

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



ĐỒ ÁN TỔNG HỢP - CÔNG NGHỆ PHẦN MỀM

SUMMERIZE REPORT

GVHD: ThS. Lê Đình Thuận

SVTH: Nhóm Tây tiến đoàn binh không mọc tóc

Trần Thanh Phong (2212571)

Đặng Quốc Phong (2212548)

Lê Vĩnh Nghiệp (2212213)

Trần Minh Quân (2212823)

Trương Anh Tuấn (2213810)

Nguyễn Lê Hoàng Phúc (2212629)

THÀNH PHỐ HỒ CHÍ MINH, THÁNG 12 2024

Mục lục

Mục lục	i
Danh sách thành viên	ii
Danh sách bảng	iii
Danh sách hình vẽ	iv
1 Quản lý tổ chức và phân công	1
1.1 Hoạch định và lập tiến độ dự án	1
1.2 Ma trận trách nhiệm	1
1.3 Kiểm soát và giám sát tiến độ	3
2 Đặc tả yêu cầu	4
2.1 Các bên liên quan	4
2.2 Yêu cầu chức năng	5
2.3 Yêu cầu phi chức năng	6
2.4 Sơ đồ Usecase (hình 2.1)	7
3 Thiết kế hệ thống	10
3.1 Mockup	10
3.2 Sơ đồ Activity (hình 3.4)	12
3.3 Sơ đồ Sequence (hình 3.5 và hình 3.6)	13
4 Các loại tài liệu	15
4.1 Báo cáo họp	15
4.2 Cấu trúc thư mục	15
4.3 Command Guide	17
5 Docker Compose và Deployment	20
5.1 Quy trình triển khai	20
5.2 Kết quả triển khai	22

Danh sách thành viên

STT	Họ tên	MSSV	Nhiệm vụ	Đóng góp	Ký tên
1	Trần Thanh Phong	2212571	1.Hiện thực toàn bộ câu lệnh + kết nối API, 2.Deploy docker	100%	
2	Đặng Quốc Phong	2212548	1.Hiện thực toàn bộ câu lệnh + kết nối API, 2.Handle lỗi OTP, 3.Usecase diagram, 4.Deploy docker	100%	
3	Lê Vĩnh Nghiệp	2212213	1.Handle lỗi hiển thị điểm, lỗi xem lịch sử điểm 2.Báo cáo, 3.Test bot	100%	
4	Trần Minh Quân	2212823	1.Handle lỗi hiển thị thông tin cá nhân, 2.Deploy docker	100%	
5	Trương Anh Tuấn	2213810	1.Handle lỗi register, 2.Activity diagram	100%	
6	Nguyễn Lê Hoàng Phúc	2212629	1.Xây dựng cấu trúc thư mục, 2.Handle lỗi login, 3.Báo cáo	100%	
Thông tin PO:				Ký tên	
Họ và tên: Nguyễn Lê Hoàng Phúc Sđt: 0766909533 Email: phuc.nguyenlehoang707@hcmut.edu.vn					
Nhận xét của giảng viên:				Ký tên	
(Đây là khu vực nhận xét của giảng viên)					

Danh sách bảng

2.1	Yêu cầu chức năng của bot Telegram	5
2.2	Yêu cầu phi chức năng của bot Telegram	6
2.3	Bảng đặc tả Usecase 2 - Login.	8
2.4	Bảng đặc tả Usecase 6 - AllGrade.	9
4.1	Bảng tổng hợp các câu lệnh trong bot Telegram.	18

Danh sách hình vẽ

2.1	Sơ đồ Usecase mô tả chức năng của toàn bộ hệ thống.	7
3.1	Mockup giao diện giao tiếp trên điện thoại.	10
3.2	Mockup giao diện giao tiếp trên Laptop.	10
3.3	Mockup giao diện giao tiếp trên máy tính bảng.	11
3.4	Sơ đồ Activity mô tả luồng hoạt động chính của hệ thống.	12
3.5	Sơ đồ Sequence mô tả chi tiết hoạt động hệ thống khi đăng ký, đăng nhập.	13
3.6	Sơ đồ Sequence mô tả chi tiết hoạt động hệ thống khi xem toàn bộ điểm và xem điểm một môn.	14
5.1	Dockerfile	20
5.2	Lệnh tạo docker image	21
5.3	Lệnh đặt tên cho image	21
5.4	Lệnh đẩy image lên Docker Hub	21
5.5	docker-compose.yml	21

1. Quản lý tổ chức và phân công

1 Quản lý tổ chức và phân công

1.1 Hoạch định và lập tiến độ dự án

Nhóm sinh việc thực hiện việc trao đổi thông tin và giao tiếp với nhau thông qua GoogleSheet: [Link GoogleSheet](#).

