TRƯỜNG ĐẠI HỌC PHENIKAA

KHOA CÔNG NGHỆ THÔNG TIN

⸎⸎⸎⸎⸎



Báo cáo bài tập lớn môn Lập trình cho thiết bị di động

Đề tài : Xây dựng App

|  |  |
| --- | --- |
| **Nhóm sinh viên thực hiện:** | Nhóm 06 |
| **Lớp tín chỉ:** | Lập trình cho thiết bị di động-1-2-23(N06) |
| **Giảng viên hướng dẫn:** | Nghiêm Việt Cường |
|  |  |

Hanoi, ngày tháng năm 2024

# Mục lục

[Mục lục 2](#_Toc162551474)

[Hình ảnh 3](#_Toc162551475)

[Bảng biểu 4](#_Toc162551476)

[Phân công công việc 4](#_Toc162551477)

[Kế hoạch phát triển thời gian còn lại: 5](#_Toc162551478)

[1. Mở đầu 5](#_Toc162551479)

[1.1: Đặt vấn đề 5](#_Toc162551480)

[1.2: Giải pháp 5](#_Toc162551481)

[1.3: Phạm vi đề tài 5](#_Toc162551482)

[2. Công nghệ sử dụng 6](#_Toc162551483)

[2.1: Tổng quan về Flutter 6](#_Toc162551484)

[2.2: Tổng quan về Dart 8](#_Toc162551485)

[2.3: Tổng quan về Figma 9](#_Toc162551486)

[3. Mô hình hệ thống (Use-case model) 10](#_Toc162551487)

[3.1: Mô tả chi tiết use-case Đăng nhập 10](#_Toc162551488)

[3.2: Mô tả chi tiết use-case Thông tin 11](#_Toc162551489)

[3.3: Mô tả chi tiết use-case Tổng điểm 12](#_Toc162551490)

[3.4: Mô tả chi tiết use-case Điểm chi tiết 13](#_Toc162551491)

[4. Phát triển và triển khai ứng dụng 15](#_Toc162551492)

[4.1: Giao diện chính 15](#_Toc162551493)

[4.2: Giao diện chính 18](#_Toc162551494)

[4.3: Giao diện Thông tin cá nhân 19](#_Toc162551495)

[4.4: Giao diện Lịch học 20](#_Toc162551496)

[4.5: Giao diện Tổng điểm 21](#_Toc162551497)

[4.6: Giao diện Điểm chi tiết 22](#_Toc162551498)

[5. Kết luận 23](#_Toc162551499)

[6. Tài liệu tham khảo 24](#_Toc162551500)

# Hình ảnh

[Hình 1:Framework Flutter 6](#_Toc162551276)

[Hình 2 Dart programming language 7](#_Toc162551277)

[Hình 3 Figma 8](#_Toc162551278)

[Hình 4 Activity Diagram Đăng nhập 9](#_Toc162551279)

[Hình 5 Activity Diagram Thông tin 11](#_Toc162551280)

[Hình 6 Activity Diagram Tổng điểm 12](#_Toc162551281)

[Hình 7 Activity Diagram Chi tiết điểm 13](#_Toc162551282)

[Hình 8 Giao diện Login 16](#_Toc162551283)

[Hình 9 Giao diện Login 17](#_Toc162551284)

[Hình 10 Giao diện chính 18](#_Toc162551285)

[Hình 11 Giao diện Thông tin 19](#_Toc162551286)

[Hình 12 Giao diện Lịch học 20](#_Toc162551287)

[Hình 13 Giao diện Tổng điểm 21](#_Toc162551288)

[Hình 14 Giao diện kỳ học 22](#_Toc162551289)

# Bảng biểu

[Bảng 1 Phân công công việc 4](#_Toc162551326)

[Bảng 2 Use-case Đăng nhập 11](#_Toc162551327)

[Bảng 3 Use-case Thông tin 12](#_Toc162551328)

[Bảng 4 Use-case Tổng điểm 13](#_Toc162551329)

[Bảng 5 Bảng Use-case Chi tiết điểm 14](#_Toc162551330)

# Phân công công việc

|  |  |  |  |
| --- | --- | --- | --- |
| Tên công việc | Vi Đăng Quang | Lê Hoàng Ngọc Tú | Hoàng Thế Cường |
| Tạo màn hình đăng nhập | 100% |  |  |
| Tìm ra lỗ hổng client của phương pháp đăng nhập bằng microsoft của web quản lý đào tạo | 100% |  |  |
| Vẽ UI người dùng |  | 100% |  |
| Call dữ liệu thông tin sinh viên đã đăng nhập về app | 100% |  |  |
| Call dữ liệu chi tiết thông tin điểm sinh viên về app |  | 100% |  |
| Call dữ liệu lịch học về app | 30% |  |  |
| Màn hình thông tin sinh viên | 100% |  | 50% (giao việc tiếp theo không làm và đã rời nhóm) |
| Màn hình chi tiết thông tin điểm sinh viên |  | 50% |  |
| Màn hình điểm tông sinh viên |  | 30% |  |

