|  |  |
| --- | --- |
| ĐOÀN VĂN QUÂN  CÔNG NGHỆ THÔNG TIN | **BỘ CÔNG THƯƠNG**  **TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**  **---------------------------------------**  ĐỒ ÁN, KLTN ĐẠI HỌC  NGÀNH CÔNG NGHỆ THÔNG TIN  **PHÁT TRIỂN ỨNG DỤNG ĐA NỀN TẢNG QUẢN LÝ ĐIỂM CHO HỌC SINH TRUNG HỌC PHỔ THÔNG**  **CBHD: T.s Nguyễn Bá Nhiễn**  **Sinh viên: Đoàn Văn Quân**  **Mã số sinh viên: 20206022342**  Hà Nội – Năm 2024 |
| ĐOÀN VĂN QUÂN  CÔNG NGHỆ THÔNG TIN | **BỘ CÔNG THƯƠNG**  **TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**  **---------------------------------------**  ĐỒ ÁN, KLTN ĐẠI HỌC  NGÀNH CÔNG NGHỆ THÔNG TIN  **PHÁT TRIỂN ỨNG DỤNG ĐA NỀN TẢNG QUẢN LÝ ĐIỂM CHO HỌC SINH TRUNG HỌC PHỔ THÔNG**  **CBHD: T.s Nguyễn Bá Nhiễn**  **Sinh viên: Đoàn Văn Quân**  **Mã số sinh viên: 20206022342**  Hà Nội – Năm 2024 |

**MỤC LỤC**

[DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT](#_Toc166792221)

[DANH MỤC HÌNH ẢNH](#_Toc166792222)

[DANH MỤC BẢNG BIỂU](#_Toc166792223)

[LỜI CẢM ƠN](#_Toc166792224)

[LỜI MỞ ĐẦU 1](#_Toc166792225)

[CHƯƠNG 1. TỔNG QUAN VỀ NỘI DUNG NGHIÊN CỨU 4](#_Toc166792226)

[1.1. Giới thiệu chung 4](#_Toc166792227)

[1.2. Giới thiệu về dự án phần mềm 4](#_Toc166792228)

[1.3. Cơ sở lý thuyết 5](#_Toc166792229)

[1.3.1. Tổng quan về lập trình đa nền tảng 5](#_Toc166792230)

[1.3.2. Giới thiệu công cụ lập trình Microsoft Visual Studio 6](#_Toc166792231)

[1.3.3. Ngôn ngữ lập trình C# 7](#_Toc166792232)

[1.3.4. Giới thiệu Framework Aavalonia 8](#_Toc166792233)

[1.3.5. Tổng quan về mô hình MVVM 9](#_Toc166792234)

[CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 11](#_Toc166792235)

[2.1. Phân tích yêu cầu 11](#_Toc166792236)

[2.1.1. Về hệ thống 11](#_Toc166792237)

[2.1.2. Khảo sát 11](#_Toc166792238)

[2.1.3. Yêu cầu về chức năng 13](#_Toc166792239)

[2.2. Thiết kế hệ thống 13](#_Toc166792240)

[2.2.1. Mô hình hóa Usecase 13](#_Toc166792241)

[2.2.2. Mô tả chi tiết các Usecase 17](#_Toc166792242)

[2.2.3. Biểu đồ lớp hệ thống 54](#_Toc166792243)

[2.3. Thiết kế cơ sở dữ liệu 55](#_Toc166792244)

[2.3.1. Thiết kế cơ sở dữ liệu mức vật lý 55](#_Toc166792245)

[2.3.2. Thiết kế cơ sở dữ liệu mức logic 56](#_Toc166792246)

[2.4. Thiết kế giao diện 58](#_Toc166792247)

[2.4.1. Giao diện đăng nhập 58](#_Toc166792248)

[2.4.2. Giao diện cài đặt tài khoản 59](#_Toc166792249)

[2.4.3. Giao diện xem điểm thành phần 59](#_Toc166792250)

[2.4.4. Giao diện xem điểm tổng kết 60](#_Toc166792251)

[2.4.5. Giao diện quản lý tài khoản 60](#_Toc166792252)

[2.4.6. Giao diện quản lý điểm 61](#_Toc166792253)

[2.4.7. Giao diện quản lý lớp học 61](#_Toc166792254)

[2.4.8. Giao diện quản lý khóa học 62](#_Toc166792255)

[2.4.9. Giao diện quản lý học sinh 62](#_Toc166792256)

[2.4.10. Giao diện quản lý giáo viên 63](#_Toc166792257)

[CHƯƠNG 3. KẾT QUẢ 64](#_Toc166792258)

[3.1. Cài đặt môi trường 64](#_Toc166792259)

[3.2. Kết quả thu được 65](#_Toc166792260)

[3.2.1. Cài đặt tài khoản 65](#_Toc166792261)

[3.2.2. Xem điểm thành phần 66](#_Toc166792262)

[3.2.3. Xem điểm tổng kết 67](#_Toc166792263)

[3.2.4. Quản lý tài khoản 68](#_Toc166792264)

[3.2.5. Quản lý điểm 69](#_Toc166792265)

[3.2.6. Quản lý lớp học 70](#_Toc166792266)

[3.2.7. Quản lý khóa học 71](#_Toc166792267)

[3.2.8. Quản lý học sinh 72](#_Toc166792268)

[3.2.9. Quản lý giáo viên 73](#_Toc166792269)

[3.3. Kết quả kiểm thử 74](#_Toc166792270)

[3.3.1. Kỹ thuật kiểm thử hộp đen (Black box testing) 74](#_Toc166792271)

[3.3.2. Kiểm thử chức năng 75](#_Toc166792272)

[KẾT LUẬN 82](#_Toc166792273)

[TÀI LIỆU THAM KHẢO 84](#_Toc166792274)

DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| **STT** | **Viết tắt** | **Dịch nghĩa** |
| 1 | SQL | Structured Query Language |
| 2 | XML | Extensible Markup Language |
| 3 | RDBMS | Hệ quản trị cơ sở dữ liệu quan hệ |
| 4 | MVC | Model-View-Controller |
| 5 | MVVM | Model-View-ViewModel |
| 6 | CSDL | Cơ sở dữ liệu |
| 7 | SDK | Software Development Kit |

DANH MỤC HÌNH ẢNH

[Hình 1.1. Mô hình MVVM 9](#_Toc166791859)

[Hình 2.1. Biểu đồ Usecase chính 14](#_Toc166791868)

[Hình 2.2. Biểu đồ phân rã Usecase phía Frontend 15](#_Toc166791869)

[Hình 2.3. Biểu đồ phân rã Usecase phía Backend. 16](#_Toc166791870)

[Hình 2.4. Biểu đồ lớp phân tích use case đăng nhập 18](#_Toc166791871)

[Hình 2.5. Biểu đồ trình tự Usecase đăng nhập 19](#_Toc166791872)

[Hình 2.6. Biểu đồ lớp phân tích Usecase cài đặt tài khoản 21](#_Toc166791873)

[Hình 2.7. Biểu đồ trình tự Usecase cài đặt tài khoản. 23](#_Toc166791874)

[Hình 2.8. Biểu đồ lớp phân tích Usecase xem điểm thành phần 25](#_Toc166791875)

[Hình 2.9. Biểu đồ trình tự Usecase xem điểm thành phần 26](#_Toc166791876)

[Hình 2.10. Biểu đồ lớp phân tích Usecase xem điểm tổng kết 28](#_Toc166791877)

[Hình 2.11. Biểu đồ trình tự usecase Xem điểm tổng kết 29](#_Toc166791878)

[Hình 2.12. Biểu đồ lớp phân tích Usecase quản lý tài khoản 31](#_Toc166791879)

[Hình 2.13. Biểu đồ trình tự Usecase quản lý tài khoản 32](#_Toc166791880)

[Hình 2.14. Biểu đồ lớp phân tích Usecase quản lý điểm 36](#_Toc166791881)

[Hình 2.15. Biểu đồ trình tự usecase quản lý điểm 38](#_Toc166791882)

[Hình 2.16. Biểu đồ lớp phân tích Usecase quản lý lớp học 40](#_Toc166791883)

[Hình 2.17. Biểu đồ trình tự Usecase quản lý lớp học 41](#_Toc166791884)

[Hình 2.18. Biểu đồ lớp phân tích usecase quản lý khóa học. 44](#_Toc166791885)

[Hình 2.19. Biểu đồ trình tự usecase Quản lý khóa học. 46](#_Toc166791886)

[Hình 2.20. Biểu đồ lớp phân tích usecase Quản lý học sinh. 48](#_Toc166791887)

[Hình 2.21. Biểu đồ trình tự usecase Quản lý học sinh 49](#_Toc166791888)

[Hình 2.22. Biểu đồ lớp phân tích usecase quản lý giáo viên. 52](#_Toc166791889)

[Hình 2.23. Biểu đồ trình tự usecase Quản lý giáo viên 53](#_Toc166791890)

[Hình 2.24. Biểu đồ lớp hệ thống 54](#_Toc166791891)

[Hình 2.25. Bảng thiết kế cơ sở dữ liệu mức vật lý. 55](#_Toc166791892)

[Hình 2.26. Biểu đồ diagram của hệ thống. 56](#_Toc166791893)

[Hình 2.27. Bảng thiết kế cơ sở dữ liệu mức logic. 57](#_Toc166791894)

[Hình 2.28. Giao diện đăng nhập 58](#_Toc166791895)

[Hình 2.29. Giao diện đăng nhập tài khoản. 59](#_Toc166791896)

[Hình 2.30. Giao diện xem điểm thành phần 59](#_Toc166791897)

[Hình 2.31. Giao diện xem điểm tổng kết 60](#_Toc166791898)

[Hình 2.32. Giao diện quản lý tài khoản 60](#_Toc166791899)

[Hình 2.33. Giao diện quản lý điểm. 61](#_Toc166791900)

[Hình 2.34. Giao diện quản lý lớp học. 61](#_Toc166791901)

[Hình 2.35. Giao diện quản lý khóa học. 62](#_Toc166791902)

[Hình 2.36. Giao diện quản lý học sinh. 62](#_Toc166791903)

[Hình 2.37. Giao diện quản lý giáo viên. 63](#_Toc166791904)

[Hình 3.1. Microsoft Visual Studio 2022 64](#_Toc166791905)