Với sơ đồ Gantt, nhóm tác giả có thể dễ dàng chủ động được quá trình làm việc, đảm bảo tiến độ dự án hoàn thành đúng và đạt hiệu quả cao. Trong sơ đồ này, có các công tác Gantt (hình thành một đường Gantt), các công tác này không được phép trễ tiến độ vì bất kỳ sự chậm trễ nào của chúng đều khiến dự án kết thúc chậm hơn thời gian hoạch định ban đầu. Mặc dù đây là phương pháp quản lý dự án thông thường, được áp dụng khi các gói công việc cần làm đều tường minh (khác xa so với lĩnh vực quản lý dự án phần mềm đặc trưng bởi sự linh hoạt trong quá trình hoàn thiện sản phẩm), tuy nhiên phương pháp Gantt giúp nhóm tác giả có thể định hình được bộ khung làm việc ngay từ đầu với các giới hạn về thông tin và nguồn lực.

1.2 Ma trận trách nhiệm

Chú thích các ký hiệu:

- R (Responsible): Chịu trách nhiệm thực hiện chính.
- A (Accountable): Thông qua kết quả cuối cùng và chịu trách nhiệm về kết quả, đảm bảo hoàn thành.
- C (Consulted): Tham gia tư vấn, hỗ trợ thực hiện.
- I (Informed): Được thông báo và theo dõi tiến độ.

	Số cấu trúc	Tiêu đề công việc	Trần Thanh Phong	Đặng Quốc Phong	Lê Vĩnh Nghiệp	Trần Minh Quân	Trương Anh Tuấn	Nguyễn Lê Hoàng Phúc
1. Phân tích và thiết kế	1.1	Đặc tả yêu cầu	R	R	R	R	R	R,A
	1.2	Xây dựng cấu trúc lệnh	R	R	R	R	R	R,A

Tiếp theo trang sau

1. Quản lý tổ chức và phân công

	Số cấu trúc	Tiêu đề công việc	Trần Thanh Phong	Đặng Quốc Phong	Lê Vĩnh Nghiệp	Trần Minh Quân	Trương Anh Tuấn	Nguyễn Lê Hoàng Phúc
	1.3	Thiết kế kiến trúc	I	I	I	R	R	A
	1.4	Mô hình hóa ứng dụng (diagram)	I	R	I	R	R	C,A
2. Phát triển bot	2.1	Xây dựng cấu trúc thư mục	I	I	I	I	I	R
	2.2	Xây dựng khởi tạo bot	I	I	R	I	I	R
	2.3	Triển khai logic chi tiết	R	R	I	I	I	C,A
	2.4	Kết nối API backend	R	R	I	I	I	C,A
	2.5	Xử lý lỗi và tối ưu hiệu năng	R	R	R	R	R	R,A
	2.6	Tích hợp Docker	R	R	I	I	I	A
3. Kiểm thử và triển khai	3.1	Kiểm thử (Test)	I	I	R	I	I	A
	3.2	Triển khai thực thể (Deploy)	R	R	I	I	I	C,A
4. Báo cáo và tổng kết	4.1	Hoàn thiện báo cáo	I	I	R	R	R	A
	4.2	Chuẩn bị slide thuyết trình	I	I	I	I	I	R

1. Quản lý tổ chức và phân công

1.3 Kiểm soát và giám sát tiến độ

Theo yêu cầu, nhóm tác giả thực hiện việc kiểm soát sự án và đảm bảo dự án hoàn thành đúng tiến độ bằng cách họp mặt thành viên. Các buổi họp sẽ diễn ra online và có ghi chú lại thông tin sau mỗi buổi. Với tần suất 1 lần/tuần, diễn ra trong 10 tuần học (thường thời điểm họp là tối thứ 6 hàng tuần), các buổi họp này được tổ chức nhằm đảm bảo các thành viên hoàn thành đúng nhiệm vụ của mình được giao trong khoảng thời gian cho phép. Quá trình họp diễn ra như sau:

1. Điểm danh thành viên.
2. Tổng kết, đánh giá kết quả tuần hiện tại (có tuyên dương, phê bình).
3. Phân công nhiệm vụ tuần tiếp theo.
4. Ghi chú nội dung buổi họp (do PO thực hiện).