Bảng 1 Phân công công việc

Link source code của nhóm:

github.com/QuangViDang/LTMOBILE\_QLDT\_PKA\_N06

# Kế hoạch phát triển thời gian còn lại:

Trong thời gian còn lại, chúng em dự định phát triển hoàn thiện, công thức tính điểm trung bình, chuyển động của màn hình lịch học, đăng nhập một cách thuận tiện hơn. Hoàn thiện các màn hình và logic trước đó.

# 1. Mở đầu

## 1.1: Đặt vấn đề

Khoa học công nghệ ngày càng phát triển nhiều người dùng cũng có thói quen dùng smartphone thay vì lên website bằng máy tính để phục vụ mua hàng, quản lí,..v.v. Hay vấn đề không đâu xa xôi là chính các bạn học sinh, sinh viên thế hệ 4.0, 5.0 hiện nay cũng là một trong những người dùng smartphone phổ biến nhất. Việc các bạn sinh viên ngày nay tiếp xúc khá nhiều từ mạng xã hội facebook, shoppe,tiktok.. đều có những ứng dụng trên smartphone tối ưu nhất nhưng việc trường học thì lại có những vẫn đề bất cập về ứng dụng quản lí điểm, lịch học, thông tin cá nhân.. thì hầu như các bạn đều phải lên website để xem và website trường nhiều khi không đáp ứng được nhu cầu cho các bạn.

Với việc phải xem lí thông tin, lịch học, điểm .. một cách tiện nhất thì chúng tôi sẽ phát triển một ứng dụng để giúp các bạn có góc nhìn phù hợp nhất, tiện nhất về điểm, thông tin, lịch học.

## 1.2: Giải pháp

Ứng dụng chúng tôi nhắm tới giúp cho học sinh, sinh viên dễ dàng quản lí điểm, lịch học,.. một cách tiện lợi nhất. Hệ thống sẽ dựa trên lượng dữ liệu của trường thông qua Microsoft nên việc thông tin của sinh viên, học sinh sẽ không bị nhầm lẫn hay sai sót. Chúng tôi thiết kế một ứng dụng dễ nhìn, tiện lợi và sử dụng trên hầu hết smartphone.

## 1.3: Phạm vi đề tài

Chúng tôi sẽ tìm hiểu về:

1. Tìm hiểu về lập trình Thiết bị di động
2. Tìm hiểu về Flutter, Dart
3. Phân trang và xây dựng bố cục
4. Thiết kế các trang ứng dụng chính
5. Xây dựng ứng dụng di động sử dụng Flutter

# 2. Công nghệ sử dụng

Đối với phát triển ứng dụng di động iOS và Android là hai nền tảng chính. Mỗi nền tảng này cần các loại code khác nhau để phát triển ứng dụng. Từ đó các công ty phát triển ứng dụng di động đã dẫn đến việc phát triển ứng dụng đa nền tảng, có nghĩa là viết một và chạy trên nhiều nền tảng khác nhau.

Việc phát triển ứng dụng đa nền tảng là phát triển các ứng dụng di động theo cách mà chúng có thể chạy trên nhiều nền tảng khác nhau nhưng chỉ code 1 lần. Trong kiểu phát triển này chỉ cần code một lần và theo cách mà ứng dụng Android, iOS hoặc Windows hỗ trợ. Phát triển đa nền tảng đã trở nên khá phổ biến vì các tính năng, sự tiện lợi và công cụ được các developer yêu thích.

Một số công cụ phổ biến để phát triển đa nền tảng bao gồm là Xamarin của Microsoft, React Native của Facebook và PhoneGap từ Adobe. Mỗi công cụ phát triển đa nền tảng này có các tính năng khác nhau cùng với ưu và nhược điểm khác nhau. Các developer luôn tìm kiếm một cái gì đó mới và được cải tiến để làm cho các quy trình phát triển ứng dụng trở nên hiệu quả và tiết kiệm thời gian khi làm ứng dụng đa nền tảng.

