 **BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC SƯ PHẠM KĨ THUẬT TP.HCM  
KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

**Xây dựng ứng dụng di động quản lý học tập**

**LỚP: ADMP431879\_23\_2\_04CLC (Chiều thứ năm)**

**Môn: Lập trình di động nâng cao**

**NHÓM: 3**

**GVHD: Th.S Nguyễn Hữu Trung**

***Thành phố Hồ Chí Minh, Tháng 4 năm 2024***

**DANH SÁCH NHÓM**

HỌC KÌ II, NĂM HỌC: 2023-2024

**Tên đề tài: Xây dựng ứng dụng di động quản lý học tập**

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Họ và tên | MSSV | Mức độ đóng góp |
| 1 | Võ Trần Bảo Nguyên | 20110138 | 100% |
| 2 | Huỳnh Hồ Thọ Tỷ | 20110597 | 100% |

**Nhận xét của giáo viên:**

..................................................................................................................................................

..................................................................................................................................................

..................................................................................................................................................

..................................................................................................................................................

*Ngày ........tháng........năm.......*

*Giáo viên chấm điểm*

**Mục lục**

[1. Tổng quan 4](#_Toc164689773)

[1.1. Mô tả 4](#_Toc164689774)

[1.2. Phân tích nghiệp vụ 4](#_Toc164689775)

[1.3. Công nghệ sử dụng 5](#_Toc164689776)

[1.3.1. Công nghệ Front-end: React Native 5](#_Toc164689777)

[1.3.2. Công nghệ Back-end: ExpressJs 5](#_Toc164689778)

[1.3.3. Hệ quản trị cơ sở dữ liệu: MongoDB 6](#_Toc164689779)

[1.4. Công cụ và môi trường lập trình 6](#_Toc164689780)

[2. Thiết kế và xây dựng hệ thống 9](#_Toc164689781)

[2.1. Class diagram 9](#_Toc164689782)

[2.2. Usecase diagram 10](#_Toc164689783)

[2.3. Thiết kế các API cho hệ thống 10](#_Toc164689784)

[2.4. Thiết kế giao diện cho hệ thống 10](#_Toc164689785)

[3. Thực nghiệm và hướng dẫn sử dụng 11](#_Toc164689786)

[3.1. Đăng ký 11](#_Toc164689787)

[3.2. Đăng nhập 13](#_Toc164689788)

[3.3. Quên mật khẩu 15](#_Toc164689789)

[3.4. Đổi mật khẩu 17](#_Toc164689790)

[3.5. Xem danh sách lớp (tài khoản giảng viên) 19](#_Toc164689791)

[3.6. Xem chi tiết lớp 21](#_Toc164689792)

[3.7. Chỉnh sửa lớp (tài khoản giảng viên) 23](#_Toc164689793)

[3.8. Tạo bài tập (tài khoản giảng viên) 27](#_Toc164689794)

[3.9. Nộp bài tập (tài khoản sinh viên) 28](#_Toc164689795)

[3.10. Xem lịch học/giảng dạy 30](#_Toc164689796)

[3.11. Xem thống kê tình hình học tập của sinh viên (tài khoản giảng viên) 32](#_Toc164689797)

[3.12. Xem tình hình học tập của bản thân(tài khoản sinh viên) 34](#_Toc164689798)

[3.13. Nhận xét, đánh giá lớp học 36](#_Toc164689799)

[3.14. Nhận thông báo 38](#_Toc164689800)

[3.15. Điểm danh 39](#_Toc164689801)

[4. Kết luận 43](#_Toc164689802)

[4.1. Đánh giá 43](#_Toc164689803)

[4.2. Định hướng phát triển 43](#_Toc164689804)

# Tổng quan

## Mô tả

Xây dựng một hệ thống trên nền tảng di động (ứng dụng hay app) dùng để quản lý quá trình học tập và giảng dạy của sinh viên và giảng viên thông qua các chức năng cơ bản như lớp học, lịch dạy, điểm danh, nộp bài, tạo bài nộp, tạo các hoạt động giảng dạy,… Nhằm khiến trải nghiệm học tập và giảng dạy của của sinh viên lẫn giảng viên ở mức tốt nhất.

## Phân tích nghiệp vụ

***Yêu cầu:*** xây dựng một hệ thống dùng để quản lý việc học tập và giảng dạy

***Các loại người dùng cuối (vai trò):*** sinh viên và giảng viên.

***Các chức năng tương ứng với từng loại người dùng:***

|  |  |
| --- | --- |
| Giảng viên | Sinh viên |
| - Đăng nhập.  - Lấy lại mật khẩu (quên mật khẩu).  - Thay đổi mật khẩu.  - Đăng xuất. | |
| - Tạo, chỉnh sửa hoặc xoá các hoạt động học tập trong lớp học, tạo các thông báo, tải lên các tài liệu kham thảo cho sinh viên.  - Xem lịch giảng dạy theo các lớp, theo ngày, theo tuần,…  - Tạo các bài tập cho sinh viên trong từng hoạt động.  - Tạo và tắt phiên điểm danh trong lớp học.  - Xem các nhận xét trong lớp học.  - Khi có nhận xét mới về lớp học, nhận được thông báo.  - Trong lớp học, xem thống kê về tình trạng học tập của các sinh viên trong lớp đó. | - Tham gia các hoạt động trong lớp học, xem các nội dụng từng hoạt động, tải xuống tài nguyên như tài liệu kham thảo.  - Xem lịch học theo các lớp, theo ngày, theo tuần,…  - Khi có bài tập mới được tạo, nhận được thông báo về bài tập mới.  - Thực hiện và nộp các bài tập được giao trong lớp.  - Điểm danh trong lớp học.  - Nhận xét, đánh giá lớp học.  - Xem thống kê về tình hình học tập của bản thân ở các lớp đang tham gia. |

## Công nghệ sử dụng

### Công nghệ Front-end: React Native

React Native là một framework phát triển ứng dụng di động, cho phép các nhà phát triển sử dụng JavaScript và React để xây dựng ứng dụng di động đa nền tảng. React Native được phát triển bởi Facebook và ra mắt lần đầu vào năm 2015. Các đặc điểm nổi bật có thể kể đến như: Sử dụng Javascript để xây dựng ứng dụng di động; Kiến trúc dựa trên các Component; Xây dựng ứng dụng di động đa nền tảng; Cộng đồng và hệ sinh thái phong phú.

### Công nghệ Back-end: ExpressJs

Express.js là một framework phát triển ứng dụng web được xây dựng trên nền tảng Node.js. Nó cho phép nhà phát triển xây dựng các ứng dụng web một cách nhanh chóng và hiệu quả bằng cách sử dụng JavaScript. Các đặc điểm nổi bật có thể kể đến như:

* Minimal và Unopinionated: Express.js là một framework tối giản và không ràng buộc, cho phép nhà phát triển tự do lựa chọn cách tổ chức mã nguồn của họ và sử dụng các thư viện và công nghệ mà họ muốn.
* Routing: Express.js cung cấp một hệ thống routing mạnh mẽ, giúp nhà phát triển dễ dàng xác định các endpoint (điểm cuối) và xử lý các yêu cầu từ phía client.
* Middleware: Middleware là một phần quan trọng trong Express.js, cho phép xử lý các yêu cầu và phản hồi trước khi chúng được chuyển đến các hàm xử lý chính. Điều này cho phép nhà phát triển thực hiện các chức năng như xác thực, ghi nhật ký, và nhiều hơn nữa.
* Template Engine Support: Express.js hỗ trợ nhiều template engine như Pug, EJS, và Handlebars, giúp nhà phát triển dễ dàng tạo ra các trang web động.
* Middleware và Routing lồng nhau: Express.js cho phép middleware và routing được lồng nhau, giúp tạo ra mã nguồn dễ đọc và dễ bảo trì.

### Hệ quản trị cơ sở dữ liệu: MongoDB

MongoDB là một hệ quản trị cơ sở dữ liệu NoSQL mã nguồn mở, được thiết kế để lưu trữ một lượng lớn dữ liệu và cho phép bạn làm việc với nguồn dữ liệu đó một cách hiệu quả. MongoDB là một cơ sở dữ liệu hướng tài liệu, có nghĩa là dữ liệu được lưu trữ trong các tài liệu theo định dạng JSON. Tài liệu có thể có cấu trúc linh hoạt, bao gồm các trường với các kiểu dữ liệu khác nhau. MongoDB có một số ưu điểm nổi bật, bao gồm:

* Khả năng mở rộng: MongoDB có thể mở rộng theo chiều ngang, giúp đáp ứng nhu cầu lưu trữ dữ liệu ngày càng tăng.
* Hiệu suất cao: MongoDB có hiệu suất cao trong việc lưu trữ và truy vấn dữ liệu.
* Linh hoạt: MongoDB có cấu trúc dữ liệu linh hoạt, phù hợp với nhiều loại ứng dụng.

***Ngoài các ưu điểm trên, nó cũng tồn tại vài điểm yếu*** như do việc không có ràng buộc chặt chẽ dễ gây ra các lỗi tiềm tàng + khả năng truy vấn kém, hiệu suất kém khi ở ứng dụng cấp độ lớn và khó bảo trì.

## Công cụ và môi trường lập trình

**Javascript:** ngôn ngữ lập trình mạnh mẽ và linh hoạt có thể được sử dụng để tạo các trang web và ứng dụng web tương tác. JavaScript là một ngôn ngữ cần thiết cho bất kỳ nhà phát triển web nào muốn tạo ra các trang web chất lượng cao. Các đặc điểm nổi bật: (1) JavaScript là một ngôn ngữ lập trình kịch bản, nghĩa là nó được thực thi bởi trình duyệt web. (2) JavaScript là một ngôn ngữ định kiểu yếu, nghĩa là không yêu cầu người lập trình xác định kiểu dữ liệu của các biến.(3) JavaScript là một ngôn ngữ đa mô hình, có thể được sử dụng để viết mã theo các mô hình lập trình khác nhau, chẳng hạn như lập trình hướng đối tượng, lập trình hàm, và lập trình mệnh lệnh. (REF: Wikipedia)

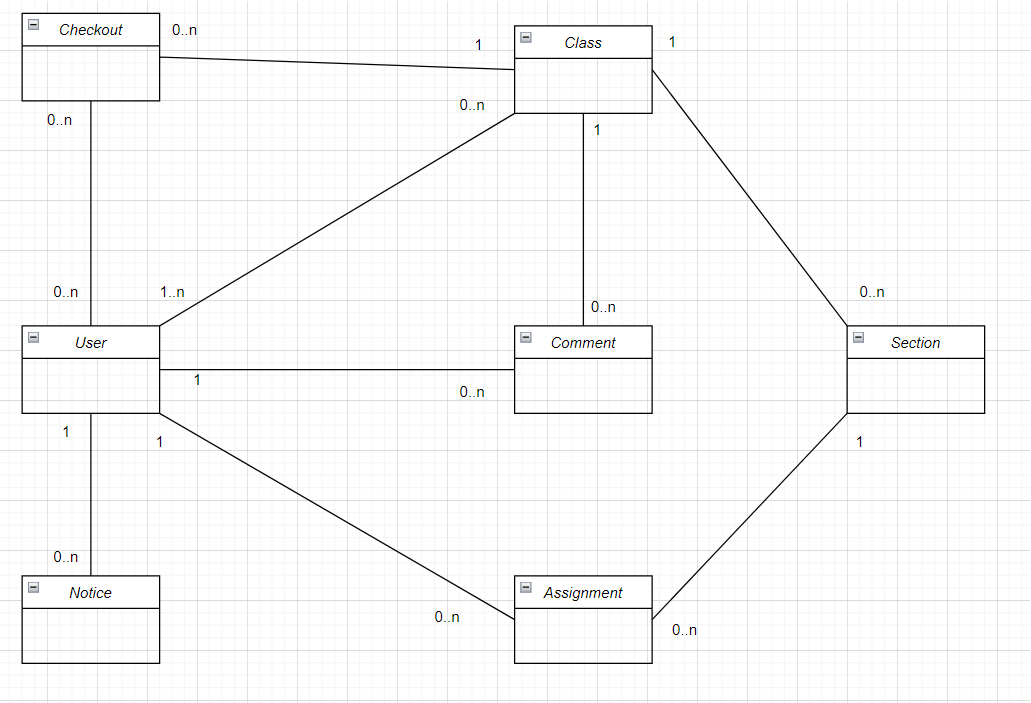
**Công cụ lập trình:** sử dụng Text Editor Visual Code, đây là một trình soạn thảo mã nguồn được phát triển bởi Microsoft dành cho Windows, Linux và macOS. Nó hỗ trợ chức năng debug, đi kèm với Git, có chức năng nổi bật cú pháp (syntax highlighting), tự hoàn thành mã thông minh, snippets, và cải tiến mã nguồn. Nó cũng cho phép tùy chỉnh, do đó, người dùng có thể thay đổi theme, phím tắt, và các tùy chọn khác. Trong cuộc khảo sát vào năm 2018, Visual Studio Code được xếp hạng là trình biên tập mã phổ biến nhất, với 34.9% của 75398 người trả lời tuyên bố sử dụng nó (REF: Wikipedia)

**Môi trường lập trình:** sử dụng NodeJs, đây là một môi trường runtime mạnh mẽ và linh hoạt, chạy JavaScript đa nền tảng và có mã nguồn mở, được sử dụng để chạy các ứng dụng web bên ngoài trình duyệt của client, do đó nó được sử dụng để xây dựng nhiều loại ứng dụng khác nhau. Node.js là một lựa chọn phổ biến cho các nhà phát triển web và ứng dụng muốn tạo ra các ứng dụng hiệu quả và có khả năng mở rộng. Các đặc điểm chính: (1) Node.js được xây dựng trên V8 JavaScript engine của Chrome. (2) Node.js sử dụng kiến trúc hướng sự kiện (event-driven), mô hình I/O không đồng bộ (non-blocking I/O) làm cho nó nhẹ và hiệu quả hơn. (REF: wikipedia).