[Hình 3.2. Cài đặt Avalonia for Visual Studio 2022. 65](#_Toc166791906)

[Hình 3.3. Màn hình cài đặt tài khoản trên nền tảng Windows. 65](#_Toc166791907)

[Hình 3.4. Màn hình cài đặt tài khoản trên nền tảng Android, IOS. 66](#_Toc166791908)

[Hình 3.5. Màn hình xem điểm thành phần trên nền tảng Windows. 66](#_Toc166791909)

[Hình 3.6. Màn hình xem điểm thành phần trên nền tảng Android, IOS 67](#_Toc166791910)

[Hình 3.7. Màn hình xem điểm tổng kết trên nền tảng Windows. 67](#_Toc166791911)

[Hình 3.8. Màn hình xem điểm tổng kết trên nền tảng Android, IOS. 68](#_Toc166791912)

[Hình 3.9. Màn hình quản lý tài khoản trên nền tảng Windows. 68](#_Toc166791913)

[Hình 3.10. Màn hình quản lý tài khoản trên nền tảng Android, IOS. 69](#_Toc166791914)

[Hình 3.11. Màn hình quản lý điểm trên nền tảng Windows. 69](#_Toc166791915)

[Hình 3.12. Màn hình quản lý điểm trên nền tảng Android, IOS. 70](#_Toc166791916)

[Hình 3.13. Màn hình quản lý lớp học trên nền tảng Windows. 70](#_Toc166791917)

[Hình 3.14. Màn hình quản lý lớp học trên nền tảng Android, IOS. 71](#_Toc166791918)

[Hình 3.15. Màn hình quản lý khóa học trên nền tảng Windows. 71](#_Toc166791919)

[Hình 3.16. Màn hình quản lý khóa học trên nền tảng Android, IOS. 72](#_Toc166791920)

[Hình 3.17. Màn hình quản lý học sinh trên nền tảng Windows. 72](#_Toc166791921)

[Hình 3.18. Màn hình quản lý học sinh trên nền tảng Android, IOS. 73](#_Toc166791922)

[Hình 3.19. Màn hình quản lý giáo viên trên nền tảng Windows. 73](#_Toc166791923)

[Hình 3.20. Màn hình quản lý giáo viên trên nền tảng Android, IOS. 74](#_Toc166791924)

DANH MỤC BẢNG BIỂU

[Bảng 2.1. Mô tả usecase đăng nhập 17](#_Toc166791988)

[Bảng 2.2. Mô tả Usecase cài đặt tài khoản. 19](#_Toc166791989)

[Bảng 2.3. Mô tả Usecase Xem điểm thành phần 23](#_Toc166791990)

[Bảng 2.4. Mô tả Usecase Xem điểm tổng kết 26](#_Toc166791991)

[Bảng 2.5. Mô tả Usecase Quản lý tài khoản 29](#_Toc166791992)

[Bảng 2.6. Mô tả Usecase Quản lý điểm 32](#_Toc166791993)

[Bảng 2.7. Mô tả Usecase Quản lý lớp học 38](#_Toc166791994)

[Bảng 2.8. Mô tả Usecase Quản lý khóa học. 41](#_Toc166791995)

[Bảng 2.9. Mô tả Usecase Quản lý học sinh 46](#_Toc166791996)

[Bảng 2.10. Mô tả Usecase Quản lý giáo viên 49](#_Toc166791997)

[Bảng 3.1. Bảng kiểm thử chức năng 75](#_Toc166791998)

LỜI CẢM ƠN

Trong suốt thời gian học tập và thực hiện đồ án tốt nghiệp em luôn nhận được sự hướng dẫn nhiệt tình từ phía nhà trường và các thầy cô giáo. Vì thế, lời đầu tiên em xin chân thành cảm ơn trường Đại học Công nghiệp Hà Nội, thầy cô giáo khoa Công nghệ thông tin đã tạo điều kiện thuận lợi và truyền dạy kiến thức cho em trong thời gian qua để em hoàn thành đề tài một cách tốt nhất. Và hơn hết em xin bày tỏ lòng biết ơn sâu sắc nhất tới thầy giáo hướng dẫn **TS. Nguyễn Bá Nghiễn** đã tận tình giúp đỡ, định hướng em rất nhiều trong suốt quá trình tìm hiểu nghiên cứu và hoàn thành đồ án tốt nghiệp.

Khi thực hiện nghiên cứu đề tài, em đã hoàn thành tốt nhất báo cáo đồ án tốt nghiệp. Em rất mong nhận được sự đóng góp của tất cả các thầy cô giáo để đồ án của em được đầy đủ và hoàn chỉnh hơn.

Em xin chân thành cảm ơn!

LỜI MỞ ĐẦU

1. **Lý do chọn đề tài**

Trong bối cảnh công nghệ thông tin phát triển mạnh mẽ, việc ứng dụng công nghệ vào giáo dục ngày càng trở nên cần thiết và quan trọng. Quản lý điểm học sinh là một phần không thể thiếu trong quá trình giáo dục, góp phần đánh giá hiệu quả học tập và sự tiến bộ của học sinh. Tuy nhiên, việc quản lý điểm một cách truyền thống thường gặp nhiều khó khăn và hạn chế, đặc biệt là trong việc lưu trữ, truy xuất và chia sẻ thông tin một cách hiệu quả.

Với sự ra đời và phát triển của các framework lập trình đa nền tảng như Avalonia, việc xây dựng các ứng dụng quản lý điểm học sinh trở nên dễ dàng và tiện lợi hơn bao giờ hết. Các ứng dụng đa nền tảng không chỉ giúp tiết kiệm thời gian và công sức của người phát triển, mà còn đảm bảo sự đồng nhất và mượt mà khi chạy trên các hệ điều hành khác nhau như Windows, macOS, Linux, Android và iOS.

Vì vậy, em chọn đề tài "Phát triển ứng dụng đa nền tảng quản lý điểm cho học sinh trung học phổ thông" nhằm mục đích nghiên cứu và xây dựng một giải pháp hiệu quả để hỗ trợ giáo viên và nhà trường trong việc quản lý điểm số. Ứng dụng sẽ cung cấp các tính năng như nhập điểm, tính toán điểm trung bình, quản lý hồ sơ học sinh, và tạo báo cáo học tập. Đặc biệt, việc áp dụng công nghệ đa nền tảng sẽ giúp ứng dụng dễ dàng tiếp cận và sử dụng trên nhiều thiết bị khác nhau, từ máy tính để bàn đến điện thoại di động, mang lại sự thuận tiện và linh hoạt cho người dùng.

1. **Đối tượng, mục tiêu và phạm vi nghiên cứu**
2. ***Mục tiêu nghiên cứu***

Mục tiêu nghiên cứu của đề tài " Phát triển ứng dụng đa nền tảng quản lý điểm cho học sinh trung học phổ thông " bao gồm:

* Nghiên cứu và phân tích các nhu cầu và yêu cầu của người dùng về ứng dụng quản lý điểm.
* Tìm hiểu và áp dụng Framework Avalonia để thực hiện xây dựng và phát triển ứng dụng đa nền tảng quản lý điểm cho học sinh trung học phổ thông.
* Đảm bảo tính hiệu quả của ứng dụng thông qua việc kiểm thử và đánh giá, đồng thời đảm bảo tính ổn định và bảo mật thông tin cá nhân của người dùng trong quá trình sử dụng ứng dụng.
* Tìm hiểu và áp dụng các công nghệ mới và tiên tiến trong quá trình phát triển ứng dụng để nâng cao tính năng và trải nghiệm của người dùng.

1. ***Đối tượng nghiên cứu***

* Ngôn ngữ lập trình C# để phát triển ứng dụng trên Flutter, cùng với các công cụ hỗ trợ như Visual Studio Code
* Mô hình MVVM (Model-View-ViewModel) được sử dụng để phát triển ứng dụng.
* Một số design pattern như: Dependency Injection, Singleton, …
* Quy trình phát triển phần mềm
* Các vấn đề liên quan đến quản lý điểm cho học sinh trung học phổ thông.

1. ***Phạm vi nghiên cứu***

Phạm vi nghiên cứu của đề tài " Phát triển ứng dụng đa nền tảng quản lý điểm cho học sinh trung học phổ thông " bao gồm:

* Phát triển một ứng dụng đa nền tảng bằng Avalonia để hỗ trợ nhà trường trong quản lý điểm cho học sinh.
* Các tính năng của ứng dụng bao gồm quản lý điểm, quản lý học sinh, quản lý giáo viên, quản lý lớp học, xem điểm thành phần, tổng kết.
* Phát triển giao diện người dùng thân thiện và dễ sử dụng.

1. **Phương pháp nghiên cứu**

* ***Phương pháp nghiên cứu về mặt lý thuyết***
* Tiến hành nghiên cứu và thu thập các tài liệu liên quan đến đề tài, bao gồm các tài liệu về quản lý điểm, ứng dụng đa nền tảng, Avalonia, và các công nghệ liên quan.
* Tổng hợp và phân tích các tài liệu thu thập được.
* Chọn lọc và sử dụng các tài liệu nghiên cứu để báo cáo đồ án tốt nghiệp.
* ***Phương pháp nghiên cứu trong thực nghiệm***
* Tìm hiểu các tiêu chí kiểm thử chức năng và cách thức thực hiện.
* Phân tích các ứng dụng có chức năng tương tự để tìm hiểu các tiêu chuẩn thiết kế và triển khai ứng dụng.
* Tiến hành xây dựng ứng dụng theo phân tích và yêu cầu thực tế.
* Tìm hiểu về các dịch vụ điện toán đám mây như Azure.
* Tổng hợp kiến thức đã học và hoàn thành báo cáo đồ án tốt nghiệp.

1. **Cấu trúc của báo cáo**

Ngoài hai phần Mở đầu và Kết luân, Báo cáo đồ án tốt nghiệp còn bao gồm năm chương như sau:

* Nội dung chương 1: Giới thiệu tổng quan về dự án phát triển ứng dụng đa nền tảng quản lý điểm cho học sinh trung học phổ thông
* Nội dung chương 2: Phân tích và thiết kế hệ thống gồm phân tích yêu cầu, mô hình hóa Usecase, mô tả chi tiết Usecase, thiết kế giao diện ứng dụng.
* Nội dung chương 3: Thực hiện cài đặt môi trường phát triển ứng dụng, thực hiện kiểm thử.