## 2.1: Tổng quan về Flutter

Flutter là một framework phát triển ứng dụng di động open-source được phát triển bởi bởi Google. Nó được sử dụng để xây dựng ứng dụng di động chạy trên nhiều nền tảng, bao gồm cả Android và iOS, từ một mã nguồn duy nhất. Flutter sử dụng ngôn ngữ lập trình Dart để viết mã nguồn

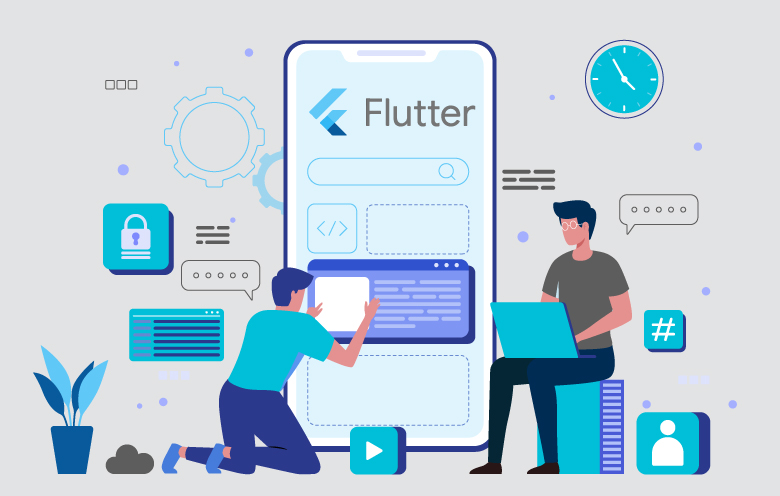
Một trong những đặc điểm nổi bật của Flutter là khả năng tạo giao diện người dùng đẹp mắt và linh hoạt, với khả năng tương tác cao. Nó cung cấp một loạt các widget để xây dựng giao diện người dùng, và đặc biệt nổi tiếng với tính năng "hot reload," cho phép nhà phát triển xem ngay lập tức các thay đổi trong mã nguồn của họ mà không cần khởi động lại ứng dụng

Flutter

Flutter đang được sử dụng bởi các developer, điều đó chứng tỏ rằng có một cái gì đó độc đáo đã tồn tại và thu hút các developer. Dưới đây là một trong những điều đó:

* Khả năng hỗ trợ nhiều API Firebase hơn
* Có công cụ riêng Sửa lỗi
* Tài liệu cải tiến Hỗ trợ phát triển windows
* Hiệu suất được nâng cao
* Công cụ cho Android Studio và Visual Studio Code
* Các tính năng bổ sung như video, biểu đồ và quảng cáo

Phát triển ứng dụng với Flutter



Hình 1:Framework Flutter

*Nguồn ảnh “https://intech.vietnamworks.com/”*

Từ quan điểm của chủ sở hữu ứng dụng, việc phát triển một ứng dụng với Flutter rất nhanh và ít tốn kém. Và những lợi thế khác của việc sử dụng Flutter để phát triển ứng dụng như:

Ưu điểm:

* Đa nền tảng (Cross-platform): Flutter cho phép phát triển ứng dụng cho cả Android và iOS từ một mã nguồn duy nhất, giảm độ phức tạp và chi phí của quá trình phát triển.
* Widget đa dạng: Flutter cung cấp một loạt các widget tích hợp sẵn để xây dựng giao diện người dùng, giúp tạo ra các ứng dụng với giao diện đẹp và linh hoạt.
* Hot Reload: Tính năng này cho phép nhà phát triển xem ngay lập tức kết quả của các thay đổi trong mã nguồn mà không cần khởi động lại ứng dụng, giúp tăng tốc quá trình phát triển.
* Dart language: Dart, ngôn ngữ lập trình của Flutter, được thiết kế để đơn giản và hiệu quả, với cú pháp tương đối dễ đọc và hiểu.
* Cộng đồng đông đảo: Cộng đồng người sử dụng Flutter ngày càng lớn mạnh, cung cấp nguồn hỗ trợ, tài nguyên và thư viện đa dạng

Nhược điểm: Mặc dù có nhiều ưu điểm, qua việc thử nghiệm, Flutter có những điểm yếu cố định

* Flutter vẫn chưa thật hoàn thiện.Vì là một ứng dụng mới nên Flutter vẫn chưa đạt mức hoàn hảo. Thực tế, nhiều tính năng nâng cao của Flutter vẫn chưa được hỗ trợ; nhiều thư viện chưa được thử nghiệm chính thức còn tồn tại hạn chế khi so sánh với các bản sao gốc (như là Google Maps).
* Dart khá "non nớt"**.** Về cơ bản Dart khá giống với Swift và Kotlin, nhưng có ít tính năng hơn, hoặc những tính năng hiện có chưa được toàn diện
* Các ứng dụng Flutter khá "nặng". Chúng chiếm nhiều dung lượng và mất nhiều thời gian để tải xuống hoặc cập nhật.
* Giao diện không giống 100% so với phiên bản gốc. Về cơ bản, Flutter không tạo ra các thành phần gốc mà sao chép không hoàn toàn các thiết kế Material Design của Android và các thành phần riêng của iOS bằng thư viện Cupertino
* Hướng dẫn phát triển ứng dụng Flutter chưa được đồng nhất, điều này có thể gây nhiều khó khăn khi xây dựng các phần mềm mang tính phức tạp.
* Framework thay đổi nhanh chóng gây khó khăn cho việc duy trì mã. Thêm vào đó, Flutter chưa chắc sẽ được ứng dụng trong tương lai khi Google liên tục loại bỏ các dự án của ứng dụng này.