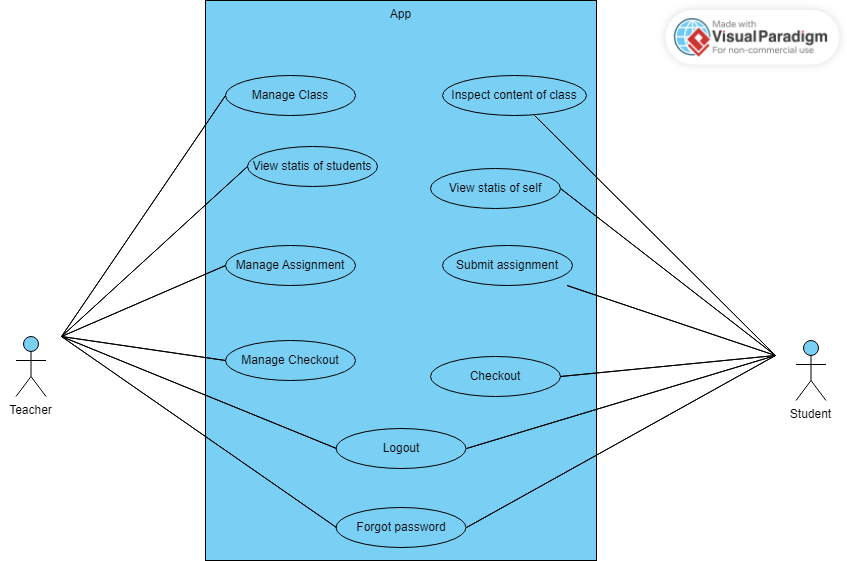
**Công cụ phát triển và kiểm thử:** Postman là một ứng dụng chuyên dụng cho việc phát triển và kiểm thử các API (Application Programming Interface). Được thiết kế với giao diện người dùng thân thiện, Postman giúp nhà phát triển dễ dàng tạo, chia sẻ, và kiểm thử các yêu cầu API một cách hiệu quả. Các tính năng chính của Postman bao gồm khả năng tạo các loại yêu cầu HTTP khác nhau, quản lý môi trường và biến môi trường, tự động hóa các bước kiểm thử, và tạo và chia sẻ các bộ kiểm thử. Nó cung cấp một giao diện đồ họa trực quan và dễ sử dụng, giúp nhóm phát triển tăng cường hiệu suất làm việc với API trong quá trình phát triển phần mềm. (REF: Wikipedia)

# Thiết kế và xây dựng hệ thống

## Class diagram



## Usecase diagram



## Thiết kế các API cho hệ thống

***Tài liệu chi tiết các API được sử dụng trong app:***

[***https://documenter.getpostman.com/view/26836985/2sA3Bq2qEU***](https://documenter.getpostman.com/view/26836985/2sA3Bq2qEU)

## Thiết kế giao diện cho hệ thống

***Tài liệu chi tiết các thiết kế của giao diện trong app:***

[***https://www.figma.com/file/fkAs9x2BdJtUDGiKbGCjaZ/LT%C4%90?type=design&node-id=0-1&mode=design***](https://www.figma.com/file/fkAs9x2BdJtUDGiKbGCjaZ/LT%C4%90?type=design&node-id=0-1&mode=design)

# Thực nghiệm và hướng dẫn sử dụng

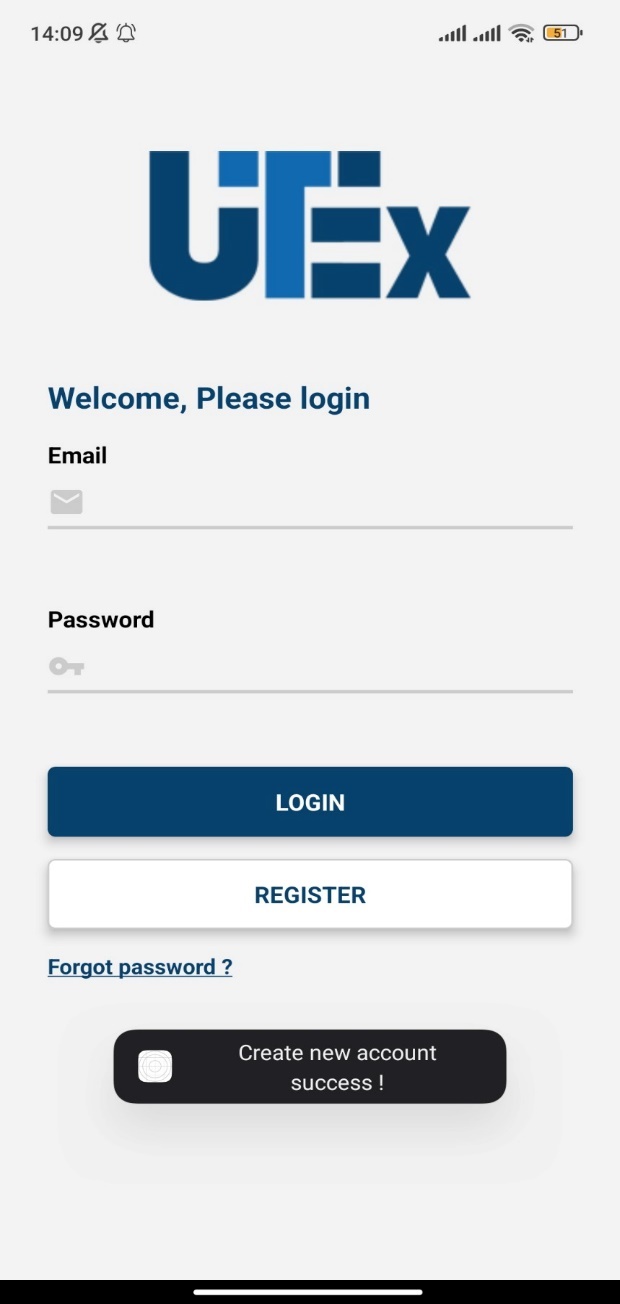
***Tài khoản thử nghiệm:***

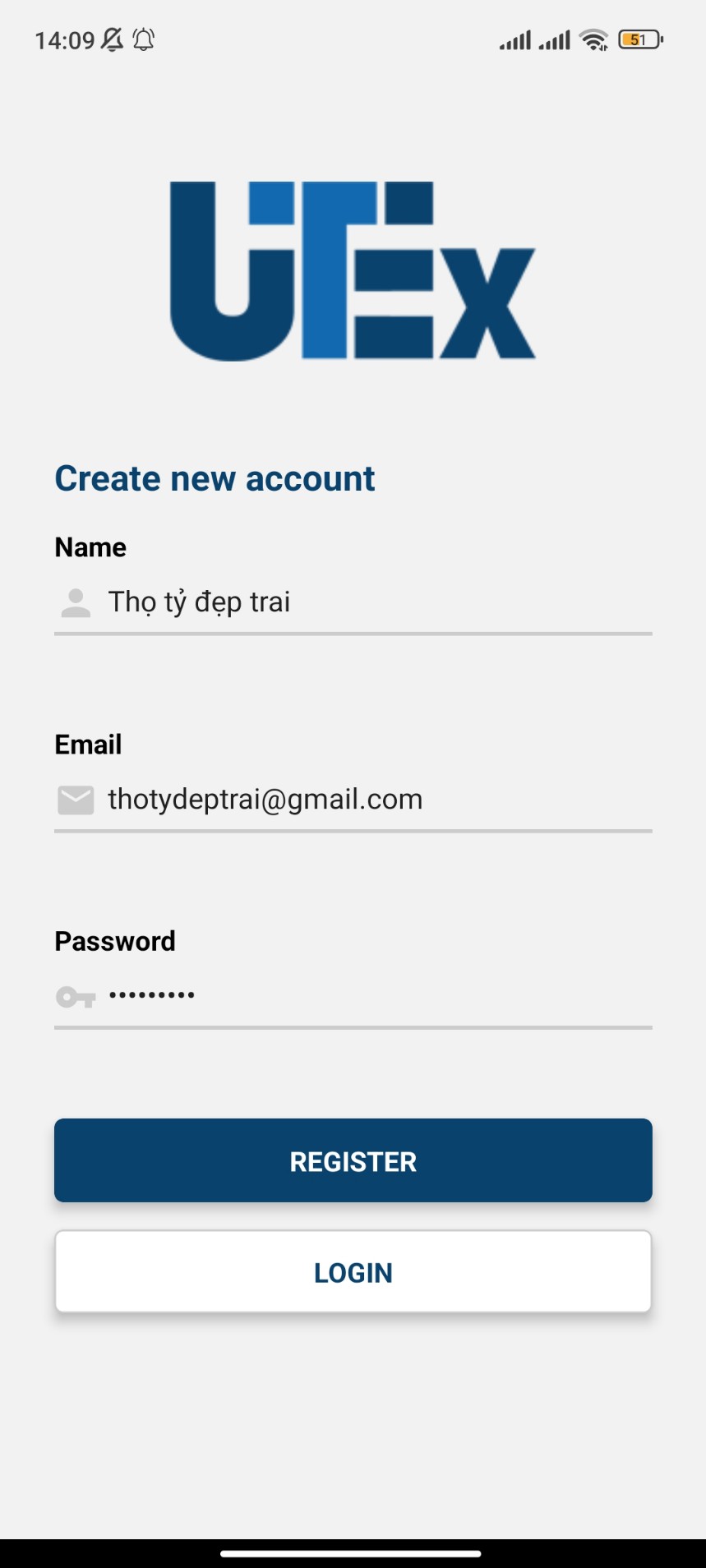
|  |  |
| --- | --- |
| Giảng viên | Email: [huynhhothoty.gpt@gmail.com](mailto:huynhhothoty.gpt@gmail.com)  Password: 123456789 |
| Sinh viên | Email: [huynhhothoty.gpt8@gmail.com](mailto:huynhhothoty.gpt8@gmail.com)  Password: 123456 |

## Đăng ký

- Flow event:

|  |  |
| --- | --- |
| Use case | Đăng ký |
| Actor | Sinh viên |
| Pre-conditions | Vào được App và hiển thị màn hình đăng nhập. |
| Post-conditions | Đăng ký thành công |
| Basic flow | 1. Điền Name, Email và Password  2. Nhấn Register  3. Nếu thông tin đúng, thực hiện A1, còn không thực hiện A2. |

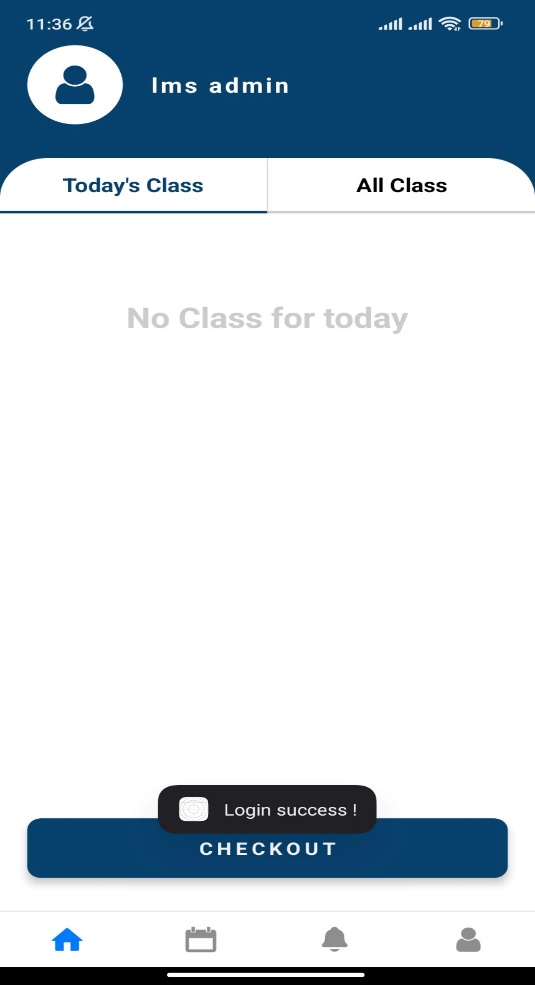
- Hình ảnh minh hoạ:

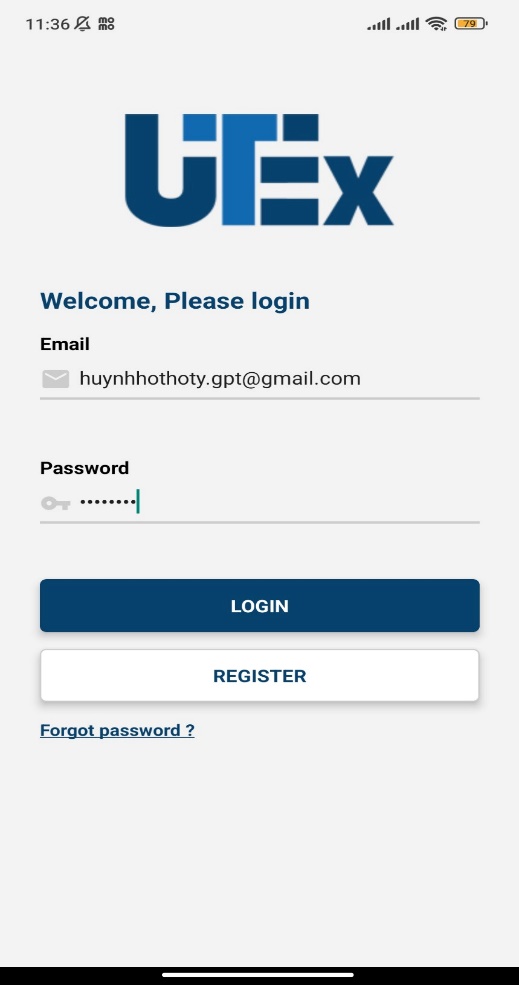


## Đăng nhập

- Flow event

|  |  |
| --- | --- |
| Use case | Đăng nhập |
| Actor | Sinh viên và giảng viên |
| Pre-conditions | Vào được App và hiển thị màn hình đăng nhập. |
| Post-conditions | Đăng nhập thành công |
| Basic flow | 1. Điền Email và Password  2. Nhấn Login  3. Nếu thông tin đúng, thực hiện A1, còn không thực hiện A2. |
| Alternative flow | A1: Nhập đúng thông tin |
|  | 1. Hiển thị thông báo đăng nhập thành công và chuyển đến màn hình ***Home***. |
|  | A2: Nhập sai thông tin |
|  | 1. Hiển thị thông báo nhập sai thông tin, yêu cầu nhập lại.  2. Nhập lại Email và Password.  3. Nếu thông tin đúng, thực hiện A1, còn không thực hiện lại A2. |

- Hình ảnh minh hoạ:

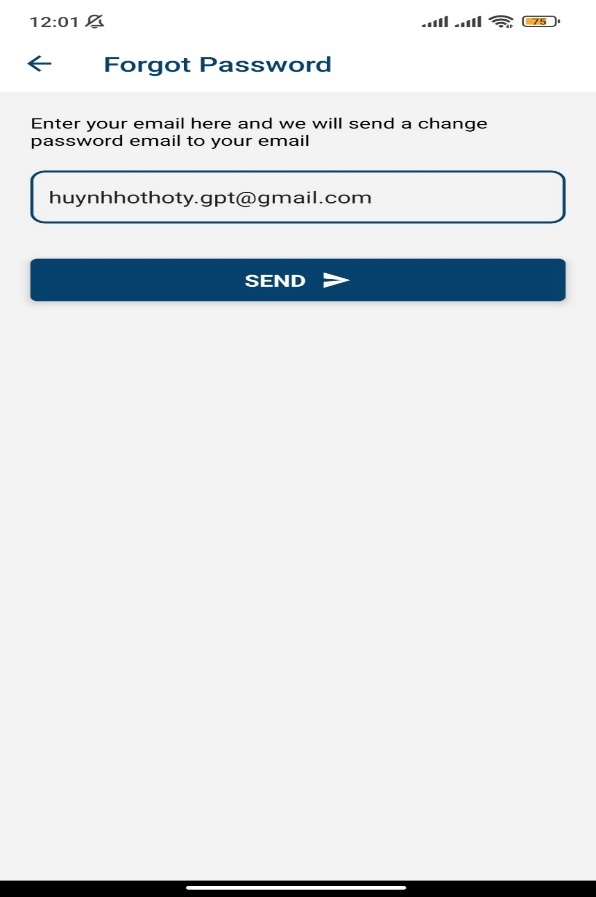
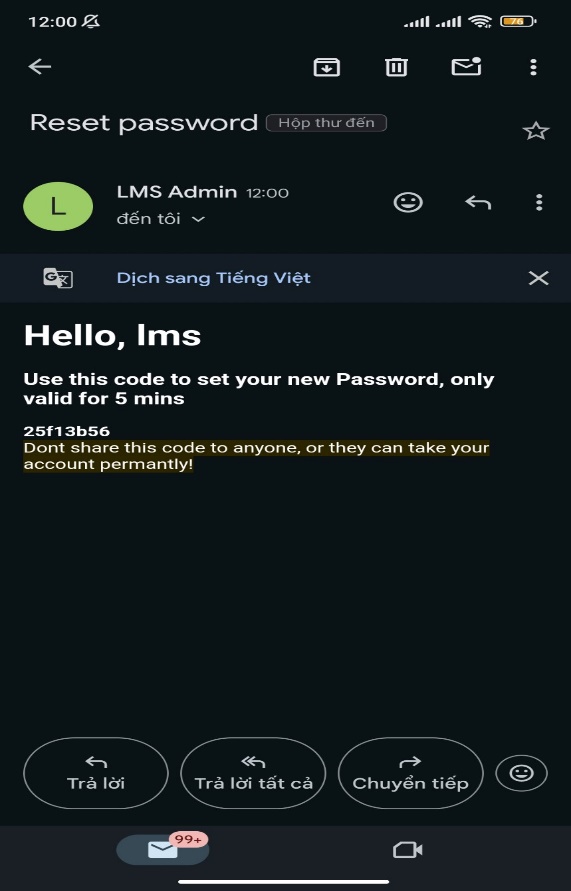
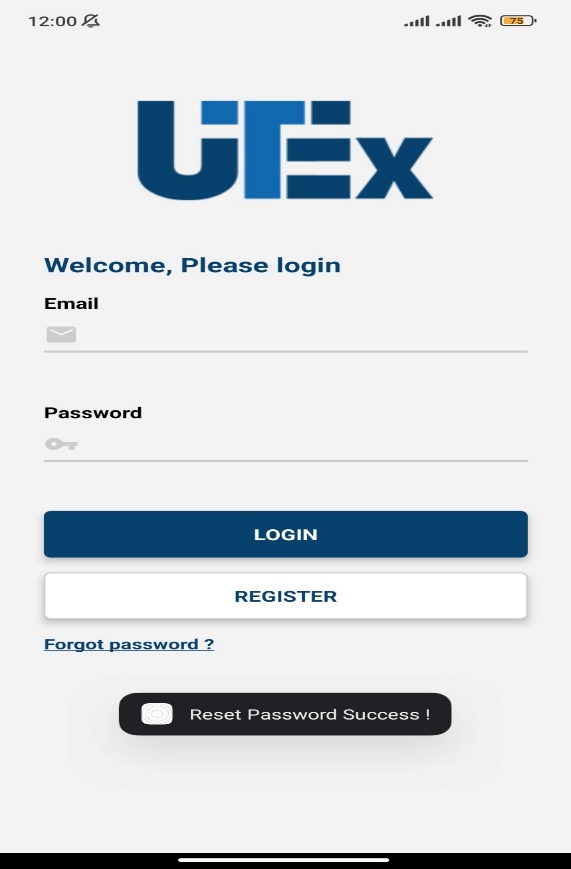
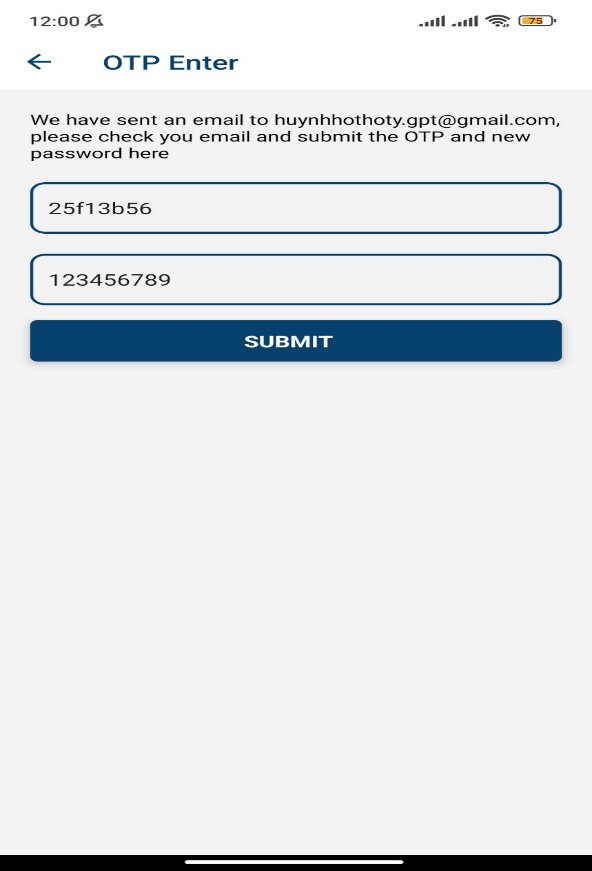


## Quên mật khẩu

- Flow event:

|  |  |
| --- | --- |
| Use case | Quên mật khẩu |
| Actor | Student, teacher |
| Pre-conditions | Vào được App và hiển thị màn hình đăng nhập. |
| Post-conditions | Đặt được mật khẩu mới |
| Basic flow | 1. Điền Email và nhấn send  2. Vào mail lấy OTP  3. Nhập OTP cùng mật khẩu mới và nhấn Submit.  4. Nếu OTP đúng, thực hiện A1, ngược lại thực hiện A2 |
| Alternative flow | A1: Nhập đúng thông tin |
|  | 1. Hiển thị thông báo đặt lại mật khẩu thành công và đưa về màn hình đăng nhập |
|  | A2: Nhập sai thông tin |
|  | 1. Hiển thị thông báo nhập sai thông tin OTP, yêu cầu nhập lại. |

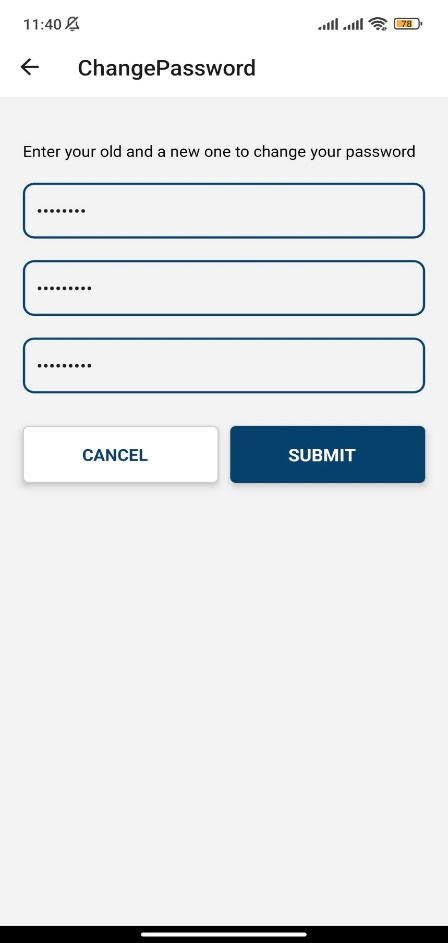
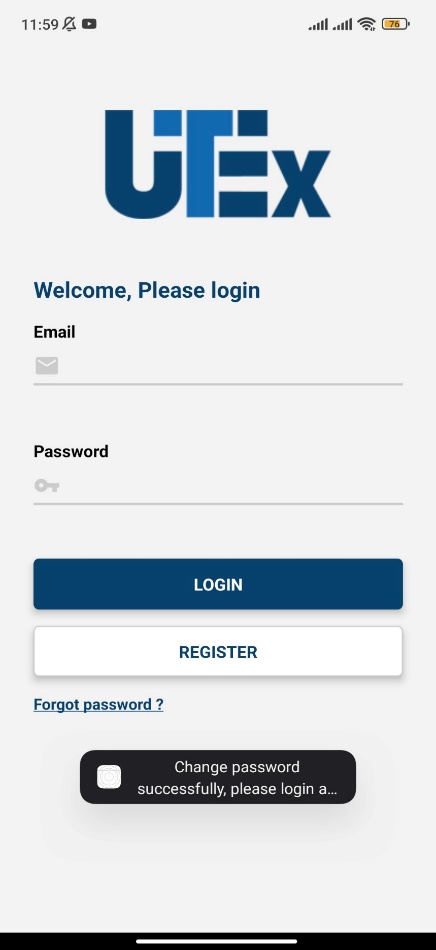
- Hình ảnh minh hoạ:

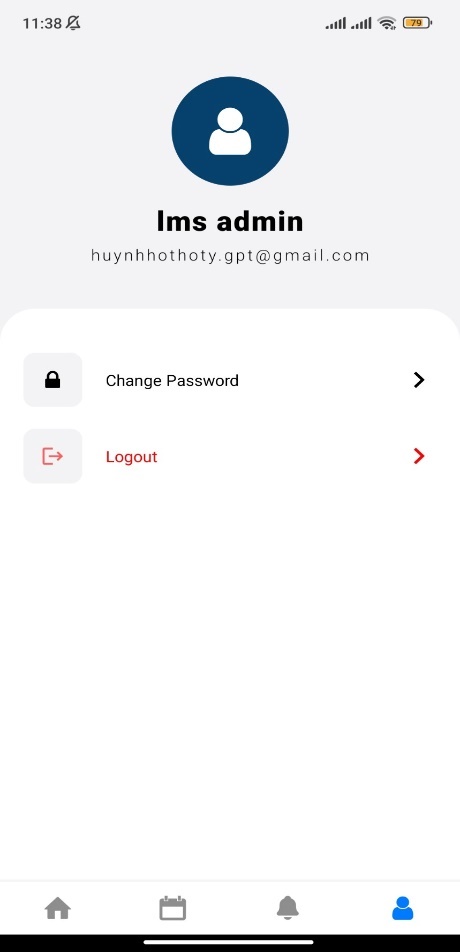


## Đổi mật khẩu

- Flow event:

|  |  |
| --- | --- |
| Use case | Đổi mật khẩu |
| Actor | Giảng viên, Sinh viên |
| Pre-conditions | Giảng viên hoặc Sinh viên đã đăng nhập được vào app. |
| Post-conditions | Đổi mật khẩu thành công. |
| Basic flow | 1. Chọn tab ***User***.  2. Nhấn ***Change Password***.  3. Nhập mật khẩu cũ, mật khẩu mới, xác nhận mật khẩu.  4. Nhấn Submit  5. Nếu mật khẩu cũ sai, mật khẩu mới khác với mật khẩu xác nhận, thực hiện A1: Nhập sai thông tin, nếu không, thực hiện tiếp bước 6.  6. Hiển thị thông báo thay đổi mật khẩu thành công. |
| Alternative flow | A1: Nhập sai thông tin |
|  | 1. Nhập lại mật khẩu cũ, mật khẩu mới, xác nhận mật khẩu.  2. Nhấn Submit |