# TỔNG QUAN VỀ NỘI DUNG NGHIÊN CỨU

## Giới thiệu chung

Một số thông tin chung về dự án:

* Tên đề tài: Phát triển ứng dụng đa nền tảng quản lý điểm cho học sinh trung học phổ thông.
* Đối tượng sử dụng: Trường Trung Học Phổ Thông Việt Yên Số 2.
* Người triển khai: Đoàn Văn Quân
* Thời gian bắt đầu:
* Thời gian kết thúc:

## Giới thiệu về dự án phần mềm

Trong bối cảnh sự phát triển không ngừng của công nghệ và sự gia tăng về yêu cầu đối với quản lý giáo dục, việc áp dụng công nghệ vào việc quản lý điểm thi cho học sinh trung học phổ thông đang trở thành một xu hướng không thể phủ nhận. Để đáp ứng nhu cầu này, các ứng dụng đa nền tảng quản lý điểm cho học sinh trung học phổ thông sẽ trở thành một giải pháp hiệu quả và tiện lợi.

Việc sử dụng Framework Avalonia để phát triển ứng dụng đa nền tảng quản lý điểm cho học sinh trung học phổ thông, các em học sinh và các thầy cô có thể sử dụng ứng dụng trên nhiều nền tảng khác nhau như Windows, Browsers, Adroid, IOS,… Các em học sinh có thể theo dõi bảng điểm của từng môn học theo kỳ học và từng năm học, xem điểm tổng kết,... Ngoài ra, ứng dụng còn cung cấp nhiều tính năng hữu ích khác như quản lý tài khoản, quản lý lớp học, quản lý khóa học, và nhiều tính năng khác.

Trong tổng quan, việc sử dụng Frameword Avalonia để phát triển ứng dụng đa nền tảng quản lý điểm cho học sinh trung học phổ thông là một giải pháp hiệu quả và tiện lợi.

## Cơ sở lý thuyết

### Tổng quan về lập trình đa nền tảng

Đa nền tảng là một thuật ngữ thường được sử dụng trong lĩnh vực công nghệ thông tin, dùng để chỉ các phần mềm hoặc một phương thức điện toán nào đó có thể vận hành trên các nền tảng khác nhau.

Nhờ có lập trình đa nền tảng mà các ứng dụng tạo ra có thể vận hành mượt mà trên các hệ điều hành, từ Android, iOS đến BlackBerry OS hay Windows Phone và những kiến trúc máy tính đang được người dùng sử dụng.

Hiện nay, có khá nhiều các ứng dụng đa nền tảng đang được sử dụng, nhưng thực tế chúng chỉ có 2 loại nhất định:

* Loại có thể chạy trực tiếp trên các nền tảng mà không cần đến sự hỗ trợ của trình biên dịch hay thông dịch. Những ứng dụng này thường được tạo ra bởi những ngôn ngữ thông dịch hay đã được dịch sẵn qua mã bytecode.

Loại còn lại được tạo ra khá đơn giản, nhưng đòi hỏi chương trình đã có bước thiết kế hoặc biên dịch phù hợp với các nền tảng mà nó phục vụ.

#### Ưu điểm

* Sản phẩm được tạo ra từ lập trình đa nền tảng sẽ nhanh chóng tiếp cận được đến nhiều đối tượng người dùng hơn.
* Tiết kiệm chi phí cho chủ sở hữu khi thực hiện.
* Tiết kiệm thời gian công sức cho các lập trình viên trong việc xây dựng và phát triển ứng dụng phục vụ nhu cầu người dùng.
* Đa phần các ứng dụng được tạo ra bởi lập trình đa nền tảng đều ứng dụng những công nghệ hiện đại và tiên tiến, mang đến những trải nghiệm tốt nhất cho người dùng.
* Thuận tiện hơn trong việc triển khai và duy trì hoạt động hóa ứng dụng trên các nền tảng thiết bị.
* Các đoạn code xây dựng chương trình có thể tái sử dụng lại nhiều lần.

#### Nhược điểm

* Việc lập trình đa nền tảng sẽ khó khai thác triệt để thư viện trong những nền tảnh lớn, phổ biến là Android và iOS.
* Khó khăn trong việc đảm bảo chất lượng cho ứng dụng khi hoạt động trên các nền tảng khác nhau. Có thể là về giao diện hiển thị, công cụ hay ngôn ngữ trong nền tảng API và một số yêu cầu xử lý từ các bên liên quan.
* Bị giới hạn trong khả năng linh hoạt so với các ứng dụng được xây dựng dành riêng cho một nền tảng nhất định.

### Giới thiệu công cụ lập trình Microsoft Visual Studio

Microsoft Visual Studio là môi trường phát triển tích hợp (IDE) được thiết kế dành cho giới lập trình viên và các nhà phát triển ứng dụng. Đây là công cụ hỗ trợ phát triển phần mềm mạnh mẽ của Microsoft, cho phép người dùng viết, dịch mã và gỡ lỗi các ứng dụng dựa trên nhiều ngôn ngữ lập trình khác nhau như C++, C#, Visual Basic, Python, JavaScript.

Microsoft Visual Studio bao gồm một trình biên tập mã nguồn, các công cụ gỡ lỗi và xây dựng ứng dụng đa nền tảng.

Dưới đây là một số tính năng và công cụ hỗ trợ của Visual Studio:

* Hỗ trợ đa ngôn ngữ lập trình.
* Hỗ trợ gỡ lỗi mạnh mẽ.
* Tích hợp GitHub để giúp bạn xây dựng các tính năng ứng dụng phổ biến và nhập mã mẫu.
* Tự động hoàn thành mã.
* Hỗ trợ đa nền tảng: Window, Android, IOS, Web, macOS, Linux.

Tóm lại, VSCode là một công cụ lập trình đa nền tảng và rất phổ biến, với nhiều tính năng hữu ích giúp cho việc phát triển phần mềm trở nên dễ dàng hơn.

### Ngôn ngữ lập trình C#

C# là một ngôn ngữ lập trình đơn giản, được phát triển bởi đội ngũ kỹ sư của Microsoft vào năm 2000. C# là một ngôn ngữ lập trình hiện đại, hướng đối tượng, và được xây dựng trên nền tảng của hai ngôn ngữ mạnh nhất là C++ và Java.

C# là một phần của .NET Framework và .NET Core, cho phép phát triển các ứng dụng đa nền tảng trên các hệ điều hành khác nhau như Windows, Linux, MacOS,… Điều này tạo điều kiện thuận lợi cho việc chia sẻ mã nguồn giữa các nền tảng.

* ***Ưu điểm***
* Tính năng hiện đại: C# được thiết kế với các tính năng hiện đại như hướng đối tượng, lập trình linh hoạt, kiểu dữ liệu an toàn, quản lý bộ nhớ tự động (garbage collection) và nhiều tính năng khác để hỗ trợ phát triển ứng dụng lớn và phúc tạp.
* Đơn giản và dễ đọc: Cú pháp của C# rất gần với ngôn ngữ lập trình C++ và Java, điều này giúp cho những lập trình viên đã có kinh nghiệm trong các ngôn ngữ này dễ dàng học và sử dụng C#.
* Tính đa năng: C# là một ngôn ngữ đa mục đích, có khả năng phát triển các loại ứng dụng từ ứng dụng máy tính đến ứng dụng web và cả ứng dụng di động.
* Tính tương thích: C# là một phần của .NET Framework và .NET Core, cho phép phát triển ứng dụng trên nhiều nền tảng như Windows, macOS và Linux. Điều này tạo điều kiện thuận lợi cho việc chia sẻ mã nguồn giữa các nền tảng.
* ***Nhược điểm***
* Phụ thuộc vào môi trường Microsoft: C# thường phụ thuộc vào các công cụ và framework của Microsoft, điều này có thể giới hạn tính đa nền tảng của ứng dụng nếu bạn muốn chạy trên các hệ thống không phải của Microsoft.
* Tài liệu hạn chế: So với một số ngôn ngữ lập trình khác như Java hoặc Python, tài liệu và cộng đồng hỗ trợ C# có thể hạn chế hơn.
* Tốn thời gian biên dịch: Do C# là một ngôn ngữ tĩnh, mã nguồn cần được biên dịch thành mã máy trước khi chạy, điều này có thể làm tăng thời gian phát triển so với các ngôn ngữ biên dịch tức thời.
* Giới hạn cho các ứng dụng đặc biệt: C# thường được sử dụng cho phát triển các ứng dụng Windows, trò chơi, ứng dụng di động (qua Xamarin), và các ứng dụng web ASP.NET. Nếu bạn muốn phát triển một loại ứng dụng khác, có thể có những ngôn ngữ phù hợp hơn.

### Giới thiệu Framework Aavalonia

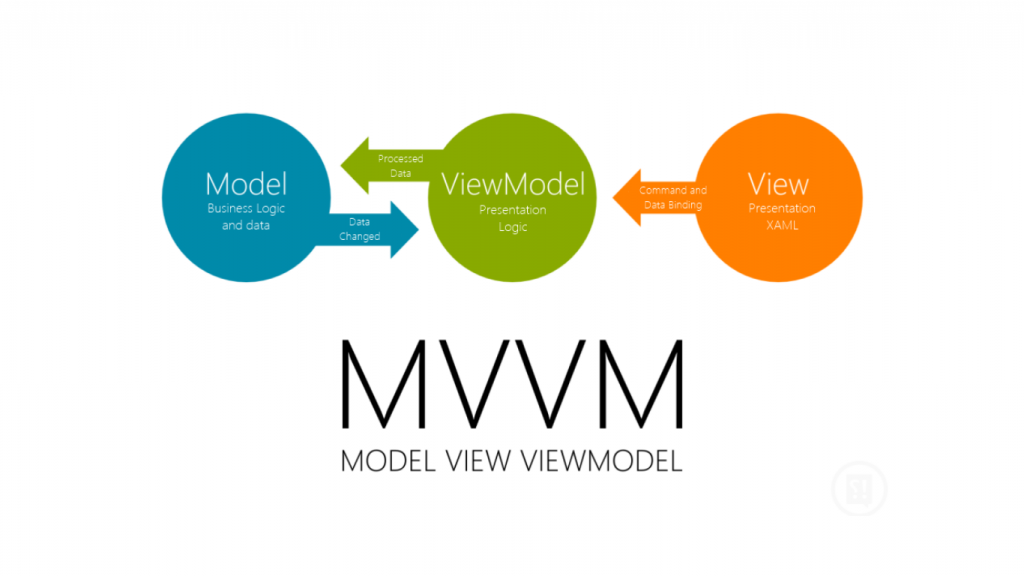
Avalonia là một framework mạnh mẽ cho phép các lập trình viên tạo ứng dụng đa nền tảng bằng .NET. Nó sử dụng công cụ kết xuất của riêng mình để kết xuất giao diện người dùng (UI), đảm bảo sự nhất quán về hình thức và hành vi trên các nền tảng khác nhau, bao gồm Windows, MacOS, Linux, Android, IOS và WebAssembly. Điều này có nghĩa là các lập trình viên, có thể chia sẻ mã giao diện người dung của mình và duy trì một giao diện và cảm giác thống nhất bất kể nền tảng nào.

