

ĐẠI HỌC QUỐC GIA TP.HCM
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA CÔNG NGHỆ PHẦN MỀM



BÁO CÁO ĐỒ ÁN MÔN HỌC



LẬP TRÌNH TRÊN THIẾT BỊ DI ĐỘNG

**Đề tài: PHÁT TRIỂN ỨNG DỤNG HỖ TRỢ TƯƠNG TÁC
VÀ ĐIỂM DANH TRONG LỚP HỌC – CHECKMEIN**

Giảng viên hướng dẫn:

ThS. Huỳnh Tuấn Anh

Nhóm thực hiện: CheckMeln

Nguyễn Hồng Phúc - 16520951

Nguyễn Cao Luyện - 16520713

Lớp SE346.K11

TPHCM, Tháng 12 năm 2019

Lời cảm ơn

Trong quá trình học tập môn Lập trình trên thiết bị di động, chúng em đã học hỏi được rất nhiều kiến thức về mặt lý thuyết, cũng như các phương pháp thực hành thực tế trong việc xây dựng và phát triển phần mềm, để có thể phát triển thành công một sản phẩm như ngày hôm nay.

Chúng em xin gửi lời cảm ơn chân thành đến thầy **Huỳnh Tuấn Anh** đã vô cùng tận tình trong việc hướng dẫn và truyền đạt nội dung môn học đến với chúng em.

Trong quá trình làm đồ án khó tránh khỏi những sai sót. Chúng em rất mong nhận được sự góp ý của thầy để có thể hoàn thiện đồ án tốt hơn nữa.

Thành phố Hồ Chí Minh, tháng 12 năm 2019

Mục lục :

Nội dung

Chương 1 : Tổng quan.	7
1.1. Hiện trạng :.....	7
1.2. Mục đích và yêu cầu đề tài :.....	7
1.2.1. Mục đích :	7
1.2.2. Yêu cầu :	8
1.2.3. Đối tượng người dùng :	8
1.3. Phương pháp và công nghệ sử dụng:	8
1.3.1. Phương pháp:	8
1.3.2. Công nghệ:.....	8
Chương 2 : Đặc tả hệ thống.....	10
2.1. Thiết kế thành phần xử lý:.....	10
2.1.1. Danh sách các actor.....	11
2.1.2 Danh sách các usecase	11
2.1.3 Đặc tả use case:.....	12
2.2. Thiết kế thành phần dữ liệu.....	26
2.2.1. Bảng person.....	26
2.2.2. Bảng Event	27
2.2.3. Bảng Registration	27
2.2.4. Bảng Attendance.....	28
2.2.5. Bảng Account	28
2.2.6. Bảng Organization.....	29
2.2.7. Bảng Collaborator.....	29
2.2.8. Bảng Multiple_choice_Q.....	30
2.2.9. Bảng Answer	30
2.2.10 Bảng Answ_MQ	31
2.2.11 Bảng Essay_Q.....	31
2.2.12 Bảng Answ_EQ	32
Chương 3 : Thiết kế kiến trúc.	32
3.1. Kiến trúc hệ thống.....	32
3.2. Mô tả chi tiết các thành phần trong hệ thống	34
3.2.1. Sơ đồ.....	34

3.2.2. Bảng mô tả.....	34
Chương 4 : Thiết kế giao diện.	40
4.1. Danh sách các màn hình.....	40
4.2. Mô tả chi tiết từng màn hình	42
4.2.1. Màn hình đăng nhập	42
4.2.2 Màn hình đăng ký tài khoản	43
4.2.3. Màn hình quên mật khẩu.....	44
4.2.4. Màn hình chính (trang chủ).....	45
4.2.5. Màn hình tạo mới tổ chức	46
4.2.6. Màn hình danh sách hoạt động	47
4.2.7. Màn hình tạo hoạt động	48
4.2.8. Màn hình hiển thị danh sách người tham gia.....	49
2.2.9. Màn hình nhập danh sách người tham gia	50
4.2.9. Màn hình gửi mail.....	51
4.2.10. Màn hình hiển thị danh sách câu hỏi (View người tham gia).....	52
4.2.11. Màn hình hiển thị danh sách câu hỏi (View người tổ chức).....	53
4.2.12. Màn hình thêm/ chỉnh sửa câu hỏi cho một event	54
4.2.13. Màn hình hiển thị câu trả lời của người tham gia.....	56
4.2.14. Màn hình hiển thị danh sách hoạt động mà người dùng đã đăng ký/ tham gia .57	
4.2.15. Màn hình hiển thị thông tin người dùng	57
Chương 5 : Giới thiệu các thư viện, công nghệ liên quan.	59
5.1 Tesseract.....	59
5.2. Firebase.....	60
5.2.1. Realtime Database	60
5.2.2. Firebase Authentication	60
Chương 6 : Các thành phần xử lý chính.	61
6.1. Bộ xử lý dữ liệu trung tâm	61
6.2. Bộ nhận nhận diện ký tự	61
6.3. Xử lý hình ảnh từ camera.....	63
6.4. Adaptor.....	64
6.5. Bộ xử lý logic và cập nhật giao diện.....	65
Chương 7 : Cài đặt và sử dụng.....	67
7.1. Cài đặt	67
7.2. Sử dụng	67
7.2.1. Đăng ký tài khoản	67

7.2.2	Đăng nhập	67
7.2.3	Tham gia bằng mã code	67
7.2.4	Tạo tổ chức	68
7.2.5	Tạo sự kiện.....	68
7.2.6	Tạo bộ câu hỏi.....	68
7.2.7	Xem thống kê câu trả lời.....	68
7.2.8	Xem thông tin người dùng	68
7.2.9	Gửi email	68
Chương 8 : Kết luận.....		69
8.1.	Nhận xét.....	69
8.1.1.	Kết quả đạt được	69
8.1.2.	Ưu điểm	69
8.1.3.	Hạn chế	69
8.1.4.	Hướng phát triển trong tương lai.	69

Chương 1 : Tổng quan.

1.1. Hiện trạng :

Hiện nay, trong các hoạt động giảng dạy nói chung và các hoạt động khác nói riêng trên nhà trường, chúng ta vẫn có một khoảng cách nhất định giữa người tổ chức (thông thường là Giảng viên) và người tham gia (thông thường là Sinh viên). Một tình trạng thực tế có thể nói đến hiện nay, như việc nhiều sinh viên vẫn đang gặp khó khăn trong vấn đề theo dõi quá trình giảng dạy, gây ra tình trạng không hiểu bài nhưng lại không phát biểu ý kiến trước đám đông. Điều đó đã gây ra không ít khó khăn cho người tổ chức hoạt động trong việc cập nhật nội dung, để phù hợp với người tham gia.

Ngoài ra, chúng ta vẫn còn một vấn đề thường gặp trong các hoạt động nói trên : Đó chính là vấn đề điểm danh người tham gia. Thông thường, chúng ta sử dụng máy điểm danh để lấy thông tin điểm danh của người dùng. Nhưng chính máy điểm danh vẫn còn một số hạn chế nhất định. Ví dụ như, chúng ta chưa thể quản lý việc theo dõi bài tập của sinh viên, chưa có thống kê trực tiếp được các hoạt động mà sinh viên tham gia,... Ngoài ra, máy điểm danh còn bị hạn chế về mặt số lượng, vì thế dễ gây ùn tắc khi điểm danh vào đầu giờ/cuối giờ. Cùng với đó, số hoạt động tổ chức được cùng 1 lúc cũng sẽ bị hạn chế vì nguyên nhân trên.

Áp dụng tin học hoá (thông qua việc phát triển ứng dụng) vào việc tổ chức các hoạt động giảng dạy, cũng như các hoạt động khác và thực hiện điểm danh cho sinh viên là một hướng đi tốt để giải quyết các vấn đề trên, phục vụ trực tiếp cho cả lợi ích của sinh viên, giảng viên và nhà trường.

1.2. Mục đích và yêu cầu đề tài :

1.2.1. Mục đích :

- Tạo ra một hệ thống tiện dụng kết nối người tổ chức hoạt động và người tham gia hoạt động :
 - Cơ chế đặt câu hỏi/trả lời câu hỏi : Giúp người tổ chức nắm rõ hơn mức độ theo dõi của người tham gia.
 - Cơ chế gửi mail tự động đến những người đã đăng ký/đã tham gia : Cho phép người trong 1 tổ chức có thể gửi mail nhanh chóng đến danh sách nêu trên.

- Tạo ra một hệ thống tiện dụng để người tổ chức hoạt động quản lý các hoạt động của mình :
 - Cơ chế điểm danh bằng quét thẻ sinh viên (nhận diện MSSV bằng OCR) : Thực hiện điểm danh nhanh chóng sinh viên không phụ thuộc vào máy điểm danh.
 - Cơ chế quản lý và thống kê danh sách người tham gia : Cho phép người tổ chức nhanh chóng nắm được tình hình tham gia hoạt động của tổ chức.
 - Cơ chế quản lý người trả lời câu hỏi : Cho phép người tổ chức thống kê nhanh chóng tình hình nắm nội dung của người tham gia.
- Tạo ra một hệ thống tiện dụng cho người tham gia nắm bắt hoạt động của mình (thông qua cơ chế quản lý hoạt động đã tham gia).

1.2.2. Yêu cầu :

- Ứng dụng tiện dụng, dễ dùng cho cả người sử dụng và người tổ chức. Cả người có nền tảng tin học và người không có nền tảng tin học.
- Hoạt động trong cả môi trường offline và online.
- Bảo mật cao. Không gây rò rỉ thông tin người dùng.
- Độ chính xác cao.

1.2.3. Đối tượng người dùng :

- Người tổ chức hoạt động (Host).
- Người hỗ trợ cho hoạt động (Collaborator).
- Người tham gia hoạt động (Participant).

1.3. Phương pháp và công nghệ sử dụng:

1.3.1. Phương pháp:

- Quy trình Agile (Scrum):

Nhóm phát triển phần mềm thông qua các phân đoạn lặp (sprint) kéo dài khoảng 2 tuần. Tất cả các thành viên cùng nhau làm việc từ công đoạn thu thập, phân tích yêu cầu, tạo product backlog, lên kế hoạch, thực hiện các chức năng trong mỗi sprint cho đến việc testing.

1.3.2. Công nghệ:

- Ngôn ngữ lập trình: Java

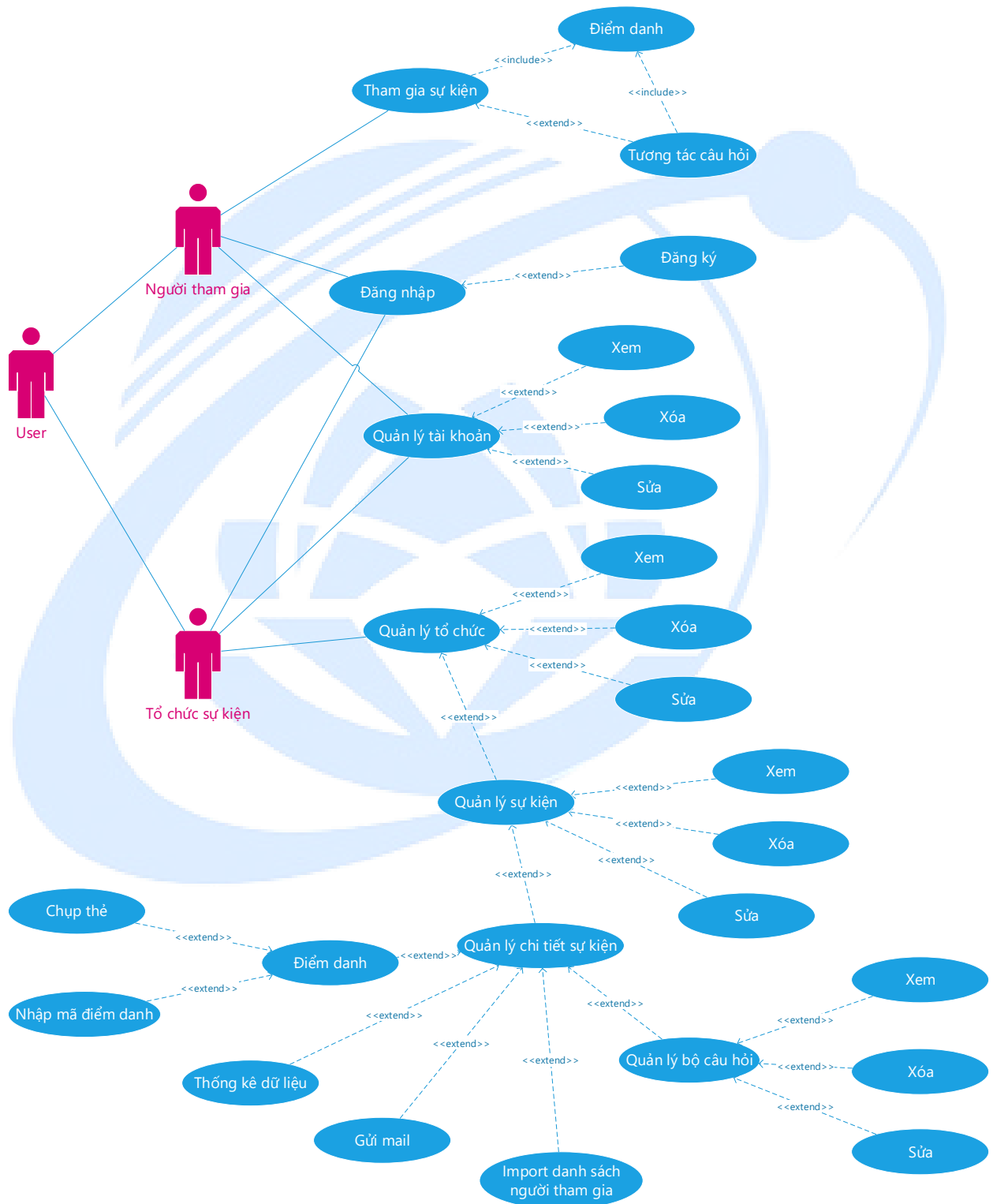
- Nền tảng: Android
- Thư viện hỗ trợ: Tesseract
- Cơ sở dữ liệu: Firebase
- Công cụ thiết kế: Visio, Figma.
- IDE: Android Studio



Chương 2 : Đặc tả hệ thống.

2.1. Thiết kế thành phần xử lý:

*** Use case:**



Sơ đồ 2-1: Use case Check Me In

2.1.1. Danh sách các actor

STT	Tên Actor	Ý nghĩa/Ghi chú
1	Người tham gia	Những người tham gia sự kiện: sinh viên, khách mời, ...
2	Tổ chức sự kiện	Tổ chức hoặc nhóm tạo sự kiện

2.1.2 Danh sách các usecase

STT	Tên Use-case	Ý nghĩa/Ghi chú
1	Đăng nhập	Người dùng đăng nhập vào hệ thống để có thể sử dụng các tính năng được cung cấp
2	Đăng ký	Người dùng đăng ký tài khoản để sử dụng, tài khoản phải đúng quy tắc và chưa tồn tại
3	Quản lý tổ chức	Người dùng tạo mới, chỉnh sửa, xóa tổ chức đã tồn tại
4	Thêm sự kiện	Người dùng tạo ra các sự kiện của tổ chức theo định dạng yêu cầu
5	Tạo bộ câu hỏi	Người dùng có thể tạo và quản lý các câu hỏi khi sự kiện bắt đầu
6	Tham gia sự kiện	Người dùng tham gia sự kiện dựa trên event-code
7	Xem thông tin User	Người dùng quản lý thông tin cá nhân và có thể tìm kiếm thông tin người dùng khác dựa trên mã số sinh viên.
8	Xem thống kê của tổ chức	
9	Gửi mail nhắc nhở/ cảm ơn	Người dùng gửi email cho những người có trong danh sách của sự kiện hiện tại
10	Quét thẻ điểm danh	Người dùng quét thẻ bằng camera điện thoại
11	Tạo danh sách từ file	Nhận dữ liệu danh sách tham gia từ file excel
12	Điểm danh trực tiếp	Người dùng nhập mã số sinh viên trực tiếp để xác nhận điểm danh

2.1.3 Đặc tả use case:

2.1.3.1 Use case đăng nhập:

- Tóm tắt:

Use case bắt đầu khi người dùng chưa từng đăng nhập, và người dùng chạy ứng dụng. Hệ thống sẽ cho phép người dùng thực hiện đăng nhập.

- Dòng sự kiện chính:

- B1: Người dùng mở ứng dụng
- B2: Hệ thống kiểm tra thông tin đăng nhập cũ, nếu tìm thấy thông tin đăng nhập cũ, đi đến B9.
- B3: Hệ thống hiển thị view đăng nhập
- B4: Người dùng nhập tên đăng nhập, mật khẩu.
- B5: Hệ thống kiểm tra cú pháp tên đăng nhập và mật khẩu của người dùng.
- B6: Nếu tên cú pháp không đúng, hiển thị thông báo và quay lại B1.
- B7: Hệ thống kiểm tra tài khoản, nếu cặp tên đăng nhập & mật khẩu không hợp lệ thì thông báo và quay lại B1.
- B8: Lưu lại thông tin đăng nhập của người dùng (dưới dạng đã mã hoá) để tự động đăng nhập cho lần sau.
- B9: Hiển thị giao diện sử dụng tương ứng với loại tài khoản người dùng nhập.
- B10: Kết thúc.