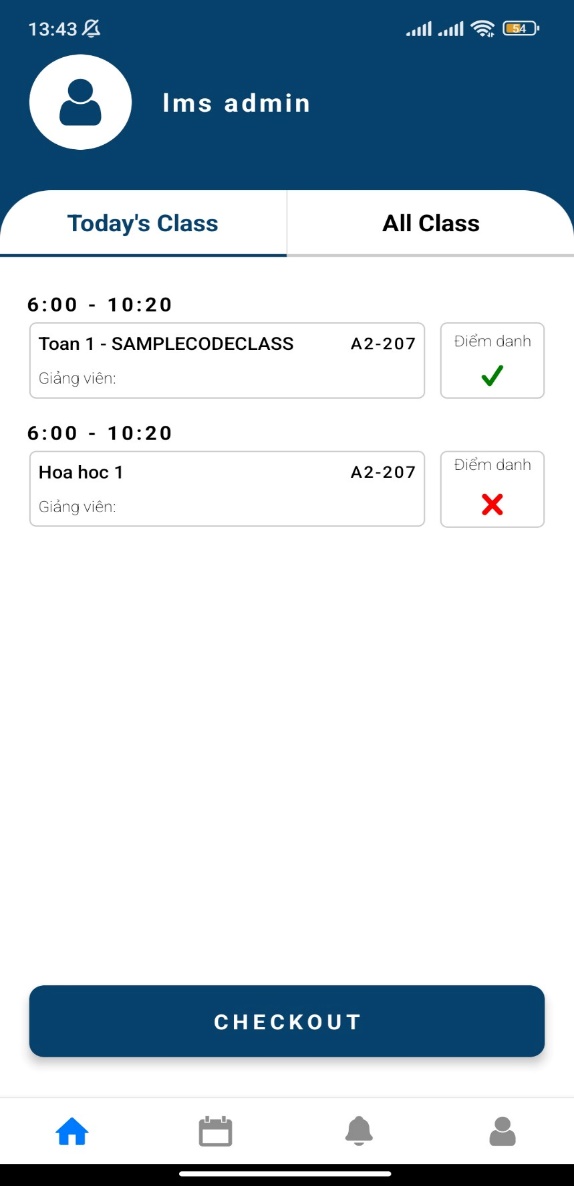
- Hình ảnh minh hoạ

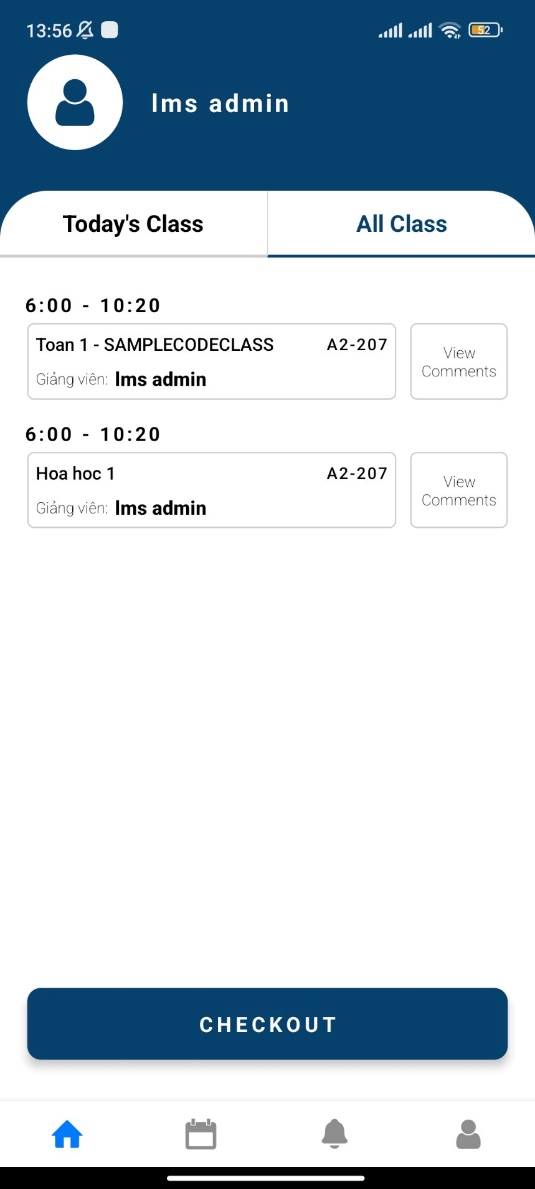


## Xem danh sách lớp (tài khoản giảng viên)

- Flow event:

|  |  |
| --- | --- |
| Use case | Xem danh sách lớp |
| Actor | Giảng viên, Sinh viên |
| Pre-conditions | Giảng viên hoặc Sinh viên đã đăng nhập được vào app. |
| Post-conditions | Xem được danh sách lớp đang dạy (giảng viên) và danh sách lớp đang học (sinh viên) |
| Basic flow | 1. Chọn tab Home.  2. Chọn tab tiêu chí lọc ( theo toàn bộ lớp hoặc chỉ các lớp của ngày hôm nay.  3. Hiển thị danh sách lớp theo tiêu chí lọc |

- Hình ảnh minh hoạ:

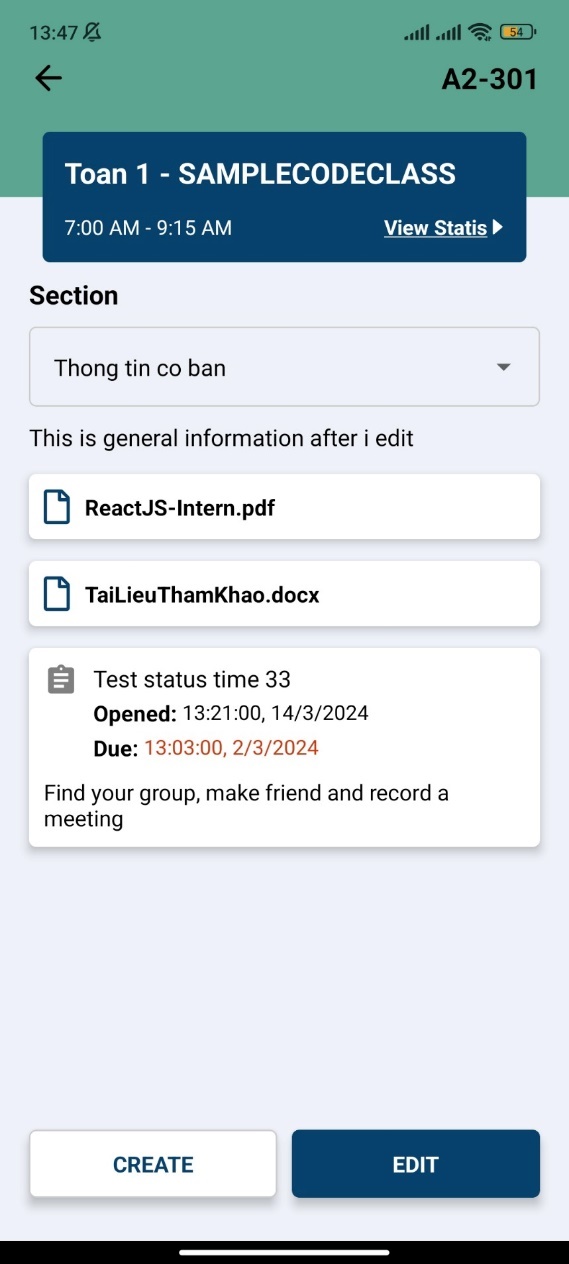


## Xem chi tiết lớp

- Flow event:

|  |  |
| --- | --- |
| Use case | Xem thông tin chi tiết lớp. |
| Actor | Giảng viên, sinh viên |
| Pre-conditions | Người dùng đăng nhập được vào app và đã xem được danh sách lớp |
| Post-conditions | xem được thông tin chi tiết lớp. |
| Basic flow | 1. Từ danh sách lớp, nhấn vào lớp cần xem thông tin  2. Từ dropdown, chọn 1 trong các hoạt động của lớp để xem (mặc định sẽ là hoạt động đầu tiên)  3. Xem được thông tin hoạt động đã chọn. |

- Hình ảnh minh hoạ:

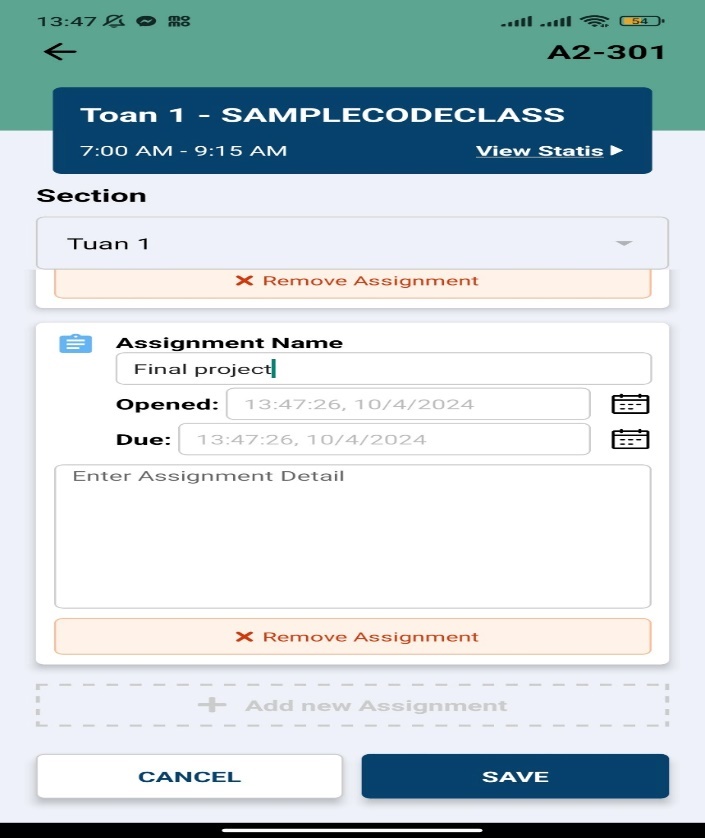
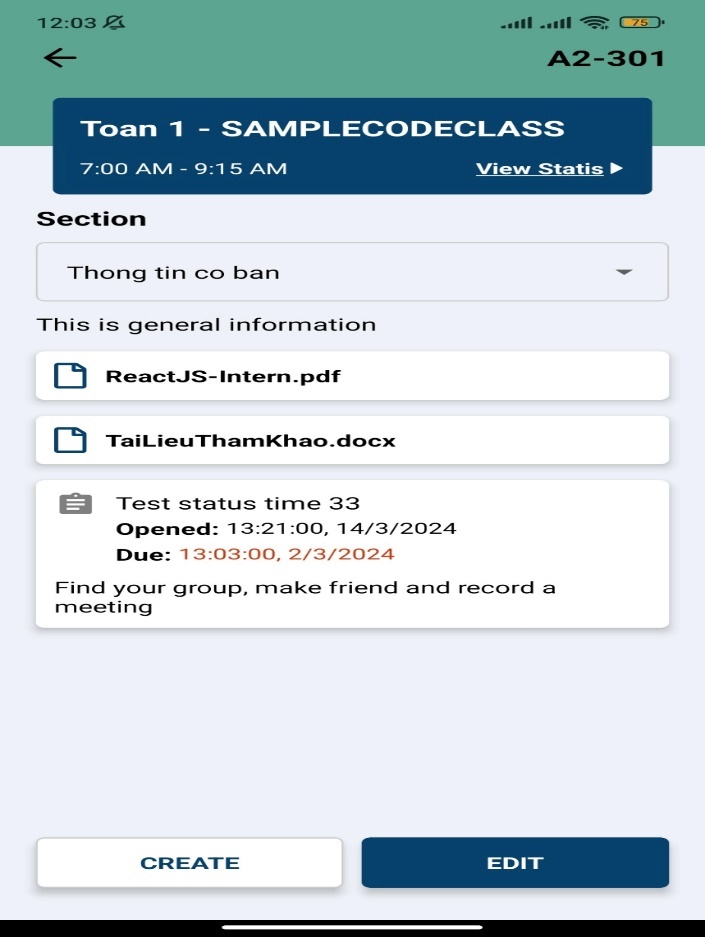
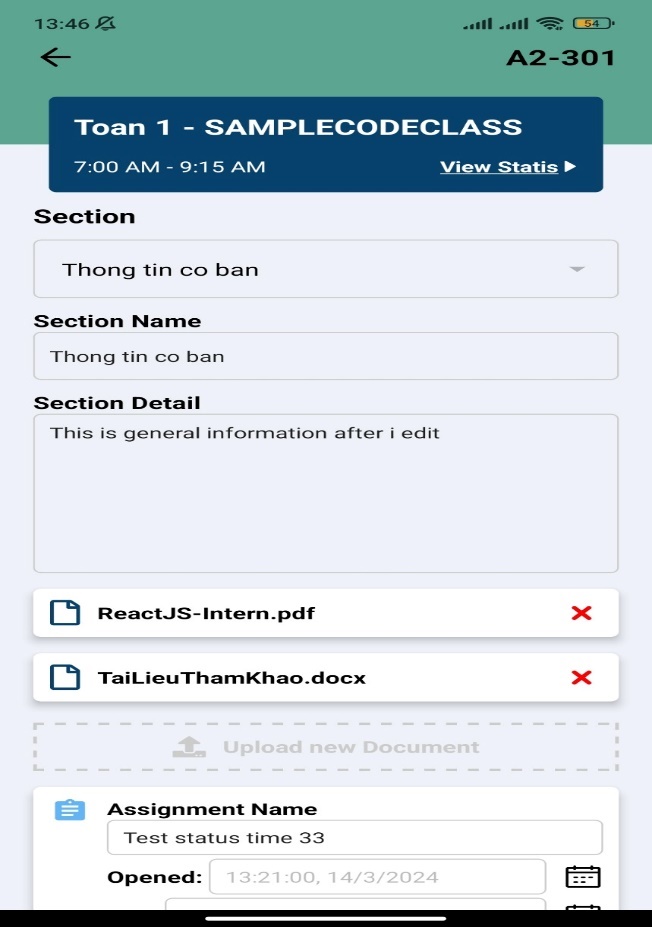
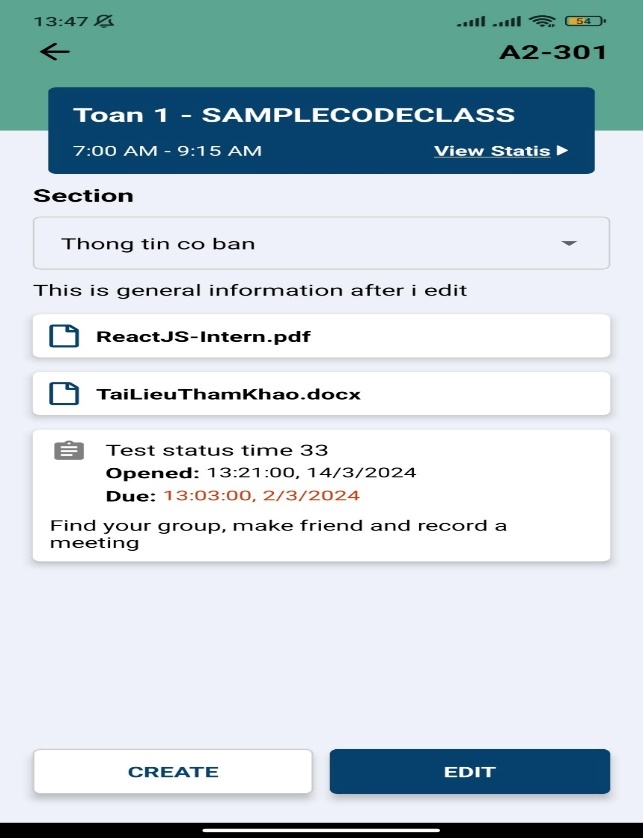