2. Đặc tả yêu cầu

2 Đặc tả yêu cầu

2.1 Các bên liên quan

• Nhóm quản lý hệ thống (System manager)

1. Quản trị viên (Administrator):

- Vai trò (Role): Quản lý và kiểm soát các hoạt động của Telegram Bot cũng như hệ thống backend liên quan.
- Nhu cầu (Need)
 - * Đảm bảo toàn hệ thống hoạt động ổn định, trong đó có Telegram bot.
 - * Cần có khả năng giám sát hiệu suất.
- Lợi ích được nhận (Benefit):
 - * Giảm tải cho hệ thống hỗ trợ khi sinh viên có thể tự tra cứu thông tin.

2. Đội ngũ phát triển backend toàn hệ thống (System backend developer team):

- Vai trò (Role): Cung cấp API để Telegram Bot kết nối và truy vấn dữ liệu
- Nhu cầu (Need):
 - * Cần Telegram Bot sử dụng đúng API để truy xuất dữ liệu điểm số chính xác.
- Lợi ích được nhận (Benefit):
 - * Telegram Bot giúp kiểm tra và khai thác API đã xây dựng, đảm bảo backend hoạt động đúng chức năng.
 - * Telegram Bot hoạt động như một công cụ bổ sung cho hệ thống, giúp sinh viên có thêm lựa chọn truy cập thông tin mà không cần phải luôn sử dụng giao diện web.

3. Đội ngũ phát triển Telegram-bot (Telegram-bot developer team):

- Vai trò (Role): Thiết kế và triển khai các chức năng của Telegram-bot, đảm bảo tính bảo mật của hệ thống và dữ liệu trong quá trình sử dụng Telegram-bot.
- Nhu cầu (Need):
 - * Cần có khả năng giám sát hiệu suất và xử lý lỗi phát sinh.
 - * Bảo trì, cập nhật và đổi mới module liên tục để đáp ứng nhu cầu sử dụng ngày càng nhiều của sinh viên.
 - * Cần khả năng theo dõi các vi phạm bảo mật và xử lý nhanh chóng.
- Lợi ích được nhận (Benefit):

2. Đặc tả yêu cầu

- * Telegram Bot được bảo mật và duy trì đúng tiêu chuẩn, tránh các vi phạm bảo mật.
- * Bot hoạt động hiệu quả và ổn định, mang lại sự hài lòng cho người dùng.

• Nhóm người dùng cuối (End-user)

1. Sinh viên (Students):

- Vai trò (Role): Người sử dụng chính của Telegram Bot.
- Nhu cầu (Need):
 - * Tra cứu điểm thi và điểm quá trình một cách nhanh chóng và dễ dàng qua nền tảng nhắn tin.
 - * Đảm bảo thông tin về cá nhân sinh viên, điểm số... được bảo mật trong quá trình trao đổi qua bot.
- Lợi ích được nhận (Benefit):
 - * Telegram Bot cung cấp cho sinh viên một kênh tiện lợi để tra cứu điểm, không cần đăng nhập vào hệ thống website chính thức của nhà trường.
 - * Thông báo cập nhật điểm ngay lập tức qua tin nhắn giúp sinh viên theo dõi kết quả học tập nhanh chóng và dễ dàng hơn.
 - * Telegram Bot sử dụng cơ chế bảo mật phù hợp, giúp bảo vệ dữ liệu sinh viên.

2.2 Yêu cầu chức năng

1. STUDENT	1.1	Sinh viên có thể xem điểm số (điểm thành phần, điểm thi cuối kỳ, và điểm trung bình môn) theo từng môn, từng học kỳ của cá nhân thông qua việc gửi tin nhắn đúng cú pháp cho Telegram bot
	1.2	Sinh viên có thể kiểm tra thông tin cá nhân của phiên đăng nhập hiện tại (tên + MSSV).
	1.3	Sinh viên phải thông qua hệ thống xác thực tài khoản - Authentication (Mỗi phiên đăng nhập chỉ tương ứng với một tài khoản của một sinh viên, đăng nhập với mật khẩu và OTP).

