**TRƯỜNG ĐẠI HỌC HẠ LONG**

**KHOA CÔNG NGHỆ THÔNG TIN**

- - - - 🙞🙞🟔🙜🙜 - - - -

Logo

Description automatically generated

**BÁO CÁO KẾT THÚC HỌC PHẦN**

***TÊN ĐỀ TÀI:***

**XÂY DỰNG PHẦN MỀM LẬP KẾ HOẠCH HOẠT ĐỘNG NHÓM**

|  |  |
| --- | --- |
| **Sinh viên thực hiện:** | **Nguyễn Văn Thạch**  **Phạm Minh Thiên** |
| **Giảng viên hướng dẫn:** | **TS. Phạm Thanh Huyền** |

*Quảng Ninh, tháng 12 năm 2023*



**TRƯỜNG ĐẠI HỌC HẠ LONG**

**KHOA CÔNG NGHỆ THÔNG TIN**

- - - - 🙞🙞🟔🙜🙜 - - - -

Logo

Description automatically generated

**BÁO CÁO KẾT THÚC HỌC PHẦN**

***TÊN ĐỀ TÀI:***

**XÂY DỰNG PHẦN MỀM LẬP KẾ HOẠCH HOẠT ĐỘNG NHÓM**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Mã sinh viên** | **Họ và Tên** | **Ngày sinh** | **Điểm** | |
| **Bằng số** | **Bằng chữ** |
| **1** | **20DH03059** | **Phạm Minh Thiên** | **01/07/2002** |  |  |
| **2** | **20DH03056** | **Nguyễn Văn Thạch** | **16/02/2002** |  |  |

|  |  |
| --- | --- |
| **CÁN BỘ CHẤM THI 1** | **CÁN BỘ CHẤM THI 2** |

*Quảng Ninh, tháng 12 năm 2023*



**LỜI CẢM ƠN**

Chúng em xin gửi lời cảm ơn chân thành nhất đến cô Phạm Thanh Huyền đã giảng dạy chúng em học phần lập trình Python.

Qua học phần này, chúng em đã có cơ hội được học hỏi rất nhiều về ngôn ngữ lập trình Python cũng như các kiến thức liên quan đến phát triển phần mềm. Những bài giảng sôi nổi và phong phú của cô đã giúp chúng em hiểu sâu và áp dụng thành thạo những khái niệm và kỹ thuật mà môn học mang lại.

Chúng em muốn bày tỏ lòng biết ơn sâu sắc đến cô đã luôn tận tâm hướng dẫn và giải đáp mọi thắc mắc của em trong quá trình học. Sự kiên nhẫn và tận tâm của thầy/cô đã tạo điều kiện thuận lợi cho em phát triển kỹ năng lập trình của mình.

Đồng thời, chúng em cũng muốn bày tỏ lòng biết ơn đặc biệt đến các bạn học trong lớp, vì đã tạo nên một môi trường học tập tích cực và sôi động. Những buổi thảo luận và làm bài tập nhóm không chỉ giúp chúng em nắm bắt kiến thức một cách chặt chẽ mà còn mở rộng tầm nhìn về ứng dụng của lập trình Python trong thực tế.

Cuối cùng, chúng em xin chân thành cảm ơn cô và mọi người trong đội ngũ giáo vụ đã tạo ra một môi trường học tập tích cực và chuyên nghiệp. Chúng em tin rằng những kiến thức và kỹ năng em đạt được từ môn học này sẽ là nền tảng vững chắc cho sự phát triển trong tương lai.

|  |  |
| --- | --- |
| Sinh viên  Phạm Minh Thiên | Sinh viên  Nguyễn Văn Thạch |

# **MỤC LỤC**

[**MỤC LỤC** 4](#_Toc153835796)

[**MỤC LỤC BẢNG BIỂU** 6](#_Toc153835797)

[**MỤC LỤC HÌNH ẢNH** 7](#_Toc153835798)

[**MỞ ĐẦU** 8](#_Toc153835799)

[**1.** **Lý do chọn đề tài** 8](#_Toc153835800)

[**2.** **Lịch sử các sản phẩm liên quan** 8](#_Toc153835801)

[**3.** **Mục tiêu nghiên cứu** 11](#_Toc153835802)

[**4.** **Dự kiến sản phẩm** 11](#_Toc153835803)

[**5.** **Khả năng áp dụng sản phẩm** 12](#_Toc153835804)

[**6.** **Bố cục báo cáo** 12](#_Toc153835805)

[**CHƯƠNG 1. TÌM HIỂU BÀI TOÁN** 13](#_Toc153835806)

[**1.1.** **Nội dung bài toán** 13](#_Toc153835807)

[**1.1.1.** **Mục tiêu** 13](#_Toc153835808)

[**1.1.2.** **Mục đích** 13](#_Toc153835809)

[**1.2.** **Đề xuất xây dựng chương trình phần mềm theo đối tượng người dùng** 13](#_Toc153835810)

[**1.2.1.** **Chức năng cơ bản cho Kế hoạch Cá Nhân** 13](#_Toc153835811)

[**1.2.2.** **Chức năng cơ bản cho Kế hoạch Đội Nhóm** 13](#_Toc153835812)

[**1.3.** **Các kiến thức python phục vụ xây dựng phần mềm** 13](#_Toc153835813)

[**1.3.1.** **Ngôn ngữ lập trình python:** 13](#_Toc153835814)

[**1.3.2.** **Lập trình hướng đối tượng (OOP):** 13](#_Toc153835815)

[**1.3.3.** **Thư viện đồ hoạ và Giao diện người dùng (GUI):** 13](#_Toc153835816)

[**1.3.4.** **Làm việc với Database:** 14](#_Toc153835817)

[**1.4.** **Các nền tảng, công cụ phục vụ xây dựng phần mềm** 14](#_Toc153835818)

[**1.4.1. Giao diện người dùng (Qt Designer):** 14](#_Toc153835819)

[**1.4.2. Lập trình cơ sở dữ liệu (SQL Server):** 14](#_Toc153835820)

[**1.4.3. Lập trình python:** 15](#_Toc153835821)

[**CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ PHẦN MỀM** 16](#_Toc153835822)

[**2.1.** **Giới thiệu mô hình bài toán** 16](#_Toc153835823)

[**2.2.** **Phân tích về sản phẩm** 16](#_Toc153835824)

[**2.2.1.** **Mục tiêu chính** 16](#_Toc153835825)

[**2.2.2.** **Chức năng chính** 16](#_Toc153835826)

[**2.2.3.** **Quản lý quyền truy cập** 16](#_Toc153835827)

[**2.2.4.** **Giao diện người dùng** 16](#_Toc153835828)

[**2.2.5.** **Kiểm thử** 17](#_Toc153835829)

[**2.3.** **Thiết kế cơ sở dữ liệu** 17](#_Toc153835830)

[**2.3.1.** **CaNhan** 17](#_Toc153835831)

[**2.3.2.** **KeHoach** 17](#_Toc153835832)

[**2.3.3.** **NVnKH** 18](#_Toc153835833)

[**2.3.4.** **TaiKhoan** 18](#_Toc153835834)

[**2.4.** **Thiết kế giao diện** 18](#_Toc153835835)

[**2.4.1.** **Giao diện trang chủ của phần mềm** 19](#_Toc153835836)

[**2.4.2.** **Giao diện chức năng cho cá nhân** 19](#_Toc153835837)

[**2.4.3.** **Giao diện đăng nhập cho chức năng đội nhóm** 20](#_Toc153835838)

[**2.4.4.** **Giao diện đăng ký cho chức năng đội nhóm** 21](#_Toc153835839)

[**2.4.5.** **Giao diện admin hay trưởng nhóm cho chức năng đội nhóm** 22](#_Toc153835840)

[**2.4.6.** **Giao diện nhân viên cho chức năng đội nhóm** 24](#_Toc153835841)

[**CHƯƠNG 3. KẾT QUẢ TRIỂN KHAI XÂY DỰNG PHẦN MỀM** 26](#_Toc153835842)

[**3.1.** **Quy trình triển khai xây dựng phần mềm** 26](#_Toc153835843)

[**3.1.1.** **Cài đặt công cụ và nền tảng** 26](#_Toc153835844)

[**3.1.2.** **Tổ chức triển khai kết nối dữ liệu và công cụ lập trình** 26](#_Toc153835845)

[**3.1.3.** **Lập trình tính năng** 26](#_Toc153835846)

[**3.2.** **Các kết quả xây dựng phần mềm** 26](#_Toc153835847)

[**3.2.1.** **Giao diện người dùng** 26](#_Toc153835848)

[**3.2.2.** **Tính năng quản lý kế hoạch cá nhân** 27](#_Toc153835849)

[**3.2.3.** **Tính năng quản lý kế hoạch đội nhóm** 27](#_Toc153835850)

[**3.3.** **Triển khai thực chạy chương trình phần mềm** 27](#_Toc153835851)

[**3.3.1** **Tạo cấu hình cơ sở dữ liệu** 27](#_Toc153835852)

[**3.3.2** **Kiểm tra hoạt động của phần mềm** 27](#_Toc153835853)

[**3.3.3** **Kết quả triển khai** 27](#_Toc153835854)

[**KẾT LUẬN** 28](#_Toc153835855)

[**1.** **Các nhiệm vụ đã thực hiện** 28](#_Toc153835856)

[**2.** **Ưu điểm và nhược điểm của sản phẩm** 29](#_Toc153835857)

[ **Ưu điểm** 29](#_Toc153835858)

[ **Nhược điểm** 29](#_Toc153835859)

[**3.** **Dự định phát triển tương lai** 29](#_Toc153835860)

[**TÀI LIỆU THAM KHẢO** 31](#_Toc153835861)

[**PHỤ LỤC** 32](#_Toc153835862)

# **MỤC LỤC BẢNG BIỂU**

Bảng 1. dbo.CaNhan 16

Bảng 2. dbo.KeHoach 17

Bảng 3. dbo.NVnKH 17

Bảng 4. dbo.TaiKhoan 17

# **MỤC LỤC HÌNH ẢNH**

Hình 1.1. Giao diện phần mềm Qt Designer 12

Hình 1.2. Giao diện phần mềm SQL Server 13

Hình 1.3. Giao diện phần mềm Visual Code 13

Hình 2.1. Giao diện trang chủ phần mềm 17

Hình 2.2. Giao diện chức năng cho cá nhân 18

Hình 2.3. Giao diện đăng nhập cho chức năng đội nhóm 19

Hình 2.4. Giao diện đăng ký cho chức năng đội nhóm 20

Hình 2.5. Giao diện cập nhật kế hoạch cho đội nhóm 21

Hình 2.6. Giao diện giao việc cho các nhân viên trong đội nhóm 22

Hình 2.7. Giao diện nhân viên cho chức năng đội nhóm 23

# **MỞ ĐẦU**

1. **Lý do chọn đề tài**

Trong thời đại công nghệ thông tin phát triển như hiện nay, việc sử dụng phần mềm để hỗ trợ công việc trở nên phổ biến hơn bao giờ hết. Trong lĩnh vực quản lý, việc sử dụng phần mềm để lập kế hoạch hoạt động nhóm cũng là một xu hướng tất yếu.