- Các dòng sự kiện khác:

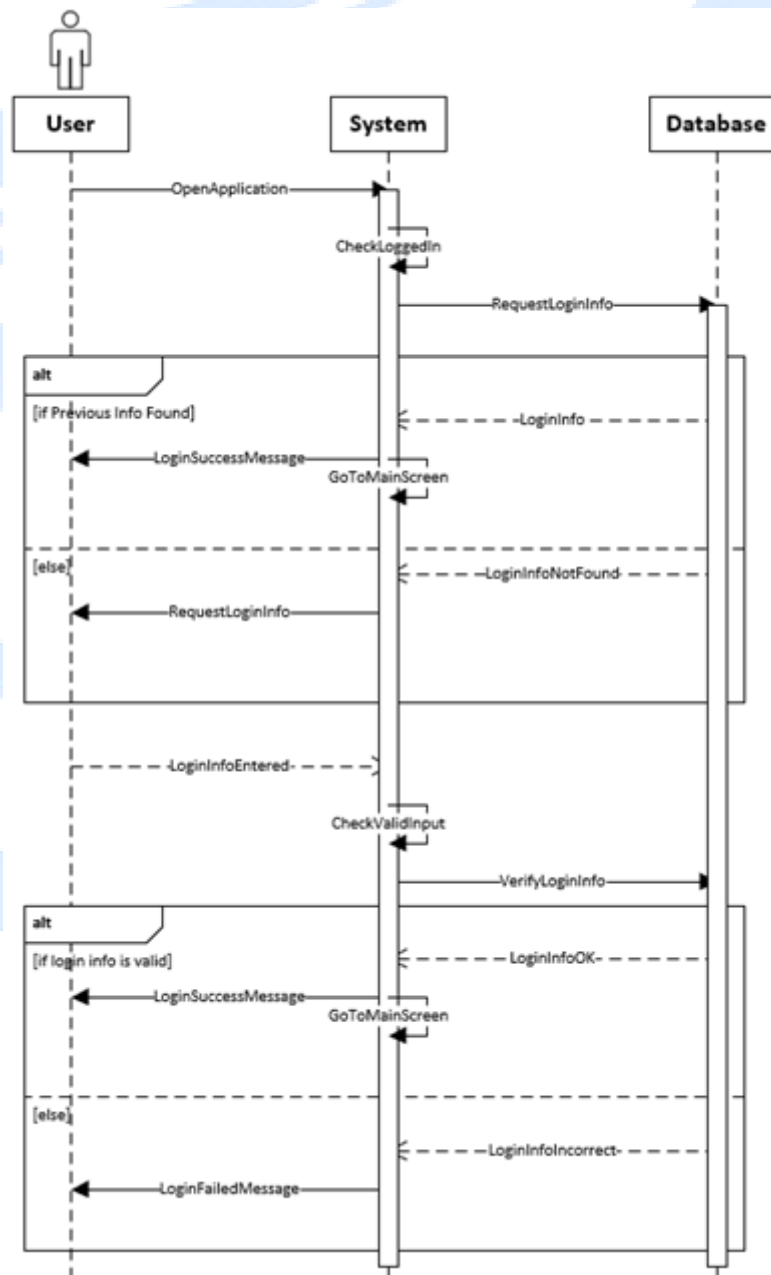
1. Xử lý việc đăng nhập không hợp lệ.
2. Xử lý việc đăng nhập quá nhiều lần.

- Các yêu cầu đặc biệt: Không có

- Tình trạng hệ thống khi bắt đầu use case:

1. Thiết bị hiện tại phải đang có kết nối Internet.
2. Ứng dụng phải kết nối thành công với CSDL Firebase.

- Tình trạng hệ thống sau khi thực hiện use case:
 1. Ứng dụng phải vào được màn hình tương tác chính tương ứng với tài khoản hiện tại hoặc vào activity đăng ký (trong TH người dùng ấn đăng ký mới tài khoản).
 2. Thông tin đăng nhập của người dùng được lưu lại dùng cho lần mở ứng dụng sau.
- Điểm mở rộng: Use case đăng ký cho trường hợp người dùng chưa có tài khoản.
- Sơ đồ tuần tự:



2.1.3.2 Đặc tả use case đăng ký:

- Tóm tắt:

Use case bắt đầu khi người dùng khởi chạy ứng dụng mà chưa đăng nhập. Khi đó nút đăng ký sẽ hiện ra (ở view đăng nhập), người dùng có thể ấn vào nút đăng ký để đi đến hoạt động đăng ký.

- Dòng sự kiện chính:

- B1: Người dùng nhấn chọn nút đăng ký trong view hiển thị login.
- B2: Hệ thống trả về view đăng ký.
- B3: Người dùng nhập thông tin đầy đủ cho phần đăng ký, bao gồm các mục như sau :
 - MSSV
 - Mật khẩu.
 - Email.
 - ID
 - Display name
 - Avatar
- B4: Người dùng ấn nút đăng ký. Hệ thống sẽ kiểm tra tính hợp lệ của thông tin mà người dùng đưa vào. Nếu thông tin không đúng, quay lại B3.
- B5: Hệ thống sẽ gửi thông tin người dùng đã nhập lên server online để lưu trữ.
- B6: Hệ thống lưu thông tin đăng ký local cho tự động đăng nhập và chuyển đến màn hình trang chủ.
- B7: Kết thúc.

- Các dòng sự kiện khác:

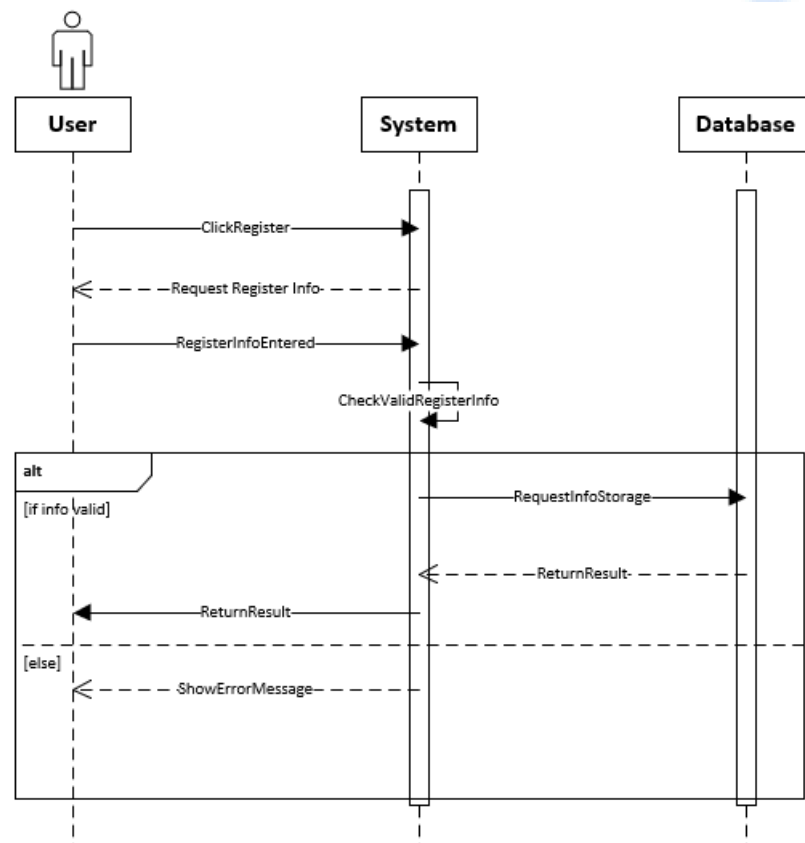
1. Xử lý cho thông tin người dùng nhập không hợp lệ.

- Các yêu cầu đặc biệt: Không có

- Trạng thái hệ thống khi bắt đầu thực hiện Use-case:

1. Người dùng đang có một kết nối Internet đến server.

2. Người dùng hiện đang không đăng nhập vào hệ thống.
- Trạng thái hệ thống sau khi thực hiện Use-case:
 1. Tài khoản người dùng được tạo và lưu trữ trên hệ thống.
 2. Thông tin tài khoản người dùng được lưu lại trên thiết bị để đăng nhập nhanh cho lần truy cập ứng dụng sau.
 - Điểm mở rộng: Không có
 - Sơ đồ tuần tự:



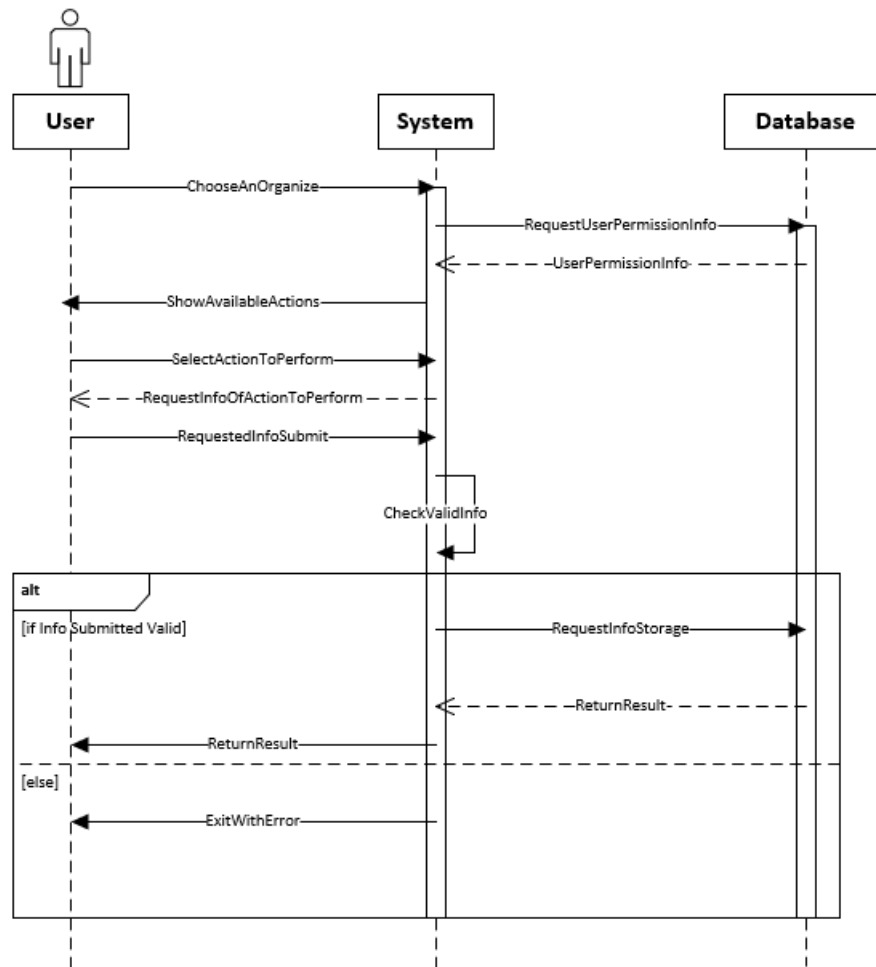
2.1.3.3 Đặc tả Use-case “Quản lý tổ chức”

- Tóm tắt:

Khi người dùng đăng nhập thành công, họ có thể chọn một tổ chức mà họ thuộc về để hiển thị view các event. Đồng thời, họ cũng có thể tạo ra 1 tổ chức mới của riêng họ.
- Dòng sự kiện chính
 - B1: Người dùng chọn một tổ chức đã có sẵn trong danh sách (người dùng đã gia nhập vào các tổ chức kể trên), hoặc tạo mới tổ chức.

- B2: Hệ thống sẽ trả về view Organizer Manager.
- B3: Tùy thuộc vào quyền hạn, người dùng lựa chọn hoạt động phù hợp với nhu cầu. Có thể kể đến một số hoạt động như :
 - Quản lý thành viên:
 - Xem thành viên của tổ chức.
 - Thêm thành viên vào tổ chức.
 - Xoá thành viên khỏi tổ chức.
 - Cấp quyền khác nhau cho thành viên trong tổ chức.
 - Quản lý hoạt động :
 - Tạo mới hoạt động.
 - Chỉnh sửa hoạt động.
 - Xoá hoạt động.
 - B4: Ứng với từng hoạt động sẽ có các phương thức nhập liệu và thực hiện khác nhau.
 - B5: Xác thực tính đúng đắn của hoạt động. Sau đó, ứng dụng sẽ thực hiện lưu trữ các thay đổi đến server.
 - B6: Kết thúc.
- Các dòng sự kiện khác:
 1. Từ chối truy cập đối với người dùng không đủ quyền hạn.
- Các yêu cầu đặc biệt: Không có
- Trạng thái hệ thống khi bắt đầu thực hiện Use-case:
 1. Người dùng đã đăng nhập vào hệ thống.
 2. Người dùng là admin của tổ chức.
- Trạng thái hệ thống sau khi thực hiện Use-case:

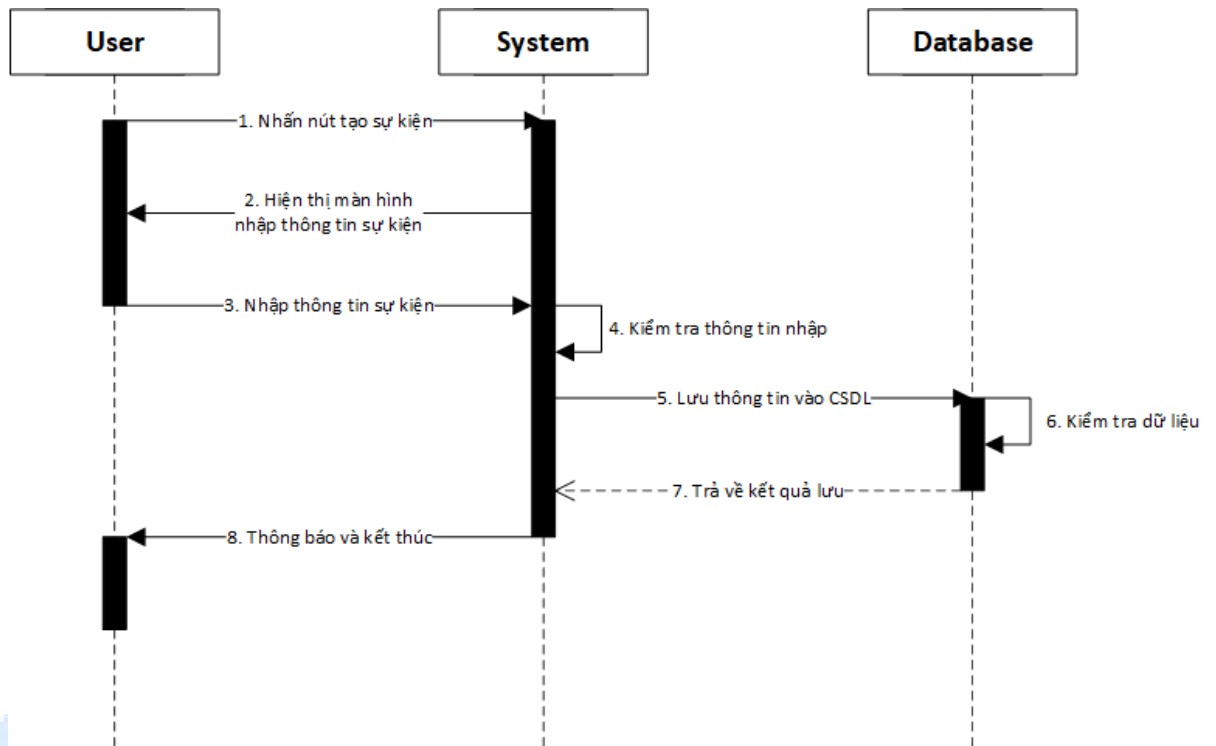
Những thay đổi được cập nhật và lưu lại ở phía server.
- Điểm mở rộng: Không có
- Sơ đồ tuần tự:



2.1.3.4 Đặc tả use case “Thêm sự kiện”

- Tóm tắt:
Người dùng thêm sự kiện của một tổ chức.
- Dòng sự kiện chính:
 - B1: Người dùng nhấn button tạo sự kiện
 - B2: Hệ thống hiển thị view các ô nhập thông tin sự kiện
 - B3: Người dùng nhập thông tin sự kiện
 - B4: Hệ thống kiểm tra tính hợp lệ (cú pháp) thông tin nhập vào. Nếu không đúng quay lại B3
 - B5: Hệ thống lưu thông tin xuống CSDL và hiển thị thông báo cho người dùng. Nếu không thành công quay lại bước 1.
- Các dòng sự kiện khác: Không có
- Các yêu cầu đặc biệt: Không có
- Trạng thái hệ thống khi bắt đầu thực hiện Use-case:
 1. Đăng nhập thành công

- Trạng thái hệ thống sau khi thực hiện Use-case:
 1. Thông tin lưu xuống CSDL thành công và thông báo.
- Điểm mở rộng: Không có
- Sơ đồ tuần tự:



2.1.3.5 Đặc tả use case “Tạo bộ câu hỏi”

- Tóm tắt:
Người dùng tạo bộ câu hỏi cho sự kiện.
- Dòng sự kiện chính:
 - B1: Người dùng nhấn button tạo bộ câu hỏi
 - B2: Hệ thống hiển thị view các ô nhập thông tin bộ câu hỏi
 - B3: Người dùng nhập thông tin bộ câu hỏi
 - B4: Hệ thống kiểm tra tính hợp lệ (cú pháp) thông tin nhập vào. Nếu không đúng quay lại B3
 - B5: Hệ thống lưu thông tin xuống CSDL và hiển thị thông báo cho người dùng. Nếu không thành công quay lại B1.