## 2.2: Tổng quan về Dart

Dart là một ngôn ngữ lập trình được phát triển bởi Google, được thiết kế chủ yếu để xây dựng ứng dụng di động, web và máy chủ. Dart thường được sử dụng để phát triển ứng dụng sử dụng framework Flutter để xây dựng giao diện người dùng đa nền tảng



Hình 2 Dart programming language

*Nguồn ảnh “https://viblo.asia/”*

## 2.3: Tổng quan về Figma

Figma là một ứng dụng thiết kế giao diện người dùng(UI/UX) dựa trên web và cộng tác đa người dùng. Nó cho phép các nhà thiết kế, nhà phát triển và bất kì ai đang làm việc trên dự án thiết kế cùng nhau tạo, chỉnh sửa và chia sẻ các thiết kế một cách dễ dàng.



Hình 3 Figma

*Nguồn ảnh “https://www.figma.com/”*

Điểm nổi bật của Figma:

* Giao diện dựa trên website: Figma hoạt động trên trình duyệt web, loại bỏ như cầu cài đặt phần mềm và cho phép truy cập bất kỳ thiết bị nào kết nối internets.
* Cộng tác đa người dùng thời gian thực: Nhiều người dùng có thể làm việc trên cùng một tài liệu Figma cùng một lúc, với các thay đổi được phản ánh ngay lập tức và có thể xem trực tiếp.
* Tích hợp công cụ thiết kế đa dạng: Figma cung cấp nhiều công cụ về hình học, vector cũng như tính năng như protoyping và animation.
* Componenet và thư viện: Figma cho phép tạo ra các component tái sử dụng và quản lí chúng trong thư viện. Điều này giúp giữ các yếu tố thiết kế nhất quán và dễ dàng cập nhật trên toàn bộ dự án.
* Bảo mật và quản lí dữ liệu: Figma cung cấp các tính năng bảo mật như quản lý quyền truy cập và kiểm soát phiên bản để đảm bảo an toàn cho dữ liệu thiết kế.

Figma trở thành một trong những công cụ thiết kế phổ biến trong công đồng thiết kế UI/UX, đặc biệt trong các dự án đòi hỏi tính cộng tác cao và khả năng truy cập từ xa.

# 3. Mô hình hệ thống (Use-case model)

## 3.1: Mô tả chi tiết use-case Đăng nhập

A screenshot of a cell phone

Description automatically generated

Hình 4 Activity Diagram Đăng nhập

|  |  |
| --- | --- |
| Use-case | Nội dung |
| Tên use-case | Đăng nhập |
| Mô tả | Người dùng xác minh danh tính để sử dụng ứng dụng |
| Tác nhân | Người dùng (Sinh viên) |
| Điều kiện kích hoạt | Người dùng chọn chức năng Đăng nhập |
| Điều kiện đầu vào | Hệ thống hiển thị giao diện Đăng nhập  Thiết bị người dùng kết nối internets |
| Luồng cơ bản | * + 1. Người dùng mở ứng dụng Quản lí thông tin     2. Màn hình hiển thị giao diện Đăng nhập     3. Chọn chức năng Đăng nhập     4. Hệ thống hiển thị giao diện Đăng nhập     5. Người dùng đăng nhập thông qua tài khoản trường cấp bằng Microsoft     6. Hệ thống xác minh thông tin người dùng đã khớp với tài khoản Microsoft và chuyển qua giao diện chính |
| Luồng ngoại lệ | Hệ thống hiển thị thông báo đăng nhập thất bại  Người dùng nhập lại thông tin không hợp lệ  Hệ thống xác nhận lại đạt yêu cầu |