Có thể thấy rằng, việc lập kế hoạch hoạt động nhóm là một công việc quan trọng, giúp nhóm đạt được mục tiêu đề ra. Tuy nhiên, việc lập kế hoạch hoạt động nhóm truyền thống thường gặp phải một số hạn chế như:

* Tốn nhiều thời gian và công sức: Việc lập kế hoạch hoạt động nhóm truyền thống thường tốn nhiều thời gian và công sức của các thành viên trong nhóm, đặc biệt là đối với các nhóm có quy mô lớn.
* Dễ xảy ra sai sót: Việc lập kế hoạch hoạt động nhóm truyền thống dễ xảy ra sai sót do thiếu tính thống nhất giữa các thành viên trong nhóm.
* Khó theo dõi và quản lý: Việc lập kế hoạch hoạt động nhóm truyền thống khó theo dõi và quản lý tiến độ thực hiện của kế hoạch.

Việc xây dựng phần mềm lập kế hoạch hoạt động nhóm sẽ giúp khắc phục được những hạn chế trên, mang lại những lợi ích sau:

* Tiết kiệm thời gian và công sức: Phần mềm lập kế hoạch hoạt động nhóm giúp tự động hóa các công việc lập kế hoạch, giúp tiết kiệm thời gian và công sức của các thành viên trong nhóm.
* Giảm thiểu sai sót: Phần mềm lập kế hoạch hoạt động nhóm giúp thống nhất thông tin giữa các thành viên trong nhóm, giảm thiểu sai sót khi lập kế hoạch.
* Dễ theo dõi và quản lý: Phần mềm lập kế hoạch hoạt động nhóm giúp theo dõi và quản lý tiến độ thực hiện của kế hoạch một cách dễ dàng.

1. **Lịch sử các sản phẩm liên quan**

Lịch sử các sản phẩm liên quan đến đề tài có thể được chia thành ba giai đoạn chính:

Giai đoạn đầu (trước năm 2000)

Trong giai đoạn này, các sản phẩm liên quan đến đề tài chủ yếu là các phần mềm quản lý dự án, tập trung vào các công việc lập kế hoạch, theo dõi và quản lý tiến độ thực hiện dự án. Một số sản phẩm tiêu biểu trong giai đoạn này bao gồm:

* Microsoft Project: Đây là một trong những phần mềm quản lý dự án phổ biến nhất hiện nay. Microsoft Project cung cấp nhiều tính năng mạnh mẽ để hỗ trợ lập kế hoạch, theo dõi và quản lý tiến độ thực hiện dự án.
* Primavera P6: Đây cũng là một phần mềm quản lý dự án phổ biến khác. Primavera P6 cung cấp nhiều tính năng cao cấp hơn Microsoft Project, phù hợp với các dự án phức tạp.
* Microsft Visio: Đây là một phần mềm vẽ sơ đồ, có thể được sử dụng để tạo sơ đồ kế hoạch dự án.

Giai đoạn giữa (2000 - 2010)

Trong giai đoạn này, các sản phẩm liên quan đến đề tài bắt đầu chú trọng đến các tính năng hỗ trợ làm việc nhóm. Một số sản phẩm tiêu biểu trong giai đoạn này bao gồm:

* Basecamp: Đây là một phần mềm quản lý dự án tập trung vào việc hỗ trợ làm việc nhóm. Basecamp cung cấp các tính năng như trao đổi thông tin, chia sẻ tài liệu, theo dõi tiến độ thực hiện dự án,...
* Asana: Đây cũng là một phần mềm quản lý dự án tập trung vào việc hỗ trợ làm việc nhóm. Asana cung cấp các tính năng tương tự Basecamp, nhưng có giao diện trực quan và dễ sử dụng hơn.
* Trello: Đây là một phần mềm quản lý dự án dựa trên thẻ (kanban). Trello cung cấp một cách đơn giản và trực quan để theo dõi tiến độ thực hiện dự án.

Giai đoạn hiện nay (2010 - nay)

Trong giai đoạn này, các sản phẩm liên quan đến đề tài tiếp tục phát triển, chú trọng đến các tính năng hỗ trợ cộng tác, trao đổi thông tin và theo dõi tiến độ thực hiện dự án một cách hiệu quả. Một số sản phẩm tiêu biểu trong giai đoạn này bao gồm:

* Slack: Đây là một phần mềm trao đổi thông tin nội bộ, được sử dụng phổ biến trong các doanh nghiệp và tổ chức. Slack cung cấp một cách nhanh chóng và dễ dàng để trao đổi thông tin giữa các thành viên trong nhóm.
* Google Workspace: Đây là một bộ ứng dụng văn phòng trực tuyến của Google, bao gồm các ứng dụng như Gmail, Google Docs, Google Sheets,... Google Workspace cung cấp các tính năng cộng tác mạnh mẽ, giúp các thành viên trong nhóm dễ dàng trao đổi thông tin và phối hợp làm việc với nhau.
* Microsoft Teams: Đây là một phần mềm hội nghị trực tuyến và cộng tác của Microsoft. Microsoft Teams cung cấp các tính năng như gọi điện video, chia sẻ màn hình, trao đổi thông tin,... giúp các thành viên trong nhóm dễ dàng kết nối và phối hợp làm việc với nhau.

Ngoài ra, còn có một số sản phẩm khác cũng có liên quan đến đề tài, như:

* Asana for Jira: Đây là một phần mềm kết nối Asana và Jira, giúp các nhóm sử dụng Asana để quản lý dự án, nhưng vẫn có thể sử dụng Jira để quản lý các vấn đề phát sinh trong dự án.
* ClickUp: Đây là một phần mềm quản lý dự án cung cấp nhiều tính năng mạnh mẽ, bao gồm các tính năng hỗ trợ làm việc nhóm, cộng tác, trao đổi thông tin,...
* Monday.com: Đây cũng là một phần mềm quản lý dự án cung cấp nhiều tính năng mạnh mẽ, bao gồm các tính năng hỗ trợ làm việc nhóm, cộng tác, trao đổi thông tin,...

1. **Mục tiêu nghiên cứu**

Mục tiêu của đề tài Xây dựng phần mềm lập kế hoạch hoạt động nhóm là xây dựng một phần mềm có thể giúp các nhóm thực hiện các công việc sau:

* Lập kế hoạch hoạt động nhóm một cách nhanh chóng, chính xác và hiệu quả.
* Theo dõi và kiểm soát tiến độ thực hiện kế hoạch một cách dễ dàng, thuận tiện.
* Giúp các thành viên trong nhóm phối hợp và làm việc hiệu quả hơn.

1. **Dự kiến sản phẩm**

**Sản phẩm 1. Phần mềm lập kế hoạch hoạt động nhóm**

Dựa trên lịch sử các sản phẩm liên quan đến đề tài và nhu cầu thực tế của các nhóm, dự kiến sản phẩm sẽ bao gồm các tính năng sau:

Tính năng lập kế hoạch đội nhóm: Sản phẩm sẽ cung cấp các tính năng để giúp các nhóm lập kế hoạch hoạt động một cách hiệu quả, bao gồm:

* + Tạo và quản lý các mục tiêu, nhiệm vụ, công việc,...
  + Phân bổ công việc cho các thành viên trong nhóm
  + Xác định thời hạn hoàn thành cho các công việc
  + Phân bổ tài nguyên cho các công việc

Tính năng lập kế hoạch cá nhân: Sản phẩm sẽ cung cấp các tính năng để giúp cá nhân lập kế hoạch hoạt động một cách hiệu quả, bao gồm:

* + Tạo và quản lý các mục tiêu, nhiệm vụ, công việc,...
  + Xác định thời hạn hoàn thành cho các công việc
  + Phân bổ tài nguyên cho các công việc

**Sản phẩm 2. Báo cáo kết quả xây dựng phần mềm**

1. **Khả năng áp dụng sản phẩm**

Khả năng áp dụng sản phẩm là rất cao. Sản phẩm có thể được áp dụng trong nhiều lĩnh vực, bao gồm:

* Giáo dục: Sản phẩm có thể được sử dụng để hỗ trợ các nhóm học sinh, sinh viên lập kế hoạch và thực hiện các dự án học tập.
* Doanh nghiệp: Sản phẩm có thể được sử dụng để hỗ trợ các nhóm trong doanh nghiệp lập kế hoạch và thực hiện các dự án kinh doanh.
* Tổ chức phi lợi nhuận: Sản phẩm có thể được sử dụng để hỗ trợ các nhóm trong các tổ chức phi lợi nhuận lập kế hoạch và thực hiện các dự án cộng đồng.

1. **Bố cục báo cáo**

Chương 1. Tìm hiểu bài toán

Chương 2. Phân tích và thiết kế phần mềm

Chương 3. Kết quả triển khai xây dựng phần mềm

# **CHƯƠNG 1. TÌM HIỂU BÀI TOÁN**

1. **Nội dung bài toán**
   * 1. **Mục tiêu**

Xây dựng phần mềm quản lý kế hoạch của cá nhân cũng như kế hoạch của mội đội nhóm.

* + 1. **Mục đích**

Giúp người dùng tổ chức công việc và nhiệm vụ một cách có hệ thống và hiệu quả hơn, giúp họ biết được mình cần làm gì và khi nào cần làm, quản lý thời gian cá nhân hoặc thời gian của nhóm làm việc. Cho phép lập kế hoạch, theo dõi tiến độ và quản lý dự án, thấy được hệ thống nhiệm vụ giúp người dùng có thể điều chỉnh và sửa đổi kế hoạch phù hợp hơn.

1. **Đề xuất xây dựng chương trình phần mềm theo đối tượng người dùng**
2. **Chức năng cơ bản cho Kế hoạch Cá Nhân**

Người dùng sẽ không cần cấp tài khoản và được hoạt động với quyền admin cho phép người dùng thêm các đầu mục công việc cần làm, chỉnh sửa các đầu mục công việc, có thể ấn hoàn thành khi đã làm xong công việc.