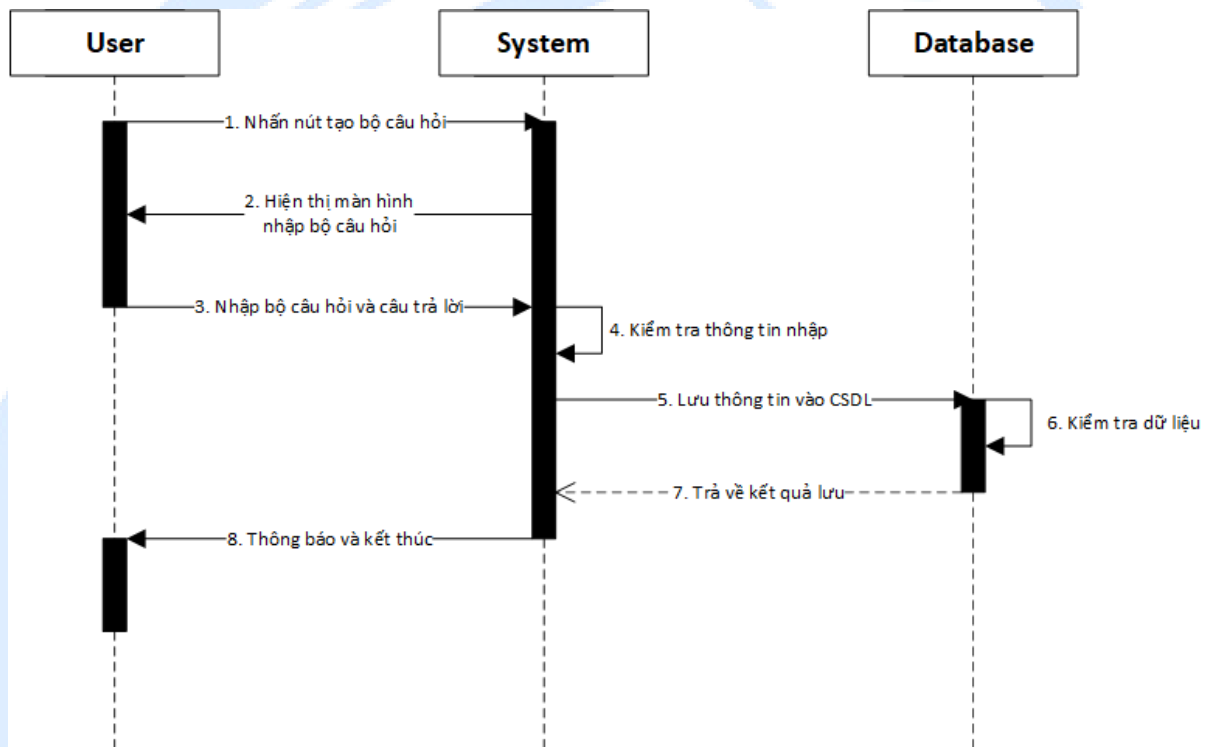
- Các dòng sự kiện khác: Không có
- Các yêu cầu đặc biệt: Không có
- Trạng thái hệ thống khi bắt đầu thực hiện Use-case:

1. Tạo sự kiện thành công.

- Trạng thái hệ thống sau khi thực hiện Use-case:

1. Thông tin lưu xuống CSDL thành công và thông báo.

- Điểm mở rộng: Không có
- Sơ đồ tuần tự:



2.1.3.6 Đặc tả use case “Tham gia sự kiện”

- Tóm tắt:

Người dùng tham gia sự kiện cần nhập mã code xác thực.

- Dòng sự kiện chính:

- B1: Người dùng nhấn button tham gia sự kiện
- B2: Hệ thống hiển thị ô nhập mã sự kiện

- B3: Người dùng nhập mã sự kiện
- B4: Hệ thống kiểm tra tính hợp lệ (cú pháp) thông tin nhập vào. Nếu không đúng quay lại bước 3
- B5: Hệ thống kiểm tra mã từ CSDL và hiển thị thông báo cho người dùng. Nếu không thành công quay lại bước 1.
- Các dòng sự kiện khác: Không có
- Các yêu cầu đặc biệt: Không có
- Trạng thái hệ thống khi bắt đầu thực hiện Use-case:
 1. Đăng nhập thành công và sự kiện có tồn tại.
- Trạng thái hệ thống sau khi thực hiện Use-case:
 1. Thông tin lưu xuống CSDL thành công và thông báo.
- Điểm mở rộng: Không có

2.1.3.7 Đặc tả use case “Xem thông tin của user”

- Tóm tắt:

Người dùng (user 1) chỉ xem được thông tin của user 2 khác khi user 2 đó đã tham gia hoạt động của user 1 tạo ra. User 1 xem thông tin user2 và những chi tiết các hoạt động (do user 1 tạo) user 2 đã tham gia.
- Dòng sự kiện chính:
 - B1: Người dùng nhấn chọn button xem thông tin user
 - B2: Hệ thống trả về view hiển thị thông tin người dùng (đang trông thông tin)
 - B3: Người dùng nhập mã số (ID) sinh viên cần xem để kiểm tra
 - B4: Hệ thống kiểm tra tính hợp lệ (cú pháp) của thông tin nhập vào. Nếu thông tin không đúng định dạng quay lại B3.
 - B5: Kiểm tra quyền xem dữ liệu của người dùng hiện tại đối với ID được nhập. Nếu không có quyền xem, thông báo không tìm thấy và đi đến B7.
 - B6: Hiển thị thông tin của ID, các hoạt động của người dùng tạo mà ID đã tham gia.

- B7: Kết thúc.

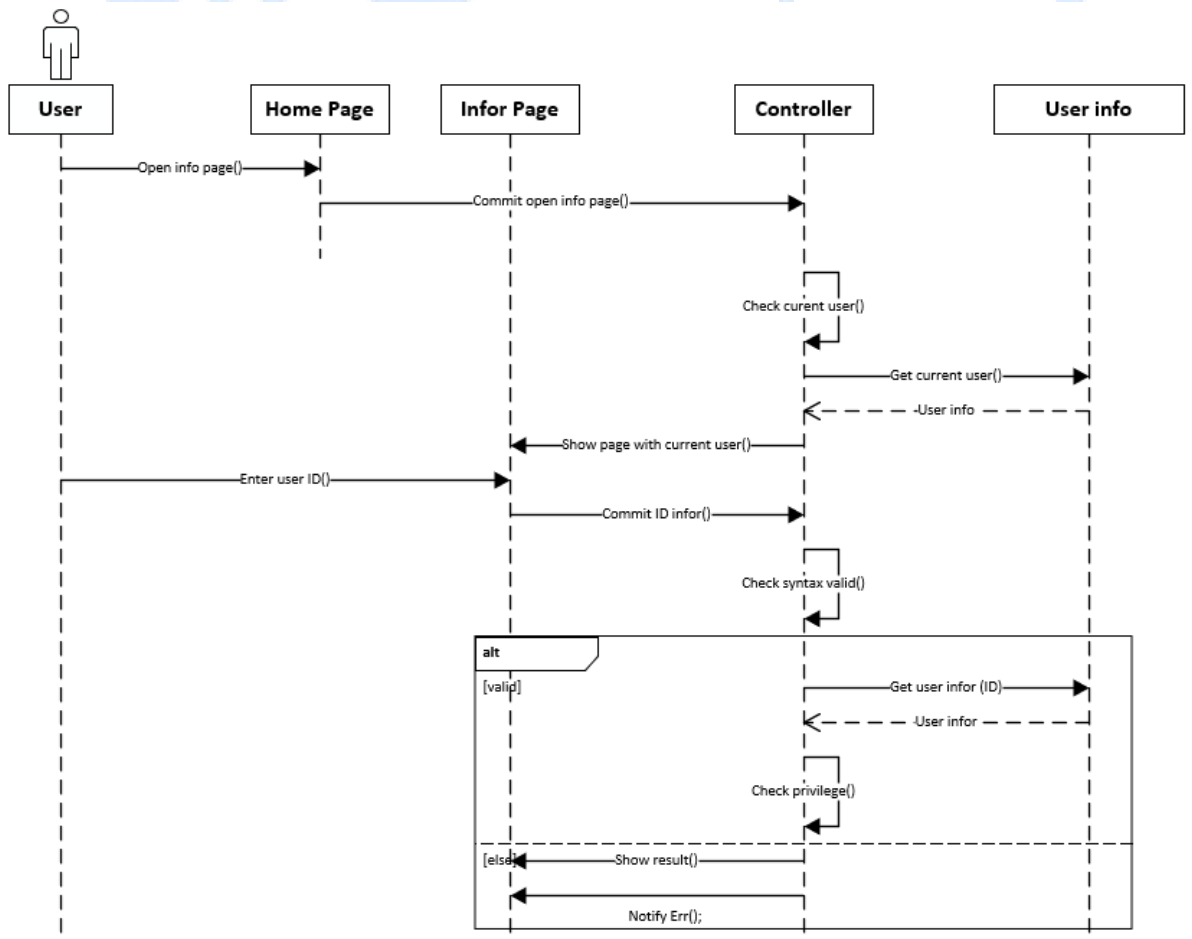
- Các dòng sự kiện khác: Không có
- Các yêu cầu đặc biệt: Không có
- Trạng thái hệ thống khi bắt đầu thực hiện Use-case:

1. Người dùng đã đăng ký tài khoản.

- Trạng thái hệ thống sau khi thực hiện Use-case:

1. Hiển thị thông tin tương ứng với từ khóa tìm kiếm.

- Điểm mở rộng: Không có
- Sơ đồ tuần tự:

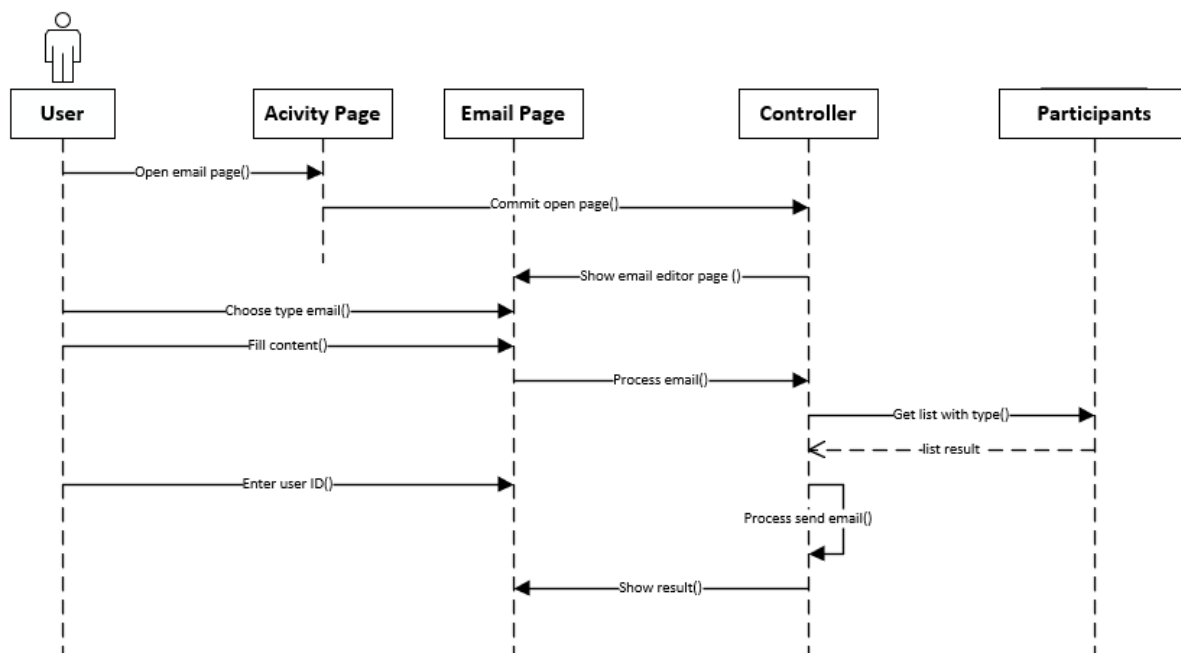


2.1.3.8 Đặc tả use case “Gửi email”

- Tóm tắt:

Người dùng cần thông báo thông tin đến người đăng ký/ tham gia bằng email đã đăng ký.

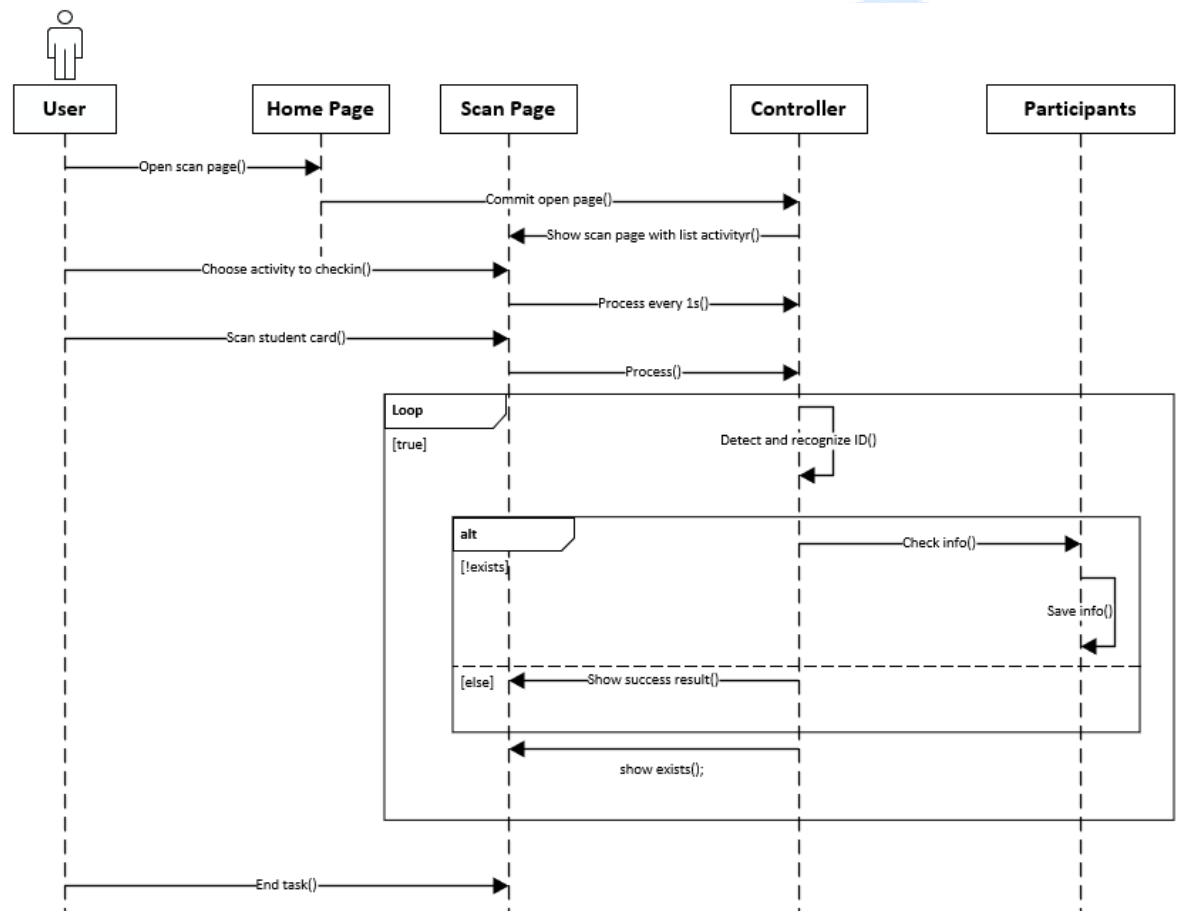
- Dòng sự kiện chính:
 - B1: Người dùng nhấn chọn button gửi mail (nằm trong giao diện của mỗi hoạt động)
 - B2: Hệ thống trả về view gửi email.
 - B3: Người dùng chọn loại gửi mail (đăng ký/ tham gia).
 - B4: Người dùng soạn nội dung email và nhấn nút send.
 - B5: Hệ thống truy vấn danh sách các email phù hợp để thêm vào người nhận và thực hiện gửi mail. Sử dụng mail gửi là email của người tạo hoạt động.
 - B6: Hệ thống thông báo kết quả
 - B7: Kết thúc.
- Các dòng sự kiện khác: Không có
- Các yêu cầu đặc biệt: Không có
- Trạng thái hệ thống khi bắt đầu thực hiện Use-case:
 1. Người dùng đã đăng ký tài khoản, có tổ chức và các hoạt động, danh sách đăng ký/ tham gia không trống.
- Trạng thái hệ thống sau khi thực hiện Use-case:
 1. Những người có mail trong danh sách gửi nhận được thông báo.
- Điểm mở rộng: Không có
- Sơ đồ tuần tự:



2.1.3.9 Đặc tả use case “Quét thẻ điểm danh”

- Tóm tắt:
Người dùng cần thực hiện điểm danh cho các user mang thẻ sinh viên.
- Dòng sự kiện chính:
 - B1: Người dùng nhấn chọn button scan (nằm trên thanh menu chính)
 - B2: Hệ thống trả về view chọn hoạt động điểm danh.
 - B3: Người dùng chọn hoạt động cần điểm danh.
 - B4: Hệ thống trả về view camera và thực hiện quét mỗi 1s.
 - B5: Người dùng đưa thẻ đặt đối diện camera và chờ hệ thống trả kết quả.
 - B6: Hệ thống xử lý thông tin từ ảnh và kiểm tra. Nếu thông tin không xác định, quay lại bước 5.
 - B7: Hệ thống thực hiện kiểm tra ID, nếu đã tồn tại thì không làm gì hết. Ngược lại thực hiện lưu vào cơ sở dữ liệu, hiển thị ID đã được lưu lên màn hình.
 - B8: Kết thúc.
- Các dòng sự kiện khác: Không có
- Các yêu cầu đặc biệt: Không có