Bảng 2 Use-case Đăng nhập

## 3.2: Mô tả chi tiết use-case Thông tin

A screenshot of a phone

Description automatically generated

Hình 5 Activity Diagram Thông tin

|  |  |
| --- | --- |
| Use-case | Nội dung |
| Tên use-case | Thông tin |
| Mô tả | Người dùng xem thông tin cá nhân của mình trên ứng dụng |
| Tác nhân | Người dùng (Sinh viên) |
| Điều kiện kích hoạt | Người dùng đã Đăng nhập và đang sử dụng ứng dụng |
| Điều kiện đầu vào | Hệ thống hiển thị giao diện Thông tin  Thiết bị người dùng kết nối internets |
| Luồng cơ bản | * + 1. Người dùng ở giao diện “Thông tin” sao khi chuyển từ giao diện chính     2. Hiển thị thông tin cơ bản của người dùng |

Bảng 3 Use-case Thông tin

## 3.3: Mô tả chi tiết use-case Tổng điểm

A screenshot of a phone

Description automatically generated

Hình 6 Activity Diagram Tổng điểm

|  |  |
| --- | --- |
| Use-case | Nội dung |
| Tên use-case | Tổng điểm |
| Mô tả | Người dùng xem thông tin cá nhân của mình trên ứng dụng |
| Tác nhân | Người dùng (Sinh viên) |
| Điều kiện kích hoạt | Người dùng đã Đăng nhập và đang sử dụng ứng dụng |
| Điều kiện đầu vào | Hệ thống hiển thị giao diện Thông tin  Thiết bị người dùng kết nối internets |
| Luồng cơ bản | * + 1. Người dùng ở giao diện “Tổng điểm” sao khi chuyển từ giao diện chính     2. Hiển thị thông tin điểm trung bình của người dùng |

Bảng 4 Use-case Tổng điểm

## 3.4: Mô tả chi tiết use-case Điểm chi tiết

A screenshot of a cell phone

Description automatically generated

Hình 7 Activity Diagram Chi tiết điểm

|  |  |
| --- | --- |
| Use-case | Nội dung |
| Tên use-case | Chi tiết điểm |
| Mô tả | Người dùng xem thông tin cá nhân của mình trên ứng dụng |
| Tác nhân | Người dùng (Sinh viên) |
| Điều kiện kích hoạt | Người dùng đã Đăng nhập và đang sử dụng ứng dụng |
| Điều kiện đầu vào | Hệ thống hiển thị giao diện Kỳ học  Thiết bị người dùng kết nối internets |
| Luồng cơ bản | * + 1. Người dùng ở giao diện “Kỳ học” sao khi chuyển từ giao diện chính     2. Hiển thị thông tin giao diện Kỳ học     3. Người dùng chọn kỳ học     4. Hệ thống hiển thị chi tiết môn học điểm số kì học đó |

Bảng 5 Bảng Use-case Chi tiết điểm

# 4. Phát triển và triển khai ứng dụng

Ứng dụng của chúng em xác định có 5 chức năng chính:

