ liệu và hiển thị bên lịch.  A screenshot of a phone  Description automatically generated  Hình giao diện AddTask | 1. Sau khi nhiệm vụ được thêm vào cơ sở dữ liệu, công việc được hiển thị theo danh sách ở giao diện lịch (Calendar   Fragment) và chọn được ngày để xem công việc được thêm  Người dụng có thể chọn xoá tạm thời công việc, có thể sửa chi tiết công việc.    Hình giao diện Calendar |
| 1. Sau khi xoá tạm thời công việc, công việc bị xoá sẽ hiển thị bên thùng rác Bin   Tại đây người dùng có thể thao tác xoá vĩnh viễn công việc hoặc chọn khôi phục lại công việc bị xoá tạm thời    Hình giao diện Bin | 1. Công việc đươc thêm vào cơ sở dữ liệu được hiển thị toàn bộ ở phần tìm kiếm FindTask.   Người dùng có thể tìm kiếm công việc theo tên    Hình giao diện FindTask |