- Trạng thái hệ thống khi bắt đầu thực hiện Use-case:
 1. Người dùng đã đăng ký tài khoản, có tổ chức và các hoạt động.
- Trạng thái hệ thống sau khi thực hiện Use-case:
 1. Những người được quét thẻ nằm trong danh sách tham gia của hoạt động.
- Điểm mở rộng: Không có
- Sơ đồ tuần tự:

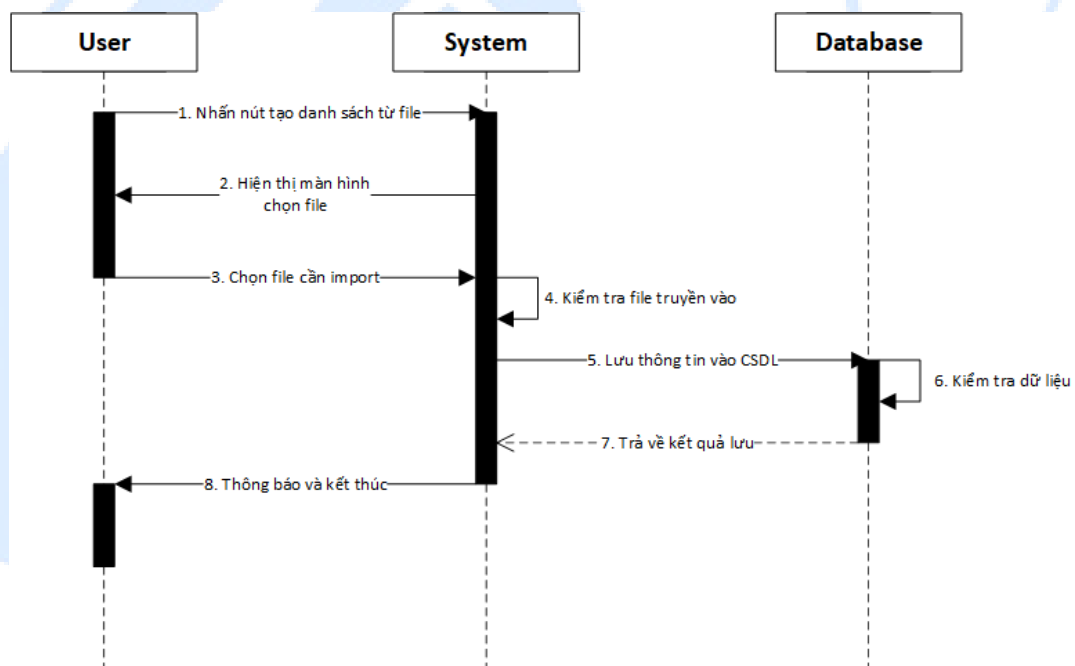


2.1.3.10 Đặc tả use case “tạo danh sách tham gia từ file”

- Tóm tắt:

Người dùng thêm thành viên vào sự kiện bằng cách chèn vào file thông tin các thành viên (file excel).
- Dòng sự kiện chính:
 - B1: Người dùng nhấn button tạo danh sách qua file
 - B2: Hệ thống hiển thị view truyền file danh sách
 - B3: Người dùng chọn file thông tin danh sách tham gia

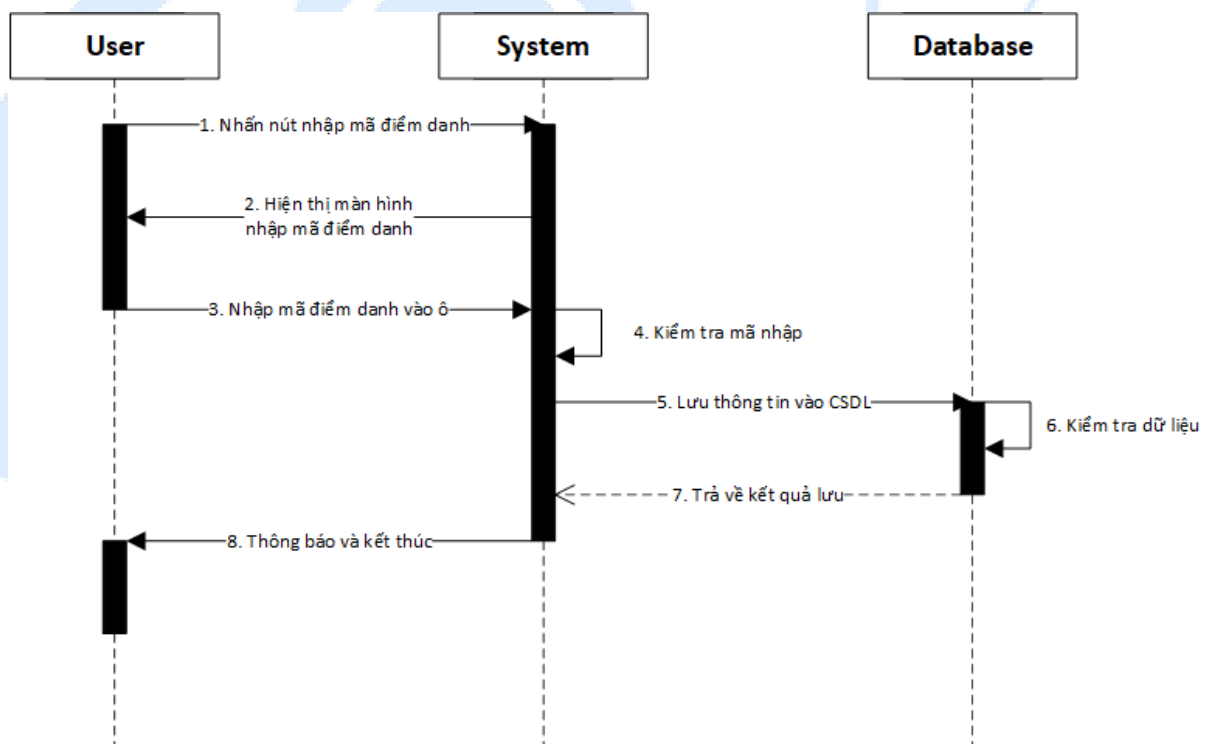
- B4: Hệ thống gửi thông tin xuống CSDL
 - B5: Thông tin được kiểm tra trước khi lưu vào CSDL. Nếu thông tin sai, quay lại bước 3 và thông báo lỗi.
 - B6: Thông báo thành công.
- Các dòng sự kiện khác: Không có
 - Các yêu cầu đặc biệt: Không có
 - Trạng thái hệ thống khi bắt đầu thực hiện Use-case:
 1. Sự kiện cần import danh sách đã được tạo thành công.
 - Trạng thái hệ thống sau khi thực hiện Use-case:
 1. Thông tin danh sách được lưu trữ vào CSDL và hiển thị thông báo.
 - Điểm mở rộng: Không có
 - Sơ đồ tuần tự:



2.1.3.11 Đặc tả use case “Điểm danh bằng mã sinh viên”

- Tóm tắt:
Người dùng nhập mã sinh viên để điểm danh.
- Dòng sự kiện chính:

- B1: Người dùng nhấn button nhập mã điểm danh
 - B2: Hệ thống hiển thị view nhập mã điểm danh
 - B3: Người dùng nhập mã điểm danh
 - B4: Hệ thống kiểm tra tính hợp lệ (cú pháp) thông tin nhập vào. Nếu không đúng quay lại bước 3
 - B5: Hệ thống lưu thông tin xuống CSDL và hiển thị thông báo cho người dùng. Nếu không thành công quay lại B1.
- Các dòng sự kiện khác: Không có
 - Các yêu cầu đặc biệt: Không có
 - Trạng thái hệ thống khi bắt đầu thực hiện Use-case:
 1. Sự kiện cần được điểm danh đã được tạo thành công.
 - Trạng thái hệ thống sau khi thực hiện Use-case:
 1. Thông tin điểm danh được lưu xuống CSDL
 - Sơ đồ tuần tự:



2.2. Thiết kế thành phần dữ liệu

2.2.1. Bảng person

STT	Tên thuộc tính	Kiểu	Ràng buộc	Ý nghĩa/ghi chú
1	mssv	String	Là khóa chính, có đúng 8 ký tự	Dùng để phân biệt giữa các person
2	Avatar	String	Dạng Url	Lưu đường dẫn đến ảnh của person
3	Displayname	String	Tối đa 50 ký tự	Dùng để hiển thị tên người dùng trong quá trình sử dụng ứng dụng
4	Class	String		Tên lớp sinh hoạt của person

2.2.2. Bảng Event

STT	Tên thuộc tính	Kiểu	Ràng buộc	Ý nghĩa/ghi chú
1	eventID	String	Là khóa chính, not null	Dùng để phân biệt giữa các event
2	eventCode	String	Có đúng 6 ký tự	Mã tham gia sự kiện dành cho người dùng có account
3	Avatar	String	Dạng URL	Lưu đường dẫn đến ảnh đại diện của sự kiện
4	beginTime	date	Not null	Thời điểm diễn ra sự kiện
5	endTime	date	Not null	Thời điểm diễn ra sự kiện
6	Location	String		Nơi diễn ra sự kiện
7	organ	String		Liên kết đến organID tổ chức tạo ra sự kiện này

2.2.3. Bảng Registration

STT	Tên thuộc tính	Kiểu	Ràng buộc	Ý nghĩa/ghi chú
1	ID	String	Là khóa chính, not null	Dùng để phân biệt giữa các registration (phân biệt giữa các lần đăng ký sự kiện)
2	MSSV	String		Liên kết đến MSSV của person đăng ký sự kiện
3	Event	String		Liên kết đến eventID của sự kiện được đăng ký
4	registerTime	dateTime		Thời điểm đăng ký

2.2.4. Bảng Attendance

STT	Tên thuộc tính	Kiểu	Ràng buộc	Ý nghĩa/ghi chú
1	ID	String	Là khóa chính, not null	Dùng để phân biệt giữa các attendance
2	MSSV	String		Liên kết đến person tham gia
3	Event	String		Liên kết đến sự kiện tham gia
4	registerTime	dateTime		Thời điểm tham gia

2.2.5. Bảng Account

STT	Tên thuộc tính	Kiểu	Ràng buộc	Ý nghĩa/ghi chú
1	Username	String	Là khóa chính, not null	Là email đăng nhập vào hệ thống
2	Password	String	Ít nhất 6 ký tự	Mật khẩu để đăng nhập hệ thống (đã được băm)

				bằng SHA trước khi lưu)
3	Person	string	Not null	Liên kết đến person (MSSV của person)
4	createDate	dateTime	Not null	Thời điểm tài khoản được tạo

2.2.6. Bảng Organization

STT	Tên thuộc tính	Kiểu	Ràng buộc	Ý nghĩa/ghi chú
1	OrganID	String	Là khóa chính, not null	Phân biệt giữa các tổ chức
2	Organ name	String		Tên tổ chức (dùng để hiển thị)
3	Avatar	String	Dạng Url, Not null	Lưu đường dẫn (URL) ảnh đại diện của tổ chức
4	Description	String		Mô tả về tổ chức
5	Host	String	Not null	Người tạo ra tổ chức (quyền cao nhất). Tham chiếu đến Username của Account người đó.

2.2.7. Bảng Collaborator

STT	Tên thuộc tính	Kiểu	Ràng buộc	Ý nghĩa/ghi chú
1	ID	String	Là khóa chính, not null	Phân biệt giữa các collaborator (cộng tác viên của tổ chức)

2	Account	String		Tham chiếu đến Username của người công tác viên đó
3	Organ	string	Dạng Url, Not null	Tham chiếu đến OrganID của tổ chức đó
4	joinTime	dateTime		Thời gian được thêm vào làm cộng tác viên

2.2.8. Bảng Multiple_choice_Q

STT	Tên thuộc tính	Kiểu	Ràng buộc	Ý nghĩa/ghi chú
1	QID	String	Là khóa chính, not null	Phân biệt giữa các câu hỏi
2	Content	String	Not null	Nội dung câu hỏi
3	publicTime	string	Dạng Url, Not null	Thời gian câu hỏi xuất hiện lên ứng dụng
4	event	String		Tham chiếu đến EventID của sự kiện chứa câu hỏi này

2.2.9. Bảng Answer

STT	Tên thuộc tính	Kiểu	Ràng buộc	Ý nghĩa/ghi chú
1	AID	String	Là khóa chính, not null	Phân biệt giữa các câu trả lời mẫu
2	Key	Integer		Thứ tự câu trả lời
3	Content	String		Nội dung câu trả lời
4	Question	String	Not null	Tham chiếu đến QID của đến câu hỏi chứa câu trả lời này

5	isCorrect	Integer		Xác định câu trả lời này là đúng hay sai
---	-----------	---------	--	--

2.2.10 Bảng Answ_MQ

STT	Tên thuộc tính	Kiểu	Ràng buộc	Ý nghĩa/ghi chú
1	Answer_MQ_ID	String	Là khóa chính, not null	Phân biệt giữa các câu trả lời của người tham gia trả lời câu hỏi
2	Respondent	String		Tham chiếu đến Username của Account của người trả lời câu hỏi này
3	Question	String		Tham chiếu đến QID của MULTIPLE-CHOICE_Q của câu hỏi mà người này trả lời
4	Answer	String		Tham chiếu đến AID của câu trả lời mà người này chọn
5	joinTime	dateTime		Thời điểm người này trả lời câu hỏi

2.2.11 Bảng Essay_Q

STT	Tên thuộc tính	Kiểu	Ràng buộc	Ý nghĩa/ghi chú
1	EID	String	Là khóa chính, not null	Phân biệt giữa các câu hỏi
2	Content	String		Nội dung câu hỏi

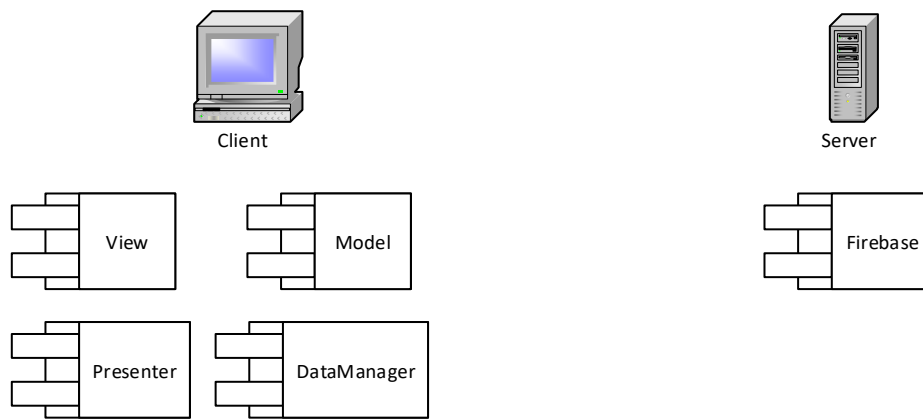
3	publicTime	dateTime		Thời gian câu hỏi xuất hiện lên ứng dụng
4	event	String		Tham chiếu đến EventID của sự kiện chứa câu hỏi

2.2.12 Bảng Answ_EQ

STT	Tên thuộc tính	Kiểu	Ràng buộc	Ý nghĩa/ghi chú
1	Answer_EQ_ID	String	Là khóa chính, not null	Phân biệt giữa các câu trả lời của người tham gia trả lời câu hỏi
2	Respondent	String		Tham chiếu đến Username của Account của người trả lời câu hỏi
3	Question	String		Tham chiếu đến EID của ESSAY_Q (câu hỏi).
4	Answer	String		Nội dung câu trả lời
5	joinTime	dateTime		Thời điểm câu hỏi được trả lời

Chương 3 : Thiết kế kiến trúc.

3.1. Kiến trúc hệ thống



Thành phần	Diễn giải
View	Quản lý việc thể hiện giao diện của các màn hình khác nhau của ứng dụng.
Model	Quản lý việc truy xuất dữ liệu đến database local (database offline của client) của từng
Presenter	Quản lý xử lý logic giữa các màn hình khác nhau của ứng dụng.
DataManager	Quản lý việc kết nối đến CSDL trên Server (database online).
FireBase	Server lưu trữ dữ liệu của ứng dụng.

Bảng 3.1: Diễn giải các thành phần hệ thống

(Hình ảnh rõ nét hơn : <http://bit.ly/37Y26IZ>).

MainActivity	Màn hình chính của ứng dụng.
OrganPresenter	Presenter của Organization (xử lý logic cho OrganView).
OrganFragment	View của Organization.
IloginView	Interface quy định Contract giữa LoginPresenter và LoginActivity
LoginPresenter	Presenter của LoginActivity (xử lý logic cho LoginActivity).
LoginActivity	View của sự kiện Login
IInfoFragmentView	Interface quy định Contract giữa InfoFragment và InfoFragmentPresenter
InfoFragment	View hiển thị giao diện dùng để hiển thị chi tiết thông tin người dùng.
InfoFragmentAccount	View dùng để hiển thị thông tin chi tiết của người dùng như MSSV, email,...
InfoFragmentAct	View hiển thị những người dùng đã đăng ký hoặc đã tham gia vào sự kiện của tổ chức.
InfoFragmentPresenter	Presenter xử lý logic của InfoFragmentAct.
ListActivityFragment	View để hiển thị danh sách các activity
ViewListActivity	View hiển thị các hoạt động mà một Organizer đã tổ chức.