1. Xem thông tin sinh viên

2. Xem thông tin điểm chi tiết sinh viên

3. Xem thông tin điểm trung bình

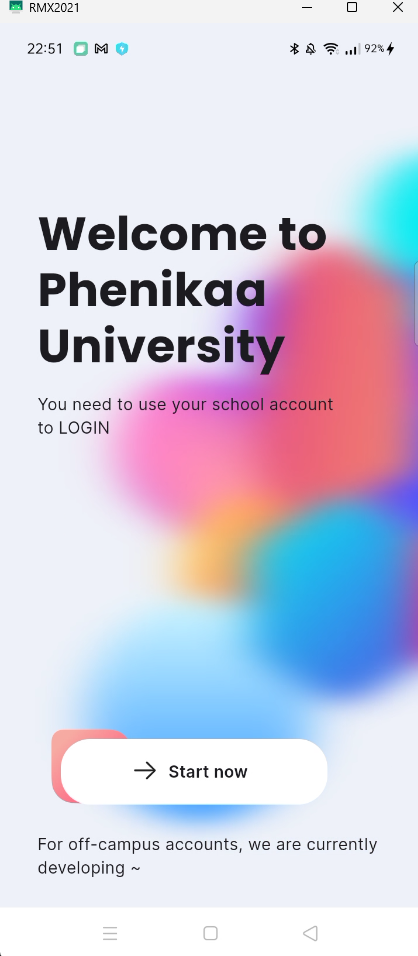
4. Xem thông tin lịch học

5. Thông báo lịch học (tính năng sẽ phát triển nếu đủ thời gian nghiên cứu)

6. Với giảng viên, các tính năng sẽ hơi khác sinh viên nên chúng em dự định phát triển sau.

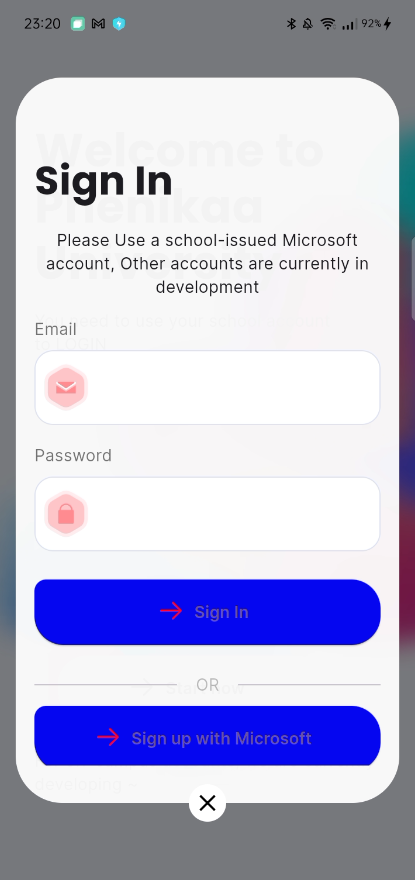
## 4.1: Giao diện chính

Giao diện đăng nhập (Login interface) là một phần quan trọng trong hầu hết các ứng dụng, từ website và hệ thống. Nơi mà người dùng nhập thông tin đăng nhập để truy cập vào tài khoản của họ. Với đề tài chúng tôi đang thực hiện thì việc thiết kế một giao điện đơn giản cho người dùng (học sinh, sinh viên) tiếp cận là một điều cần thiết. Những tính năng cơ bản chúng tôi mang lại sẽ thuận tiện cho người dùng. Dưới đây chính là giao diện chính giao diện đăng nhập.



Hình 8 Giao diện Login

Khi người dùng “click” vào ứng dụng sẽ hiển thị giao diện “Hình 5”. Tiếp đến để nhập thông tin đăng nhập “click Start now” sẽ hiển thị giao diện “Hình 6” ở dưới



Hình 9 Giao diện Login

Do đề tài chúng tôi liên quan đến tài khoản thông tin cá nhân của sinh viên nên việc xin thông tin dữ liệu của trường để phục vụ đề tài là bất khả thi. Nên ta phải dựa vào tài khoản Microsoft. Sinh viên “click – Sign up with Microsoft” sẽ chuyển đến phần đăng nhập bằng tài khoản sinh viên.

Bằng việc timg hiểu sử dụng những thư viện của Dart và Flutter như Webview, hay Provider thì việc chúng tôi đã lấy được những dữ liệu cần thiết để phục vụ đề tài.

## 4.2: Giao diện chính

Quá trình đăng nhập thông qua Microsoft hoàn tất người dùng (học sinh, sinh viên) sẽ chuyển đến giao diện chính “Hình 6”.

A screenshot of a phone

Description automatically generated

Hình 10 Giao diện chính

Ở đây sẽ có một vài mục mà chúng tôi đã thiết lập lên ý tưởng chính như: Thông tin,Thời khóa biểu ,Tổng điểm, Kì học. Mỗi một trường thông tin sẽ có những chức năng khác nhau.

## 4.3: Giao diện Thông tin cá nhân

Với đề tài quản lí thông tin cá nhân của sinh viên khi sinh viên đã đăng nhập thông qua Microsoft chuyển đến giao diện chính thì việc xem thông tin chi tiết làm ở trường thông tin ở giao diện là “Thông tin”