* ***Ưu điểm***
  + Đa nền tảng: Hỗ trợ nhiều nền tảng bao gồm Windows, MacOS, Linux, Android, IOS, và WebAssembly.
  + Tiết kiệm thời gian phát triển ứng dụng.
  + Hiệu suất cao: sử dụng công cụ riêng để kết xuất giao diện người dùng.
  + UI nhất quán.
  + Ngôn ngữ lập trình hiện đại: hỗ trợ C# và F#.
  + Tích hợp XAML.
* ***Nhược điểm***
  + Mới mẻ và ít phổ biến.
  + Hỗ trợ hạn chế cho một số nền tảng.
  + Hạn chế về thư viện bên thứ 3.
  + Cộng đồng phát triển nhỏ.

### Tổng quan về mô hình MVVM

MVVM là viết tắt của Model – View – ViewModel, đây là mô hình hỗ trợ two-way data binding giữa View và ViewModel. Cụ thể mô hình MVVM được trình bày như sau:

* ***View***
* Tương tự như trong mô hình MVC, View là phần giao diện của ứng dụng để hiển thị dữ liệu và nhận tương tác của người dùng. Một điểm khác biệt so với các ứng dụng truyền thống là View trong mô hình này tích cực hơn, nó có khả năng thực hiện các hành vi và phản hồi lại người dùng thông qua tính năng binding, command.
* ***Model***
* Cũng tương tự như trong mô hình MVC, Model là các đối tượng giúp truy xuất và thao tác trên dữ liệu thực sự.
* ***View Model***
* Là lớp trung gian giữa View và Model. View Model có thể được xem là thành phần thay thế cho Controller trong mô hình MVC. Nó chứa các mã lệnh thực hiện Data Binding, Command. Một điểm cần lưu ý là trong mô hình MVVM, các tầng bên dưới sẽ không biết được các thông tin gì về các tầng trên của nó.



Hình 1.1. Mô hình MVVM

* ***Ưu điểm***
* Thực hiện Unit testing bây giờ sẽ rất dễ dàng, vì bạn thực sự không phụ thuộc vào view
* MVVM sẽ tạo sự tương tác hiệu quả giữa designer và developer
* Tăng khả năng sử dụng lại các thành phần hay việc thay đổi giao diện chương trình mà không cần phải viết lại code quá nhiều
* Phát triển ứng dụng nhanh, đơn giản, dễ nâng cấp, bảo trì…
* ***Nhược điểm***
* Khả năng duy trì khi view có thể gán cả biến và biểu thức, các logic không liên quan sẽ tăng dần theo thời gian, ảnh hưởng đến việc thêm code vào XML
* Đối với dự án nhỏ việc áp dụng mô hình MVVM gây cồng kềnh, tốn thời gian trong quá trình phát triển. Tốn thời gian trung chuyển dữ liệu của các thành phần
* Đối với dự án lớn hơn, nó gây khó khăn và mất thời gian để thiết kế các ViewModel
* Việc liên kết dữ liệu cho tất cả các thành phần gây khó khăn trong việc debug khi cơ sở dữ liệu phức tạp

# PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

## Phân tích yêu cầu

### Về hệ thống

Ứng dụng đa nền tảng Quản lý điểm cho học sinh trung học phổ thông hướng tới một hệ thống có giao diện dễ nhìn, dễ sử dụng, dễ quản lý và đem lại sự trải nghiệm tuyệt vời nhất cho người sử dụng. Bên cạnh đó, ứng dụng cũng hướng tới sự tối ưu hiệu suất, tiết kiệm thời gian cho người sử dụng.

### Khảo sát

### Yêu cầu về chức năng

* ***Các chức năng chính của hệ thống***
* Đăng nhập
* Xem điểm thành phần
* Xem điểm tổng kết
* Quản lý điểm
* Quản lý học sinh
* Quản lý lớp học
* Quản lý tài khoản
* Quản lý giáo viên
* Quản lý khóa học
* Cài đặt tài khoản
* ***Yêu cầu phi chức năng***
* Giao diện đồ họa dễ hình, dễ thao tác
* Hoạt động của hệ thống ổn định
* Độ tin cậy của ứng dụng
* Hiệu năng của ứng dụng

## Thiết kế hệ thống

### Mô hình hóa Usecase

#### Xác định các tác nhân

* Học sinh: là những người truy cập vào ứng dụng để thực hiện các chức năng như xem điểm thành phần, xem điểm tổng kết, …
* Giáo viên: là những người truy cập vào ứng dụng để thực hiện các chức năng như quản lý điểm, cài đặt tài khoản, …
* Người quản trị: thực hiện các nhiệm vụ quản trị.

#### Xác định các Usecase

* Đăng nhập
* Cài đặt tài khoản
* Xem điểm thành phần
* Xem điểm tổng kết
* Quản lý điểm
* Quản lý lớp học
* Quản lý khóa học
* Quản lý tài khoản
* Quản lý học sinh
* Quản lý giáo viên

#### Biểu đồ Usecase

* ***Biểu đồ Usecase chính***

|  |
| --- |
|  |

Hình 2.1. Biểu đồ Usecase chính

* **Các Usecase phía frontend**

|  |
| --- |
|  |

Hình 2.2. Biểu đồ phân rã Usecase phía Frontend

* **Trong đó**
* Đăng nhập: Cho phép người dùng đăng nhập vào ứng dụng.
* Cài đặt tài khoản: Cho phép người dùng cập nhật thông tin, đổi mật khẩu tài khoản.
* Xem điểm thành phần: Cho phép học sinh theo dõi các đầu điểm của các môn học, theo từng năm học.
* Xem điểm tổng kết: Cho phép học sinh xem điểm tổng kết của tất cả học kỳ, và điểm tổng kết cuối năm theo từng năm học.
* Quản lý điểm: Cho phép giáo viên xem, thêm, sửa bảng điểm của các lớp mà giáo viên đó đang giảng dạy.
* **Các Usecase phía người quản trị**

|  |
| --- |
|  |

Hình 2.3. Biểu đồ phân rã Usecase phía Backend.

* ***Trong đó***
* Quản lý lớp học: Cho phép người quản trị xem, thêm, sửa lớp học.
* Quản lý khóa học: Cho phép người quản trị xem, thêm, sửa khóa học cho từng lớp học.
* Quản lý học sinh: Cho phép người quản trị xem, thêm học sinh.
* Quản lý giáo viên: Cho phép người quản trị xem, thêm giáo viên.
* Quản lý tài khoản: Cho phép người quản trị xem, thêm, khóa, đặt lại mật khẩu tài khoản.
* Quản lý môn học: Cho phép người quản trị xem, thêm, sửa môn học.

### Mô tả chi tiết các Usecase

#### Mô tả Usecase <Đăng nhập>

Bảng 2.1. Mô tả usecase đăng nhập

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mô tả** | | Usecase này cho phép người dùng đăng nhập để xác nhận quyền truy cập hệ thống. | | |
| **Tác nhân** | | Người dùng | | |
| **Sự kiện kích hoạt chức năng** | | Usecase bắt đầu khi người dùng mở ứng dụng. | | |
| **Tiền điều kiện** | | Người dùng sử dụng ứng dụng | | |
| **Luồng sự kiện** | **Luồng sự kiện chính** | **#** | **Thực hiện bởi** | **Hành động** |
| 1 | Hệ thống | Hiển thị form đăng nhập |
| 2 | Người dùng | Nhập thông tin tài khoản. Kích chọn đăng nhập. |
| 3 | Hệ thống | Kiểm tra thông tin. |
| 4 | Hệ thống | Hiển thị màn hình trang chủ. |
| **Luồng rẽ nhánh** | **#** | **Thực hiện bởi** | **Hành động** |
| 3a | Hệ thống | Nếu người dùng nhập sai tên đăng nhập hoặc mật khẩu, hệ thống sẽ hiển thị một thống báo “đăng nhập không thành công” và Usecase kết thúc. |
| 1..4 | Hệ thống | Tại bất kỳ thời điểm nào trong quá trình thực hiện Usecase nếu không kết nối được tới cơ sở dữ liệu, hệ thống sẽ hiển thị thông báo lỗi và Usecase kết thúc. |
| **Hậu điều kiện** | | Không | | |

* Biểu đồ lớp phân tích:

|  |
| --- |
|  |

Hình 2.4. Biểu đồ lớp phân tích use case đăng nhập

* Biểu đồ trình tự:

Hình 2.5. Minh họa trình tự hoạt động của Usecase đăng nhập.

|  |
| --- |
|  |

Hình 2.5. Biểu đồ trình tự Usecase đăng nhập

#### Mô tả Usecase <Cài đặt tài khoản>

Bảng 2.2. Mô tả Usecase cài đặt tài khoản.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mô tả** | | Usecase này cho phép người dùng xem, thay đổi thông tin cá nhân, hoặc đổi mật khẩu. | | |
| **Tác nhân** | | Người dùng | | |
| **Sự kiện kích hoạt chức năng** | | Use case này bắt đầu khi kích vào chức năng “Cài đặt tài khoản” trên thanh menu chính. | | |
| **Tiền điều kiện** | | Người dùng sử dụng ứng dụng | | |
| **Luồng sự kiện** | **Luồng sự kiện chính** | **#** | **Thực hiện bởi** | **Hành động** |
| 1 | Người dùng | Kích chọn “Cài đặt tài khoản. |
| 2 | Hệ thống | Lấy thông tin của tài khoản đang sử dụng trong bảng User như: họ đêm, tên, giới tính, số điện thoại, địa chỉ, ngày sinh, … và hiển thị lên giao diện. |
| 3 | Người dùng | Nhập thông tin cần sửa. Kích nút “Cập nhật”. |
| 4 | Hệ thống | Kiểm tra thông tin và cập nhật bảng User. Usecase kết thúc. |
| 5 | Người dùng | Kích nút “Đổi mật khẩu”. |
| 6 | Hệ thống | Hiển thị form đổi mật khẩu. |
| 7 | Người dùng | Nhập mật khẩu mới. Kích nút “OK”. |
| 8 | Hệ thống | Kiểm tra mật khẩu và cập nhật bảng User. Usecase kết thúc. |
| **Luồng rẽ nhánh** | **#** | **Thực hiện bởi** | **Hành động** |
| 3a | Người dùng | Nếu người dùng không kích nút “Cập nhật” hoặc sử dụng chức năng khác của ứng dụng. Hệ thống sẽ giữ nguyên thông tin và Usecase kết thúc. |
| 8a | Hệ thống | Nếu mật khẩu mới và xác nhận mật khẩu không giống nhau. Hệ thống sẽ hiển thị thông báo “Mật khẩu không giống nhau”. Usecase kết thúc. |
| **Hậu điều kiện** | | Không | | |