## Chỉnh sửa lớp (tài khoản giảng viên)

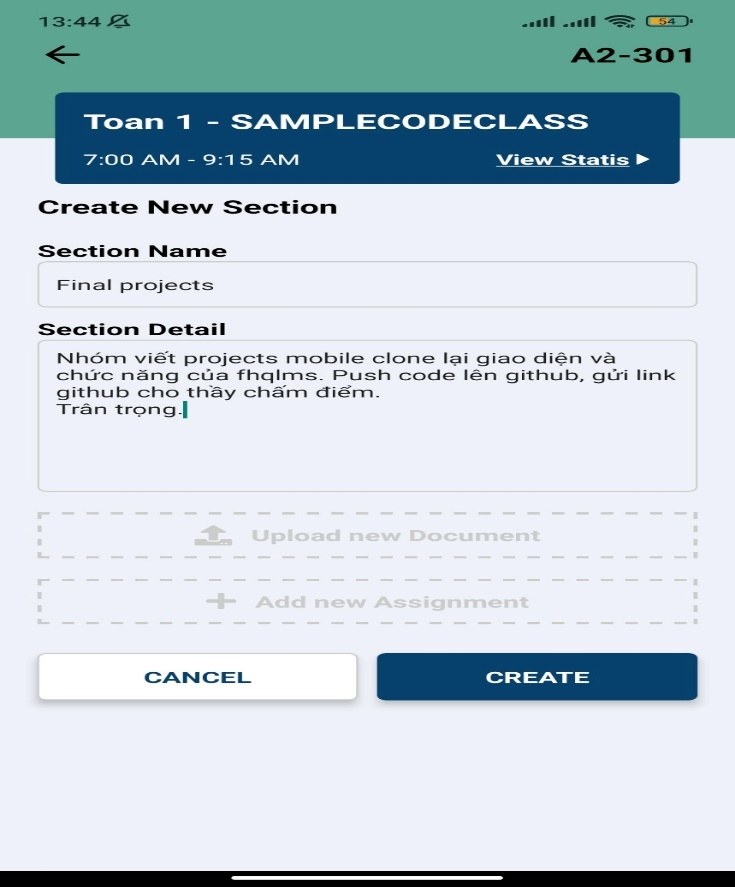
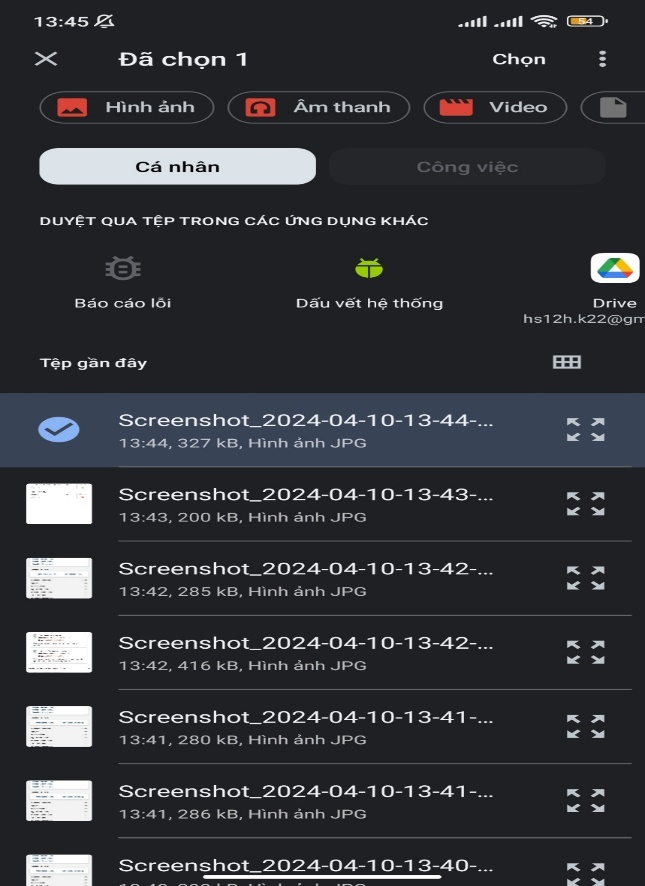
- Flow event:

|  |  |
| --- | --- |
| Use case | Quản lý (chỉnh sửa) thông tin lớp học |
| Actor | Giảng viên |
| Pre-conditions | Giảng viên đăng nhập được vào app và có role là Giảng viên. |
| Post-conditions | Chỉnh sửa thông tin của lớp học |
| Basic flow | 1. Chọn tab Home  2. Chọn một lớp từ danh sách các lớp và nhấn vào lớp cần chỉnh sửa.  3. Nếu muốn tạo hoạt động mới (***Section***), thực hiện A1.  4. Nếu cần chỉnh sửa hoạt động, thực hiện A3.  5. Nếu cần xoá hoạt động, thực hiện A4 |
| Alternative flow | A1: Tạo hoạt động mới. |
|  | 1. Nhấn nút tạo hoạt động mới.  2. Điền tên hoạt động (***Section Name***).  3. Điền nội dung hoạt động (***Section Detail***).  4. Nếu cần upload tài nguyên, thực hiện A2 từ bước 2.  5. Nếu là bài tập, thực hiện A5.  6. Nhấn OK để tạo hoạt động mới.  7. Quay lại flow gốc từ sau bước thực hiện A1 |
|  | A2: Upload tài nguyên |
|  | 1. Chọn một hoạt động (Từ dropdown).  2. Điền tên cho tài nguyên  3. Nhấn nút ***Upload new Document***.  4. Chọn loại tài nguyên (ảnh, văn bản, tệp tin,…)  5. Chọn tài nguyên từ máy hoặc điền vào.  6. Nhấn OK để thực hiện upload.  7. Quay lại flow gốc từ sau bước thực hiện A2 |
|  | A3: Chỉnh sửa hoạt động |
|  | 1. Chọn một hoạt động (Từ dropdown).  2. Nhấn ***Edit***  3. Chỉnh sửa nội dung hoạt động (***Section Name, Section Detail***).  4. Nếu cần chỉnh sửa tài nguyên upload, thực hiện A2 từ bước 2  5. Nhấn ***Save*** để xác nhận thay đổi.  6. Quay lại flow gốc từ sau bước thực hiện A3 |
|  | A4: Xoá hoạt động |
|  | 1. Chọn một hoạt động (Từ dropdown).  2. Nhấn vào nút xoá Delete bên cạnh tên của hoạt động cần xoá.  3. Nhấn xác nhận để xoá hoạt động.  4. Quay lại flow gốc từ sau bước thực hiện A4 |
|  | A5: Tạo bài tập |
|  | 1. Nhấn nút ***Add new Assignment***.  2. Điền tên bài tập  3. Chọn loại file hợp lệ để nộp.  4. Đặt thời gian deadline.  5. Nhấn OK để tạo bài tập.  6. Quay lại flow gốc từ sau bước thực hiện A5 |

- Hình ảnh minh hoạ:



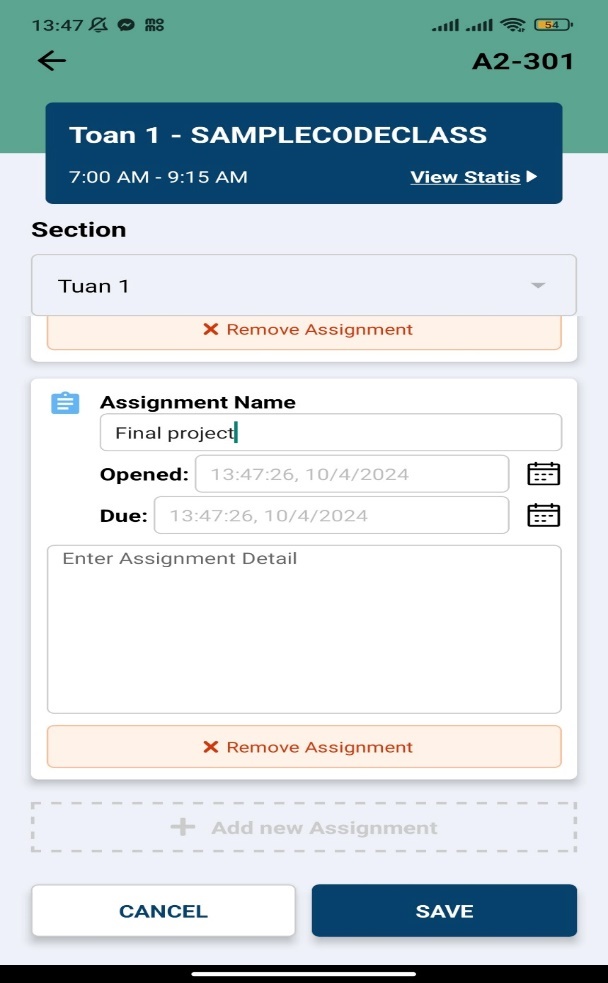
Hoặc cũng có thể tạo một hoạt động (Section) mới bằng cách nhấn CREATE, điền các thông tin cần thiết như tên và nội dụng của hoạt động, có thể đính kèm tài liệu nếu cần.



## Tạo bài tập (tài khoản giảng viên)

- Flow event: bao gồm bên trong flow event của ***mục 3.7***

- Hình ảnh minh hoạ: Tương tự như chỉnh sửa lớp, mỗi hoạt động có thể có một hoặc nhiều bài tập.