1. **Chức năng cơ bản cho Kế hoạch Đội Nhóm**

Nhóm trưởng sẽ là người được cấp quyền admin còn mọi thành viên khác sẽ được cấp quyền thành viên chỉ có thể xem xét kế hoạch, ấn nút hoàn thành khi đã hoàn thành công việc.

1. **Các kiến thức python phục vụ xây dựng phần mềm**
2. **Ngôn ngữ lập trình python:**

Hiểu rõ về cú pháp, cấu trúc dữ liệu, và xử lý ngoại lệ trong Python.

1. **Lập trình hướng đối tượng (OOP):**

Sử dụng lập trình hướng đối tượng để tổ chức mã nguồn một cách cấu trúc và dễ quản lý.

1. **Thư viện đồ hoạ và Giao diện người dùng (GUI):**

Sử dụng thư viện PyQt6 để tạo giao diện người dùng thân thiện và dễ sử dụng.

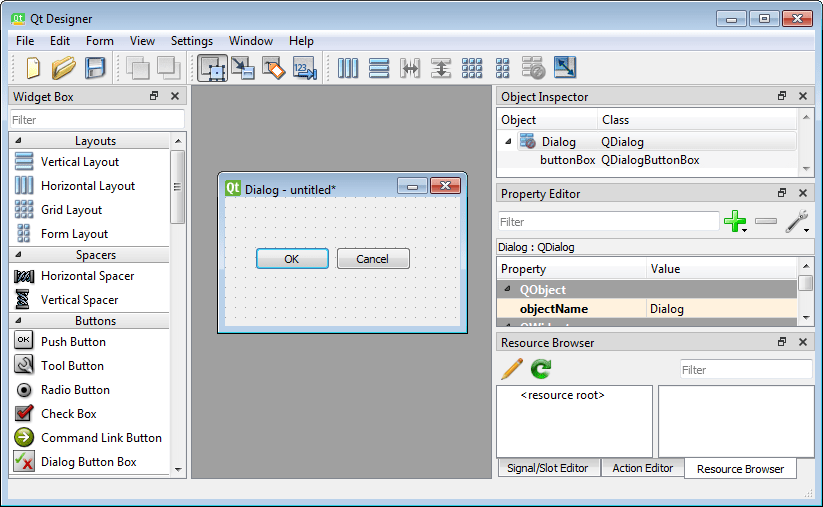
1. **Làm việc với Database:**

Sử dụng SQL Server để lưu trữ và quản lý dữ liệu về kế hoạch, dự án và thông tin người dùng.

1. **Các nền tảng, công cụ phục vụ xây dựng phần mềm**

**1.4.1. Giao diện người dùng (Qt Designer):**

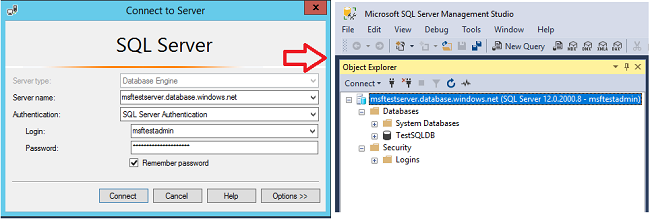
Qt Designer là một công cụ mạnh mẽ của Qt Framework, được thiết kế để tạo và thiết kế giao diện người dùng (GUI) cho ứng dụng sử dụng Qt. Nó giúp phát triển ứng dụng dễ dàng hơn thông qua việc tạo và chỉnh sửa các thành phần giao diện người dùng một cách trực quan, mà không cần viết mã nguồn.



Hình 1.1. Giao diện phần mềm Qt Designer

**1.4.2. Lập trình cơ sở dữ liệu (SQL Server):**

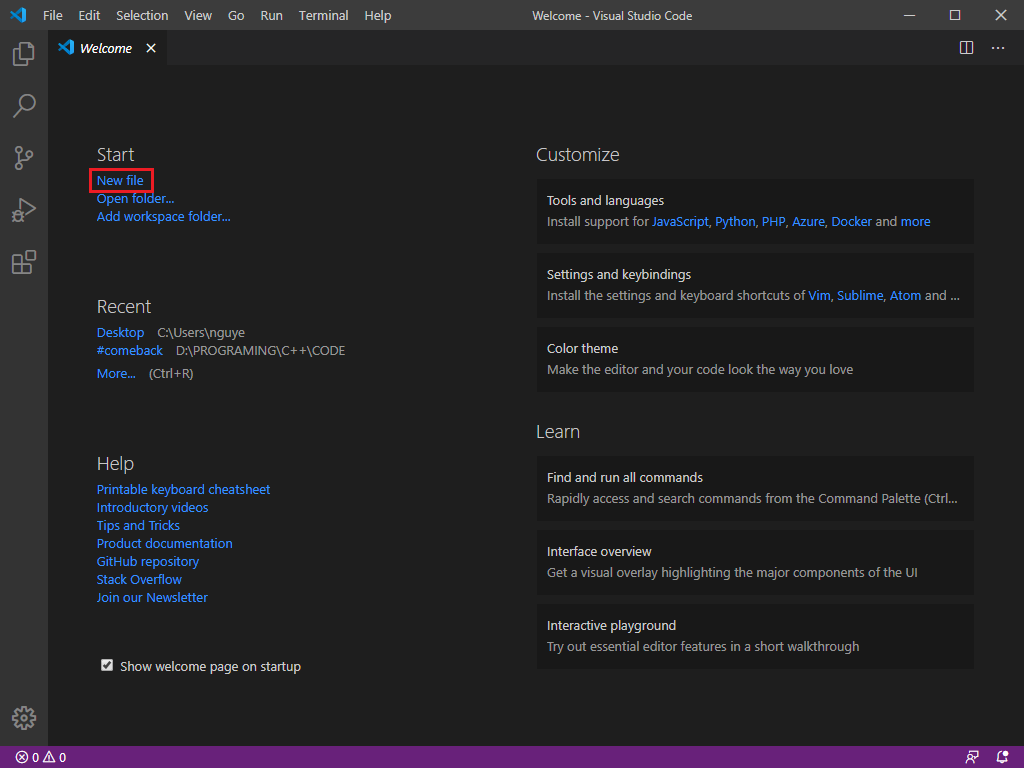
Sử dụng SQL Server làm hệ quản trị cơ sở dữ liệu (DBMS). Điều này giúp lưu trữ và truy vấn dữ liệu một cách hiệu quả.



Hình 1.2. Giao diện phần mềm SQL Server

**1.4.3. Lập trình python:**

Sử dụng phần mềm Visual Code để lập trình chương trình chính của phần mềm bằng ngôn ngữ python.



Hình 1.3. Giao diện phần mềm Visual Code

**CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ PHẦN MỀM**

1. **Giới thiệu mô hình bài toán**

Trong môi trường công việc ngày nay, việc quản lý thời gian và công việc cá nhân cũng như trong đội nhóm là một thách thức. Để giải quyết vấn đề này, chúng em đề xuất xây dựng một ứng dụng phần mềm sử dụng Python và Qt Framework. Chúng em sẽ sử dụng Python làm ngôn ngữ lập trình chính, kết hợp với Qt Framework để xây dựng giao diện người dùng. Cơ sở dữ liệu sẽ được tích hợp để lưu trữ thông tin về công việc và dự án.

1. **Phân tích về sản phẩm**
2. **Mục tiêu chính**

Xây dựng một ứng dụng phần mềm đa nhiệm có khả năng quản lý kế hoạch cá nhân và đội nhóm một cách hiệu quả. Mục tiêu là tối ưu hóa tổ chức công việc, cải thiện hiệu suất làm việc và tăng cường tương tác trong nhóm.

1. **Chức năng chính**

**Chức năng cho Kế hoạch Cá Nhân**

Người dùng sẽ không cần cấp tài khoản và được hoạt động với quyền admin cho phép người dùng thêm các đầu mục công việc cần làm, chỉnh sửa các đầu mục công việc, có thể ấn hoàn thành khi đã làm xong công việc.

**Chức năng cho Kế hoạch Đội Nhóm**

Nhóm trưởng sẽ là người được cấp quyền admin còn mọi thành viên khác sẽ được cấp quyền thành viên chỉ có thể xem xét kế hoạch, ấn nút hoàn thành khi đã hoàn thành công việc.

1. **Quản lý quyền truy cập**

Xác định và quả lý quyển truy cập của từng thành viên trong đội nhóm.

1. **Giao diện người dùng**

Thiết kế giao diện đẹp, dễ sử dụng và hiển thị một cách rõ ràng và tổng quan về kế hoạch và dự án.

1. **Kiểm thử**

Thực hiện kiểm thử chất lượng để đảm bảo tính ổn định và an toàn.

1. **Thiết kế cơ sở dữ liệu**
2. **CaNhan**

Chứa các thông tin trong khi người dùng sử dụng chức năng cho cá nhân.

Các thuộc tính: stt, tg, noidung, mota, trangthai, tht

|  |  |  |
| --- | --- | --- |
| Tên thuộc tính | Diễn giải | Kiểu dữ liệu |
| stt | Số thứ tự | int |
| tg | Thời gian | date |
| noidung | Nội dung kế hoạch | nvarchar(50) |
| mota | Mô tả kế hoạch | nvarchar(100) |
| trangthai | Trạng thái kế hoạch | nvarchar(50) |
| tht | Ngày hoàn thành | date |

Bảng 1. dbo.CaNhan

1. **KeHoach**

Chứa các thông tin của kế hoạch đội nhóm khi người dùng sử dụng chức năng cho đội nhóm.

Các thuộc tính: idkh, ngay, noidung, mota, trangthai, ngayht

|  |  |  |
| --- | --- | --- |
| Tên thuộc tính | Diễn giải | Kiểu dữ liệu |
| idkh | Mã số kế hoạch | int |
| ngay | Thời gian | date |
| noidung | Nội dung kế hoạch | nvarchar(50) |
| mota | Mô tả kế hoạch | nvarchar(100) |
| trangthai | Trạng thái kế hoạch | nvarchar(50) |
| ngayht | Ngày hoàn thành | date |

Bảng 2. dbo.KeHoach

1. **NVnKH**

Mối liên kết giữa nhân viên và kế hoạch trong một đội nhóm khi người dùng sử dụng chức năng cho đội nhóm.

Các thuộc tính: idNVKH, id, idKH, tkgiao

|  |  |  |
| --- | --- | --- |
| Tên thuộc tính | Diễn giải | Kiểu dữ liệu |
| idNVKH | Mã số liên kết nhân viên và kế hoạch | int |
| id | Mã số nhân viên | int |
| idKH | Mã số kế hoạch | int |
| tkgiao | Tài khoản giao việc | nvarchar(50) |

Bảng 3. dbo.NVnKH

1. **TaiKhoan**

Chứa các thông tin về tài khoản và thông tin cơ bản của các nhân viên trong đội nhóm khi người dùng sử dụng chức năng cho đội nhóm.