ViewSetContentFragment	View hiển thị mục để người dùng điền vào mô tả cho hoạt động.
ViewSetNameFragment	View hiển thị mục để người dùng điền vào tên của hoạt động
ViewSetTimeFragment	View hiển thị mục để người dùng có thể cài đặt ngày diễn ra hoạt động và thời gian bắt đầu/kết thúc của hoạt động đó.
ParticipantAnalyticsListFragment	View hiển thị kết quả trả lời câu hỏi của những người đã tham gia hoạt động.
ParticipantAnalyticsListFragmentContract	Contract giữa ParticipantAnalyticsListFragment và ParticipantAnalyticsListFragmentPresenter
ParticipantAnalyticsListFragmentPresenter	Presenter xử lý logic cho ParticipantAnalyticsListFragment.
ParticipantListFragment	View hiển thị danh sách những người đăng ký/tham gia vào một hoạt động.
ExtendDialogFragment	View hiển thị các tương tác có thể thực hiện ứng với những người tham gia hoạt động (bao gồm gửi mail cho những người đã đăng ký, gửi mail cho những người đã tham gia, gửi mail cho tất cả những người đã đăng ký hoặc đã tham gia, nhập danh sách người đăng ký/tham gia).

ImportDialogFragment	View hiển thị tiến độ trong việc nhập (Import) danh sách người tham gia.
QuestionListParticipantViewFragment	View hiển thị danh sách câu hỏi của từng hoạt động cụ thể
QuestionManagementFragment	View hiển thị cửa sổ quản lý chung cho hoạt động.
QuestionListContract	Contract giữa QuestionListFragment và QuestionListPresenter
QuestionListFragment	View hiển thị danh sách câu hỏi
QuestionListPresenter	Presenter xử lý logic cho QuestionListFragment
QuestionListParticipantViewFragment	View hiển thị danh sách câu hỏi (của người tham gia event)
QuestionManagementFragment	View quản lý câu hỏi (thêm/xoá/sửa).
HomeFragment	View hiển thị trang chủ.
ILoginView	Contract giữa LoginActivity và LoginPresenter
LoginActivity	View hiển thị cho phép người dùng đăng nhập
LoginPresenter	Xử lý logic cho LoginActivity
RegisterActivity	View hiển thị cho phép người dùng đăng ký
ResetPasswordActivity	View hiển thị cho phép người dùng thao tác khôi phục mật khẩu.

SendEmailFragment	View cho phép người tổ chức gửi mail đến người tham gia sự kiện.
DatePickerFragment	View cho phép người dùng thao tác chọn ngày
DateTimeDialogFragment	View cho phép người dùng thao tác chọn cả ngày và giờ.
CameraPreview	Một Surface View dùng để hiển thị những gì mà Camera đang quét được.
CardScannerFragment	View cho phép người dùng thực hiện tương tác quét mã sinh viên
AnswerListCustomAdapter	Custom Adapter cho việc hiển thị danh sách câu hỏi.
CollaborationAdapter	Custom Adapter cho việc hiển thị danh sách các cộng tác viên của một tổ chức.
EventAdapter	Custom Adapter cho việc hiển thị danh sách các hoạt động mà người dùng đã tham gia.
PageAdapter	Custom Adapter cho ViewPager trong cửa sổ hiển thị thông tin người sử dụng.
PageAdapterListActivity	Custom Adapter cho ViewPager trong cửa sổ hiển thị các hoạt động của tổ chức.
PageAdapterListParticipant	
ParticipantAdapter	Custom Adapter cho việc hiển thị danh sách những người đã tham gia/đã đăng ký vào một hoạt động.

ParticipantAnswerDetailsCustomAdapter	Custom Adapter cho việc hiển thị kết quả trả lời câu hỏi của tất cả những người đã tham gia vào hoạt động.
ParticipantListCustomAdapter	
QuestionListCustomAdapter	Custom Adapter cho việc hiển thị danh sách câu hỏi của một hoạt động.
QuestionManagerViewPagerAdapter	Custom Adapter cho ViewPager dùng trong hiển thị chi tiết của một hoạt động (Danh sách người đăng ký/tham gia, các câu hỏi trắc nghiệm, câu trả lời của người tham gia).
Account	Model của tài khoản người dùng.
Answer	Model của một câu trả lời (của một câu hỏi).
Event	Model của một sự kiện
Organization	Model của tổ chức.
PaticipantAnswerDetails	Hiển thị thông tin về câu trả lời của người tham dự.
Person	Model thể hiện một người (MSSV,...).
Question	Model thể hiện một câu hỏi.
DataManager	Quản lý kết nối CSDL đến Firebase.
ResizeWidthAnimation	Hỗ trợ thay đổi kích cỡ của layout dựa trên Animation
Tesseract	Hỗ trợ việc nhận diện ký tự (OCR) của MSSV trong thẻ.

Bảng 3.2: Diễn giải các lớp trong hệ thống

Chương 4 : Thiết kế giao diện.

4.1. Danh sách các màn hình

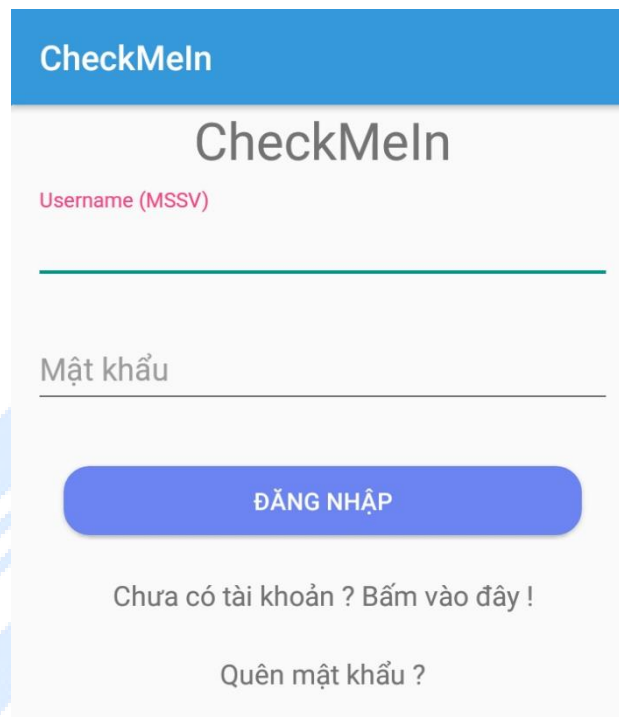
STT	Tên màn hình	Ý nghĩa/Ghi chú
1	Màn hình đăng nhập	Màn hình hiển thị các ô EditText cho phép người dùng thực hiện đăng nhập (nếu người dùng chưa từng đăng nhập trước đó).
2	Màn hình đăng ký tài khoản	Màn hình hiển thị các ô EditText cho phép người dùng nhập vào thông tin để đăng ký tài khoản mới.
3	Màn hình quên mật khẩu	Màn hình hiển thị ô nhập Email, cho phép người dùng nhập vào Email của mình để hệ thống gửi một liên kết cho người dùng tạo mới lại mật khẩu.
4	Màn hình chính (Trang chủ)	Màn hình chính cho người dùng tổng quan về tổ chức (Organize) của người đó. Đồng thời, cho phép người dùng tham gia nhanh vào một hoạt động thông qua mã hoạt động (Event Code).
5	Màn hình tạo mới tổ chức (Create Organization View)	Màn hình cho phép người dùng tạo mới một tổ chức.
6	Màn hình danh sách hoạt động (Activity List)	Hiển thị danh sách các hoạt động của một tổ chức (Organizer).
7	Màn hình tạo hoạt động	Màn hình cho phép người chủ tổ chức (Host) tạo một hoạt động mới cho tổ chức của mình.
8	Màn hình hiển thị danh sách người tham gia	Màn hình cho phép người trong một tổ chức có thể xem được danh sách những người đã tham gia (hoặc đăng ký) vào hoạt động của mình.
9	Màn hình nhập danh sách người tham gia	Màn hình cho phép người dùng thao tác để nhập danh sách người tham gia từ file Excel (thay vì cho người dùng đăng ký hoặc quét thẻ).

10	Màn hình gửi Email	Màn hình soạn thảo Email để gửi ngay hoặc hẹn lịch gửi đến các đối tượng đã được chọn trước. Bao gồm Tiêu đề Email, Nội dung email, Thời gian gửi.
11	Màn hình hiển thị danh sách câu hỏi (View của người tham gia Event)	Màn hình hiển thị danh sách những câu hỏi để người dùng có thể tham gia trả lời câu hỏi của Event.
12	Màn hình hiển thị danh sách câu hỏi (View của người tổ chức Event)	Màn hình hiển thị danh sách những câu hỏi để người tạo Event có thể quản lý (thêm/xoá/sửa).
13	Màn hình thêm/chỉnh sửa câu hỏi cho một Event	Màn hình cho phép người đứng đầu tổ chức (người host) tạo ra (hoặc chỉnh sửa, tùy vào cách khởi chạy màn hình này) một câu hỏi và các câu trả lời tương ứng cho Event.
14	Màn hình hiển thị câu trả lời của người tham gia	Màn hình hiển thị thành tích trả lời câu hỏi của tất cả những người đã tham gia vào việc trả lời câu hỏi của Event. Bao gồm tên người trả lời, Xếp hạng của người đó, Số câu trả lời đúng / Tổng số câu, và Thời gian hoàn thành.
15	Màn hình hiển thị danh sách hoạt động mà người dùng đã đăng ký/tham gia	Màn hình hiển thị danh sách hoạt động mà người dùng đã đăng ký/tham gia, bao gồm tên hoạt động, ngày tổ chức hoạt động.
16	Màn hình hiển thị thông tin của người dùng	Màn hình hiển thị thông tin của người dùng, bao gồm Ảnh đại diện, MSSV, Email, Số điện thoại. Cho phép người dùng chỉnh sửa thông tin của mình ở đây.

Bảng 4.1. Danh sách các màn hình

4.2. Mô tả chi tiết từng màn hình

4.2.1. Màn hình đăng nhập



Hình 4.1: Màn hình đăng nhập

Mô tả cách sử dụng và xử lý trên màn hình :

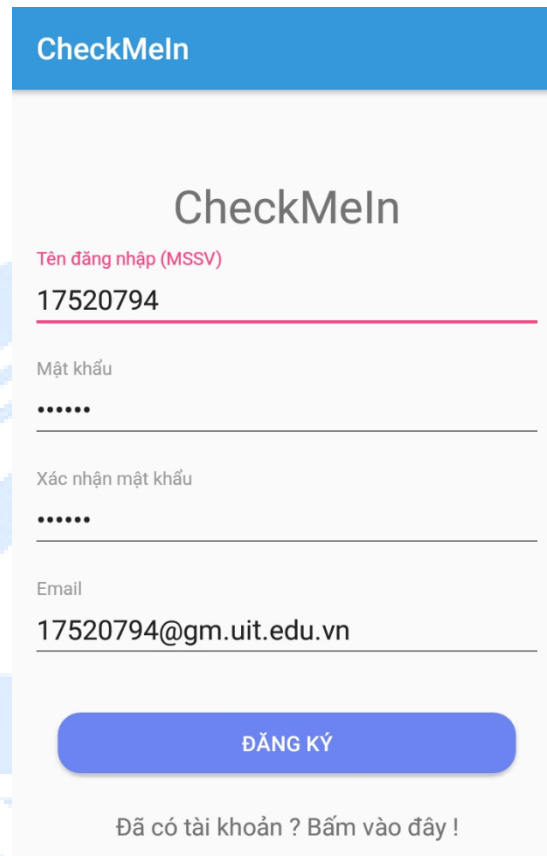
Khi người dùng mở ứng dụng, hệ thống sẽ xem xét người dùng có lưu tài khoản đăng nhập trước đó hay chưa (đã đăng nhập hay chưa). Nếu có hệ thống sẽ bỏ qua màn hình này và đi thẳng đến màn hình giao diện chính của ứng dụng. Nếu không tìm thấy thông tin đăng nhập trước đó thì hệ thống sẽ hiển thị màn hình này cho phép người dùng đăng nhập. Khi đó, người dùng nhập đầy đủ thông tin và nhấn **Đăng nhập**. Nếu thông tin đăng nhập hợp lệ, người dùng sẽ được chuyển đến **Màn hình chính**. Nếu thông tin đăng nhập không hợp lệ, ứng dụng sẽ hiển thị thông báo để người dùng thử lại.

Ngoài ra, ở màn hình này còn có 2 mục phụ : “**Chưa có tài khoản ? Bấm vào đây !**” và mục “**Quên mật khẩu ?**”.

Khi chưa có tài khoản, người dùng ấn vào mục đầu tiên để tạo mới một tài khoản cho mình. Khi đó, ứng dụng sẽ mở **Màn hình đăng ký tài khoản**.

Trường hợp người dùng đã có mật khẩu mà quên thông tin đăng nhập, người dùng có thể dễ dàng lấy lại bằng cách bấm vào mục thứ 2. Khi đó, ứng dụng sẽ mở **Màn hình quên mật khẩu**.

4.2.2 Màn hình đăng ký tài khoản



Hình 4.2: Màn hình đăng ký

Mô tả cách sử dụng và xử lý trên màn hình :

Người dùng nhập đầy đủ các trường như trên để thực hiện việc đăng ký. Nếu người dùng có nhập thông tin đăng nhập tại trang đăng nhập thì khi ấn đăng ký, hệ thống sẽ tự động chuyển những thông tin đó từ màn hình đăng nhập sang màn hình đăng ký để người dùng không mất công phải nhập lại. Bao gồm các trường **MSSV**, **Mật khẩu**, **Xác nhận mật khẩu** và **Email**.

Khi người dùng ấn **Đăng ký**, ứng dụng sẽ lần lượt kiểm tra các mục sau, nếu hợp lệ sẽ thực hiện lưu lại thông tin người dùng và gửi thông tin đăng ký đến **Firestore** :

- **Mật khẩu và Xác nhận mật khẩu** : Phải giống nhau.

- **Tên đăng nhập (MSSV)** : Chỉ cho nhập kiểu số, bắt buộc 8 số và phải đúng format MSSV UIT.
- **Email** : Phải ở định dạng Email hợp lệ.

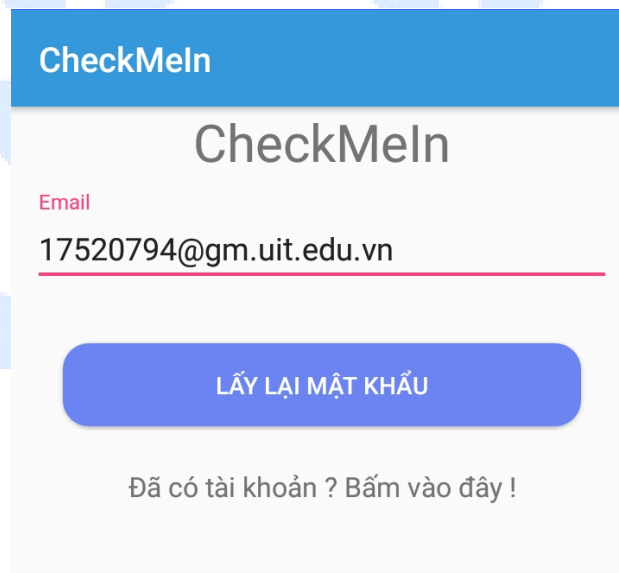
Sau khi gửi thông tin đăng ký lên **Firestore**, ứng dụng sẽ chờ phản hồi xem đăng ký có thành công hay không và hiển thị lỗi thông báo tương ứng cho người dùng nếu đăng ký thất bại. Một trong những lỗi thường gặp có thể kể đến là :

- Tên đăng nhập đã tồn tại : Người dùng với MSSV trên đã có tài khoản tương ứng trên hệ thống.
- Kết nối lỗi : Không thể kết nối đến server Firestore.

Nếu đăng ký thành công, ứng dụng sẽ tự động đăng nhập với thông tin đăng ký và lưu lại thông tin cho lần khởi động sau để ứng dụng tự động đăng nhập.

Ngược lại, nếu do nhầm lẫn mà người dùng vào màn hình đăng ký dù đã có tài khoản, người dùng có thể bấm vào mục **“Đã có tài khoản, bấm vào đây !”** để trở về màn hình đăng nhập.

4.2.3. Màn hình quên mật khẩu



Hình 4.3. Màn hình lấy lại mật khẩu

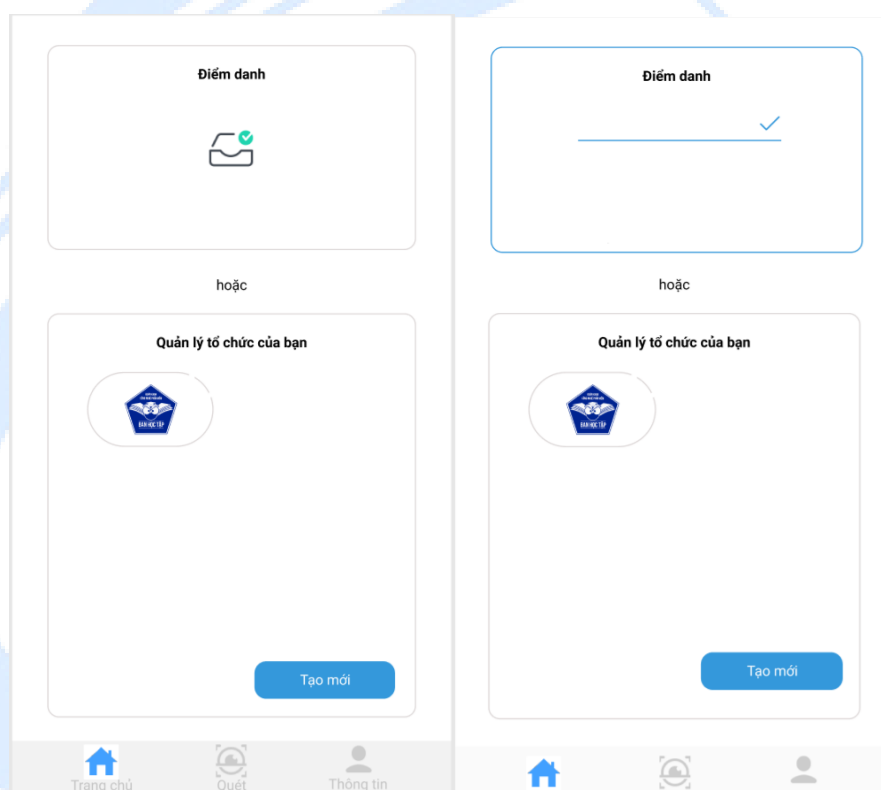
Mô tả cách sử dụng và xử lý trên màn hình :

Người dùng khi quên mật khẩu sẽ vào màn hình này. Ở đây có một trường **Email (EditText)** để điền email của tài khoản muốn khôi phục thông tin. Sau khi điền email, người dùng bấm nút “**Lấy lại mật khẩu**” để tiến hành khôi phục lại thông tin đăng nhập trên tài khoản của mình.