* Biểu đồ lớp phân tích

|  |
| --- |
|  |

Hình 2.6. Biểu đồ lớp phân tích Usecase cài đặt tài khoản

* Biểu đồ trình tự

Hình 2.7. Mô tả trình tự hoạt động của Usecase cài đặt tài khoản

|  |
| --- |
|  |

|  |
| --- |
|  |

Hình 2.7. Biểu đồ trình tự Usecase cài đặt tài khoản.

#### Mô tả Usecase <Xem điểm thành phần>

Bảng 2.3. Mô tả Usecase Xem điểm thành phần

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mô tả** | | Usecase này cho phép học sinh xem điểm thành phần của các môn học | | |
| **Tác nhân** | | Học sinh | | |
| **Sự kiện kích hoạt chức năng** | | Usecase bắt đầu khi học sinh bấm vào “Xem điểm thành phần” trên thanh menu chính. | | |
| **Tiền điều kiện** | | Học sinh sử dụng ứng dụng | | |
| **Luồng sự kiện** | **Luồng sự kiện chính** | **#** | **Thực hiện bởi** | **Hành động** |
| 1 | Học sinh | Kích chọn “Xem điểm thành phần”. |
| 2 | Hệ thống | Hệ thống hiển thị danh sách năm học tới thời điểm hiện tại. Năm bắt đầu được lấy từ bảng User. |
| 3 | Học sinh | Kích chọn năm học hiện tại. |
| 4 | Hệ thống | Hiển thị danh sách bảng điểm lên màn hình gồm những thông tin: tên môn học lấy từ bảng Subject, học kỳ lấy từ bảng Course, điểm thường xuyên 1, điểm thường xuyên 2, điểm thường xuyên 3, điểm thường xuyên 4, điểm giữa kỳ, điểm cuối kỳ từ bảng GradeSheet. Usecase kết thúc |
| **Luồng rẽ nhánh** | **#** | **Thực hiện bởi** | **Hành động** |
| 1..4 | Hệ thống | Tại bất kỳ thời điểm nào trong quá trình thực hiện Usecase nếu không kết nối được tới cơ sở dữ liệu, hệ thống sẽ hiển thị thông báo lỗi và Usecase kết thúc. |
| **Hậu điều kiện** | | Không | | |

* Biểu đồ lớp phân tích

|  |
| --- |
|  |

Hình 2.8. Biểu đồ lớp phân tích Usecase xem điểm thành phần

* Biểu đồ trình tự

Hình 2.9 mô tả trình tự hoạt động của Usecase xem điểm thành phần.

|  |
| --- |
|  |

Hình 2.9. Biểu đồ trình tự Usecase xem điểm thành phần

#### Mô tả Usecase <Xem điểm tổng kết>

Bảng 2.4. Mô tả Usecase Xem điểm tổng kết

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mô tả** | | Usecase này cho phép học sinh xem điểm tổng kết của từng học kỳ và điểm tổng kết cuối năm. | | |
| **Tác nhân** | | Học sinh | | |
| **Sự kiện kích hoạt chức năng** | | Usecase bắt đầu khi người dùng bấm vào “Xem điểm tổng kết” trên thanh menu chính. | | |
| **Tiền điều kiện** | | Học sinh sử dụng ứng dụng | | |
| **Luồng sự kiện** | **Luồng sự kiện chính** | **#** | **Thực hiện bởi** | **Hành động** |
| 1 | Học sinh | Kích chọn “Xem điểm tổng kết”. |
| 2 | Hệ thống | Hệ thống sẽ hiển thị danh sách các năm học của học sinh đó tính tới thời điểm hiện tại. Năm bắt đầu được lấy từ bảng User. |
| 3 | Học sinh | Kích chọn năm học hiện tại. |
| 4 | Hệ thống | Hệ thống sẽ lấy thông tin mã lớp học từ bảng StudentAssignment. Hệ thống sẽ lấy tất cả khóa học của lớp học trong năm đó khi đó sẽ lấy điểm trung bình từ bảng GradeSheet theo từng khóa học và tính điểm trung bình cả học kỳ. Usecase kết thúc. |
| **Luồng rẽ nhánh** | **#** | **Thực hiện bởi** | **Hành động** |
| 1..4 | Hệ thống | Tại bất kỳ thời điểm nào trong quá trình thực hiện Usecase nếu không kết nối được tới cơ sở dữ liệu, hệ thống sẽ hiển thị thông báo lỗi và Usecase kết thúc. |
| **Hậu điều kiện** | | Không | | |

* Biểu đồ lớp phân tích

|  |
| --- |
|  |

Hình 2.10. Biểu đồ lớp phân tích Usecase xem điểm tổng kết

* Biểu đồ trình tự

Hình 2.11 mô tả trình tự hoạt động của Usecase xem điểm tổng kết

|  |
| --- |
|  |

Hình 2.11. Biểu đồ trình tự usecase Xem điểm tổng kết

#### Mô tả Use case <Quản lý tài khoản>

Bảng 2.5. Mô tả Usecase Quản lý tài khoản

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mô tả** | | Usecase này cho phép người quản trị quản lý và xem thông tin tài khoản người dùng. | | |
| **Tác nhân** | | Người quản trị | | |
| **Sự kiện kích hoạt chức năng** | | Usecase bắt đầu khi người quản trị bấm vào “Quản lý tài khoản” trên thanh menu chính. | | |
| **Tiền điều kiện** | | Người quản trị phải đăng nhập với tài khoản quản trị của hệ thống. | | |
| **Luồng sự kiện** | **Luồng sự kiện chính** | **#** | **Thực hiện bởi** | **Hành động** |
| 1 | Người quản trị | Kích chọn “Quản lý tài khoản”. |
| 2 | Hệ thống | Hiển thị danh sách các tài khoản gồm các thông tin: Họ và tên, giới tính, ngày sinh, số điện thoại, địa chỉ, hình ảnh từ bảng User lên màn hình. |
| 3 | Người quản trị | Kích chọn nút “Thêm mới”. |
| 4 | Hệ thống | Hiển thị form thêm mới tài khoản. |
| 5 | Người quản trị | Nhập thông tin tài khoản, kích nút “OK”. |
| 6 | Hệ thống | Kiểm tra thông tin và thêm mới bản ghi vào bảng User. Usecase kết thúc. |
| 7 | Người quản trị | Chọn tài khoản bất kỳ. Kích vào nút “Khóa tài khoản”. |
| 8 | Hệ thống | Thực hiện khóa tài khoản, cập nhật lại bảng User sau đó hiển thị lại danh sách tài khoản. Usecase kết thúc. |
| **Luồng rẽ nhánh** | **#** | **Thực hiện bởi** | **Hành động** |
| 1..8 | Hệ thống | Tại bất kỳ thời điểm nào trong quá trình thực hiện Usecase nếu không kết nối được tới cơ sở dữ liệu, hệ thống sẽ hiển thị thông báo lỗi và Usecase kết thúc. |
| **Hậu điều kiện** | | Không | | |