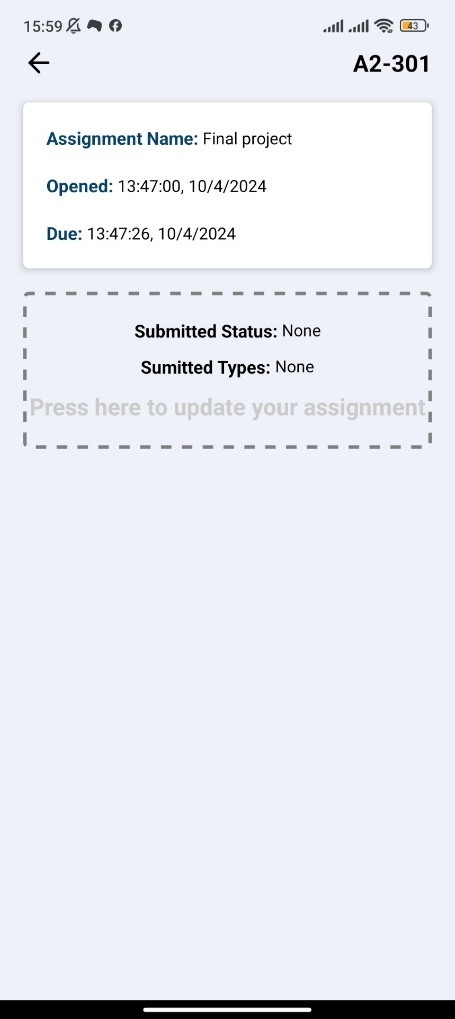
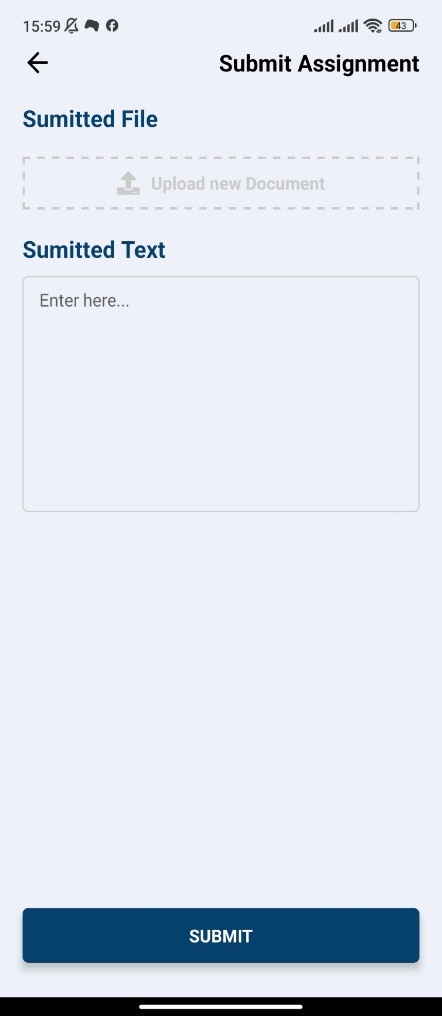
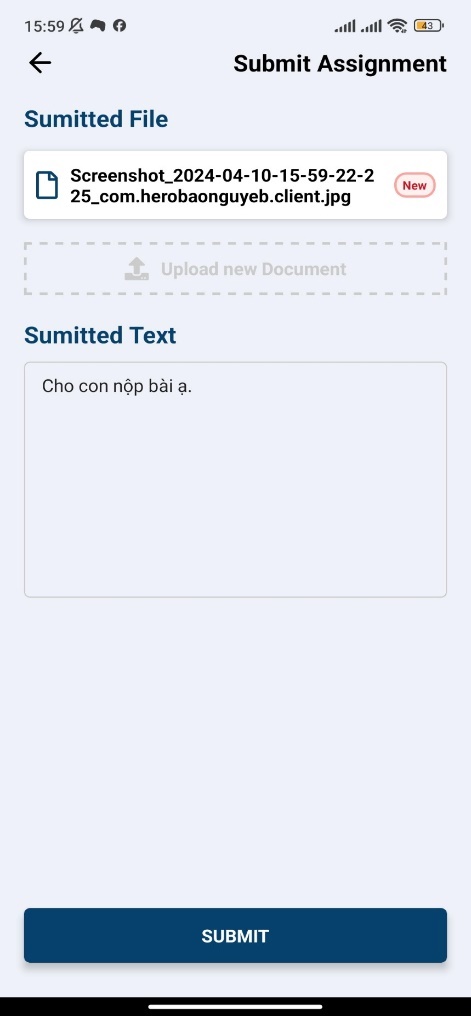


## Nộp bài tập (tài khoản sinh viên)

- Flow event:

|  |  |
| --- | --- |
| Use case | Sử dụng tài nguyên của lớp học |
| Actor | Sinh viên |
| Pre-conditions | Sinh viên đăng nhập được vào app và có role là Sinh viên. |
| Post-conditions | Sinh viên xem/ sử dụng tài nguyên từ lớp học thành công. |
| Basic flow | 1. Chọn tab ***Home***.  2. Chọn một lớp từ danh sách các lớp đang theo học và nhấn lớp muốn thao tác.  3. Chọn 1 hoạt động từ dropdown  4. Xem thông tin chi tiết lớp đã chọn.  5. Nếu muốn download tài nguyên, thực hiện A1  6. Nếu muốn nộp bài tập, thực hiện A2 |
| Alternative flow | A1: Download tài nguyên. |
|  | 1. Tìm tài nguyên cần download.  2. Nhấn chọn file đó để download về thiết bị.  3. Quay lại flow gốc từ sau bước thực hiện A1 |
|  | A2: Nộp bài tập |
|  | 1. tìm bài tập cần nộp.  2. Nhấn nút upload file từ thiết bị hoặc/và điền nội dung vào ô Nội dung  3. Nếu sai loại file quy định, thực hiện lại bước 2  4. Nhấn Submit để nộp bài tập. |

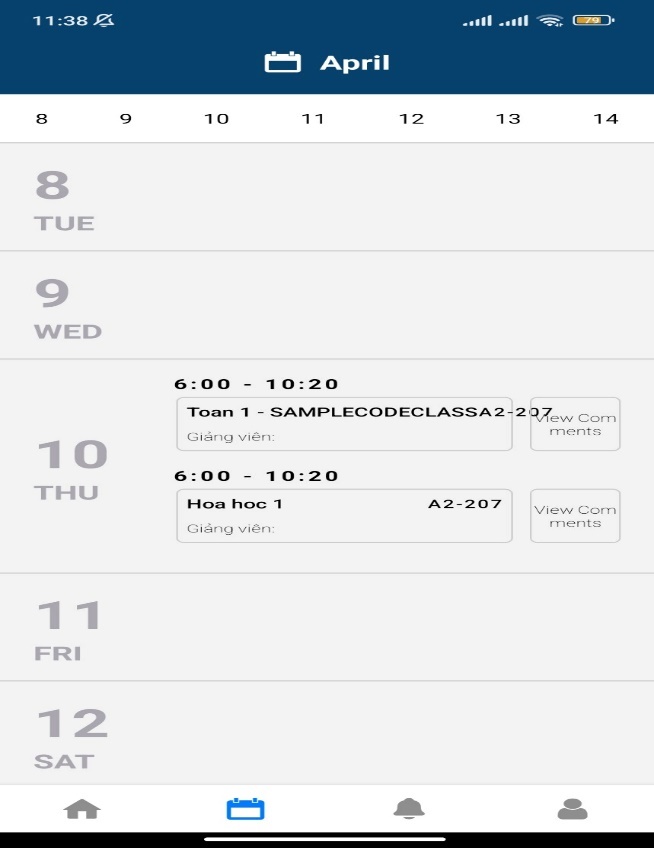
- Hình ảnh minh hoạ: Nhấn vào một bài tập bất kì của một hoạt động, điền nội dung nộp hoặc file nộp vào, sau đó nhấn SUBMIT để hoàn thành

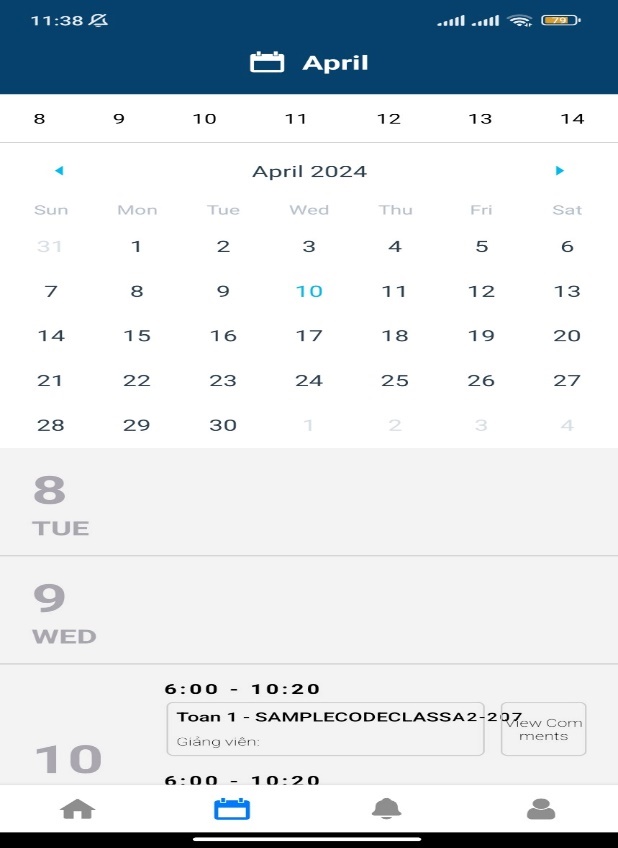


## Xem lịch học/giảng dạy

- Flow event

|  |  |
| --- | --- |
| Use case | Xem lịch học/dạy trong ngày, tuần. |
| Actor | Sinh viên, giảng viên |
| Pre-conditions | Sinh viên, giảng viên đăng nhập được vào app và có role tương ứng. |
| Post-conditions | Sinh viên/giảng viên xem được thông tin về lịch học/dạy. |
| Basic flow | 1. Chọn tab ***Calendar***.  2. Chọn chế độ xem (theo ngày trong tuần hoặc ngày cụ thể, mặc định hiển thị ngày trong tuần).  3. Xem thông tin về lịch học/dạy. |

- Hình ảnh minh hoạ: Chọn một ngày để xem lịch học/giảng dạy

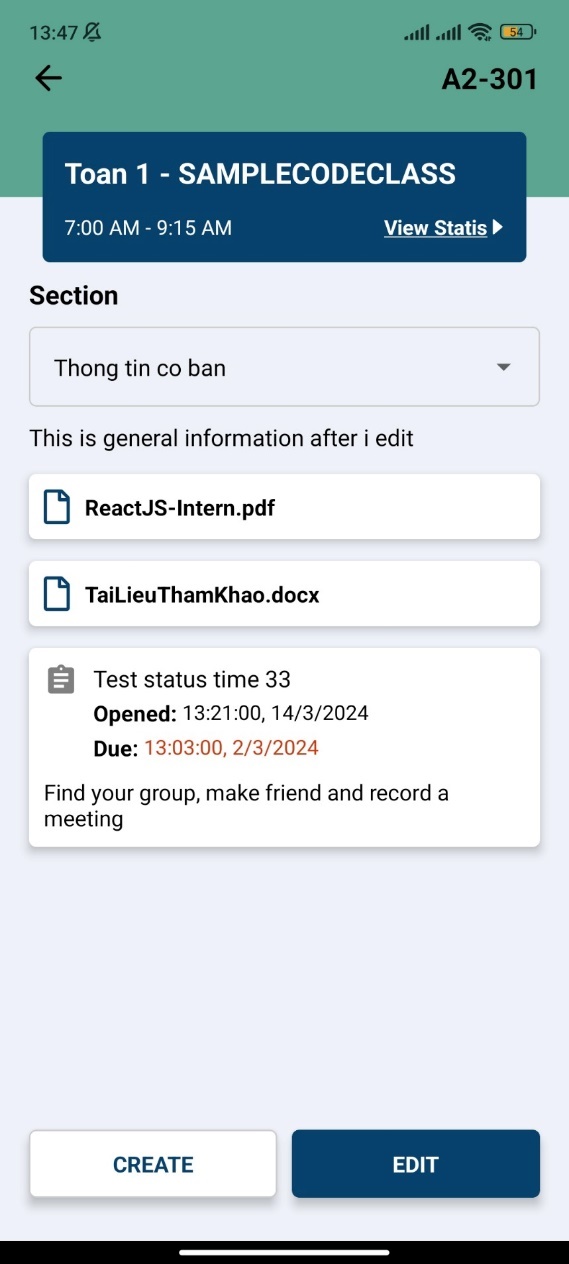
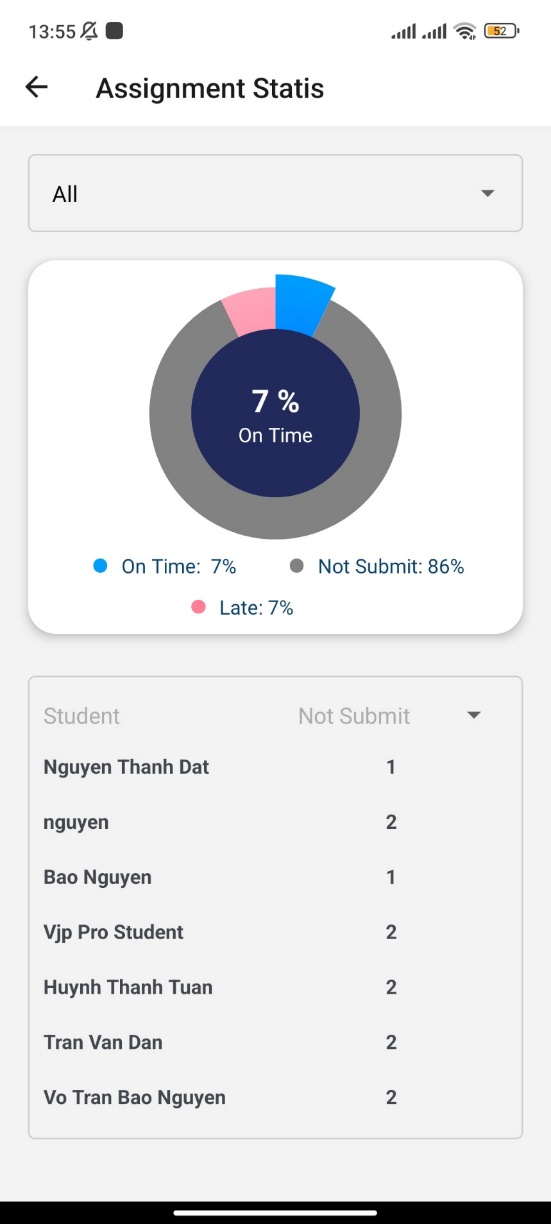


## Xem thống kê tình hình học tập của sinh viên (tài khoản giảng viên)

- Flow event:

|  |  |
| --- | --- |
| Use case | Xem thống kê tình hình học tập |
| Actor | Giảng viên |
| Pre-conditions | Giảng viên đăng nhập được vào app và có role là giảng viên. |
| Post-conditions | Giảng viên xem được thông tin về thống kê học tập. |
| Basic flow | 1. Chọn tab Home, hiển thị danh sách lớp  2. Chọn vào 1 lớp bằng cách nhấn vào lớp đó.  3. Nhấn View Statis  4. Hiển thị thông tin thống kê về tình hình học tập của lớp đã chọn |

- Hình ảnh minh hoạ: lớp có 7 người, có ***Nguyen Thanh Dat*** và ***Bao Nguyen*** không nộp 1 bài tập, các thành viên còn lại không nộp 2 bài tập.