Hình 11 Giao diện Thông tin

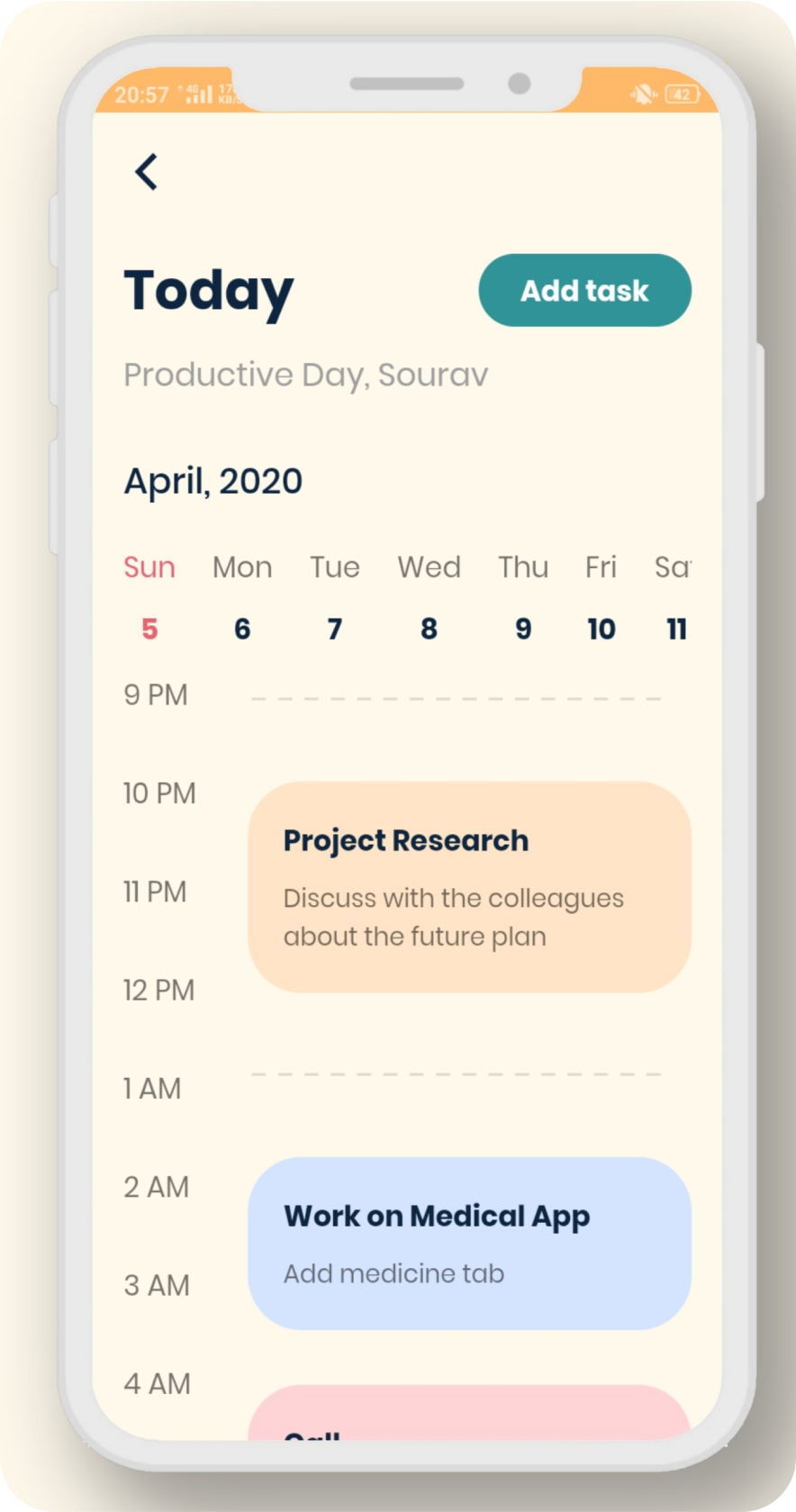
Ở đây có thể thấy việc tiếp nhận thông tin sinh viên từ trường thông qua Microsoft và sử dụng thư viện webview để tiếp nhận đúng những trường thông tin cần thiết.

Giao diện Thông tin sinh viên có thể thấy những thông tin có cơ bản như là họ tên, năm sinh, ngành học, trạng thái học, ... đúng với những gì như trên quản lí đào tạo của trường học.

Đối với phần “Active Projects” hiện tại chúng tôi đang nghiêm cứu và phát triển trong tương lai

## 4.4: Giao diện Lịch học

Bên cạnh thông tin thi phần không thể thiếu đối với mỗi sinh viên là nắm bắt lịch học. Chúng tôi đã tìm hiểu và thiết kế nên giao diện Lịch học cho sinh viên đơn giản dễ nhìn nhất.

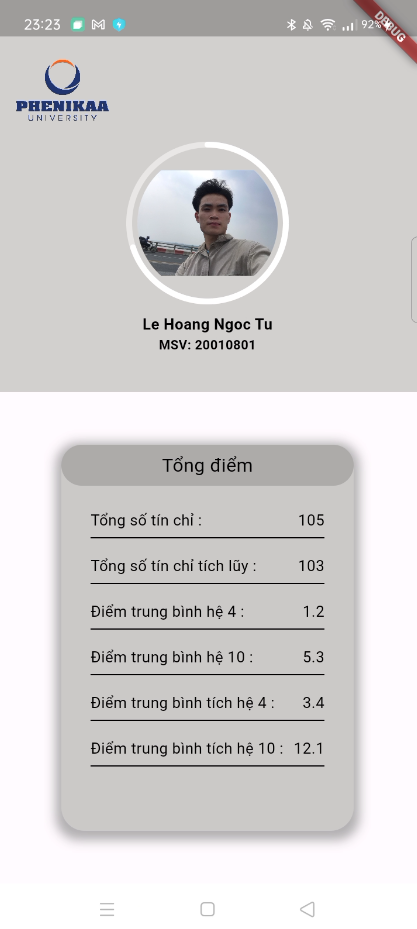


Hình 12 Giao diện Lịch học

Giao diện đơn giản với các môn học và khung giờ học theo từng sẽ không làm dối mắt của sinh viên khi phải nhìn vào website quản lí ở trường rất dối mắt.