Thuộc tính: id, tk, mk, role, hoten, sdt

|  |  |  |
| --- | --- | --- |
| Tên thuộc tính | Diễn giải | Kiểu dữ liệu |
| id | Mã số nhân viên | int |
| tk | Tài khoản | nvarchar(20) |
| mk | Mật khẩu | nvarchar(50) |
| role | Vị trí | nvarchar(20) |
| hoten | Họ và tên nhân viên | nvarchar(50) |
| sdt | Số điện thoại | nvarchar(15) |

Bảng 4. dbo.TaiKhoan

1. **Thiết kế giao diện**
2. **Giao diện trang chủ của phần mềm**

Người dùng có thể chọn chức năng cho cá nhân hoặc đội nhóm.

A screenshot of a computer

Description automatically generated

Hình 2.1. Giao diện trang chủ phần mềm

1. **Giao diện chức năng cho cá nhân**

Quản lý thông tin kế hoạch: thêm, sửa, xoá kế hoạch và chỉnh sửa trạng thái của kế hoạch.

**A screenshot of a computer

Description automatically generated**

Hình 2.2. Giao diện chức năng cho cá nhân

1. **Giao diện đăng nhập cho chức năng đội nhóm**

Tại đây trưởng nhóm (admin), nhân viên có thể nhập tài khoản và mật khẩu đã được cấp để đăng nhập, sau khi nhập tài khoản và mật khẩu có thể nhấn login. Nếu trường hợp tài khoản không hợp lệ sẽ không thể đăng nhập vào.

**A screenshot of a computer

Description automatically generated**

Hình 2.3. Giao diện đăng nhập cho chức năng đội nhóm

1. **Giao diện đăng ký cho chức năng đội nhóm**

Nếu vào trường hợp nhân viên chưa được cấp tài khoản thì có thể đăng ký tài khoàn để nhận được công việc được giao bởi trưởng nhóm (admin).

A screenshot of a computer

Description automatically generated

Hình 2.4. Giao diện đăng ký cho chức năng đội nhóm

1. **Giao diện admin hay trưởng nhóm cho chức năng đội nhóm**

Trưởng nhóm (admin) sẽ được cấp phép để thêm, sửa, xoá kế hoạch cho đội nhóm và giao việc cho các nhân viên trong đội nhóm.

A screenshot of a computer

Description automatically generated

Hình 2.5. Giao diện cập nhật kế hoạch cho đội nhóm

A screenshot of a computer

Description automatically generated

Hình 2.6. Giao diện giao việc cho các nhân viên trong đội nhóm

1. **Giao diện nhân viên cho chức năng đội nhóm**

Nhân viên sau khi đăng nhập có thể thấy các công việc mà mình đã được giao, sau khi hoàn thành có thể cập nhật trạng thái đã hoàn thành bằng cách ấn vào nút Hoàn thành.

A screenshot of a computer

Description automatically generated

Hình 2.7. Giao diện nhân viên cho chức năng đội nhóm

**CHƯƠNG 3. KẾT QUẢ TRIỂN KHAI XÂY DỰNG PHẦN MỀM**

1. **Quy trình triển khai xây dựng phần mềm**
2. **Cài đặt công cụ và nền tảng**

Triển khai xây dựng cài đặt môi trường lập trình sử dụng ngôn ngữ Python với các thư viện như PyQt, pyodbc để hỗ trợ xây dựng giao diện và quản lý cơ sở dữ liệu của phần mềm.

Cài đặt phần mềm Qt Designer và tích hợp nó vào quy trình phát triển để tạo và chỉnh sửa giao diện người dùng một cách thuận tiện.

1. **Tổ chức triển khai kết nối dữ liệu và công cụ lập trình**

Kết nối ứng dụng với cơ sở dữ liệu sử dụng thư viện pyodbc và môi trường SQL Server, đảm bảo tính ổn định và hiệu suất cao trong việc lưu trữ và truy xuất dữ liệu.

Tổ chức mã nguồn theo mô hình lập trình hướng đối tượng để tạo cấu trúc dự án rõ ràng.

Xây dựng các chức năng riêng biệt cho quản lý công việc cá nhân hay quản lý kế hoạch đội nhóm.

1. **Lập trình tính năng**

Triển khai tính năng quản lý công việc cá nhân bao gồm thêm, sửa, xoá công việc, tìm kiếm theo nội dung công việc, lọc các công việc theo trạng thái và export file tổng hợp công việc.

Xây dựng tính năng quản lý công việc cho đội nhóm với khả năng chỉnh sửa dự án, gán công việc, theo dõi tiến độ dành cho trưởng nhóm (admin) và nhận công việc dành cho nhân viên đội nhóm.

Với quản lý công việc cho đội nhóm sẽ có tính năng đăng nhập, đăng ký để phân biệt được vai trò của các thành viên trong đội nhóm.

1. **Các kết quả xây dựng phần mềm**
2. **Giao diện người dùng**

Thiết kế giao diện thân thiện với người dùng, giúp họ dễ dàng tương tác với ứng dụng.

Tối ưu hoá trải nghiệm người dùng với các tính năng như tương tác nhanh bằng các nút bấm, chế độ xem linh hoạt.

1. **Tính năng quản lý kế hoạch cá nhân**

Các tính năng quản lý công việc cá nhân như tạo, chỉnh sửa, xóa công việc đã được triển khai và kiểm thử một cách chặt chẽ.

Kèm theo đó là các tính năng như tìm kiếm công việc, lọc công việc, export file công việc đã được triển khai và kiểm thử một cách chặt chẽ.

1. **Tính năng quản lý kế hoạch đội nhóm**

Triển khai tính năng quản lý dự án đội nhóm với khả năng tạo dự án, gán công việc, và theo dõi tiến độ từng thành viên.

Tính năng đăng nhập, đăng ký để phân biệt được vai trò thành viên được triển khai và kiểm thử.

1. **Triển khai thực chạy chương trình phần mềm**
2. **Tạo cấu hình cơ sở dữ liệu**

Tạo cơ sở dữ liệu cho phần mềm.

Thêm dữ liệu mẫu vào cơ sở dữ liệu.

1. **Kiểm tra hoạt động của phần mềm**

Khởi chạy phần mềm thông qua môi trường lập trình.

Kiểm tra các tính năng của phần mềm hoạt động đúng như mong muốn.

1. **Kết quả triển khai**

Phần mềm không bị lỗi, các tính năng hoạt động ổn định, cơ sở dữ liệu không sai sót.

**KẾT LUẬN**

* 1. **Các nhiệm vụ đã thực hiện**
* **Nghiên cứu và phân tích bài toán**

Tìm Hiểu Nhu Cầu Người Dùng: Tiến hành cuộc nghiên cứu để hiểu rõ nhu cầu và mong muốn của người dùng đối với phần mềm quản lý kế hoạch đội nhóm.

Phân Tích Yêu Cầu: Đặt ra các yêu cầu cụ thể cho ứng dụng, xác định tính năng cần thiết để đáp ứng nhu cầu người dùng.

* **Thiết kế hệ thống**

Thiết Kế Giao Diện Người Dùng: Sử dụng công cụ thiết kế để vẽ giao diện người dùng dựa trên yêu cầu và phản hồi từ người dùng.

Xây Dựng Thiết Kế Cơ Sở Dữ Liệu: Định hình cấu trúc cơ sở dữ liệu, xác định bảng, quan hệ, và các truy vấn cần thiết.

* **Phát triển phần mềm**

Triển Khai Môi Trường Phát Triển: Cài đặt và cấu hình môi trường phát triển sử dụng Python, Qt, và cơ sở dữ liệu.

Lập Trình Tính Năng Cơ Bản: Xây dựng các tính năng quản lý công việc cá nhân và dự án đội nhóm, bao gồm thêm, sửa, xóa công việc.

* **Kiểm thử**

Kiểm Thử Đơn Vị: Kiểm thử từng đơn vị mã nguồn để đảm bảo tính độc lập và chính xác.

Kiểm Thử Hệ Thống: Thực hiện kiểm thử hệ thống để đảm bảo sự liên kết chặt chẽ giữa các module và tính ổn định của hệ thống.

* **Thực chạy phần mềm**

Kiểm tra tính ổn định của phần mềm, các tính năng, cơ sở dữ liệu và đảm bảo không sai sót.

* **Báo cáo**

Viết báo cáo lại quá trình làm việc phát triển phần mềm cũng như giới thiệu hướng dẫn sử dụng phần mềm.

* 1. **Ưu điểm và nhược điểm của sản phẩm**
* **Ưu điểm**

Tăng hiệu quả công việc: giúp người dùng quản lý thời gian và công việc cá nhân cũng như dự án đội nhóm một cách hiệu quả.

Giao diện người dùng thân thiện, dễ sử dụng và tích hợp các tính năng hiệu quả.

Cung cấp khả năng quản lý và theo dõi tiến độ hoàn thành công việc của các thành viên trong đội nhóm.

* **Nhược điểm**

Có thể đòi hỏi một thời gian ngắn để người dùng làm quen và học cách sử dụng một số tính năng.

Phụ thuộc vào các công nghệ cụ thể như Python, Qt và cơ sở dữ liệu, có thể tạo ra rắc rối nếu có sự thay đổi trong công nghệ.

Có thể làm cho ứng dụng trở nên phức tạp với nhiều tính năng, làm mất đi sự tập trung của người dùng.

Trong môi trường có đồng thời nhiều người sử dụng, khả năng đồng bộ hóa dữ liệu có thể gặp vấn đề.

* 1. **Dự định phát triển tương lai**

Tích hợp thêm các chức năng như giao tiếp giữa các thành viên trong nhóm, quản lý hạn cuối của công việc.

Tích hợp nền tảng đám mây: Cung cấp khả năng truy cập dữ liệu từ bất kỳ đâu, tăng tính khả dụng và đảm bảo bảo mật thông tin.

Phát triển ứng dụng di động: Cung cấp trải nghiệm người dùng liền mạch trên cả các thiết bị di động, giúp họ quản lý công việc mọi lúc, mọi nơi.

Tích hợp các công cụ tương tác và giao tiếp trực tuyến như video họp, trò chuyện để cải thiện sự liên kết trong đội nhóm.

Nâng cấp hệ thống bảo mật để đảm bảo an toàn dữ liệu và ngăn chặn truy cập trái phép.

Thêm khả năng hỗ trợ đa ngôn ngữ để mở rộng sự sử dụng cho người dùng ở nhiều quốc gia.