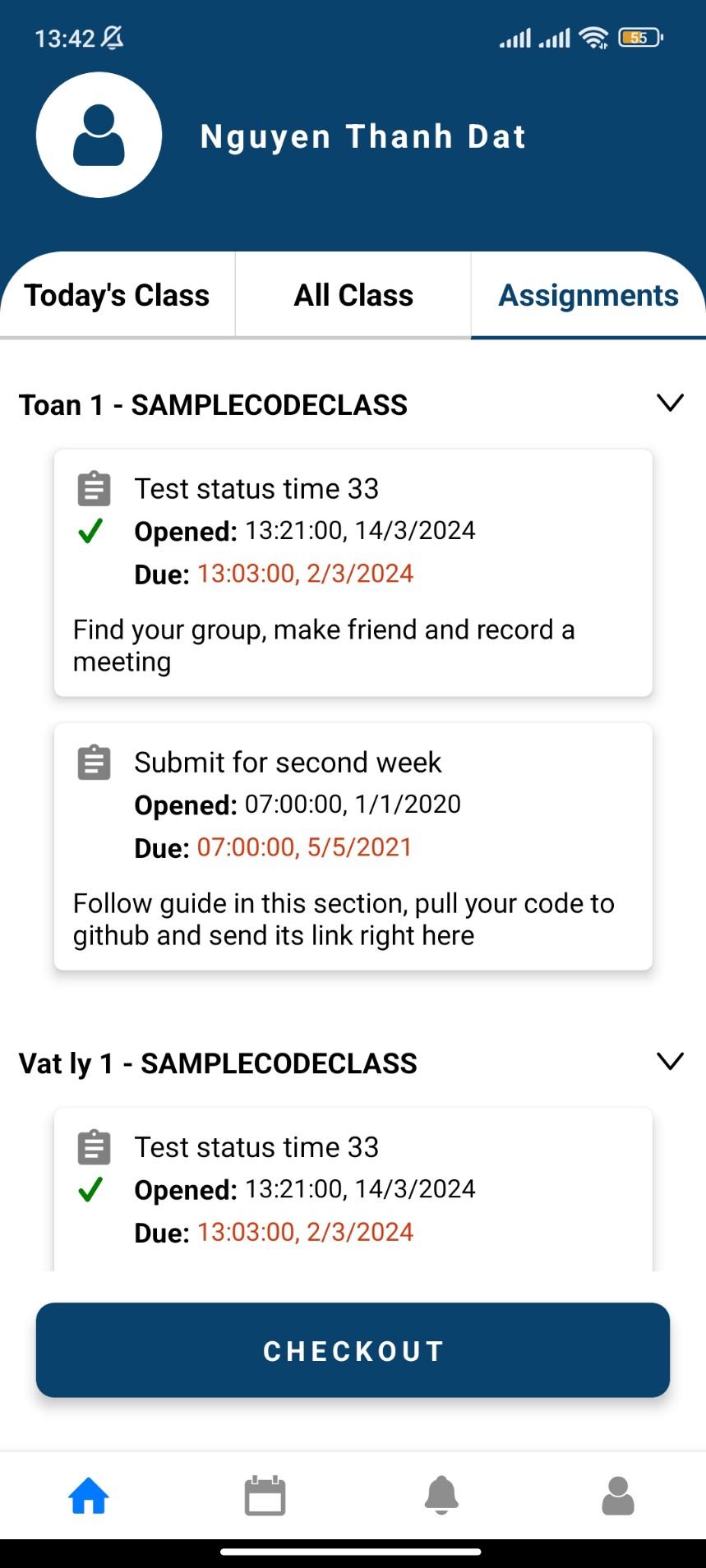


## Xem tình hình học tập của bản thân(tài khoản sinh viên)

- Flow event:

|  |  |
| --- | --- |
| Use case | Xem thống kê tình hình học tập của bản thân |
| Actor | Sinh viên |
| Pre-conditions | Sinh viên đăng nhập được vào app và có role là sinhviên. |
| Post-conditions | Sinh viên xem được thông tin về thống kê học tập. |
| Basic flow | 1. Chọn tab Home  2. chọn Assignment  3. Chọn vào 1 lớp bằng cách nhấn vào lớp đó.  4. Hiển thị thông tin thống kê về tình hình học tập của lớp đã chọn của bản thân. |

- Hình ảnh minh hoạ:



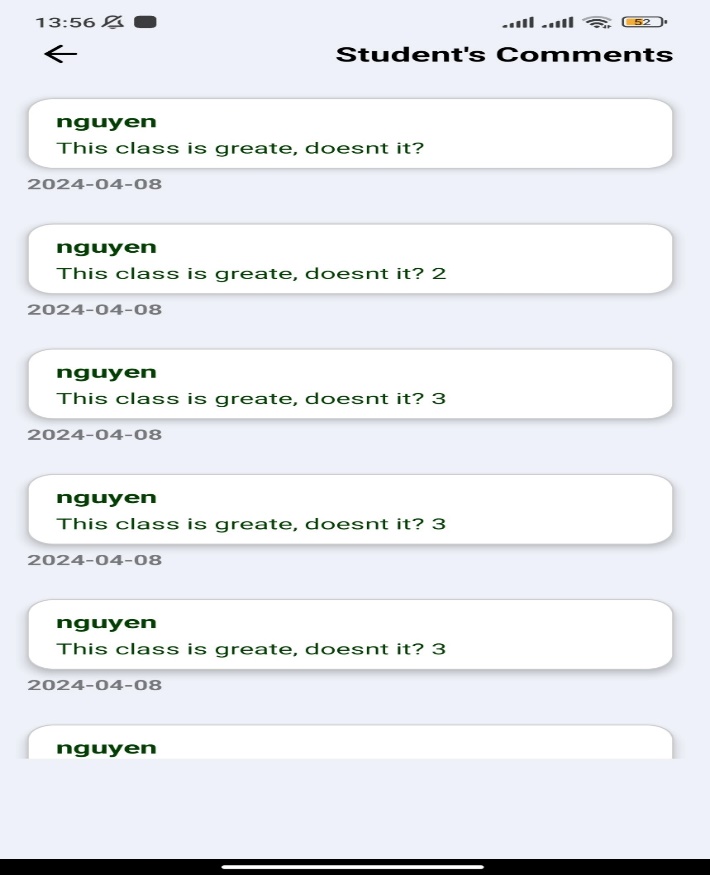
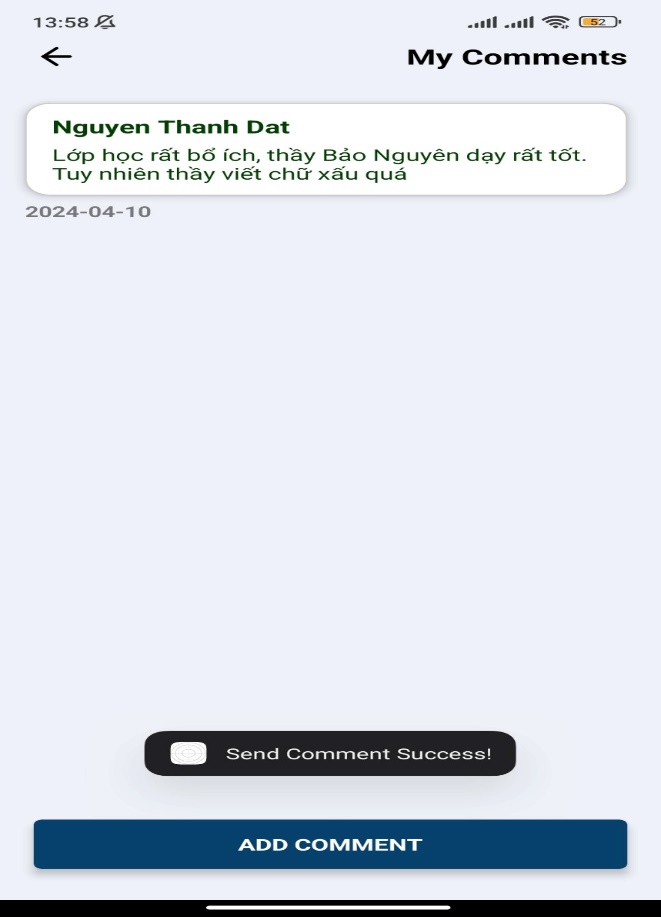
## Nhận xét, đánh giá lớp học

- Flow event:

|  |  |
| --- | --- |
| Use case | Nhận xét đánh giá lớp học |
| Actor | Sinh viên |
| Pre-conditions | Sinh viên đăng nhập được vào app và có role là sinh viên. |
| Post-conditions | Sinh viên nhận xét đánh giá lớp học thành công. |
| Basic flow | 1. Chọn tab Home, chọn filter All Class, hiển thị danh sách lớp  2. Chọn 1 lớp, nhấn vào nút comment bên cạnh lớp đó.  3. Nhấn Add comment để thêm nhận xét mới.  4. Điền nội dung nhận xét vào ô textbox, sau đó nhấn send. |

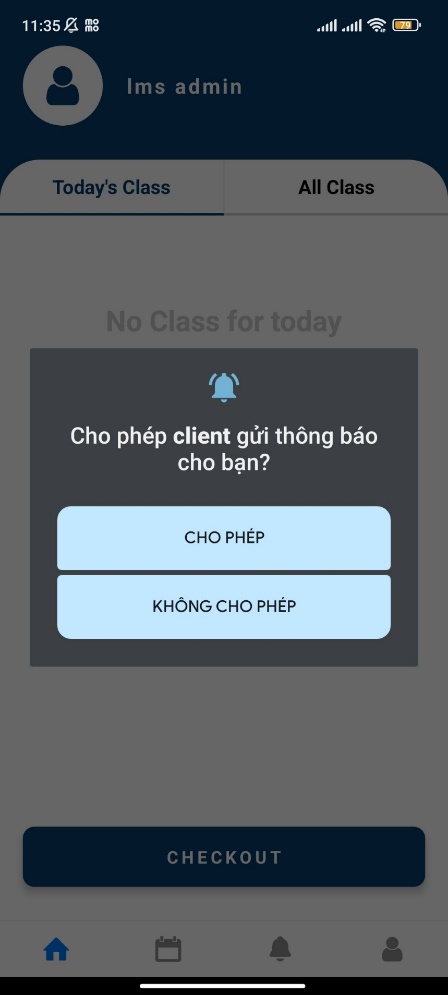
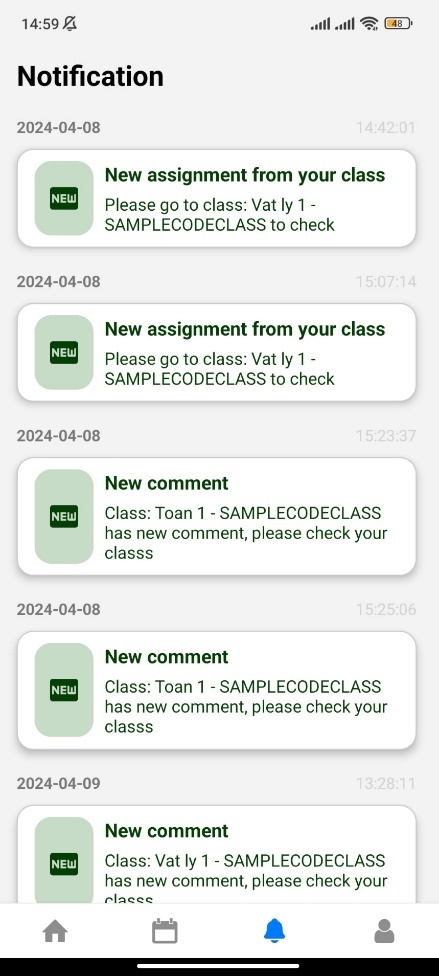
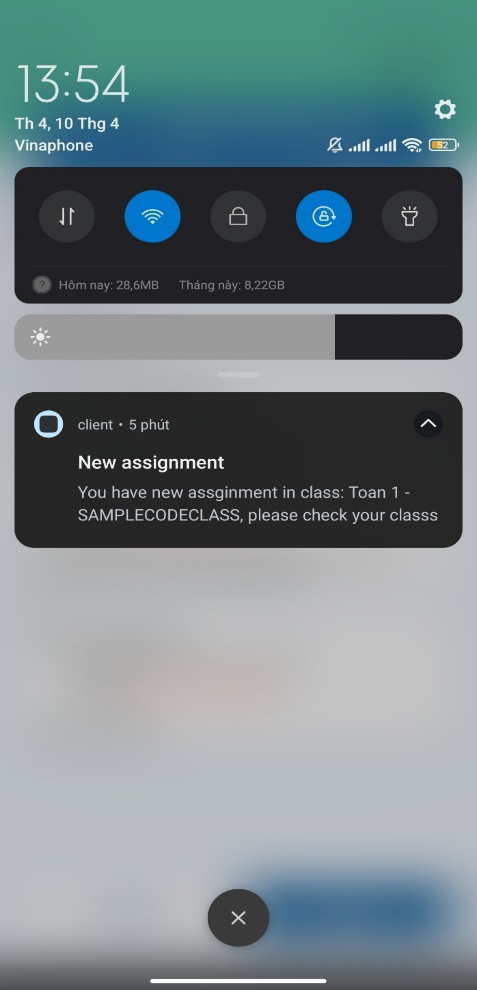
|  |  |
| --- | --- |
| Use case | Xem nhận xét đánh giá lớp học |
| Actor | Giảng viên |
| Pre-conditions | Giảng viên đăng nhập được vào app và có role là giảng viên. |
| Post-conditions | Giảng viên xem nhận xét đánh giá lớp học thành công. |
| Basic flow | 1. Chọn tab Home, chọn filter All Class, hiển thị danh sách lớp  2. Chọn 1 lớp, nhấn vào nút comment bên cạnh lớp đó.  3. Hiển thị danh sách các bình luận nhận xét của lớp học đó. |