## 4.5: Giao diện Tổng điểm

Phần không thể thiếu với sinh viên là thông tin tổng điểm, để quản lí một cách dễ dàng chúng tôi đã xây dựng giao diện tổng điểm để phục vụ cho sinh viên quản lí.

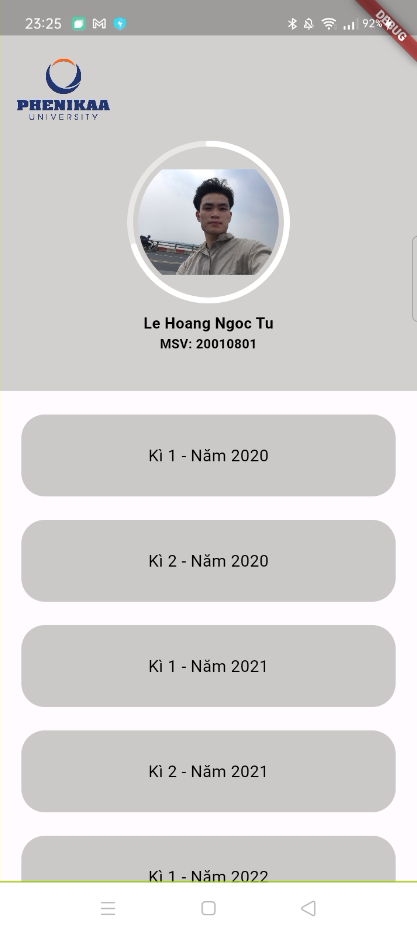


Hình 13 Giao diện Tổng điểm

Giao diện này chúng tôi sẽ lấy một vài thông tin cơ bản như tổng số tín chỉ, tổng số tín tích lũy, điểm trung bình hệ 4 và 10 và điểm tích lũy. Việc này dễ thấy nhất là xem mình đã tích lũy được bao nhiêu tin và đã trong tình trạng nợ môn hay không để biết cách quản lí để học lại.

## 4.6: Giao diện Điểm chi tiết

Để xem chi tiết môn học và từng kì thì chung tôi thiết kế nên giao diện và cũng lấy dữ liệu của sinh viên thông qua Microsoft. Xem chi tiết từng kì “Click vào từng kỳ” sẽ hiện thị ra môn và điểm thành phần của môn từng kì đó.



Hình 14 Giao diện kỳ học

# 5. Kết luận

Trong báo cáo này, chúng tôi đã thảo luận về việc phát triển một ứng dụng di động để quản lí thông tin điểm, lịch học , thông tin cá nhân cho sinh viên. Chúng tôi nhận thấy việc có một ứng dụng di động như vậy không chi giúp sinh viên tổ chức thông tin một cách hiệu quả mà còn giúp sinh quản lí được thời gian hơn ví dụ về việc xem lịch học buổi hôm nay thay vì phải xem tất cả các buổi học với lịch học chằng chịt trên website quản lí đào tạo của trường.

Trong quá trình nghiêm cứu và phát triển, chúng tôi đã đưa ra những yêu cầu cụ thể về tính năng và giao diện người dùng ứng dụng. Chúng tôi cân nhắc về việc tôi ưu hóa trải nghiệm cho người dùng, đảm bảo tính thân thiện, dễ sử dụng cũng như tính di động chỉ cần bạn có internets,

Ngoài ra chúng tôi có những thách thức rõ rệt như việc thiết kế giao diện đơn giản quá, việc tìm kiếm nguồn dữ liệu khá khó khăn, khi có dữ liệu phải xử lí thế nào để tối ưu nguồn dữ liệu đấy vừa đủ cho sử dụng trong quá trình phát triển.

Bên cạnh đó chúng tôi đã có những cơ hội tiếp cận những thông tin mới, tích lũy được khá nhiều kiến thức mới

Từ những nỗ lực nghiêm và phát triển sẽ hi vọng ứng dụng sẽ mang lại trải nghiệm tốt cho người dùng trong việc quản lí thông tin, điểm số, lịch học.....

Trong tương lai, chúng tôi sẽ tiếp tục nghiêm cứu và phát triển ứng dụng một cách tối ưu hơn nữa, cải thiện lại giao diện, và phát triển một số tính năng khác.

# 6. Tài liệu tham khảo

[1] <https://www.tutorialspoint.com/flutter/index.htm>

[2] https://www.figma.com/resource-library/design-basics/

[3] <https://www.youtube.com/@dbestech>

[4] <https://www.youtube.com/@techwithsam>