* Biểu đồ lớp phân tích

|  |
| --- |
|  |

Hình 2.12. Biểu đồ lớp phân tích Usecase quản lý tài khoản

* Biểu đồ trình tự

Hình 2.3 mô tả trình tự hoạt động của Usecase quản lý tài khoản

|  |
| --- |
|  |

Hình 2.13. Biểu đồ trình tự Usecase quản lý tài khoản

#### Mô tả Use case <Quản lý điểm>

Bảng 2.6. Mô tả Usecase Quản lý điểm

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mô tả** | | Usecase này cho phép giáo viên quản lý và xem bảng điểm của từng học sinh | | |
| **Tác nhân** | | Giáo viên | | |
| **Sự kiện kích hoạt chức năng** | | Usecase bắt đầu khi giáo viên bấm vào “Quản lý điểm” trên thanh menu chính. | | |
| **Tiền điều kiện** | | Giáo viên phải đăng nhập với tài khoản giáo viên của hệ thống. | | |
| **Luồng sự kiện** | **Luồng sự kiện chính** | **#** | **Thực hiện bởi** | **Hành động** |
| 1 | Giáo viên | Kích chọn “Quản lý điểm”. |
| 2 | Hệ thống | Hiển thị danh sách các năm giảng dạy của giáo viên tính tới thời điểm hiện tại. Năm bắt đầu giảng dạy của giáo viên được lấy từ bảng User. |
| 3 | Giáo viên | Kích chọn năm học, học kỳ hiện tại. |
| 4 | Hệ thống | Lấy toàn bộ lớp học mà giáo viên đang giảng dạy trong bảng Course và hiển thị lên màn hình. |
| 5 | Giáo viên | Kích chọn lớp học. |
| 6 | Hệ thống | Lấy danh sách bảng điểm gồm những thông tin: mã học sinh từ bảng Student, họ và tên từ bảng User, điểm thường xuyên một, điểm thường xuyên hai, điểm thường xuyên ba, điểm thường xuyên bốn, điểm giữa kỳ, điểm cuối kỳ từ bảng GradeSheet hiển thị lên màn hình. |
| 7 | Giáo viên | Chọn bảng điểm, kích nút “Cập nhật”. |
| 8 | Hệ thống | Hiển thị form cập nhật điểm. |
|  | 9 | Giáo viên | Nhập thông tin, sau đó kích nút “OK”. |
| 10 | Hệ thống | Kiểm tra thông tin và cập nhật lại bảng GradeSheet, sau đó hiển thị lại danh sách bảng điểm. Usecase kết thúc. |
| 11 | Giáo viên | Chọn bảng điểm, kích chọn “Yêu cầu sửa điểm”. |
| 12 | Hệ thống | Hiển thị form đơn sửa điểm. |
| 13 | Giáo viên | Nhập thông tin đơn sửa điểm, kích nút “OK”. |
| 14 | Hệ thống | Kiểm tra thông tin sau đó thêm mới đơn sửa điểm vào bảng EditGradeSheetForm. Usecase kết thúc. |
| 15 | Giáo viên | Chọn “File” sau đó kích chọn “Thêm mới. |
| 16 | Hệ thống | Hiển thị form nhập gồm danh sách lớp được lấy từ bảng Class, danh sách học sinh từ bảng StudentAssignment, điểm thường xuyên một, điểm thường xuyên hai, điểm thường xuyên ba, điểm thường xuyên bốn, điểm giữa kỳ, điểm cuối kỳ từ bảng GradeSheet. Usecase kết thúc. |
| 17 | Giáo viên | Chọn lớp học. |
| 18 | Hệ thống | Lấy danh sách học sinh của lớp học đã chọn và hiển thị ra màn hình. |
| 19 | Giáo viên | Nhập thông tin bảng điểm. Kích chọn nut “OK”. |
| 20 | Hệ thống | Kiểm tra thông tin và thêm mới bản ghi vào bảng GradeSheet. |
| 21 | Giáo viên | Chọn lớp học và kích chọn nút “Hoàn thành”. |
| 22 | Hệ thống | Thực hiện khóa bảng điểm của lớp được chọn. Cập nhật lại bảng GradeSheet. Usecase kết thúc. |
| 23 | Giáo viên | Chọn “File” sau đó kích chọn “Download file .xlsx”. |
| 24 | Hệ thống | Hiển thị dialog cho giáo viện chọn nơi tải xuống. |
| 25 | Giáo viên | Chọn nơi tải xuống và kích nút “OK”. |
| 26 | Hệ thống | Xuất danh sách lớp được chọn ra file excel. Usecase kết thúc. |
| **Luồng rẽ nhánh** | **#** | **Thực hiện bởi** | **Hành động** |
| 1..26 | Hệ thống | Tại bất kỳ thời điểm nào trong quá trình thực hiện Usecase nếu không kết nối được tới cơ sở dữ liệu, hệ thống sẽ hiển thị thông báo lỗi và Usecase kết thúc. |
| **Hậu điều kiện** | | Không | | |

* Biểu đồ lớp phân tích

|  |
| --- |
|  |

Hình 2.14. Biểu đồ lớp phân tích Usecase quản lý điểm

* Biểu đồ trình tự

Hình 2.15 mô tả trình tự hoạt động của Usecase quản lý điểm.

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |

Hình 2.15. Biểu đồ trình tự usecase quản lý điểm

#### Mô tả Usecase <Quản lý lớp học>

Bảng 2.7. Mô tả Usecase Quản lý lớp học

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mô tả** | | Usecase này cho phép người quản trị xem, thêm, sửa lớp học. | | |
| **Tác nhân** | | Người quản trị. | | |
| **Sự kiện kích hoạt chức năng** | | Usecase bắt đầu khi người quản trị bấm vào “Quản lý lớp học” trên thanh menu chính. | | |
| **Tiền điều kiện** | | Người quản trị phải đăng nhập với tài khoản người quản trị của hệ thống. | | |
| **Luồng sự kiện** | **Luồng sự kiện chính** | **#** | **Thực hiện bởi** | **Hành động** |
| 1 | Người quản trị | Kích chọn “Quản lý lớp học”. |
| 2 | Hệ thống | Lấy danh sách các lớp từ bản Class gồm mã lớp, tên lớp, số lượng học sinh để hiển thị lên màn hình. |
| 3 | Người quản trị | Kích chọn nút “Thêm mới”. |
| 4 | Hệ thống | Hiển thị form thêm mới lớp học. |
| 5 | Người quản trị | Nhập thông tin lớp học sau đó kích nút “OK”. |
| 6 | Hệ thống | Kiểm tra thông tin và thêm mới bản ghi vào bảng Class. Usecase kết thúc. |
| 7 | Người quản trị | Chọn lớp học bất kỳ. Kích vào nút “Cập nhật”. |
| 8 | Hệ thống | Hiển thị form cập nhật lớp học. |
| 9 | Người quản trị | Nhập thông tin vào form sau đó kích nút “OK”. |
| 10 | Hệ thống | Kiểm tra thông tin và cập nhật thông tin vào bảng Class. |
|  | **Luồng rẽ nhánh** | **#** | **Thực hiện bởi** | **Hành động** |
| 1..10 | Hệ thống | Tại bất kỳ thời điểm nào trong quá trình thực hiện Usecase nếu không kết nối được tới cơ sở dữ liệu, hệ thống sẽ hiển thị thông báo lỗi và Usecase kết thúc. |
| **Hậu điều kiện** | | Không | | |

* Biểu đồ lớp phân tích

|  |
| --- |
|  |

Hình 2.16. Biểu đồ lớp phân tích Usecase quản lý lớp học

* Biểu đồ trình tự

Hình 2.17. Mô tả trình tự hoạt động của Usecase quản lý lớp học.

|  |
| --- |
|  |

Hình 2.17. Biểu đồ trình tự Usecase quản lý lớp học

#### Mô tả Usecase <Quản lý khóa học>

Bảng 2.8. Mô tả Usecase Quản lý khóa học.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mô tả** | | Usecase này cho phép người quản trị xem, thêm, sửa khóa học. | | |
| **Tác nhân** | | Người quản trị. | | |
| **Sự kiện kích hoạt chức năng** | | Usecase bắt đầu khi người quản trị bấm vào “Quản lý khóa học” trên thanh menu chính. | | |
| **Tiền điều kiện** | | Người quản trị phải đăng nhập với tài khoản người quản trị của hệ thống. | | |
| **Luồng sự kiện** | **Luồng sự kiện chính** | **#** | **Luồng rẽ nhánh** | **Hành động** |
| 1 | Người quản trị | Kích chọn “Quản lý khóa học”. |
| 2 | Hệ thống | Hiển thị danh sách lớp học từ bảng Class. |
| 3 | Người dùng | Chọn năm học, lớp học, và kỳ học. |
| 4 | Hệ thống | Lấy thông tin các khóa học của lớp học có năm học và học kỳ đã chọn từ bảng Course. |
| 5 | Hệ thống | Lấy thông tin môn học từ bảng Subject, tên giáo viên từ bảng User, tên chương trình đào tạo từ bảng EducationProgram, thời gian bắt đầu, thời gian kết thúc, học kỳ, trạng thái từ bảng Course để hiển thị lên màn hình. Usecase kết thúc. |
| 6 | Người quản trị | Kích chọn nút “Thêm mới”. |
| 7 | Hệ thống | Hiển thị một form nhập thông tin gồm: lớp học được lấy từ bảng Class, học kỳ, môn học được lấy từ bảng Subject, giáo viên giảng dạy lấy từ bảng Teacher, chương trình học lấy từ bảng EducationProgram, ngày bắt đầu, ngày kết thúc. |
| 8 | Người quản trị | Nhập thông tin khóa học (khi chọn lớp học, hệ thống sẽ hiển thị danh sách môn học mà lớp đó chưa có, sau đó chọn môn học thì hệ thống sẽ hiển thị các giáo viên có chuyên môn về môn đã chọn) sau đó kích nút “OK”. |
| 9 | Hệ thống | Kiểm tra thông tin và thêm mới bản ghi vào bảng Course. Usecase kết thúc. |
| 10 | Người quản trị | Chọn khóa học bất kỳ. Kích vào nút “Cập nhật”. |
|  | 11 |  | Hiển thị form cập nhật khóa học. |
|  | 12 |  | Nhập thông tin vào form sau đó kích nút “OK”. |
|  | 13 |  | Kiểm tra thông tin và cập nhật thông tin vào bảng Course. |
| **Luồng rẽ nhánh** | **#** | **Luồng rẽ nhánh** | **Hành động** |
| 1..13 | Hệ thống | Tại bất kỳ thời điểm nào trong quá trình thực hiện Usecase nếu không kết nối được tới cơ sở dữ liệu, hệ thống sẽ hiển thị thông báo lỗi và Usecase kết thúc. |
| **Hậu điều kiện** | | Không | | |

* Biểu đồ lớp phân tích

|  |
| --- |
|  |

Hình 2.18. Biểu đồ lớp phân tích usecase quản lý khóa học.

* Biểu đồ trình tự

Hình 2.19 mô tả trình tự hoạt động của Usecase quản lý khóa học.

|  |
| --- |
|  |
|  |

Hình 2.19. Biểu đồ trình tự usecase Quản lý khóa học.

#### Mô tả Usecase <Quản lý học sinh>

Bảng 2.9. Mô tả Usecase Quản lý học sinh

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mô tả** | | Usecase này cho phép người quản trị xem, thêm học sinh. | | |
| **Tác nhân** | | Người quản trị. | | |
| **Sự kiện kích hoạt chức năng** | | Usecase bắt đầu khi người quản trị bấm vào “Quản lý học sinh” trên thanh menu chính. | | |
| **Tiền điều kiện** | | Người quản trị phải đăng nhập với tài khoản người quản trị của hệ thống. | | |
| **Luồng sự kiện** | **Luồng sự kiện chính** | **#** | **Thực hiện bởi** | **Hành động** |
| 1 | Người quản trị | Kích chọn “Quản lý học sinh”. |
| 2 | Hệ thống | Hiển thị danh sách học sinh gồm họ và tên, giới tính, ngày sinh, địa chỉ từ bảng User, mã học sinh từ bảng Student. |
| 3 | Người quản trị | Kích chọn nút “Thêm mới”. |
| 4 | Hệ thống | Hiển thị form thêm mới học sinh. |
| 5 | Người quản trị | Nhập thông tin học sinh sau đó kích nút “OK”. |
| 6 | Hệ thống | Kiểm tra thông tin và thêm mới bản ghi vào bảng Student. Usecase kết thúc. |
| **Luồng rẽ nhánh** | **#** | **Thực hiện bởi** | **Hành động** |
| 1..6 | Hệ thống | Tại bất kỳ thời điểm nào trong quá trình thực hiện Usecase nếu không kết nối được tới cơ sở dữ liệu, hệ thống sẽ hiển thị thông báo lỗi và Usecase kết thúc. |
| **Hậu điều kiện** | | Không | | |

* Biểu đồ lớp phân tích

|  |
| --- |
|  |

Hình 2.20. Biểu đồ lớp phân tích usecase Quản lý học sinh.