- Hình ảnh minh hoạ: Ở tài khoản giảng viên, có thể coi toàn bộ đánh giá của sinh viên. Ở tài khoản sinh viên, có thể thêm nhận xét mới cho lớp học



## Nhận thông báo

Lúc vào App sẽ có thông báo yêu cầu cấp quyền, nhấn “Cho phép” để Ứng dụng có quyền gửi thông báo. Khi có một bài tập mới được khởi tạo ở lớp nào đó mà người dùng tham gia, sẽ có một thông báo được gửi đến thiết bị người dùng, đồng thời người dùng cũng có thể coi toàn bộ các thông báo ở trong ứng dụng.



Ngoài ra, khi một học sinh nhận xét lớp học, giảng viên phụ trách lớp đó cũng sẽ nhận được thông báo.

## Điểm danh

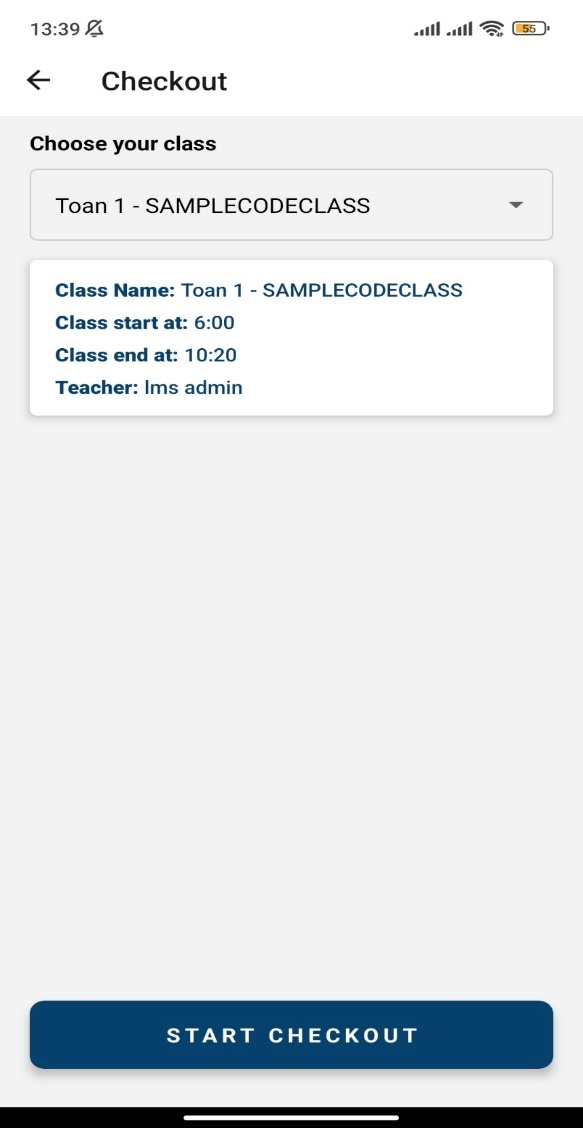
- Flow event:

|  |  |
| --- | --- |
| Use case | Tạo sự kiện điểm danh |
| Actor | Giảng viên |
| Pre-conditions | Giảng viên đăng nhập được vào app và có role là Giảng viên. |
| Post-conditions | Tạo sự kiện để cho các sinh viên có thể điểm danh. |
| Basic flow | 1. Chọn tab ***Home***, nhấn ***Checkout***.  2. Chọn một lớp từ danh sách lớp phụ trách (từ dropdown) và nhấn ***Checkout***.  3. Nhấn tạo sự kiện điểm danh. Thông báo yêu cầu quyền truy cập vị trí hiện lên.  4. Nhấn đồng ý, nếu nhấn không đồng ý thực hiện E1.  5. Chọn thời gian cho phép điểm danh (bắt đầu và kết thúc).  6. Nhấn OK để tạo sự kiện |
| Exception flow | E1: Không cho phép truy cập vị trí |
|  | 1. App hiển thị thông báo vui lòng cung cấp quyền truy cập vị trí để thực hiện hành động. |

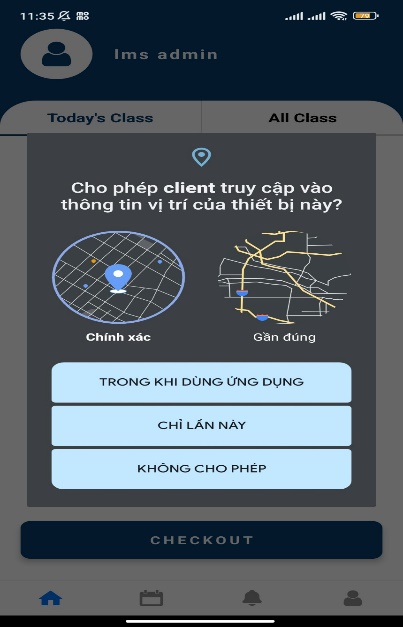
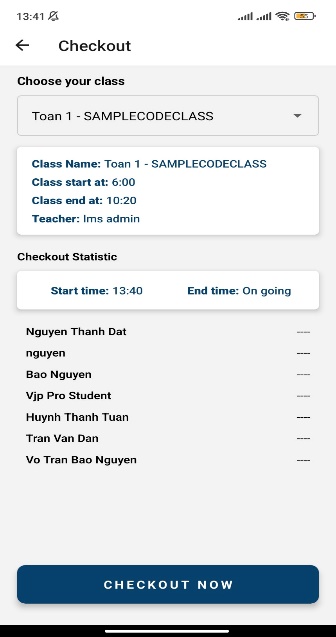
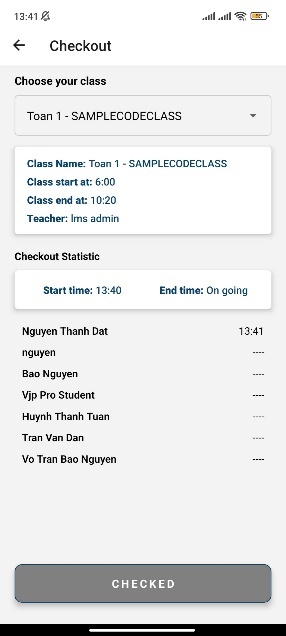
|  |  |
| --- | --- |
| Use case | Điểm danh |
| Actor | Sinh viên |
| Pre-conditions | Sinh viên đăng nhập được vào app và có role là Sinh viên. |
| Post-conditions | Điểm danh thành công. |
| Basic flow | 1. Chọn tab ***Home***, nhấn ***Checkout***.  2. Chọn một lớp từ danh sách lớp phụ trách và nhấn ***Checkout***.  3. Nhấn Điểm danh. Thông báo yêu cầu quyền truy cập vị trí hiện lên.  4. Nhấn đồng ý, nếu nhấn không đồng ý thực hiện E1.  5. Nhấn OK để điểm danh. |
| Exception flow | E1: Không cho phép truy cập vị trí |
|  | 1. App hiển thị thông báo vui lòng cung cấp quyền truy cập vị trí để thực hiện hành động. |

- Hình ảnh minh hoạ: Quy trình điểm danh như sau:

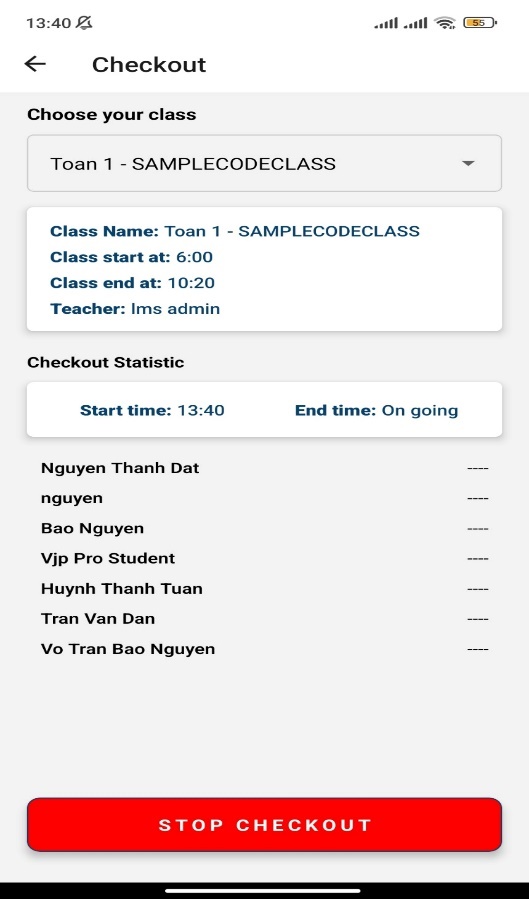
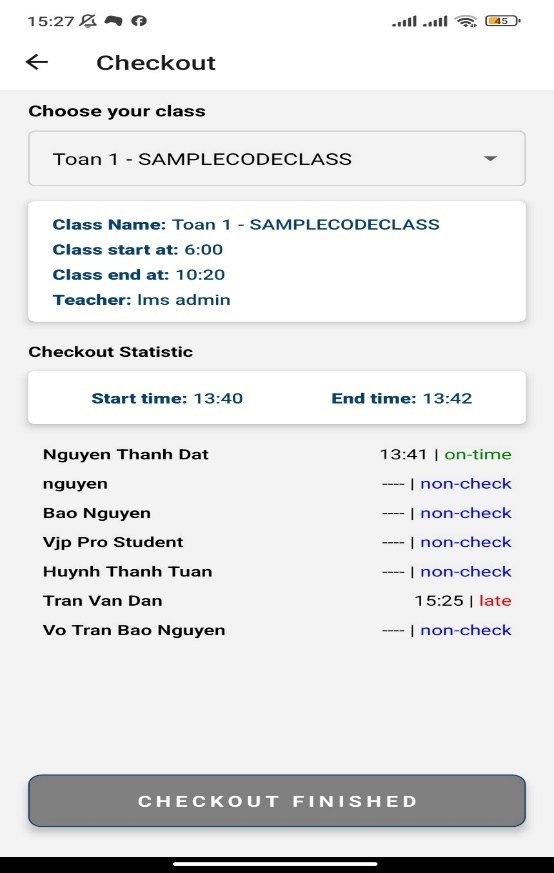
+ Giảng viên chọn Checkout ở màn hình chính, chọn một lớp sau đó nhấn START CHECKOUT để bắt đầu phiên điểm danh.



+ Sinh viên vào lớp tương ứng để điểm danh, lưu ý cần cấp quyền vị trí thì mới có thể thực hiện thao tác này.



+ Giảng viên có thể vào lớp tương ứng để kết thúc phiên điểm danh bất kì lúc nào, những học sinh điểm danh sau lúc này sẽ có trạng thái “trễ”, nếu vị trí khi điểm danh quá xa so với giảng viên sẽ bị trạng thái “quá xa”, còn lại sẽ là những người điểm danh đúng (nếu có điểm danh) và những người không có điểm danh.



# Kết luận

## Đánh giá

* Hoàn thành được các chức năng cơ bản cần thiết của một ứng dụng di động dùng để quản lý quá trình học tập lẫn giảng dạy của sinh viên và giảng viên.
* Giao diện là sự kết hợp giữa sự tối giản mang lại cảm giác thoải mái, nâng cao trải nghiệm người dùng và sự chuyên nghiệp trong phong cách thiết kế phù hợp với không khí lớp học, khơi gợi sự tò mò và hứng thú học tập ở trẻ.

## Định hướng phát triển

Thêm các chức năng mở rộng thêm cho ứng dụng như:

* Thêm role Phụ huynh, có thể theo dõi tình hình học tập của trẻ.
* Thông báo và tích hợp khả năng thanh toán học phí online bằng app.
* Ngoài các bài tập tự luận, giảng viên có thể tạo một bài trắc nghiệm trực tuyến trên lớp học.