**TÀI LIỆU THAM KHẢO**

[1] Nguyễn Văn Ba, Phân tích và thiết kế hệ thống thông tin, NXB Đại học Quốc gia Hà Nội, 2006.

[2] Mark Summerfield, Rapid GUI Programming with Python and Qt, NXB Prentice Hall, 2007.

**PHỤ LỤC**

**Chọn chức năng cá nhân hoặc đội nhóm**

class OpenW(QMainWindow):

    def \_\_init\_\_(self):

        super(OpenW,self).\_\_init\_\_()

        uic.loadUi("OpenW.ui",self)

        self.btCaNhan.clicked.connect(self.CaNhan)

        self.btNhom.clicked.connect(self.Nhom)

    def Nhom(self):

        widget.setFixedSize(425,320)

        widget.setCurrentIndex(2)

        # widget.show()

        # app.exec()

    def CaNhan(self):

        widget.setFixedSize(650,600)

        widget.setCurrentIndex(3)

**Chức năng làm việc cá nhân**

class CaNhanW(QMainWindow):

    def \_\_init\_\_(self):

        super(CaNhanW,self).\_\_init\_\_()

        uic.loadUi("CaNhanMain.ui",self)

        self.setFixedSize(650, 600)

        self.ngay.setDate(QDate.currentDate())

        self.ngay\_2.setDate(QDate.currentDate())

        self.ngay\_2.setVisible(False)

        self.Load\_data()

        self.tbCaNhan.itemSelectionChanged.connect(self.on\_item\_selection\_changed)

        self.cbBox.currentIndexChanged.connect(self.on\_combobox\_changed)

        self.btThem.clicked.connect(self.Them\_data)

        self.btSua.clicked.connect(self.Sua\_data)

        self.btXoa.clicked.connect(self.Xoa\_data)

        self.btHT.clicked.connect(self.HoanT)

        self.btTim.clicked.connect(self.Tim\_data)

        self.btEx.clicked.connect(self.export\_to\_excel)

        self.btBack.clicked.connect(self.home)

    def home(self):

        widget.setFixedSize(310,180)

        widget.setCurrentIndex(0)

    def Load\_data(self):

        self.tbCaNhan.clear()

        query=db.cursor()

        query.execute("select \* from CaNhan")

        kq=query.fetchall()

        self.tbCaNhan.setColumnHidden(0, True)

        self.tbCaNhan.setColumnCount(5)

        self.tbCaNhan.setColumnWidth(0, 30)

        self.tbCaNhan.setColumnWidth(1, 90) #10

        self.tbCaNhan.setColumnWidth(2, 130) #20

        self.tbCaNhan.setColumnWidth(3, 150) #50

        self.tbCaNhan.setColumnWidth(4, 100) #20

        self.tbCaNhan.setColumnWidth(6, 100)

        header\_labels = ['Stt', 'Ngày giao', 'Nội dung', 'Loại', 'Trạng thái', 'Ngày HT']

        self.tbCaNhan.setHeaderLabels(header\_labels)

        for row in kq:

            item = QTreeWidgetItem(self.tbCaNhan)

            for i, value in enumerate(row):

                item.setText(i, str(value))

    def Them\_data(self):

        nd=self.txtNoiDung.text()

        ngay=self.ngay.date().toString("yyyy-MM-dd")

        mota=self.txtMoTa.text()

        trangthai='Chua Hoan Thanh'

        if not nd:

            QMessageBox.warning(self, 'Lỗi', 'Vui lòng nhập phần Nội Dung!')

            return

        if not mota:

            QMessageBox.warning(self, 'Lỗi', 'Vui lòng nhập phần Mô Tả')

            return

        query=db.cursor()

        query.execute(f"INSERT INTO CaNhan (tg, noidung, mota, trangthai) VALUES ( '{ngay}', '{nd}', '{mota}', '{trangthai}')")

        db.commit()

        self.Load\_data()

        self.txtNoiDung.clear()

        self.txtMoTa.clear()

    def Sua\_data(self):

        selected\_items = self.tbCaNhan.selectedItems()

        if selected\_items:

            selected\_item = selected\_items[0].text(0)

            nd=self.txtNoiDung.text()

            ngay=self.ngay.date().toString("yyyy-MM-dd")

            mota=self.txtMoTa.text()

            #trangthai='Chua Hoan Thanh'

            if not nd:

                QMessageBox.warning(self, 'Lỗi', 'Vui lòng nhập phần Nội Dung!')

                return

            if not mota:

                QMessageBox.warning(self, 'Lỗi', 'Vui lòng nhập phần Mô Tả')

                return

            query=db.cursor()

            query.execute("UPDATE CaNhan SET tg=?, noidung=?, mota=? WHERE stt=?", ngay, nd, mota, selected\_item)

            db.commit()

            self.Load\_data()

            self.txtNoiDung.clear()

            self.txtMoTa.clear()

        else:

            QMessageBox.information(self,'Thông Báo','Bạn chưa chọn dòng muốn sửa')

    def Xoa\_data(self):

        selected\_items = self.tbCaNhan.selectedItems()

        if selected\_items:

            selected\_item = selected\_items[0].text(0)

            query=db.cursor()

            query.execute("DELETE FROM CaNhan WHERE stt=?", selected\_item)

            db.commit()

            self.Load\_data()

        else:

            QMessageBox.information(self,'Thông Báo','Bạn chưa chọn dòng muốn xóa')

    def HoanT(self):

        selected\_items = self.tbCaNhan.selectedItems()

        if selected\_items:

            selected\_item = selected\_items[0].text(0)

            ngay=self.ngay\_2.date().toString("yyyy-MM-dd")

            time=None

            if selected\_items[0].text(4)=="Hoan Thanh":

                query=db.cursor()

                query.execute("UPDATE CaNhan SET trangthai='Chua Hoan Thanh',tht=? WHERE stt=?",time, selected\_item)

                db.commit()

            if selected\_items[0].text(4)=="Chua Hoan Thanh":

                query=db.cursor()

                query.execute("UPDATE CaNhan SET trangthai='Hoan Thanh',tht=? WHERE stt=?",ngay, selected\_item)

                db.commit()

            self.Load\_data()

        else:

            QMessageBox.information(self,'Thông Báo','Bạn chưa chọn dòng')

    def on\_item\_selection\_changed(self):

        selected\_items = self.tbCaNhan.selectedItems()

        if selected\_items:

            if selected\_items[0].text(4)=="Hoan Thanh":

                self.btHT.setText("Làm Lại")

            else:

                self.btHT.setText("Hoàn thành")

            self.txtNoiDung.setText(selected\_items[0].text(2))

            self.txtMoTa.setText(selected\_items[0].text(3))

            self.ngay.setDate(QDate.fromString(selected\_items[0].text(1),"yyyy-MM-dd"))

    def on\_combobox\_changed(self):

        selected\_text = self.cbBox.currentText()

        if selected\_text == "Tat Ca":

            self.Load\_data()

        elif selected\_text == "Hoan Thanh":

            self.tbCaNhan.clear()

            query=db.cursor()

            query.execute("select \* from CaNhan Where trangthai='Hoan Thanh'")

            kq=query.fetchall()

            self.tbCaNhan.setColumnHidden(0, True)

            self.tbCaNhan.setColumnCount(5)

            self.tbCaNhan.setColumnWidth(0, 30)

            self.tbCaNhan.setColumnWidth(1, 90) #10

            self.tbCaNhan.setColumnWidth(2, 130) #20

            self.tbCaNhan.setColumnWidth(3, 150) #50

            self.tbCaNhan.setColumnWidth(4, 100) #20

            self.tbCaNhan.setColumnWidth(6, 100)

            header\_labels = ['Stt', 'Ngày giao', 'Nội dung', 'Loại', 'Trạng thái', 'Ngày HT']

            self.tbCaNhan.setHeaderLabels(header\_labels)

            for row in kq:

                item = QTreeWidgetItem(self.tbCaNhan)

                for i, value in enumerate(row):

                    item.setText(i, str(value))

        elif selected\_text == "Chua Hoan Thanh":

            self.tbCaNhan.clear()

            query=db.cursor()

            query.execute("select \* from CaNhan Where trangthai='Chua Hoan Thanh'")

            kq=query.fetchall()

            self.tbCaNhan.setColumnHidden(0, True)

            self.tbCaNhan.setColumnCount(5)

            self.tbCaNhan.setColumnWidth(0, 30)

            self.tbCaNhan.setColumnWidth(1, 90) #10

            self.tbCaNhan.setColumnWidth(2, 130) #20

            self.tbCaNhan.setColumnWidth(3, 150) #50

            self.tbCaNhan.setColumnWidth(4, 100) #20

            self.tbCaNhan.setColumnWidth(6, 100)

            header\_labels = ['Stt', 'Ngày giao', 'Nội dung', 'Loại', 'Trạng thái', 'Ngày HT']

            self.tbCaNhan.setHeaderLabels(header\_labels)

            for row in kq:

                item = QTreeWidgetItem(self.tbCaNhan)

                for i, value in enumerate(row):

                    item.setText(i, str(value))

    def Tim\_data(self):

        search\_text = self.txtTim.text()

        if search\_text:

            self.tbCaNhan.clear()

            query=db.cursor()

            sql\_command = "SELECT \* FROM CaNhan WHERE noidung LIKE ?"

            query.execute(sql\_command, ('%' + search\_text + '%',))

            kq=query.fetchall()

            self.tbCaNhan.setColumnHidden(0, True)

            self.tbCaNhan.setColumnCount(5)

            self.tbCaNhan.setColumnWidth(0, 30)

            self.tbCaNhan.setColumnWidth(1, 90) #10

            self.tbCaNhan.setColumnWidth(2, 130) #20

            self.tbCaNhan.setColumnWidth(3, 150) #50

            self.tbCaNhan.setColumnWidth(4, 100) #20

            self.tbCaNhan.setColumnWidth(6, 100)

            header\_labels = ['Stt', 'Ngày giao', 'Nội dung', 'Loại', 'Trạng thái', 'Ngày HT']

            self.tbCaNhan.setHeaderLabels(header\_labels)

            for row in kq:

                item = QTreeWidgetItem(self.tbCaNhan)

                for i, value in enumerate(row):

                    item.setText(i, str(value))

        else:

            self.Load\_data()

        self.txtTim.clear()

    def export\_to\_excel(self):

        # Tạo một Workbook

        workbook = Workbook()

        # Tạo một WorkSheet

        worksheet = workbook.active

        # Đặt tiêu đề cột

        columns = ["Ngày giao", "Nội dung", "Loại","Trạng thái", "Ngày HT"]

        worksheet.append(columns)