Bảng 2.1: Yêu cầu chức năng của bot Telegram

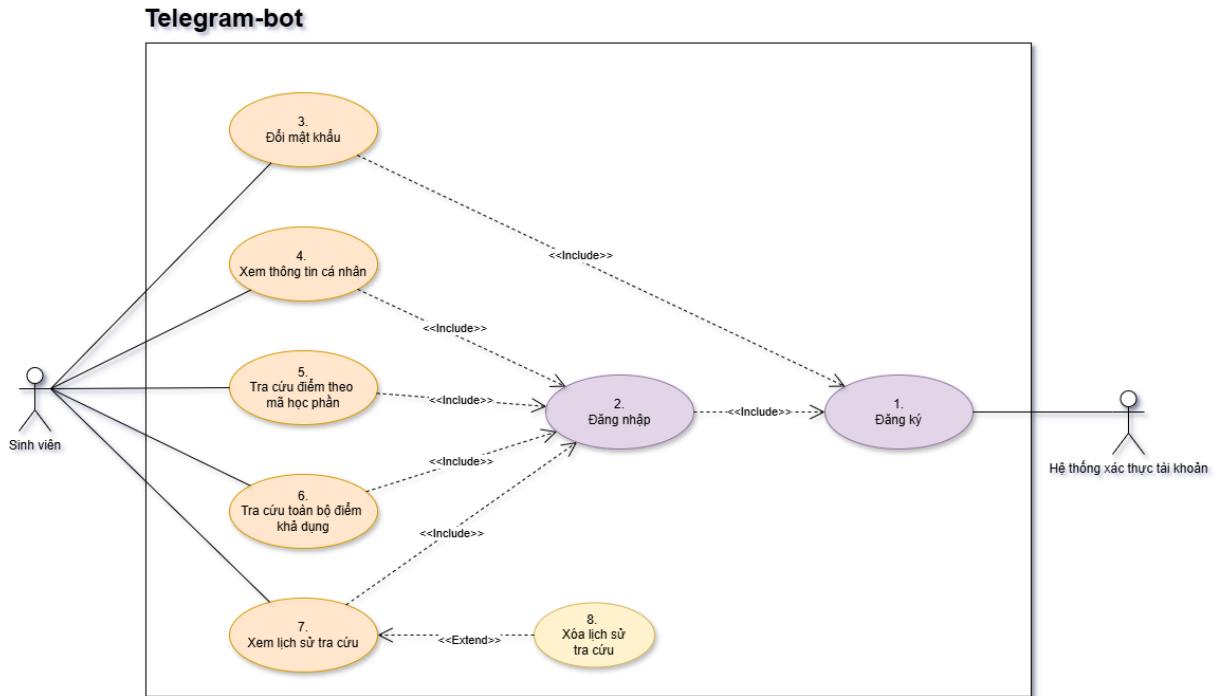
2. Đặc tả yêu cầu

2.3 Yêu cầu phi chức năng

2. PRODUCT	2.1	Hiệu suất: Thời gian phản hồi của bot không vượt quá 2s/ một yêu cầu từ phía người dùng cho mọi câu lệnh được thực hiện.
	2.2	Khả năng đáp ứng: Bot phải có khả năng xử lý ít nhất 1000 yêu cầu đồng thời mà không gây gián đoạn dịch vụ.
	2.3	Tính khả dụng: Bot phải có tính khả dụng 24/7.
	2.4	Tính tương thích: Cần tương thích với phiên bản Telegram mới nhất (phương cách sử dụng API).
	2.5	Tính hiệu dụng: Sinh viên có thể sử dụng bot chỉ với hướng dẫn nhanh, ngắn gọn từ tin nhắn bot gửi khi bắt đầu cuộc trò chuyện.
3. ORGANIZATIONAL	3.1	Tất cả các phiên bản mã nguồn của bot phải được quản lý qua GitHub, và có mô tả rõ ràng.

Bảng 2.2: Yêu cầu phi chức năng của bot Telegram

2.4 Sơ đồ Usecase (hình 2.1)



Hình 2.1: Sơ đồ Usecase mô tả chức năng của toàn bộ hệ thống.

Đặc tả Usecase 2 - Login

1	Use-case ID	UC-2
2	Use-case name	Đăng nhập (Login)
3	Description	Là người dùng, tôi muốn đăng nhập vào hệ thống bằng MSSV và mật khẩu để sử dụng các dịch vụ của bot Telegram.
4	Actors	Người dùng
5	Priority	Must have
6	Trigger	Người dùng gửi lệnh <code>/login MSSV password</code> qua Telegram bot.
7	Pre-condition	1. Người dùng đã có tài khoản trong hệ thống. 2. MSSV và mật khẩu của người dùng đã được lưu trữ trong cơ sở dữ liệu.
8	Post-condition	1. Người dùng đăng nhập thành công. 2. Bot thông báo kết quả đăng nhập thành công cho người dùng.