* Biểu đồ trình tự

Hình 2.21 mô tả trình tự hoạt động của Usecase quản lý học sinh.

|  |
| --- |
|  |

Hình 2.21. Biểu đồ trình tự usecase Quản lý học sinh

#### Mô tả Usecase <Quản lý giáo viên>

Bảng 2.10. Mô tả Usecase Quản lý giáo viên

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mô tả** | | Usecase này cho phép người quản trị xem, thêm giáo viên. | | |
| **Tác nhân** | | Người quản trị. | | |
| **Sự kiện kích hoạt chức năng** | | Usecase bắt đầu khi người quản trị bấm vào “Quản lý giáo viên” trên thanh menu chính. | | |
| **Tiền điều kiện** | | Người quản trị phải đăng nhập với tài khoản người quản trị của hệ thống. | | |
| **Luồng sự kiện** | **Luồng sự kiện chính** | **#** | **Thực hiện bởi** | **Hành động** |
| 1 | Người quản trị | Kích chọn “Quản lý giáo viên”. |
| 2 | Hệ thống | Hiển thị danh sách giáo viên gồm họ và tên, giới tính, ngày sinh, địa chỉ từ bảng User, mã giáo viên, tên phòng ban, trường đại học từng học, bằng cấp, chuyên môn,… từ bảng Teacher. |
| 3 | Người quản trị | Kích chọn nút “Thêm mới”. |
| 4 | Hệ thống | Hiển thị form thêm mới giáo viên gồm toàn bộ tài khoản là teacher từ bảng User và chưa giáo viên nào sử dụng, danh sách phòng ban gồm tên phòng ban từ bảng Department, và danh sách môn học gồm tên môn học từ bảng Subject. |
| 5 | Người quản trị | Nhập thông tin giáo viên sau đó kích nút “OK”. |
| 6 | Hệ thống | Kiểm tra thông tin và thêm mới bản ghi vào bảng Teacher. Usecase kết thúc. |
| **Luồng rẽ nhánh** | **#** | **Thực hiện bởi** | **Hành động** |
| 1..6 | Hệ thống | Tại bất kỳ thời điểm nào trong quá trình thực hiện Usecase nếu không kết nối được tới cơ sở dữ liệu, hệ thống sẽ hiển thị thông báo lỗi và Usecase kết thúc. |
| **Hậu điều kiện** | | Không | | |

* Biểu đồ lớp phân tích

|  |
| --- |
|  |

Hình 2.22. Biểu đồ lớp phân tích usecase quản lý giáo viên.

* Biểu đồ trình tự

Hình 2.23. Mô tả trình tự hoạt động của Usecase quản lý giáo viên.

|  |
| --- |
|  |

Hình 2.23. Biểu đồ trình tự usecase Quản lý giáo viên

### Biểu đồ lớp hệ thống

|  |
| --- |
|  |

Hình 2.24. Biểu đồ lớp hệ thống

## Thiết kế cơ sở dữ liệu

### Thiết kế cơ sở dữ liệu mức vật lý

Hình 2.25 trình bày thiết kế cơ sở dữ liệu mức vật lý cho ứng dụng quản lý điểm.

|  |
| --- |
|  |

Hình 2.25. Bảng thiết kế cơ sở dữ liệu mức vật lý.

Hình 2.26 trình bày biểu đồ diagram của hệ thống.

|  |
| --- |
|  |

Hình 2.26. Biểu đồ diagram của hệ thống.

### Thiết kế cơ sở dữ liệu mức logic

Hình 2.27 minh họa cơ sở dữ liệu được thiết kế ở mức logic.

|  |
| --- |
|  |

Hình 2.27. Bảng thiết kế cơ sở dữ liệu mức logic.

## Thiết kế giao diện

### Giao diện đăng nhập

|  |
| --- |
|  |

Hình 2.28. Giao diện đăng nhập

### Giao diện cài đặt tài khoản

|  |
| --- |
|  |

Hình 2.29. Giao diện đăng nhập tài khoản.

### Giao diện xem điểm thành phần

|  |
| --- |
|  |

Hình 2.30. Giao diện xem điểm thành phần

### Giao diện xem điểm tổng kết

|  |
| --- |
|  |

Hình 2.31. Giao diện xem điểm tổng kết

### Giao diện quản lý tài khoản

|  |
| --- |
|  |

Hình 2.32. Giao diện quản lý tài khoản

### Giao diện quản lý điểm

|  |
| --- |
|  |

Hình 2.33. Giao diện quản lý điểm.

### Giao diện quản lý lớp học

|  |
| --- |
|  |

Hình 2.34. Giao diện quản lý lớp học.

### Giao diện quản lý khóa học

|  |
| --- |
|  |

Hình 2.35. Giao diện quản lý khóa học.

### Giao diện quản lý học sinh

|  |
| --- |
|  |

Hình 2.36. Giao diện quản lý học sinh.

### Giao diện quản lý giáo viên

|  |
| --- |
|  |

Hình 2.37. Giao diện quản lý giáo viên.

# KẾT QUẢ

## Cài đặt môi trường

* Cài đặt VS Code
* Bước 1: Truy cập vào [link https://visualstudio.microsoft.com/vs/](link%20https:/visualstudio.microsoft.com/vs/%20) để tải xuống phiên bản Microsoft Visual Studio 2022
* Bước 2: Ấn nút download, chọn Community 2022 để thực hiện tải và cài đặt phiên bản Microsoft Visual Studio 2022

|  |
| --- |
|  |

Hình 3.1. Microsoft Visual Studio 2022

* Cài đặt Avalonia
  + Mở Microsoft Visual Studio 2022.
  + Chọn Tools > NutGet Package Manager > Package Manager Console
  + Nhập command “dotnet new install Avalonia.Templates” và nhấn enter.
  + Chọn Extensions > Manager Extensions, sau đó nhập từ khóa Avalonia.

|  |
| --- |
|  |

Hình 3.2. Cài đặt Avalonia for Visual Studio 2022.

## Kết quả thu được

### Cài đặt tài khoản

* Nền tảng Windows

|  |
| --- |
|  |

Hình 3.3. Màn hình cài đặt tài khoản trên nền tảng Windows.

* Nền tảng Android, IOS

|  |
| --- |
|  |

Hình 3.4. Màn hình cài đặt tài khoản trên nền tảng Android, IOS.

### Xem điểm thành phần

* Nền tảng Windows

|  |
| --- |
|  |

Hình 3.5. Màn hình xem điểm thành phần trên nền tảng Windows.

* Nền tảng Android, IOS

|  |
| --- |
|  |

Hình 3.6. Màn hình xem điểm thành phần trên nền tảng Android, IOS

### Xem điểm tổng kết

* Nền tảng Windows

|  |
| --- |
|  |

Hình 3.7. Màn hình xem điểm tổng kết trên nền tảng Windows.

* Nền tảng Android, IOS

|  |
| --- |
|  |

Hình 3.8. Màn hình xem điểm tổng kết trên nền tảng Android, IOS.

### Quản lý tài khoản

* Nền tảng Windows

|  |
| --- |
|  |

Hình 3.9. Màn hình quản lý tài khoản trên nền tảng Windows.

* Nền tảng Android, IOS

|  |
| --- |
|  |

Hình 3.10. Màn hình quản lý tài khoản trên nền tảng Android, IOS.

### Quản lý điểm

* Nền tảng Windows

|  |
| --- |
|  |

Hình 3.11. Màn hình quản lý điểm trên nền tảng Windows.

* Nền tảng Android, IOS

|  |
| --- |
|  |

Hình 3.12. Màn hình quản lý điểm trên nền tảng Android, IOS.

### Quản lý lớp học

* Nền tảng Windows

|  |
| --- |
|  |

Hình 3.13. Màn hình quản lý lớp học trên nền tảng Windows.

* Nền tảng Android, IOS

|  |
| --- |
|  |

Hình 3.14. Màn hình quản lý lớp học trên nền tảng Android, IOS.

### Quản lý khóa học

* Nền tảng Windows

|  |
| --- |
|  |

Hình 3.15. Màn hình quản lý khóa học trên nền tảng Windows.

* Nền tảng Android, IOS

|  |
| --- |
|  |

Hình 3.16. Màn hình quản lý khóa học trên nền tảng Android, IOS.

### Quản lý học sinh

* Nền tảng Windows

|  |
| --- |
|  |

Hình 3.17. Màn hình quản lý học sinh trên nền tảng Windows.

* Nền tảng Android, IOS

|  |
| --- |
|  |

Hình 3.18. Màn hình quản lý học sinh trên nền tảng Android, IOS.

### Quản lý giáo viên

* Nền tảng Windows.

|  |
| --- |
|  |

Hình 3.19. Màn hình quản lý giáo viên trên nền tảng Windows.

* Nền tảng Android, IOS

|  |
| --- |
|  |

Hình 3.20. Màn hình quản lý giáo viên trên nền tảng Android, IOS.

## Kết quả kiểm thử

### Kỹ thuật kiểm thử hộp đen (Black box testing)

Kiểm thử hộp đen (Black box testing) là một phương pháp kiểm thử phần mềm mà việc kiểm tra các chức năng của một ứng dụng không cần quan tâm vào cấu trúc nội bộ hoặc hoạt động của nó. Mục đích chính của kiểm thử hộp đen chỉ là để xem phần mềm có hoạt động như dự kiến trong tài liệu yêu cầu và liệu nó có đáp ứng được sự mong đợi của người dùng hay không.