Nếu việc yêu cầu lấy lại mật khẩu thành công (Email tồn tại trong hệ thống) thì ngay lập tức ứng dụng sẽ chuyển sang màn hình **Đăng nhập**.

Nếu email không tồn tại trong hệ thống thì sẽ thông báo lỗi.

4.2.4. Màn hình chính (trang chủ)



Hình 4.4 Màn hình trang chủ

(Hình 1 : Trước khi ấn nút **Điểm danh** – Hình 2 : Sau khi ấn nút **Điểm danh**).

Mô tả cách sử dụng và xử lý trên màn hình :

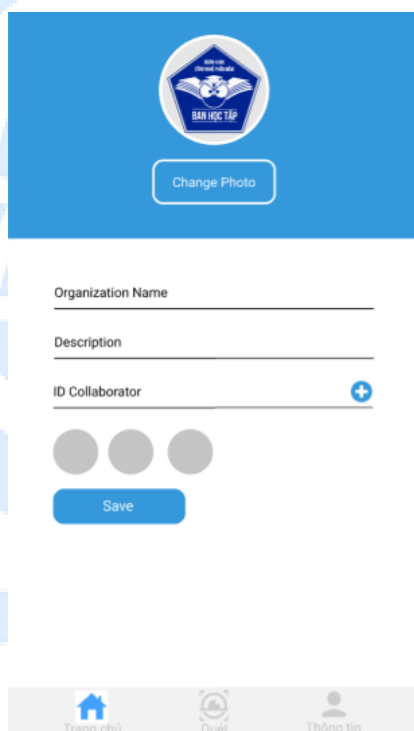
Trên màn hình gồm 3 phần :

- Phần trên cùng (**Điểm danh**) : Để người dùng có thể tham gia nhanh vào 1 hoạt động bằng mã sự kiện (**Event Code**). Người dùng chỉ cần nhập mã sự kiện vào

phần **EditText**, và bấm dấu Tick nằm kế bên. Hệ thống sẽ ghi nhận và đánh dấu sự hiện diện của người dùng.

- Phần giữa (**Quản lý tổ chức**) : Hiển thị ảnh đại diện của tất cả tổ chức mà người dùng hiện tại đang tham gia vào (Người dùng có thể là Collaborator (cộng tác viên) hoặc Host (người đứng đầu) của tổ chức đó). Người dùng chỉ cần bấm vào ảnh đại diện của một tổ chức để xem những hoạt động của tổ chức. Ngoài ra còn nút **Tạo mới**, cho phép người dùng tạo mới một tổ chức bằng cách chuyển ứng dụng sang **màn hình tạo mới tổ chức**.
- Phần dưới cùng (Thanh điều hướng – **Navigation Bar**) : Cho phép người dùng thực hiện nhanh thao tác chuyển đến View có chức năng khác như **Quét thẻ (Điểm danh)**, **Thông tin cá nhân**.

4.2.5. Màn hình tạo mới tổ chức



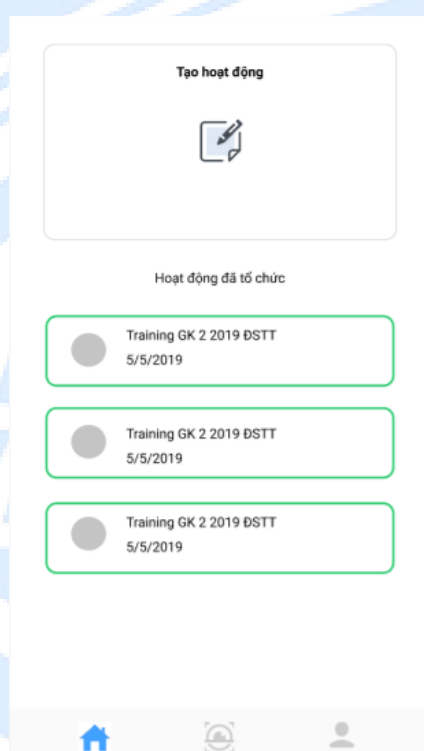
Hình 5. Màn hình tạo mới, chỉnh sửa tổ chức

Mô tả cách sử dụng và xử lý trên màn hình :

Ở màn hình này, phần trên cùng của chúng ta chính là ảnh đại diện của tổ chức. Chúng ta có thể ấn nút **Change Photo** để thực hiện việc thay đổi ảnh đại diện của tổ chức bằng một ảnh có sẵn ở thiết bị. Tiếp đó, người dùng nhập vào tên tổ chức

(**Organization Name**), mô tả tổ chức (**Description**), và nhập MSSV (**ID Collaborator**) của cộng tác viên. Người dùng có thể bấm dấu (+) để thêm cộng tác viên trên vào danh sách của tổ chức. Sau khi thêm thì ảnh đại diện của cộng tác viên sẽ được hiển thị bên dưới và trên nút **Save**. Cuối cùng, sau khi đã đủ thông tin thì người dùng bấm nút **Save** để thực hiện lưu lại thay đổi cho việc tạo mới tổ chức của mình.

4.2.6. Màn hình danh sách hoạt động



Hình 6. Màn hình danh sách các hoạt động

Mô tả cách sử dụng và xử lý trên màn hình :

Ở màn hình này hiển thị danh sách các hoạt động đã, đang hoặc sắp diễn ra của tổ chức được chọn. Gồm 2 phần : **Tạo hoạt động** và **Danh sách hoạt động đã tổ chức**.

Nếu người dùng bấm vào **Tạo hoạt động**, ứng dụng sẽ chuyển qua màn hình tạo hoạt động để người dùng có thể tạo mới hoạt động trên tổ chức của mình.

Nếu người dùng bấm vào một hoạt động bất kỳ, ứng dụng sẽ chuyển hướng đến **màn hình hiển thị danh sách người tham gia** để xem chi tiết của hoạt động đó.

4.2.7. Màn hình tạo hoạt động

The image displays three sequential mobile app screens for creating an activity. Each screen has a top bar with a back arrow and the title 'Tạo hoạt động'.
Screen 1: Features a pencil icon and a text input field for 'Tên hoạt động'. A blue 'Tiếp tục' button is at the bottom.
Screen 2: Features a pencil icon. It has input fields for 'Ngày' (22/12/2019), 'Bắt đầu - kết thúc' (18 giờ 35 phút), and another '18 giờ 35 phút' field. A blue 'Tiếp tục' button is at the bottom.
Screen 3: Features a pencil icon and a large text input field for 'Nội dung'. A blue 'Tạo' button is at the bottom.
All screens have a bottom navigation bar with icons for home, camera, and profile.

Hình 7 : Màn hình tạo hoạt động.

(1- Người dùng vừa bấm vào nút **Tạo hoạt động**).

(2- Tùy chỉnh các thông số thời gian).

(3- Cài đặt mô tả cho hoạt động).

Mô tả cách sử dụng và xử lý trên màn hình :

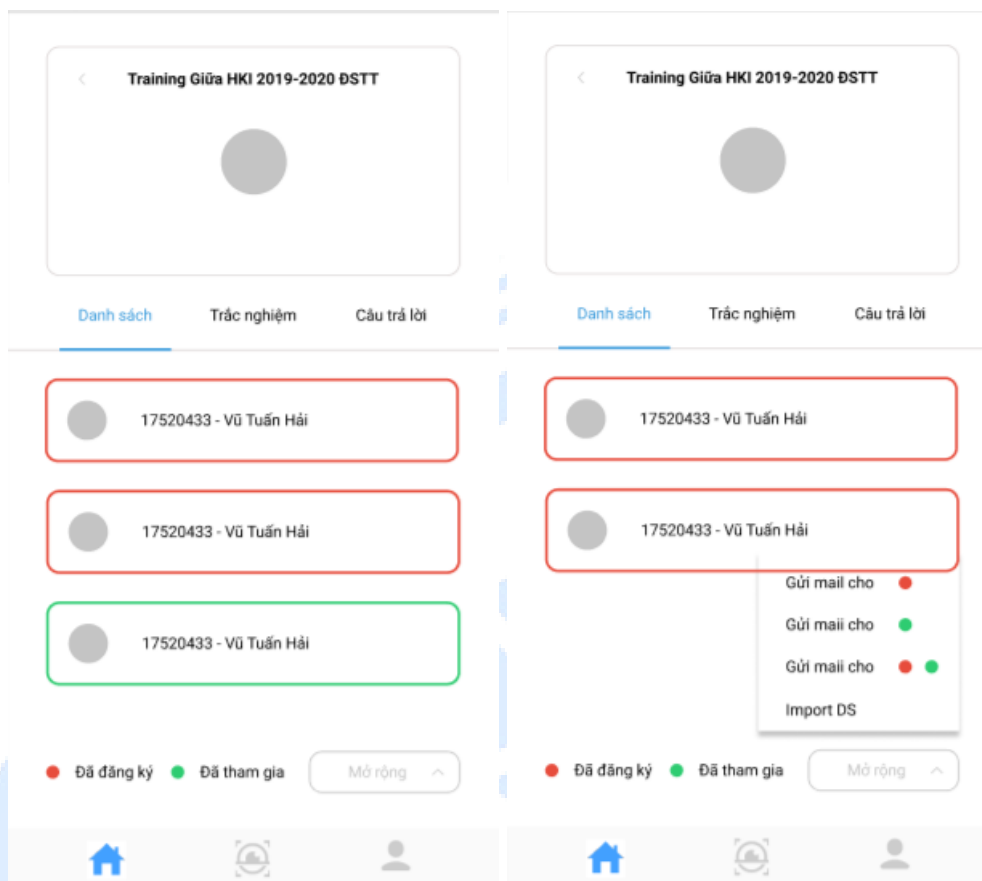
Khi người dùng nhấn vào **Tạo hoạt động**, màn hình như hình 1 sẽ được hiện ra, cho phép người dùng nhập vào tên của hoạt động. Sau khi nhập xong tên của hoạt động, người dùng tiếp tục nhấn vào nút **Tiếp theo**.

Màn hình 2 sẽ xuất hiện sau khi người dùng nhấn **Tiếp theo** ở màn hình 1. Ở màn hình này, người dùng có thể cài đặt **ngày tổ chức cho hoạt động, thời điểm bắt đầu, thời điểm kết thúc** của hoạt động. Sau khi hoàn tất việc cài đặt các thông số, người dùng nhấn vào nút **Tiếp theo**.

Màn hình 3 sẽ xuất hiện sau khi người dùng nhấn **Tiếp theo** ở màn hình 2. Ở màn hình này, người dùng nhập vào **mô tả của hoạt động**. Cuối cùng, để hoàn tất quá trình tạo hoạt động và lưu lại những thay đổi, người dùng nhấn nút **Tạo**. Sau khi nhấn **Tạo**, người dùng sẽ được đưa trở lại **màn hình danh sách hoạt động**. Ngược lại nếu

có lỗi xảy ra trong quá trình tạo mới hoạt động, ứng dụng sẽ hiển thị thông báo để người dùng được biết.

4.2.8. Màn hình hiển thị danh sách người tham gia



(Hình 8 : 1.Danh sách người tham gia hoạt động).




(2. Dialog hiển thị khi người dùng nhấn nút **Mở rộng**).

Mô tả cách sử dụng và xử lý trên màn hình :

Ở phần trên cùng của màn hình, người dùng có thể quan sát được tên của hoạt động và ảnh đại diện của hoạt động đó.

Ở phần dưới của màn hình, người dùng có thể xem **Danh sách người tham gia**. Những người tham gia được tô màu đỏ là những người tham gia **Đã đăng ký** vào hoạt động này. Những người tham gia tô màu xanh là những người tham gia **Đã tham tham** vào hoạt động này.

Có thêm một nút **Mở rộng**, người dùng có thể bấm vào để thực hiện thêm các thao tác nâng cao.

- Gửi mail cho  : Gửi mail cho những người đã đăng ký.
- Gửi mail cho  : Gửi mail cho những người đã tham gia.
- Gửi mail cho  : Gửi mail cho tất cả những người đã đăng ký / đã tham gia.
- Import DS : Nhập danh sách người đăng ký / tham gia từ file Excel. Khi người dùng ấn vào mục này, ứng dụng sẽ chuyển người dùng đến **màn hình nhập danh sách người tham gia**.

2.2.9. Màn hình nhập danh sách người tham gia



(Hình 9 : 1.Nhập đường dẫn tới google sheet).

(2. Tiến độ xử lý danh sách).

(3. Đã xử lý xong danh sách).

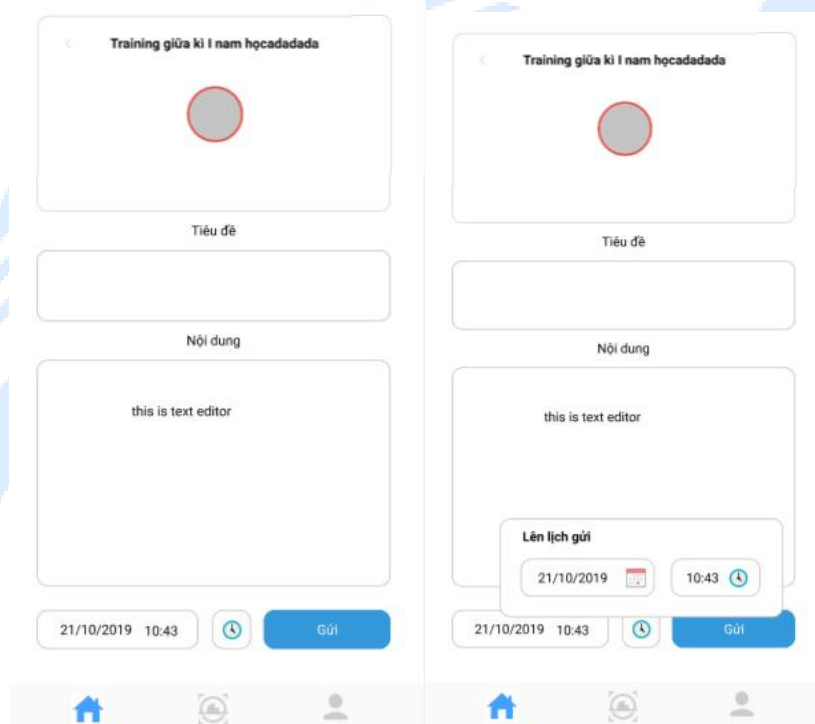
Mô tả cách sử dụng và xử lý trên màn hình :

Hình 1 sẽ xuất hiện khi người dùng ấn Import DS từ **màn hình hiển thị danh sách người tham gia**.

Hình 1 gồm 2 phần : Phần **EditText** và nút **Tiếp tục**. Khi bấm vào **EditText**, một hộp thoại cho phép người dùng chọn file sẽ được hiển thị. Sau khi chọn file, đường dẫn đến file sẽ xuất hiện trong **EditText** này. Tiếp theo, người dùng bấm nút **Tiếp tục** để thực hiện xử lý danh sách được đưa vào.

Sau khi danh sách được xử lý xong, nút **Tiếp tục** sẽ xuất hiện. Người dùng có thể bấm vào đây để trở về lại màn hình trước đó.

4.2.9. Màn hình gửi mail



(Hình 9 : 1.Màn hình gửi Email).

(2. Dialog xuất hiện khi người dùng bấm vào biểu tượng **Đồng hồ**).

Mô tả cách sử dụng và xử lý trên màn hình :

Phần trên cùng của màn hình gửi mail chính là thông tin của hoạt động, bao gồm **Tên của hoạt động**, và **ảnh đại diện của hoạt động**. Màu của vòng tròn xung quanh cho biết loại đối tượng mà mail này sẽ gửi đến là gì (Đã đăng ký : ●, đã tham gia: ●).

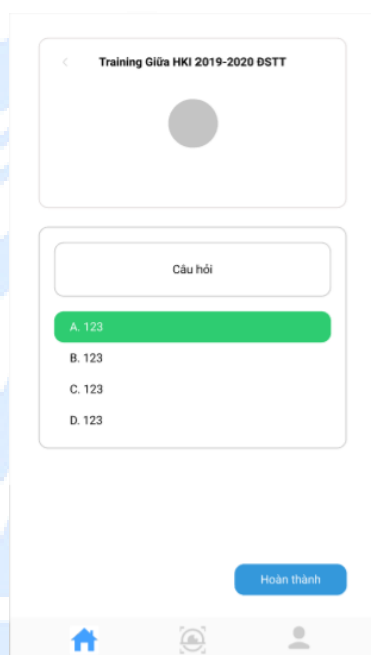
Bên dưới **thông tin hoạt động** là **Tiêu đề** của email.