        # Lặp qua dữ liệu QTreeWidget và thêm vào WorkSheet

        for row in range(self.tbCaNhan.topLevelItemCount()):

            item = self.tbCaNhan.topLevelItem(row)

            row\_data = [item.text(col) for col in range(1,item.columnCount())]

            worksheet.append(row\_data)

        file=self.txtEx.text()

        if not file:

            QMessageBox.warning(self, 'Lỗi', 'Vui lòng nhập phần tên cho File Excel')

            return

        # Lưu Workbook thành tệp Excel

        workbook.save(file+".xlsx")

        QMessageBox.information(self,'Thông Báo','Đã lưu ra File Excel')

**Chức năng đăng ký tài khoản cho làm việc đội nhóm**

class DKApp(QMainWindow):

    def \_\_init\_\_(self):

        super(DKApp,self).\_\_init\_\_()

        uic.loadUi("DangKy.ui",self)

        self.btDk\_dk.clicked.connect(self.DK)

        self.btThoat.clicked.connect(self.Thoat)

    def Thoat(self):

        widget.setFixedSize(310,180)

        widget.setCurrentIndex(0)

    def DK(self):

        un=self.txtTk\_dk.text()

        psw=self.txtMk\_dk.text()

        hoten=self.txthoten.text()

        sdt=self.txtsdt.text()

        if not un:

            QMessageBox.information(self, 'Thông báo', 'Vui lòng nhập Tài Khoản')

            return

        if not psw:

            QMessageBox.information(self, 'Thông báo', 'Vui lòng nhập Mật khẩu')

            return

        if not hoten:

            QMessageBox.information(self, 'Thông báo', 'Vui lòng nhập Họ tên')

            return

        if not sdt:

            QMessageBox.information(self, 'Thông báo', 'Vui lòng nhập Số điện thoại')

            return

        query=db.cursor()

        query.execute("select \* from TaiKhoan where tk='"+un+"' ")

        kt=query.fetchone()

        if kt:

            QMessageBox.information(self,"Register Output","Đăng ký thất bại")

        else:

            query.execute("insert into TaiKhoan values ('"+un+"','"+psw+"','customer','"+hoten+"','"+sdt+"')")

            db.commit()

            QMessageBox.information(self,"Register Output","Đăng ký thành công")

            widget.setFixedSize(425,320)

            widget.setCurrentIndex(2)

**Chức năng làm việc của đội nhóm**

class AdminApp(QMainWindow):

    def \_\_init\_\_(self):

        super(AdminApp,self).\_\_init\_\_()

        uic.loadUi("adminW.ui",self)

        self.txtte.setVisible(False)

        self.tabWidget.setCurrentIndex(2)

        self.tabWidget.tabBar().setTabText(0, "")

        self.tabWidget.tabBar().setTabText(1, "")

        self.tabWidget.setTabEnabled(0, False)

        self.tabWidget.setTabEnabled(1, False)

        self.btDn.clicked.connect(self.DN)

        self.btDk.clicked.connect(self.DangKy)

        self.tbNhom.itemSelectionChanged.connect(self.on\_item\_selection\_changed)

        self.cbBox.currentIndexChanged.connect(self.on\_combobox\_changed)

        self.btHT.clicked.connect(self.HoanT)

        self.btTim.clicked.connect(self.Tim\_data)

        self.btEx.clicked.connect(self.export\_to\_excel)

        self.btBack.clicked.connect(self.Logout)

        self.btThoat.clicked.connect(self.Thoat)

        #Phần Admin

        self.ngay\_2.setDate(QDate.currentDate())

        self.ngay\_2.setVisible(False)

        self.tbKeHoach.itemSelectionChanged.connect(self.on\_item\_selection\_changed2)

        self.cbBox\_2.currentIndexChanged.connect(self.on\_combobox\_changed\_KH)

        self.ngay.setDate(QDate.currentDate())

        self.Load\_data2()

        self.Load\_data3()

        self.btKeHoach.clicked.connect(self.HideWgKH)

        self.btNhiemVu.clicked.connect(self.HideWgNV)

        self.btThem.clicked.connect(self.Them\_data)

        self.btSua.clicked.connect(self.Sua\_data)

        self.btXoa.clicked.connect(self.Xoa\_data)

        self.add\_Item()

        self.cbTK.currentIndexChanged.connect(self.update\_hoten)

        self.cbMaKH.currentIndexChanged.connect(self.update\_hoten)

        self.btThem\_2.clicked.connect(self.Them\_data2)

        self.btXoa\_2.clicked.connect(self.Xoa\_data2)

        self.btSua\_2.clicked.connect(self.Sua\_data2)

        self.cbBox\_2.setVisible(False)

        self.load\_cbbox\_makh\_2()

        self.cbMaKH\_2.currentIndexChanged.connect(self.Load\_data\_makh2)

        self.btTimND.clicked.connect(self.Tim\_data\_NV)

    def Tim\_data\_NV(self):

        search\_text = self.txtNDTim.text()

        if search\_text:

            self.tbNhiemVu.clear()

            query=db.cursor()

            query.execute("SELECT NVnKH.idNVKH,TaiKhoan.tk,TaiKhoan.hoten,KeHoach.idkh,KeHoach.noidung FROM TaiKhoan,KeHoach,NVnKH WHERE TaiKhoan.id=NVnKH.id AND KeHoach.idkh=NVnKH.idKH AND KeHoach.noidung=?",search\_text)

            kq=query.fetchall()

            self.tbNhiemVu.setColumnHidden(0, True)

            self.tbNhiemVu.setColumnCount(5)

            self.tbNhiemVu.setColumnWidth(0, 30)

            self.tbNhiemVu.setColumnWidth(1, 100)

            self.tbNhiemVu.setColumnWidth(2, 150)

            self.tbNhiemVu.setColumnWidth(3, 150)

            self.tbNhiemVu.setColumnWidth(4, 200)

            header\_labels = ['idkh', 'Tài Khoản', 'Họ Tên','Mã Kế Hoạch', 'Nội Dung']

            self.tbNhiemVu.setHeaderLabels(header\_labels)

            for row in kq:

                item = QTreeWidgetItem(self.tbNhiemVu)

                for i, value in enumerate(row):

                    item.setText(i, str(value))

            self.update\_hoten()

        else:

            self.Load\_data3()

        self.txtTim.clear()

    def Load\_data\_makh2(self):

        selected\_idkh = self.cbMaKH\_2.currentText()

        self.tbNhiemVu.clear()

        query=db.cursor()

        query.execute("SELECT NVnKH.idNVKH,TaiKhoan.tk,TaiKhoan.hoten,KeHoach.idkh,KeHoach.noidung FROM TaiKhoan,KeHoach,NVnKH WHERE TaiKhoan.id=NVnKH.id AND KeHoach.idkh=NVnKH.idKH AND KeHoach.idkh=? ",selected\_idkh)

        kq=query.fetchall()

        self.tbNhiemVu.setColumnHidden(0, True)

        self.tbNhiemVu.setColumnCount(5)

        self.tbNhiemVu.setColumnWidth(0, 30)

        self.tbNhiemVu.setColumnWidth(1, 100)

        self.tbNhiemVu.setColumnWidth(2, 150)

        self.tbNhiemVu.setColumnWidth(3, 150)

        self.tbNhiemVu.setColumnWidth(4, 200)

        header\_labels = ['idkh', 'Tài Khoản', 'Họ Tên','Mã Kế Hoạch', 'Nội Dung']

        self.tbNhiemVu.setHeaderLabels(header\_labels)

        for row in kq:

            item = QTreeWidgetItem(self.tbNhiemVu)

            for i, value in enumerate(row):

                item.setText(i, str(value))

        self.update\_hoten()

    def load\_cbbox\_makh\_2(self):

        query=db.cursor()

        query.execute("SELECT idkh FROM KeHoach")

        kq=query.fetchall()

        for row in kq:

            self.cbMaKH\_2.addItem(str(row.idkh))

    def on\_combobox\_changed\_KH(self):

        selected\_text = self.cbBox\_2.currentText()

        if selected\_text == "Tat Ca":

            self.Load\_data2()

        elif selected\_text == "Hoan Thanh":

            self.tbKeHoach.clear()

            query=db.cursor()

            query.execute("select \* from KeHoach WHERE trangthai=?",selected\_text)

            kq=query.fetchall()

            self.tbKeHoach.setColumnHidden(0, True)

            self.tbKeHoach.setColumnCount(5)

            self.tbKeHoach.setColumnWidth(0, 30)

            self.tbKeHoach.setColumnWidth(1, 90) #10

            self.tbKeHoach.setColumnWidth(2, 130) #20

            self.tbKeHoach.setColumnWidth(3, 150) #50

            self.tbKeHoach.setColumnWidth(4, 130) #20

            self.tbKeHoach.setColumnWidth(5, 100)

            header\_labels = ['Stt', 'Ngày giao', 'Nội dung', 'Loại', 'Trạng thái', 'Ngày HT']

            self.tbKeHoach.setHeaderLabels(header\_labels)

            for row in kq:

                item = QTreeWidgetItem(self.tbKeHoach)

                for i, value in enumerate(row):

                    item.setText(i, str(value))

        elif selected\_text == "Chua Hoan Thanh":

            self.tbKeHoach.clear()

            query=db.cursor()

            query.execute("select \* from KeHoach WHERE trangthai=?",selected\_text)

            kq=query.fetchall()

            self.tbKeHoach.setColumnHidden(0, True)

            self.tbKeHoach.setColumnCount(5)

            self.tbKeHoach.setColumnWidth(0, 30)

            self.tbKeHoach.setColumnWidth(1, 90) #10

            self.tbKeHoach.setColumnWidth(2, 130) #20

            self.tbKeHoach.setColumnWidth(3, 150) #50

            self.tbKeHoach.setColumnWidth(4, 130) #20

            self.tbKeHoach.setColumnWidth(5, 100)

            header\_labels = ['Stt', 'Ngày giao', 'Nội dung', 'Loại', 'Trạng thái', 'Ngày HT']

            self.tbKeHoach.setHeaderLabels(header\_labels)

            for row in kq:

                item = QTreeWidgetItem(self.tbKeHoach)

                for i, value in enumerate(row):

                    item.setText(i, str(value))

    def Sua\_data2(self):

        selected\_items = self.tbNhiemVu.selectedItems()

        if selected\_items:

            selected\_item = selected\_items[0].text(0)

            nd=self.txtND.text()

            ten=self.txtTenNV.text()

            selected\_tk = self.cbTK.currentText()

            selected\_idkh = self.cbMaKH.currentText()

            if not nd:

                QMessageBox.warning(self, 'Lỗi', 'Vui lòng chọn lại Mã Kế Hoạch')

                return

            if not ten:

                QMessageBox.warning(self, 'Lỗi', 'Vui lòng chọn lại Tài Khoản')

                return

            query=db.cursor()

            query.execute("UPDATE NVnKH SET idKH=?,id=(SELECT id FROM TaiKhoan WHERE tk = ?) WHERE idNVKH=?", selected\_idkh,selected\_tk,selected\_item)

            db.commit()

            self.Load\_data3()

            self.Load\_data()

        else:

            QMessageBox.information(self,'Thông Báo','Bạn chưa chọn dòng muốn sửa')

    def Xoa\_data2(self):

        selected\_items = self.tbNhiemVu.selectedItems()

        if selected\_items:

            selected\_item = selected\_items[0].text(0)

            query=db.cursor()

            query.execute("DELETE FROM NVnKH WHERE idNVKH=?", selected\_item)

            db.commit()

            self.Load\_data3()

            self.Load\_data()

        else:

            QMessageBox.information(self,'Thông Báo','Bạn chưa chọn dòng muốn xóa')

    def Them\_data2(self):

        nd=self.txtND.text()

        ten=self.txtTenNV.text()

        a=self.txtte.text()

        selected\_tk = self.cbTK.currentText()

        selected\_idkh = self.cbMaKH.currentText()

        if not nd:

            QMessageBox.warning(self, 'Lỗi', 'Vui lòng chọn lại Mã Kế Hoạch')

            return

        if not ten:

            QMessageBox.warning(self, 'Lỗi', 'Vui lòng chọn lại Tài Khoản')

            return

        query\_admin=db.cursor()

        query\_admin.execute("select \* from NVnKH where idKH=? and id=(SELECT id FROM TaiKhoan WHERE tk = ?)",selected\_idkh,selected\_tk)

        kt=query\_admin.fetchone()

        if kt:

            QMessageBox.information(self,"Thông báo","Nhân viên "+ten+" đã được giao kế hoạch "+nd+" rồi!")

        else:

            query=db.cursor()

            query.execute("INSERT INTO NVnKH (idKH,id,tkgiao) VALUES (?,(SELECT id FROM TaiKhoan WHERE tk = ?),?)",selected\_idkh,selected\_tk,a)

            db.commit()

        self.Load\_data3()

        self.Load\_data2()

        self.Load\_data()

    def Load\_data3(self):

        self.tbNhiemVu.clear()

        a=self.txtte.text()

        query=db.cursor()

        query.execute("SELECT NVnKH.idNVKH,TaiKhoan.tk,TaiKhoan.hoten,KeHoach.idkh,KeHoach.noidung FROM TaiKhoan,KeHoach,NVnKH WHERE TaiKhoan.id=NVnKH.id AND KeHoach.idkh=NVnKH.idKH AND NVnKH.tkgiao=?",a)

        kq=query.fetchall()

        self.tbNhiemVu.setColumnHidden(0, True)

        self.tbNhiemVu.setColumnCount(5)

        self.tbNhiemVu.setColumnWidth(0, 30)

        self.tbNhiemVu.setColumnWidth(1, 100)

        self.tbNhiemVu.setColumnWidth(2, 150)

        self.tbNhiemVu.setColumnWidth(3, 150)

        self.tbNhiemVu.setColumnWidth(4, 200)

        header\_labels = ['idkh', 'Tài Khoản', 'Họ Tên','Mã Kế Hoạch', 'Nội Dung']

        self.tbNhiemVu.setHeaderLabels(header\_labels)

        for row in kq:

            item = QTreeWidgetItem(self.tbNhiemVu)

            for i, value in enumerate(row):

                item.setText(i, str(value))

        self.update\_hoten()

    def add\_Item(self):

        self.cbTK.clear()

        self.cbMaKH.clear()

        query=db.cursor()

        query.execute("SELECT tk FROM TaiKhoan")

        kq=query.fetchall()

        for row in kq:

            self.cbTK.addItem(row.tk)

        query=db.cursor()

        query.execute("SELECT idkh FROM KeHoach")

        kq=query.fetchall()

        for row in kq:

            self.cbMaKH.addItem(str(row.idkh))

    def update\_hoten(self):

        # Lấy giá trị tk được chọn từ QComboBox

        selected\_tk = self.cbTK.currentText()

        selected\_idkh = self.cbMaKH.currentText()

        # Thực hiện truy vấn để lấy hoten tương ứng từ cơ sở dữ liệu

        query=db.cursor()

        query.execute("SELECT hoten FROM TaiKhoan WHERE tk=?", (selected\_tk,))

        hoten\_result = query.fetchone()

        # Hiển thị hoten tương ứng trong self.txthoten

        if hoten\_result:

            self.txtTenNV.setText(hoten\_result[0])

        else:

            self.txtTenNV.setText("")

        query=db.cursor()

        query.execute("SELECT noidung FROM KeHoach WHERE idkh=?", (selected\_idkh,))

        noidung\_result = query.fetchone()

        if noidung\_result:

            self.txtND.setText(noidung\_result[0])

        else:

            self.txtND.setText("")

    def Xoa\_data(self):

        selected\_items = self.tbKeHoach.selectedItems()

        if selected\_items:

            selected\_item = selected\_items[0].text(0)

            query=db.cursor()

            query.execute("DELETE FROM NVnKH WHERE idkh=?", selected\_item)

            db.commit()

            query.execute("DELETE FROM KeHoach WHERE idkh=?", selected\_item)

            db.commit()

            self.Load\_data2()

            self.Load\_data()

        else:

            QMessageBox.information(self,'Thông Báo','Bạn chưa chọn dòng muốn xóa')

        self.add\_Item()

        self.Load\_data3()

    def Sua\_data(self):

        selected\_items = self.tbKeHoach.selectedItems()

        if selected\_items:

            selected\_item = selected\_items[0].text(0)

            nd=self.txtNoiDung.text()

            ngay=self.ngay.date().toString("yyyy-MM-dd")

            mota=self.txtMoTa.text()

            trangthai='Chua Hoan Thanh'

            if not nd:

                QMessageBox.warning(self, 'Lỗi', 'Vui lòng nhập phần Nội Dung!')

                return

            if not mota:

                QMessageBox.warning(self, 'Lỗi', 'Vui lòng nhập phần Mô Tả')

                return

            query=db.cursor()

            query.execute("UPDATE KeHoach SET ngay=?, noidung=?, mota=?, trangthai=? WHERE idkh=?", ngay, nd, mota, trangthai, selected\_item)

            db.commit()

            self.Load\_data2()

            self.Load\_data3()

            self.Load\_data()

            self.txtNoiDung.clear()

            self.txtMoTa.clear()

        else:

            QMessageBox.information(self,'Thông Báo','Bạn chưa chọn dòng muốn sửa')

    def on\_item\_selection\_changed2(self):

        selected\_items = self.tbKeHoach.selectedItems()

        if selected\_items:

            self.txtNoiDung.setText(selected\_items[0].text(2))

            self.txtMoTa.setText(selected\_items[0].text(3))

            self.ngay.setDate(QDate.fromString(selected\_items[0].text(1),"yyyy-MM-dd"))

    def Them\_data(self):

        nd=self.txtNoiDung.text()

        ngay=self.ngay.date().toString("yyyy-MM-dd")

        mota=self.txtMoTa.text()

        trangthai='Chua Hoan Thanh'

        if not nd:

            QMessageBox.warning(self, 'Lỗi', 'Vui lòng nhập phần Nội Dung!')

            return

        if not mota:

            QMessageBox.warning(self, 'Lỗi', 'Vui lòng nhập phần Mô Tả')

            return

        query=db.cursor()

        query.execute(f"INSERT INTO KeHoach (ngay, noidung, mota, trangthai) VALUES ( '{ngay}', '{nd}', '{mota}', '{trangthai}')")

        db.commit()

        a=self.txtte.text() #Lấy ra tên tài khoản

        query.execute("INSERT INTO NVnKH (idkh, id) VALUES ((SELECT MAX(idkh) FROM KeHoach),(SELECT id FROM TaiKhoan WHERE tk = ?))",a)

        db.commit()

        self.Load\_data2()

        self.Load\_data()

        self.txtNoiDung.clear()

        self.txtMoTa.clear()

        self.add\_Item()

    def Load\_data2(self):

        self.tbKeHoach.clear()

        a=self.txtte.text()

        query=db.cursor()

        query.execute("SELECT KeHoach.idkh , KeHoach.ngay, KeHoach.noidung, KeHoach.mota, KeHoach.trangthai,KeHoach.ngayht FROM KeHoach JOIN NVnKH ON KeHoach.idkh = NVnKH.idkh WHERE NVnKH.id IN (SELECT id FROM TaiKhoan WHERE tk =?)",a)

        kq=query.fetchall()

        self.tbKeHoach.setColumnCount(5)

        self.tbKeHoach.setColumnHidden(0, True)

        self.tbKeHoach.setColumnWidth(0, 30)

        self.tbKeHoach.setColumnWidth(1, 90)

        self.tbKeHoach.setColumnWidth(2, 130)

        self.tbKeHoach.setColumnWidth(3, 150)

        self.tbKeHoach.setColumnWidth(4, 130)

        self.tbKeHoach.setColumnWidth(5, 100)

        header\_labels = [ 'idKH','Ngày giao', 'Nội dung', 'Loại', 'Trạng thái', 'Ngày HT']

        self.tbKeHoach.setHeaderLabels(header\_labels)

        for row in kq:

            item = QTreeWidgetItem(self.tbKeHoach)

            for i, value in enumerate(row):

                item.setText(i, str(value))

    def HideWgKH(self):

        self.wgKeHoach.setVisible(True)

        self.tbKeHoach.setVisible(True)

        self.wgNhiemVu.setVisible(False)

        self.tbNhiemVu.setVisible(False)

        self.cbBox\_2.setVisible(True)

        self.cbMaKH\_2.setVisible(False)

        self.txtNDTim.setVisible(False)

        self.btTimND.setVisible(False)

        self.label\_12.setText("BẢNG KẾ HOẠCH ĐÃ GIAO")

    def HideWgNV(self):

        self.wgKeHoach.setVisible(False)

        self.tbKeHoach.setVisible(False)

        self.wgNhiemVu.setVisible(True)

        self.tbNhiemVu.setVisible(True)

        self.cbBox\_2.setVisible(False)

        self.cbMaKH\_2.setVisible(True)

        self.txtNDTim.setVisible(True)

        self.btTimND.setVisible(True)

        self.label\_12.setText("BẢNG NHIỆM VỤ ĐÃ GIAO")

    def Thoat(self):

        widget.setFixedSize(310, 180)

        widget.setCurrentIndex(0)

    def DangKy(self):

        widget.setFixedSize(421, 386)

        widget.setCurrentIndex(1)

    def Logout(self):

        self.tabWidget.tabBar().setTabText(2, "Đăng Nhập")

        self.tabWidget.setTabEnabled(2, True)

        widget.setFixedSize(425,320)

        self.tabWidget.setCurrentIndex(2)

        self.tabWidget.tabBar().setTabText(0, "")

        self.tabWidget.tabBar().setTabText(1, "")

        self.tabWidget.setTabEnabled(0, False)

        self.tabWidget.setTabEnabled(1, False)

    def DN(self):

        un=self.txtTk.text()

        self.txtte.setText(un)

        print(self.txtte.text())

        psw=self.txtMk.text()

        if not un:

            QMessageBox.information(self, 'Thông báo', 'Vui lòng nhập Tài Khoản')

            return

        if not psw:

            QMessageBox.information(self, 'Thông báo', 'Vui lòng nhập Mật khẩu')

            return

        query\_admin=db.cursor()

        query\_admin.execute("select \* from TaiKhoan where tk='"+un+"' and mk='"+psw+"' and role = 'admin' ")

        kt\_admin=query\_admin.fetchone()

        if kt\_admin:

            QMessageBox.information(self,"Login Output","Đăng nhập thành công với quyền quản trị")

            self.tabWidget.tabBar().setTabText(0, "Thông Tin Kế Hoạch")

            self.tabWidget.tabBar().setTabText(1, "Quản Lý")

            self.tabWidget.tabBar().setTabText(2, "")

            self.tabWidget.setTabEnabled(0, True)

            self.tabWidget.setTabEnabled(1, True)

            self.tabWidget.setTabEnabled(2, False)

            widget.setFixedSize(691, 631)

            self.tabWidget.setCurrentIndex(1)

        query=db.cursor()

        query.execute("select \* from TaiKhoan where tk='"+un+"' and mk='"+psw+"' and role = 'customer' ")

        kt=query.fetchone()

        if kt:

            QMessageBox.information(self,"Login Output","Đăng nhập thành công")

            self.tabWidget.tabBar().setTabText(0, "Thông Tin Kế Hoạch")

            self.tabWidget.tabBar().setTabText(1, "")

            self.tabWidget.tabBar().setTabText(2, "")

            self.tabWidget.setTabEnabled(0, True)

            self.tabWidget.setTabEnabled(1, False)

            self.tabWidget.setTabEnabled(2, False)

            widget.setFixedSize(691, 631)

            self.tabWidget.setCurrentIndex(0)

        if not kt\_admin and not kt:

            QMessageBox.information(self,"Thông báo","Đăng nhập thất bại")

        self.Load\_data()

        self.Load\_data2()

        self.Load\_data3()

        self.cbTK.clear()

        self.cbMaKH.clear()

        self.add\_Item()

    def Load\_data(self):

        self.tbNhom.clear()

        a=self.txtte.text()

        query=db.cursor()

        query.execute("SELECT KeHoach.idkh , KeHoach.ngay, KeHoach.noidung, KeHoach.mota, KeHoach.trangthai,KeHoach.ngayht FROM KeHoach JOIN NVnKH ON KeHoach.idkh = NVnKH.idkh WHERE NVnKH.id IN (SELECT id FROM TaiKhoan WHERE tk =?)",a)

        kq=query.fetchall()

        self.tbNhom.setColumnCount(5)

        self.tbNhom.setColumnHidden(0, True)

        self.tbNhom.setColumnWidth(0, 30)

        self.tbNhom.setColumnWidth(1, 90)

        self.tbNhom.setColumnWidth(2, 130)

        self.tbNhom.setColumnWidth(3, 150)

        self.tbNhom.setColumnWidth(4, 130)

        self.tbNhom.setColumnWidth(5, 100)

        header\_labels = [ 'idKH','Ngày giao', 'Nội dung', 'Loại', 'Trạng thái', 'Ngày HT']

        self.tbNhom.setHeaderLabels(header\_labels)

        for row in kq:

            item = QTreeWidgetItem(self.tbNhom)

            for i, value in enumerate(row):

                item.setText(i, str(value))

    def HoanT(self):

        selected\_items = self.tbNhom.selectedItems()

        if selected\_items:

            selected\_item = selected\_items[0].text(0)

            ngay=self.ngay\_2.date().toString("yyyy-MM-dd")

            time=None

            if selected\_items[0].text(4)=="Hoan Thanh":

                query=db.cursor()

                query.execute("UPDATE KeHoach SET trangthai='Chua Hoan Thanh',ngayht=? WHERE idkh=?",time, selected\_item)

                db.commit()

            if selected\_items[0].text(4)=="Chua Hoan Thanh":

                query=db.cursor()

                query.execute("UPDATE KeHoach SET trangthai='Hoan Thanh',ngayht=? WHERE idkh=?",ngay, selected\_item)

                db.commit()

            self.Load\_data()

        else:

            QMessageBox.information(self,'Thông Báo','Bạn chưa chọn dòng')

        self.Load\_data2()

    def on\_item\_selection\_changed(self):

        selected\_items = self.tbNhom.selectedItems()

        if selected\_items:

            if selected\_items[0].text(4)=="Hoan Thanh":

                self.btHT.setText("Làm Lại")

            else:

                self.btHT.setText("Hoàn thành")

    def on\_combobox\_changed(self):

        selected\_text = self.cbBox.currentText()

        if selected\_text == "Tat Ca":

            self.Load\_data()

        elif selected\_text == "Hoan Thanh":

            self.tbNhom.clear()

            a=self.txtte.text()

            query=db.cursor()

            query.execute("SELECT KeHoach.idkh , KeHoach.ngay, KeHoach.noidung, KeHoach.mota, KeHoach.trangthai,KeHoach.ngayht FROM KeHoach JOIN NVnKH ON KeHoach.idkh = NVnKH.idkh WHERE NVnKH.id IN (SELECT id FROM TaiKhoan WHERE tk =?) AND KeHoach.trangthai='Hoan Thanh'",a)

            kq=query.fetchall()

            self.tbNhom.setColumnCount(5)

            self.tbNhom.setColumnHidden(0, True)

            self.tbNhom.setColumnWidth(0, 30)

            self.tbNhom.setColumnWidth(1, 90)

            self.tbNhom.setColumnWidth(2, 130)

            self.tbNhom.setColumnWidth(3, 150)

            self.tbNhom.setColumnWidth(4, 130)

            self.tbNhom.setColumnWidth(5, 100)

            header\_labels = [ 'idKH','Ngày', 'Nội dung', 'Loại', 'Trạng thái', 'Ngày HT']

            self.tbNhom.setHeaderLabels(header\_labels)

            for row in kq:

                item = QTreeWidgetItem(self.tbNhom)

                for i, value in enumerate(row):

                    item.setText(i, str(value))

        elif selected\_text == "Chua Hoan Thanh":

            self.tbNhom.clear()

            a=self.txtte.text()

            query=db.cursor()

            query.execute("SELECT KeHoach.idkh , KeHoach.ngay, KeHoach.noidung, KeHoach.mota, KeHoach.trangthai,KeHoach.ngayht FROM KeHoach JOIN NVnKH ON KeHoach.idkh = NVnKH.idkh WHERE NVnKH.id IN (SELECT id FROM TaiKhoan WHERE tk =?) AND KeHoach.trangthai='Chua Hoan Thanh'",a)

            kq=query.fetchall()

            self.tbNhom.setColumnCount(5)

            self.tbNhom.setColumnHidden(0, True)

            self.tbNhom.setColumnWidth(0, 30)

            self.tbNhom.setColumnWidth(1, 90)

            self.tbNhom.setColumnWidth(2, 130)

            self.tbNhom.setColumnWidth(3, 150)

            self.tbNhom.setColumnWidth(4, 130)

            self.tbNhom.setColumnWidth(5, 100)

            header\_labels = [ 'idKH','Ngày', 'Nội dung', 'Loại', 'Trạng thái', 'Ngày HT']

            self.tbNhom.setHeaderLabels(header\_labels)

            for row in kq:

                item = QTreeWidgetItem(self.tbNhom)

                for i, value in enumerate(row):

                    item.setText(i, str(value))

    def Tim\_data(self):

        search\_text = self.txtTim.text()

        if search\_text:

            self.tbNhom.clear()

            a=self.txtte.text()

            query=db.cursor()

            query.execute("SELECT KeHoach.idkh , KeHoach.ngay, KeHoach.noidung, KeHoach.mota, KeHoach.trangthai, KeHoach.ngayht FROM KeHoach JOIN NVnKH ON KeHoach.idkh = NVnKH.idkh WHERE NVnKH.id IN (SELECT id FROM TaiKhoan WHERE tk =?) AND KeHoach.noidung=?",a,search\_text)

            kq=query.fetchall()

            self.tbNhom.setColumnCount(5)

            self.tbNhom.setColumnHidden(0, True)

            self.tbNhom.setColumnWidth(0, 30)

            self.tbNhom.setColumnWidth(1, 90)

            self.tbNhom.setColumnWidth(2, 130)

            self.tbNhom.setColumnWidth(3, 150)

            self.tbNhom.setColumnWidth(4, 130)

            self.tbNhom.setColumnWidth(5, 100)

            header\_labels = [ 'idKH','Ngày', 'Nội dung', 'Loại', 'Trạng thái', 'Ngày HT']

            self.tbNhom.setHeaderLabels(header\_labels)

            for row in kq:

                item = QTreeWidgetItem(self.tbNhom)

                for i, value in enumerate(row):

                    item.setText(i, str(value))

        else:

            self.Load\_data()

        self.txtTim.clear()

    def export\_to\_excel(self):

        # Tạo một Workbook

        workbook = Workbook()

        # Tạo một WorkSheet

        worksheet = workbook.active

        # Đặt tiêu đề cột

        columns = ["Ngày", "Nội dung", "Loại","Trạng thái", "Ngày HT"]

        worksheet.append(columns)

        # Lặp qua dữ liệu QTreeWidget và thêm vào WorkSheet

        for row in range(self.tbNhom.topLevelItemCount()):

            item = self.tbNhom.topLevelItem(row)

            row\_data = [item.text(col) for col in range(1,item.columnCount())]

            worksheet.append(row\_data)

        file=self.txtEx.text()

        if not file:

            QMessageBox.warning(self, 'Lỗi', 'Vui lòng nhập phần tên cho File Excel')

            return

        # Lưu Workbook thành tệp Excel

        workbook.save(file+".xlsx")

        QMessageBox.information(self,'Thông Báo','Đã lưu ra File Excel')