* Các loại kiểm thử hộp đen
* Kiểm thử chức năng (Functional Testing): Là loại kiểm thử có liên quan tới chức năng của toàn bộ hệ thống bên trong sản phẩm, phần mềm. Trong quá trình kiểm thử với Functional Testing chức năng sẽ được kiểm tra kỹ càng trong luồng dữ liệu đầu vào, sau đó đánh giá kết quả đầu ra mà không lo bị ảnh hưởng tới cấu trúc bên trong ứng dụng.
* Kiểm thử phi năng (Non - Functional Testing): Là loại kiểm thử nhằm mục đích kiểm thử các vấn đề về hiệu suất, độ tin cậy và mức độ có thể sử dụng trong 1 sản phẩm phần mềm, ứng dụng bất kỳ.
* Kiểm thử hồi quy (Regression testing): Là kiểm thử hồi quy thường được sử dụng khi đã xác định được lỗi, đã nâng cấp và bảo trì hệ thống thành công.
* Kiểm thử hộp đen có 8 kỹ thuật phổ biến và có nhiều ưu điểm nhất:
* Phân vùng tương đương (equivalence partitioning)
* Phân tích giá trị biên (boundary value analysis)
* Bảng quyết định (decision tables)
* Kiểm thử chuyển đổi trạng thái (state transition testing).
* Kỹ thuật kiểm thử các bộ n thần kỳ (Pairwise)
* Kỹ thuật phân tích vùng miền (Domain analysis)
* Kỹ thuật dựa trên đặc tả Use case (Use case)
* Kỹ thuật dùng lược đồ quan hệ nhân quả (Cause- Effect Diagram)

### Kiểm thử chức năng

Phần mềm được thực hiện kiểm thử chức năng (Functional Testing). Kiểm thử chức năng là một loại kiểm thử hộp đen (Black box testing) và test case của nó được dựa trên đặc tả của ứng dụng phần mềm/thành phần đang test. Các chức năng được test bằng cách nhập vào các giá trị và kiểm tra kết quả đầu ra, ít quan tâm đến cấu trúc bên trong của ứng dụng.

Bảng 3.1. Bảng kiểm thử chức năng

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Chức năng | Trường hợp kiểm thử | Bước thực hiện | Dữ liệu kiểm thử | Đầu ra thực tế | Đầu ra mong muốn | Kết quả |
| TQ1 | Xem thông tin tổng quan | Xem thông tin tổng quan | 1. Bấm vào nút “Tổng quan” trên màn hình |  | Hệ thống hiển thị các thông tin tổng quan về Nguồn tiền, báo cáo chi tiêu và các giao dịch gần đây | Hệ thống hiển thị các thông tin tổng quan về Nguồn tiền, báo cáo chi tiêu và các giao dịch gần đây | Pass |
| SK1 | Quản lý sự kiện | Thêm sự kiện | 1. bấm vào nút thêm sự kiện (Dấu cộng màu xanh lục) trên màn hình “Quản lý sự kiện”  2. Điền thông tin về tên sự kiện, số tiền, chọn ngày, chọn giờ  3. Nhấn vào “Lưu” bên góc phải trên của màn hình | Tên sự kiện: Ăn uống  Số tiền: 100.000đ  Ghi chú: Tiền ăn  Chọn ngày: 07/05/2023  Chọn thời gian: 07:15 | Màn hình hiển thị “Thêm mới sự kiện thành công” | Màn hình hiển thị “Thêm mới sự kiện thành công” | Pass |
| SK2 | Xem danh sách sự kiện | 1. bấm vào nút “Sự kiện” trên màn hình | Thông tin sự kiện vừa thêm ở trên | Màn hình hiển thị các sự kiện | Màn hình hiển thị các sự kiện | Pass |
| SK3 | Xem chi tiết sự kiện | 1. Ở màn hình danh sách sự kiện, khi người dùng bấm vào một sự kiện, hệ thống hiển thị màn hình “Chi tiết sự kiện” |  | Hiển thị thông tiên chi tiết của sự kiện | Hiển thị thông tiên chi tiết của sự kiện | Pass |
| SK4 | Sửa sự kiện | 1. Trên màn hình “Chi tiết sự kiện”, người dùng nhập thông tin cần thay đổi và nhấn “Sửa” |  | Màn hình hiển thị “Sửa sự kiện thành công” | Màn hình hiển thị “Sửa sự kiện thành công” | Pass |
| GD1 | Quản lý giao dịch | Thêm giao dịch | 1. Bấm vào nút thêm giao dịch (Dấu cộng màu xanh lam)  2. Điền thông tin về số tiền, nhập ghi chú, chọn danh mục, thời gian, nguồn tiền, sự kiện, hình ảnh  3. Nhấn vào “Lưu” bên góc phải trên của màn hình | Số tiền: 100.000đ  Ghi chú: Tiền ăn  Chọn danh mục: Ăn uống  Chọn thời gian: 07/05/2023 – 07:15  Chọn nguồn tiền: Tiền mặt  Chọn sự kiện: Ăn uống | Màn hình hiển thị “Thêm giao dịch thành công” | Màn hình hiển thị “Thêm giao dịch thành công” | Pass |
| GD2 | Xem giao dịch | 1. Bấm vào “Lịch sử” | Xem được thông tin giao dịch vừa thêm ở trên | Màn hình hiển thị thông tin các cuộc giao dịch trong tháng này | Màn hình hiển thị thông tin các cuộc giao dịch trong tháng này | Pass |
| GD3 | Sửa giao dịch | 1. Trên màn hình “Chi tiết giao dịch”, người dùng nhập thông tin cần thay đổi và nhấn “Sửa”.  2. Sửa thông tin  3. Trên màn hình xác bấm nhấn vào nút “Sửa” | Sửa số tiền thành: 150.000đ | Màn hình hiển thị “Sửa giao dịch thành công” | Màn hình hiển thị “Sửa giao dịch thành công” | Pass |
| GD4 | Xoá giao dịch | 1. Trong màn hình hiển thị các giao dịch, kéo giao dịch sang trái sẽ hiển thị nút “Xóa”.  2. Nhấn vào “Xoá”  3. Tại màn hình xác nhận nhấn nút “Xoá” | Xoá giao dịch vừa thêm ở trên | Màn hình hiển thị “Xoá giao dịch thành công” | Màn hình hiển thị “Xoá giao dịch thành công” | Pass |
| GD5 | Tìm kiếm giao dịch | 1. Trên màn hình quản lý giao dịch, người dùng nhấn vào thanh tìm kiếm và nhập tên loại giao dịch | Thông tin tìm kiếm: Ăn uống | Màn hình hiển thị thông tin giao dịch “Ăn uống” | Màn hình hiển thị thông tin giao dịch “Ăn uống” | Pass |
| NT1 | Quản lý nguồn tiền | Xem danh sách các nguồn tiền | 1. bấm vào “Xem tất cả” trong mục “Ví của tôi” trên màn hình tổng quan |  | Màn hình hiển thị danh sách các nguồn tiền. | Màn hình hiển thị danh sách các nguồn tiền. | Pass |
| NT2 | Thêm nguồn tiền | 1. bấm vào nút thêm giao dịch (Dấu cộng) trên màn hình “Quản lý nguồn tiền”  2. Nhập thông tin nguồn tiền  3. Bấm “Lưu” | Tên ví: Momo  Số tiền: 1.000.000đ  Chọn loại ví: Ví cơ bản | Màn hình hiển thị “Thêm mới Ví tiền thành công” | Màn hình hiển thị “Thêm mới Ví tiền thành công” | Pass |
| NT3 | Xem danh sách giao dịch sử dụng nguồn tiền | 1. Trên màn hình hiển thị danh sách các nguồn tiền, khi người dùng bấm vào một nguồn tiền |  | Màn hình hiển thị ra các giao dịch sử dụng nguồn tiền đó | Màn hình hiển thị ra các giao dịch sử dụng nguồn tiền đó | Pass |
| TB1 | Xem thông báo | Xem thông báo | 1. bấm vào nút “Thông báo” trên màn hình |  | Màn hình hiển thị danh sách thông báo. | Màn hình hiển thị danh sách thông báo. | Pass |

KẾT LUẬN

* **Đánh giá kết quả**

Dựa trên những thông tin em cung cấp, đánh giá kết quả đề tài "Ứng dụng đa nền tảng quản lý điểm cho học sinh trung học phổ thông" như sau:

* Phần hoàn thành:
* Đã nắm bắt và áp dụng thành công quy trình để xây dựng một ứng dụng đa nền tảng, đồng thời sử dụng thành thạo trình soạn thảo Microsoft Visual Studio.
* Hoàn thiện bài khảo sát và phân tích thiết kế hệ thống " Quản lý điểm cho học sinh trung học phổ thông ".
* Phát triển thành thạo với ngôn ngữ C# và Framework Avalonia.
* Phần chưa hoàn thành:
* Tuy nhiên, cần hoàn thiện thêm giao diện để khách hàng dễ sử dụng. Điều này rất quan trọng vì giao diện đóng vai trò quan trọng trong trải nghiệm người dùng và ảnh hưởng đến sự thành công của ứng dụng. Em có thể đưa thêm các tính năng và cải thiện trải nghiệm người dùng để tăng tính hữu dụng của ứng dụng.

Tóm lại, em đã hoàn thành một phần lớn của đề tài "Ứng dụng đa nền tảng quản lý điểm cho học sinh trung học phổ thông", đồng thời đã có kiến thức và kinh nghiệm để phát triển các ứng dụng đa nền tảng. Tuy nhiên, để đạt được sự thành công về mặt thương mại, em cần phải hoàn thiện giao diện và trải nghiệm người dùng của ứng dụng.

* **Hướng phát triển**
* Thiết kế và lập trình mở rộng các chức năng: Đồng bộ tự động tích hợp hệ thống Back End, Tích hợp thêm thanh toán học phí online, phát triển thêm một số chức năng như điểm danh, tạo thời khóa biểu, …
* Cải thiện sự mượt mà của ứng dụng. Nâng cao trải nghiệm người dùng.
* Em rất mong tiếp tục nhận được sự giúp đỡ và tạo điều kiện của thầy cô và nhà trường để em có cơ hội phát triển, hoàn thiện ứng dụng tốt hơn trong thời gian tới.

TÀI LIỆU THAM KHẢO

[1]. Tổ HTTT Đại học Công Nghiệp Hà Nội – Giáo trình hệ quản trị cơ sở dữ liệu SQL.

[2]. Tổ HTTT Đại học Công Nghiệp Hà Nội – Giáo trình phân tích thiết kế hệ thống.

[3]. Vũ Thị Dương, Phùng Đức Hòa, Nguyễn Thị Hương Lan (2015), “Giáo trình Phân tích thiết kế hướng đối tượng”, NXB Khoa học và Kỹ thuật.

{Thomas, 2000 #1}