Bên dưới **Tiêu đề** của email là **Nội dung** email.

Để bổ sung thêm cho tính năng gửi mail, ứng dụng còn có tính năng **lên lịch cho mail** thông qua việc cài đặt thời gian ở phần dưới cùng của màn hình.

Sau khi mọi thứ đã được chuẩn bị xong, người dùng đơn giản bấm vào nút **Gửi**.

4.2.10. Màn hình hiển thị danh sách câu hỏi (View người tham gia)



Hình 10: Màn hình danh sách câu hỏi view người tham gia

Mô tả cách sử dụng và xử lý trên màn hình :

Phần trên là thông tin hoạt động như đã được giải thích rõ ở các màn hình trước.

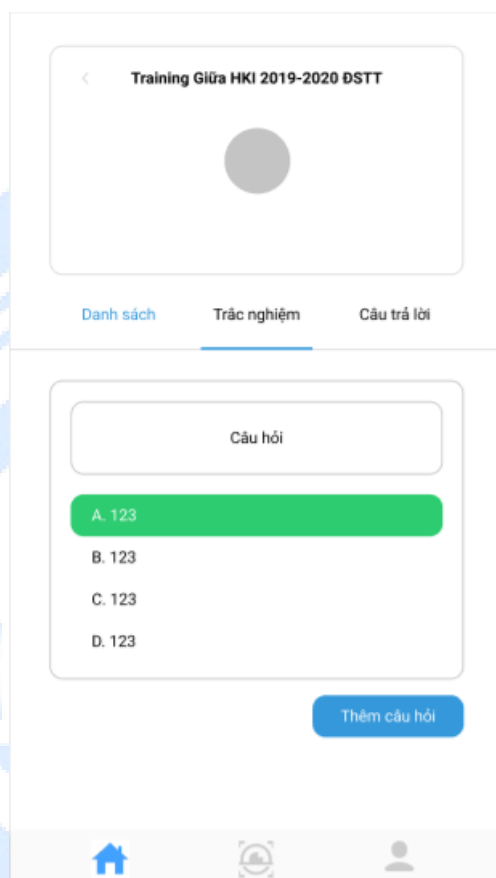
Ở phần dưới là danh sách các câu hỏi để người dùng trả lời.

Ứng với mỗi câu hỏi sẽ có 2 loại :

- **Trắc nghiệm** : Mỗi câu hỏi sẽ có từ 2 -> Nhiều đáp án. Người dùng có thể chọn 1 hoặc nhiều câu trả lời.
- **Tự luận** : Phần câu trả lời sẽ là **Text** (thay vì cho người dùng lựa chọn như **Trắc nghiệm**).

Sau khi trả lời các câu hỏi, người dùng bấm nút **Hoàn thành** để ghi nhận phần trả lời của mình lên hệ thống. Nếu trước khi người dùng bấm **Hoàn thành** mà hệ thống ghi nhận đã hết thời gian để trả lời các câu hỏi, hệ thống sẽ tự động lưu lại thông tin trả lời của người dùng và khoá nút **Hoàn thành**.

4.2.11. Màn hình hiển thị danh sách câu hỏi (View người tổ chức)



Hình 11: Màn hình danh sách câu hỏi với góc nhìn người tổ chức

Mô tả cách sử dụng và xử lý trên màn hình :

Ở đây hiển thị các câu hỏi mà đã được thêm vào trước đó. Khi bấm vào một câu hỏi, ứng dụng sẽ chuyển đến **màn hình thêm/chỉnh sửa câu hỏi cho một Event**. Đối với câu hỏi trắc nghiệm, câu trả lời được tô màu xanh là câu trả lời đúng.

Ngoài ra, nếu muốn thêm một câu hỏi mới, người dùng có thể bấm vào nút Thêm câu hỏi. Khi đó, ứng dụng cũng sẽ chuyển đến **màn hình thêm/chỉnh sửa câu hỏi cho một Event**.

Các thao tác của người dùng sẽ được tự động lưu lại.

4.2.12. Màn hình thêm/ chỉnh sửa câu hỏi cho một event

The image displays three screenshots of a mobile application interface for managing event questions and answers. The top two screenshots show the 'Thêm câu trả lời' (Add answer) screen, and the bottom screenshot shows the 'Đồng ý' (Agree) screen.

Screenshot 1 (Top Left): The screen has a title bar 'Thêm câu trả lời'. Below it is a text input field labeled 'Nhập câu hỏi'. There are four answer options, each with a checkbox and a label: 'Câu trả lời thứ 1', 'Câu trả lời thứ 2', 'Câu trả lời thứ 3', and 'Câu trả lời thứ 4'. Each option has a red 'X' mark to its right. At the bottom, there is a blue button labeled 'THÊM CÂU TRẢ LỜI' and a blue button labeled 'ĐỒNG Ý'. A note at the bottom says 'Bấm vào câu trả lời đúng để chọn câu trả lời đúng.'

Screenshot 2 (Top Right): Similar to Screenshot 1, but with five answer options: 'Câu trả lời thứ 1' through 'Câu trả lời thứ 5'. The blue button 'THÊM CÂU TRẢ LỜI' is visible at the bottom.

Screenshot 3 (Bottom): The screen shows the 'Đồng ý' screen. It has a title bar 'Đồng ý'. Below it is a text input field labeled 'Nhập câu hỏi'. There are three answer options: 'Câu trả lời thứ 1', 'Câu trả lời thứ 2', and 'Câu trả lời thứ 3'. Each option has a red 'X' mark to its right. At the bottom, there is a blue button labeled 'THÊM CÂU TRẢ LỜI' and a blue button labeled 'ĐỒNG Ý'. A note at the bottom says 'Bấm vào câu trả lời đúng để chọn câu trả lời đúng.'

(Hình 12 : 1. Màn hình thêm/chỉnh sửa câu hỏi cho một Event).

(2. Màn hình thêm/chỉnh sửa câu hỏi cho một Event sau khi bấm **Thêm câu trả lời**).

(2. Màn hình thêm/chỉnh sửa câu hỏi cho một Event sau khi bấm **Xoá 2 câu trả lời** từ hình 2).

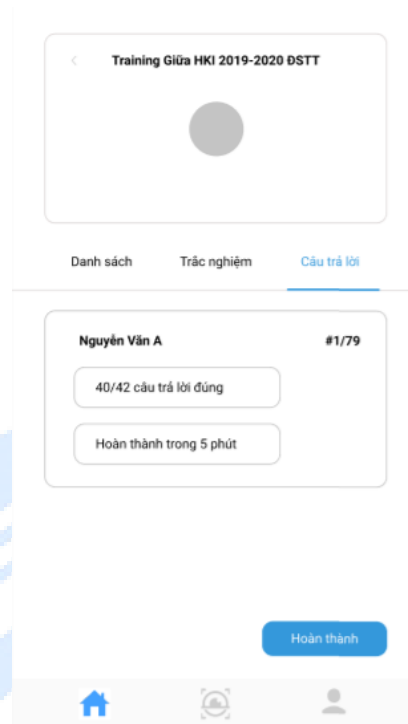
Mô tả cách sử dụng và xử lý trên màn hình :

Giao diện trực quan với 3 phần phân rõ rệt :

- Phần câu hỏi : Người dùng nhập vào câu hỏi.
- Phần câu trả lời : Người dùng có thể thêm không giới hạn số lượng câu trả lời cho một câu hỏi.
 - Ở bên trái từng câu hỏi là một **dấu Tích**. Dấu tích đó tượng trưng cho câu trả lời chính xác. Người dùng có thể chọn 1 câu trả lời đúng hoặc nhiều câu trả lời đúng (câu hỏi chỉ được tính là câu trả lời đúng nếu như người dùng chọn tất cả câu trả lời đúng).
 - Ở bên phải câu hỏi là một **dấu X**. Dấu X đó tượng trưng cho thao tác xoá câu hỏi
- Phần thao tác : Có hai thao tác chính :
 - **Thêm câu trả lời** : Người dùng khi bấm vào đây, ứng dụng sẽ hiển thị thêm một trường câu trả lời để người dùng có thể nhập vào.

Đồng ý : Khi người dùng nhấn vào đây, cửa sổ **Thêm/chỉnh sửa câu hỏi** sẽ đóng lại và kết quả người dùng nhập vào sẽ được lưu lại.

4.2.13. Màn hình hiển thị câu trả lời của người tham gia

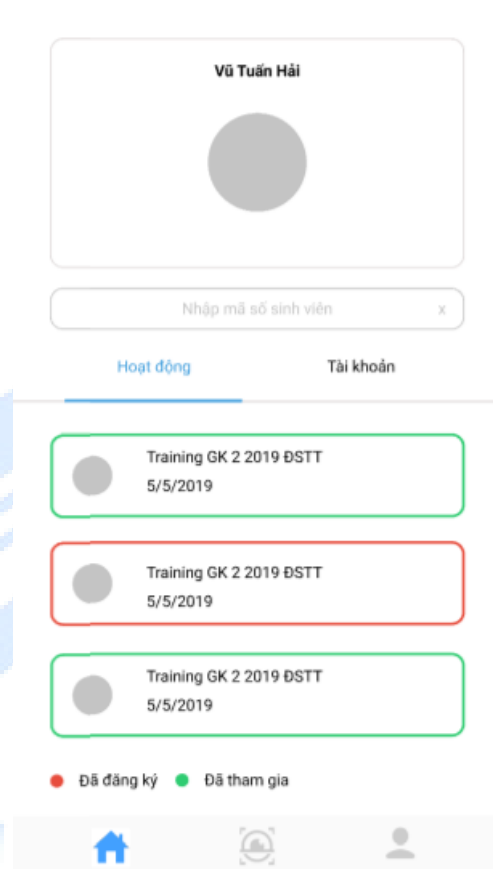


Hình 13: Màn hình thống kê câu trả lời

Mô tả cách sử dụng và xử lý trên màn hình :

Màn hình này hiển thị kết quả trả lời của từng người trong hoạt động được chọn. Những thông tin hiển thị bao gồm : **Tên người tham gia hoạt động, Số câu trả lời đúng, Thời gian hoàn thành câu hỏi và Xếp hạng của người dùng** (dựa vào số câu trả lời đúng). Màn hình này người dùng chỉ có thể xem chứ không thể chỉnh sửa thêm.

4.2.14. Màn hình hiển thị danh sách hoạt động mà người dùng đã đăng ký/ tham gia



Hình 14: Màn hình danh sách hoạt động của user

Mô tả cách sử dụng và xử lý trên màn hình :

Màn hình liệt kê danh sách các hoạt động mà người dùng đã đăng ký/tham gia. Thông tin được liệt kê gồm có **Tên hoạt động**, **Ngày tổ chức hoạt động**, **Hình thức tham gia hoạt động** (Đăng ký ●, Tham gia ●). **Hoạt động được viền khung màu đỏ** là những hoạt động người dùng đã đăng ký. Những **hoạt động được viền màu xanh** là những hoạt động mà người dùng đã tham gia.

4.2.15. Màn hình hiển thị thông tin người dùng

Hình 15: Màn hình thông tin cá nhân của user

Mô tả cách sử dụng và xử lý trên màn hình :

Đây là nơi mà người dùng có thể xem và chỉnh sửa các thông tin liên quan đến tài khoản cá nhân của mình. Những thông tin mà người dùng có thể chỉnh sửa bao gồm : **Tên người dùng, Ảnh đại diện, MSSV, Email, Số điện thoại.**

Khi người dùng muốn chỉnh sửa bất kỳ mục nào trong đây, người dùng chỉ cần đơn giản nhấn vào mục đó. Khi đó, biểu tượng **Cây viết chì màu đen** sẽ được hiển thị ở kế bên trường thông tin mà người dùng nhấn vào. Khi người dùng bắt đầu thực hiện chỉnh sửa, lập tức **Cây viết chì màu đen** sẽ được chuyển thành **dấu Tick**. Người dùng chỉnh sửa xong có thể bấm vào **dấu Tick** để lưu lại những thông tin đã được chỉnh sửa ở trường đó. Nếu người dùng không muốn thực hiện chỉnh sửa nữa, người dùng có thể bấm ra ngoài. Khi đó **dấu Tick** sẽ biến mất và người dùng sẽ cần thao tác lại từ đầu (như đã ghi ở trên) nếu muốn chỉnh sửa.

Tất cả các trường sau khi chỉnh sửa sẽ được kiểm tra thông tin hợp lệ hay không. Nếu hợp lệ, hệ thống sẽ ghi nhận lại những thay đổi. Nếu không hợp lệ, hệ thống sẽ xuất hiện thông báo lỗi và huỷ bỏ những thay đổi đó của người dùng.

Chương 5 : Giới thiệu các thư viện, công nghệ liên quan.

5.1 Tesseract

Nhận dạng ký tự quang học (Optical Character Recognition) là công việc xử lý hình ảnh, trích xuất văn bản thành các ký tự có thể sử dụng trong máy tính. Một cách dễ hiểu, với input là 1 bức ảnh với định dạng bất kỳ, phương pháp này sẽ tạo ra output là hệ thống các ký tự (chữ cái, chữ số, accs ký hiệu) có trong bức ảnh.

Trên lý thuyết đó, Tesseract được ra đời là một phần mềm mã nguồn mở được nghiên cứu và phát triển tại Hewlett Packet có độ chính xác nhận diện đứng đầu về bộ nhận dạng OCR.

Để hoạt động, Tesseract nhận vào 1 bức ảnh màu hoặc ảnh xám, sau đó chuyển đổi về ảnh nhị phân bằng phương pháp phân tích ngưỡng động (Adaptive thresholding). Sau đó nó sẽ thực hiện tuần tự các bước sau:

Bước 1: Phân tích các thành phần liên thông, giúp đưa ra đường bao quanh của các ký tự.

Bước 2: Tìm hàng và tìm từ, tạo ra đường bao giữa các hàng chữ, các ký tự trong vùng văn bản.

Bước 3: Nhận dạng ký tự đã được bao ở bước trước đó. Bước này được thực hiện qua 2 giai đoạn. Giai đoạn 1 phân loại từ theo lượt. Những từ thỏa mãn sẽ được chuyển qua bộ phân loại thích ứng (Adaptive classifier) để làm train data, nhờ đó nó sẽ nhận diện tốt hơn ở phần sau của trang. Sau khi bộ phân loại thích ứng đã học được các thông tin hữu ích từ giai đoạn 1 thì giai đoạn 2 sẽ được thực hiện. Giai đoạn này thực hiện quét toàn bộ

trang, các từ không được nhận diện chính xác ở giai đoạn 1 sẽ được nhận diện lại. Cuối cùng sẽ tổng hợp thông tin và đưa ra kết quả hoàn chỉnh.

Tesseract hỗ trợ nhận diện trên nhiều ngôn ngữ và nhiều font chữ khác nhau, ta có thể sử dụng data đã được huấn luyện và sử dụng. Hoặc cũng có thể tự train dữ liệu bằng cách để các tập tin dữ liệu trong thư mục tessdata theo định dạng: ten-ngon-ngu.ten-tep-tin.

Trong phạm vi đồ án, nhóm đã sử dụng dữ liệu train sẵn dạng ngôn ngữ là tiếng anh. Ngay khi ứng dụng khởi chạy lần đầu, thông tin dữ liệu đã train sẽ truyền qua lưu tại bộ nhớ thiết bị và sử dụng để nhận diện cho những lần tiếp theo.

5.2. Firebase

Firebase là nền tảng giúp lập trình ứng dụng di động và web với các công cụ, hạ tầng được thiết kế để hỗ trợ lập trình viên phát triển các ứng dụng chất lượng.

5.2.1. Realtime Database

Firebase lưu trữ dữ liệu database dưới dạng JSON và thực hiện đồng bộ database với tất cả client theo thời gian thực. Chúng ta có thể xây dựng ứng dụng đa nền tảng và sử dụng cùng 1 cơ sở dữ liệu đồng bộ với nhau. Firebase cũng hỗ trợ phân quyền bằng cách viết code tương tự javascript.

Tất cả dữ liệu được truyền qua kết nối an toàn SSL với 1 chứng nhận 2048 bit. Cơ sở dữ liệu truy vấn và việc xác nhận được điều khiển tại một cấp độ chi tiết sử dụng theo một số các quy tắc mềm dẻo security rules language. Tất cả các logic bảo mật dữ liệu của bạn được tập trung ở một chỗ để dễ dàng cho việc cập nhật và kiểm thử.

Firebase hỗ trợ lưu trữ dữ liệu local khi bị trục trặc với internet, ngay sau khi được kết nối lại, client sẽ được đồng bộ ngay với server.

5.2.2. Firebase Authentication

Với Firebase, chúng ta có thể dễ dàng xác thực người dùng từ ứng dụng của bạn trên Android, iOS và JavaScript SDKs chỉ với một vài đoạn mã. Firebase đã xây dựng chức năng cho việc xác thực người dùng với Email, Facebook, Twitter,

GitHub, Google, và xác thực nặc danh. Các ứng dụng sử dụng chức năng xác thực của Firebase có thể giải quyết được vấn đề khi người dùng đăng nhập, nó sẽ tiết kiệm thời gian và rất nhiều các vấn đề phức tạp về phần backend. Hơn nữa bạn có thể tích hợp xác thực người dùng với các chức năng backend đã có sẵn sử dụng custom auth tokens.

Chương 6 : Các thành phần xử lý chính.

6.1. Bộ xử lý dữ liệu trung tâm

DataManager là lớp xử lý tất cả dữ liệu liên quan đến firebase như truy vấn dữ liệu, thực hiện xác thực tài khoản người dùng.

6.2. Bộ nhận diện diện ký tự

Để sử dụng tesseract chúng ta cần có 1 thể hiện của lớp TessBaseAPI Có sẵn khi import Tess-two 9.0.0.

Ngoài ra, để quá trình nhận diện được thực thi ta cần chuyển dữ liệu đã huấn luyện qua bộ nhớ ứng dụng.

```
20 public class Tesseract {
21
22     private final TessBaseAPI mTesseract;
23
24     String mPathSaveData="/tesseract/tessdata/";
25     String mPathToInitTesseract;
26     String mPathFileSave;
27     String mTrainedDataNamel="eng.traineddata";
28
29     static final Integer READ_STORAGE_PERMISSION_REQUEST_CODE=0x3;
30
31     public static boolean checkPermissionForReadExtertalStorage(Context _context) {
32         if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
33             int result = _context.checkSelfPermission(Manifest.permission.READ_EXTERNAL_STORAGE);
34             return result == PackageManager.PERMISSION_GRANTED;
35         }
36         return false;
37     }
38
39     public static void requestPermissionForReadExtertalStorage(Context _context) throws Exception {
40         try {
41             ActivityCompat.requestPermissions((Activity) _context, new String[]{Manifest.permission.READ_EXTERNAL_STORAGE},
42                 READ_STORAGE_PERMISSION_REQUEST_CODE);
43         } catch (Exception e) {
44             e.printStackTrace();
45             throw e;
46         }
47     }
48 }
```

```

50     public Tesseract(Context context,String language) {
51
52         mTesseract=new TessBaseAPI();
53         AssetManager assetManager=context.getAssets();
54         InputStream inputFile=null;
55         FileOutputStream outputFile=null;
56
57         //add pre root dir
58         mPathSaveData=context.getFilesDir()+mPathSaveData;
59         mPathToInitTesseract=context.getFilesDir()+"/tesseract";
60         mPathFileSave=mPathSaveData+mTrainedDataName1;
61
62         //start transfer data from asset to dir
63         boolean isCreated=false;
64         try {
65             //open source data
66             inputFile=assetManager.open(mTrainedDataName1);
67
68             //create folder to save data
69             File fileToSave=new File(mPathSaveData);
70
71             if(!fileToSave.exists()){
72                 if(!fileToSave.mkdirs()){
73                     Toast.makeText(context, "Can't init data from OCR", Toast.LENGTH_SHORT).show();
74                 }
75
76                 //create stream file to save
77                 outputFile=new FileOutputStream(new File(mPathFileSave));
78             }

```

```

79         } else {
80             isCreated=true;
81         }
82     } catch (IOException e) {
83         e.printStackTrace();
84     }
85     finally {
86
87         //if file already create and transfer data
88         if(isCreated){
89             if(inputFile!=null) {
90                 try {
91                     inputFile.close();
92                 } catch (IOException e) {
93                     e.printStackTrace();
94                 }
95             }
96             mTesseract.init(mPathToInitTesseract,language);
97             return;
98         }
99
100         if (inputFile != null && outputFile != null) {
101             try {
102                 //copy file
103                 byte[] buf = new byte[1024];
104                 int len;
105                 while ((len = inputFile.read(buf)) != -1) {
106                     outputFile.write(buf, 0, len);
107                 }
108                 inputFile.close();
109                 outputFile.close();
110                 mTesseract.init(mPathToInitTesseract, language);

```

```

111             } catch (Exception ex) {
112                 ex.printStackTrace();
113             }
114         } else {
115             Toast.makeText(context, "Can't transfer data OCR", Toast.LENGTH_SHORT).show();
116         }
117     }
118
119 }
120

```

Việc còn lại chỉ đơn giản là lấy chuỗi ký tự kết quả mà ta nhận diện được từ hình ảnh và hủy bỏ thể hiện khi không còn dùng đến.

```
121     public String getOCRResult(Bitmap bitmap) {
122         mTesseract.setImage(bitmap);
123         return mTesseract.getUTF8Text();
124     }
125
126
127     public void onDestroy() {
128         if (mTesseract != null) mTesseract.end();
129     }
```

6.3. Xử lý hình ảnh từ camera

Lớp cameraPreview có nhiệm vụ quét và lấy mã số sinh viên dựa trên các khu vực được chỉ định và Tesseract API.

```
26 public class CameraPreview extends SurfaceView implements SurfaceHolder.Callback, Camera.AutoFocusCallback,
27
28     //Biến này dùng để quản lý marginTop và marginRight của khung dùng để cắt MSSV (theo phần trăm).
29     //VD : Nếu giá trị = 0.2 thì tức là chỉ lấy 20% độ cao màn hình.
30     final double percentHeightMSSV = 0.2;
31     final double percentWidthMSSV = 0.2;
32
33     //Kích cỡ của Focus Area.
34     final float focusAreaSize = 72;
35
36     //Delay để lấy một frame.
37     final double frameCaptureDelay = 3;
38
39     //Tính thời gian đã qua.
40     double timeHolder = System.currentTimeMillis()/1000;
41
42     //Giữ frame hiện tại, để chúng ta có thể thao tác với frame hiện tại.
43     private final Lock lock = new ReentrantLock();
44     private byte[] mPreviewFrameBuffer;
45
46     private SurfaceHolder mHolder;
47
48     private static Camera cameraInstance;
49
50     private Camera mCamera;
51     private boolean meteringAreaSupported = false;
52
53     Tesseract tesseract;
```

```

57     public CameraPreview(Context context) {
58         super(context);
59         this.context = context;
60
61         //Ảnh xạ.
62         tesseract = new Tesseract(context, "eng");
63
64         Log.d("onCreateCameraPreview", "Camera Preview Created !");
65
66         mHolder = getHolder();
67
68         mHolder.setType(SurfaceHolder.SURFACE_TYPE_PUSH_BUFFERS);
69
70         //Chỉnh cài đặt cho sự kiện chạm vào màn hình để lấy nét.
71         onTouchListener touchListener = new onTouchListener() {
72             @Override
73             public boolean onTouch(View v, MotionEvent event) {
74                 Log.d("onTouchCameraPreview", "Camera Preview Touched !");
75                 focusOnTouch(event);
76                 return false;
77             }
78         };
79         this.setOnTouchListener(touchListener);
80     }

```

```

83     //Hàm này sẽ được gọi khi người dùng chạm tay vào màn hình để lấy nét.
84     protected void focusOnTouch (MotionEvent event){
85         if (mCamera != null){
86             Log.d("focusOnTouch", "Success !");
87             mCamera.cancelAutoFocus();
88             Rect focusRect = getTapArea(event.getX(), event.getY(), 1f);
89             Rect meteringRect = getTapArea(event.getX(), event.getY(), 1.5f);
90
91             //Danh sách những vùng sẽ được Camera focus.
92             List focusArea = new ArrayList<Camera.Area>();
93             focusArea.add(new Camera.Area(focusRect, 1000));
94             List meteringArea = new ArrayList<Camera.Area>();
95             meteringArea.add(new Camera.Area(meteringRect, 1000));
96
97             //Chỉnh lại chế độ focus của camera.
98             Camera.Parameters parameters = mCamera.getParameters();
99             parameters.setFocusMode(Camera.Parameters.FOCUS_MODE_AUTO);
100
101             //Cài đặt vùng focus của Camera.
102             parameters.setFocusAreas(focusArea);
103
104             if (meteringAreaSupported){
105                 parameters.setMeteringAreas(meteringArea);
106             }
107
108             mCamera.setParameters(parameters);
109             mCamera.autoFocus(this);
110         }
111         else{
112             Log.d("focusOnTouch", "Camera is null");
113         }
114     }

```

6.4. Adaptor

Mỗi listview đều có 1 adaptor lưu trữ thông tin riêng biệt, tuy nhiên đều được kế thừa từ ArrayAdapter. Để việc load các listview item mượt hơn, nhóm sử dụng kỹ thuật dùng viewholder để lấy row template 1 lần sau đó sử dụng lại.


```

18 public class EventAdapter extends ArrayAdapter {
19     Context context;
20     List<Event> lsEvent;
21
22     public EventAdapter(@NonNull Context context, List<Event> ls) {
23         super(context, R.layout.custom_row_event, ls);
24         this.context=context;
25         lsEvent=ls;
26     }
27

```

```

29 @NonNull
30 @Override
31 public View getView(int position, @Nullable View convertView, @NonNull ViewGroup parent) {
32     LayoutInflater inflater =(LayoutInflater)context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
33
34     View viewRow= convertView;
35
36     if(viewRow==null){
37         viewRow=inflater.inflate(R.layout.custom_row_event,parent,false);
38         ViewHolder holder=new ViewHolder();
39         holder.imageView =viewRow.findViewById(R.id.avtRowEvent);
40         holder.title =viewRow.findViewById(R.id.nameRowEvent);
41         holder.date =viewRow.findViewById(R.id.dateRowEvent);
42         holder.border =viewRow.findViewById(R.id.rowEvent);
43
44         viewRow.setTag(holder);
45     }
46
47     ViewHolder viewHolder= (ViewHolder) viewRow.getTag();
48
49     //change color if this is event just register and not take part in
50     if(position==3)
51         viewHolder.border.setBackgroundResource(R.drawable.custom_row_event_orange);
52
53     viewHolder.imageView.setImageResource(R.drawable.ninja_avt);
54
55     Event e=lsEvent.get(position);
56     viewHolder.title.setText(e.getEvent_name());
57     viewHolder.date.setText(e.getEvent_day());
58
59     return viewRow;
60 }

```

View holder dành cho listview sự kiện:

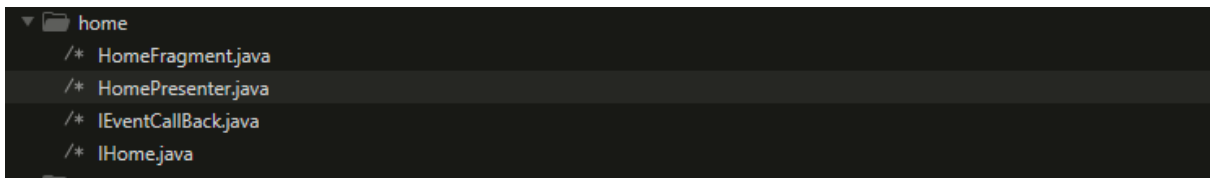
```

62 public static class ViewHolder{
63     View border;
64     CircularImageView imageView;
65     TextView title;
66     TextView date;
67 }

```

6.5. Bộ xử lý logic và cập nhật giao diện

Nhờ cấu trúc chương trình theo mô hình MVP, các thành phần xử lý được tách biệt độc lập nhất có thể để dễ dàng chỉnh sửa, nâng cấp và kiểm thử. Nhìn chung các màn hình đều có cấu trúc như sau:



- Fragment chịu trách nhiệm hiển thị view và các cập nhật liên quan đến giao diện.
- Presenter chịu trách nhiệm xử lý logic, có quan hệ 1-1 với Fragment.
- IHome là interface tạo các ràng buộc để tránh sai sót trong quá trình triển khai.
- Callback thực hiện theo Observer pattern giúp tự động thông báo kết quả xử lý dữ liệu dưới tầng network lên Presenter để cập nhật giao diện khi cần.

Đoạn code về việc callback khi xử lý dữ liệu, ngay lập tức view sẽ được cập nhật.

```
50 //callback event
51 @Override
52 public void OnLoadEventComplete(Event event) {
53     if(event!=null){
54         if(DataManager.Instance().SaveAttendance(DataCenter.UserID,event.getEvent_id(),DataCenter.UserDisplayName))
55             view.OnJoinEventSucess(event);
56         else
57             view.OnJoinEventFail(event);
58     }
59 }
60
61 else
62     view.OnJoinEventFail(event);
63 }
```

Chương 7 : Cài đặt và sử dụng.

7.1. Cài đặt

1. Clone mã nguồn chương trình với cmd bằng lệnh: git clone <https://github.com/the-document/Attendance-Manager.git>
2. Mở mã nguồn bằng Visual studio hoặc IDE khác có hỗ trợ.
3. [Tùy chọn] Build và chạy trên máy ảo
4. [Tùy chọn] Build và chạy trên điện thoại. Thiết bị phải để ở chế độ dành cho nhà phát triển
5. [Tùy chọn] Build thành file apk và cài đặt lên thiết bị muốn sử dụng.

7.2. Sử dụng

7.2.1. Đăng ký tài khoản

Tại màn hình đăng nhập, người dùng chưa có tài khoản sẽ nhấn vào dòng chữ “Chưa có tài khoản ? Bấm vào đây !” để chuyển sang màn hình đăng ký. Sau khi đã điền đầy đủ tất cả các thông tin, người dùng nhấn đăng ký tài khoản. Hệ thống sẽ gửi mail xác nhận đến tài khoản email đã đăng ký xác thực và sau khi người dùng xác thực thông qua email thì tài khoản đã được đăng ký thành công.

7.2.2 Đăng nhập

Người dùng đăng nhập bằng email và mật khẩu đã tạo lúc đăng ký tài khoản. Sau đó, nhấn nút đăng nhập và ứng dụng sẽ chuyển về trang chủ nếu đăng nhập thành công.

7.2.3 Tham gia bằng mã code

Tại màn hình trang chủ, phần mục điểm danh nằm trên cùng. Khi người dùng nhấn vào và nhập mã code tham gia hoạt động, ứng dụng hiển thị một dấu tick xanh để người dùng có thể click vào xác nhận mã code vừa nhập.

7.2.4 Tạo tổ chức

Tại màn hình trang chủ, người dùng nhấn vào button “Tạo mới” tổ chức trong mục “Quản lý tổ chức của bạn”. Sau đó, màn hình nhập thông tin tổ chức được hiển thị và người dùng nhập đầy đủ các thông tin. Nhấn lưu lại và kết thúc.

7.2.5 Tạo sự kiện

Người dùng chọn một tổ chức cần tạo hoạt động ở màn hình chính. Tiếp theo, ở phía trên màn hình quản lý hoạt động là mục “Tạo hoạt động”. Sau khi người dùng thao tác với mục tạo, ứng dụng hiển thị sang trang nhập thông tin cho hoạt động. Lướt sang trái để nhập đầy đủ thông tin ở từng trang và nhấn “Tạo” ở trang cuối cùng.

7.2.6 Tạo bộ câu hỏi

Người dùng chọn hoạt động cần tạo bộ câu hỏi trong mục “Hoạt động đã tổ chức” ở màn hình quản lý hoạt động. Tiếp theo, chọn mục “Trắc nghiệm” và nhấn vào button dấu cộng (+) góc dưới bên phải màn hình để thêm bộ câu hỏi.

7.2.7 Xem thống kê câu trả lời

Sau khi chọn hoạt động cần xem thống kê ở màn hình quản lý hoạt động, người dùng tiếp tục chọn mục “Câu trả lời” ở màn hình tiếp theo.

7.2.8 Xem thông tin người dùng

Người dùng nhấn vào mục “Thông tin” ở thanh Navigation Bar, sau đó tiếp tục chọn mục “Tài khoản” để được chuyển đến màn hình thông tin người dùng.

7.2.9 Gửi email

Sau khi chọn hoạt động ở màn hình quản lý hoạt động, người dùng nhấn vào button “Mở rộng” ở mục “Danh sách” trong màn hình chi tiết hoạt động. Tiếp tục chọn mục “Gửi mail cho người tham gia”, ứng dụng sẽ chuyển sang trang nhập thông tin email (tiêu đề, nội dung). Sau khi nhấn button “Gửi”, bạn sẽ được chuyển đến ứng dụng gửi mail thứ ba và có thể tiếp tục chỉnh sửa nội dung mail trước khi gửi.

Chương 8 : Kết luận.

8.1. Nhận xét

8.1.1. Kết quả đạt được

Thông qua đồ án môn học, dưới sự hướng dẫn của thầy, nhóm đã cùng nhau tìm hiểu, học tập được thêm nhiều kiến thức mới về lập trình ứng dụng trên thiết bị di động, cách xây dựng phần mềm theo hướng đối tượng, các kiến thức mới về design pattern (Singleton, Observer), lập trình đa luồng, kiến thức về xử lý ảnh thông qua sử dụng api của google cung cấp.

Cùng với đó, chỉ trong thời gian hạn chế, phần mềm được phát triển tương đối hoàn thiện. Có khả năng đáp ứng nhu cầu quản lý công việc điểm danh và hỗ trợ quản lý hoạt động cho các tổ chức có nhu cầu.

8.1.2. Ưu điểm

- Phần mềm có thể chạy trên các thiết bị android mà không cần cài đặt thêm bất cứ phần mềm thứ 3.
- Ứng dụng được thiết kế tối giản, dễ dàng sử dụng.
- Hỗ trợ nhiều phương pháp điểm danh nhanh chóng và tiện lợi.
- Độ tin cậy cao, tất cả tài khoản được xác thực trước khi sử dụng, chống spam.

8.1.3. Hạn chế

- Phương pháp điểm danh bằng quét thẻ hiệu suất chưa cao.
- Tốc độ tải dữ liệu còn chịu độ trễ, phụ thuộc quá nhiều vào tốc độ mạng.

8.1.4. Hướng phát triển trong tương lai.

Trong tương lai, ứng dụng sẽ được cung cấp thêm các tính năng hỗ trợ tương tác nhiều hơn:

- Cho phép người dùng tìm và đăng ký tham gia sự kiện.
- Thông báo trực tiếp trên ứng dụng các thông tin về sự kiện đã đăng ký.
- Cải thiện hiệu suất quét thẻ.
- Thêm chế độ đồ họa thứ 2 là DarkMode.
- Hỗ trợ upload tài liệu