2. ĐẶC TẢ YÊU CẦU

9	Basic flow	<p>1. Người dùng gửi lệnh <code>/login MSSV password</code>.</p> <p>2. Bot tiếp nhận lệnh và gửi yêu cầu xác thực tới cơ sở dữ liệu.</p> <p>3. Hệ thống xác thực MSSV và mật khẩu</p> <p>4. Bot thông báo kết quả đăng nhập thành công.</p> <p><i>Use case kết thúc.</i></p>
10	Alternative flow	<p>Quên mật khẩu</p> <p>1a. Người dùng thực hiện lệnh <code>/getOTP MSSV</code>.</p> <p>1b. Bot gửi yêu cầu tới cơ sở dữ liệu để cấp OTP tương ứng với MSSV.</p> <p>1c. Hệ thống gửi OTP tới email người dùng.</p> <p>1d. Người dùng sử dụng OTP với lệnh <code>/resetpassword newPW OTP</code>.</p> <p>1e. Hệ thống cập nhật mật khẩu mới.</p> <p><i>Use case tiếp tục bước 1</i></p>
11	Exception flow	<p>Thông tin đăng nhập không hợp lệ</p> <p>3a. Hệ thống phát hiện thông tin MSSV hoặc mật khẩu sai: Bot thông báo "MSSV hoặc mật khẩu không chính xác." Gợi ý người dùng nhập lại thông tin hoặc chọn <code>/resetpassword</code> nếu quên mật khẩu.</p> <p><i>Use case kết thúc</i></p>
12	Business rules	<p>1. MSSV phải hợp lệ theo định dạng 7 chữ số.</p> <p>2. Mật khẩu cần được mã hóa trước khi gửi tới cơ sở dữ liệu để xác thực.</p>
13	Non-functional requirements	Thời gian phản hồi không quá 2 giây kể từ lúc người dùng gửi lệnh.

Bảng 2.3: Bảng đặc tả Usecase 2 - Login.

ĐẶC TẢ USECASE 6 - XEM TOÀN BỘ ĐIỂM KHẢ DỤNG

1	Use-case ID	UC-6
2	Use-case name	Xem điểm tất cả các môn
3	Description	Là người dùng, tôi muốn xem toàn bộ điểm các môn học của mình để kiểm tra kết quả học tập.

2. Đặc tả yêu cầu

4	Actors	Người dùng, Telegram Bot, Hệ thống cơ sở dữ liệu từ xa
5	Priority	Optional
6	Trigger	Người dùng gửi lệnh <code>/allgrade</code> qua Telegram Bot
7	Pre-condition	1. Người dùng đã đăng ký tài khoản trên hệ thống. 2. Người dùng đã đăng nhập vào bot Telegram. 3. Bot Telegram có quyền truy xuất dữ liệu từ cơ sở dữ liệu.
8	Post-condition	1. Danh sách điểm của tất cả các môn học được hiển thị cho người dùng. 2. Nếu xảy ra lỗi, thông báo lỗi được gửi về người dùng.
9	Basic flow	1. Người dùng gửi lệnh <code>/allgrade</code> qua Telegram Bot. 2. Telegram Bot nhận lệnh và gửi yêu cầu truy xuất toàn bộ điểm từ cơ sở dữ liệu. 3. Hệ thống cơ sở dữ liệu trả về danh sách điểm của các môn học. 4. Telegram Bot hiển thị danh sách điểm cho người dùng. <i>Use case kết thúc.</i>
10	Alternative flow	3a. Nếu không có điểm nào trong cơ sở dữ liệu, Telegram Bot hiển thị thông báo "Không tìm thấy điểm nào trong cơ sở dữ liệu". <i>Use case kết thúc</i>
11	Exception flow	Nhập sai định dạng lệnh 2a1. Telegram Bot hiển thị thông báo "Vui lòng nhập đúng định dạng: <code>/allgrade</code> ". <i>Use case kết thúc</i> Không kết nối được với cơ sở dữ liệu từ xa 2a2. Telegram Bot gửi thông báo lỗi "Không thể truy xuất dữ liệu, vui lòng thử lại sau". <i>Use case kết thúc</i>
12	Business rules	None
13	Non-functional requirements	Thời gian phản hồi không quá 2 giây kể từ lúc người dùng gửi lệnh.

Bảng 2.4: Bảng đặc tả Usecase 6 - AllGrade.

3. Thiết kế hệ thống

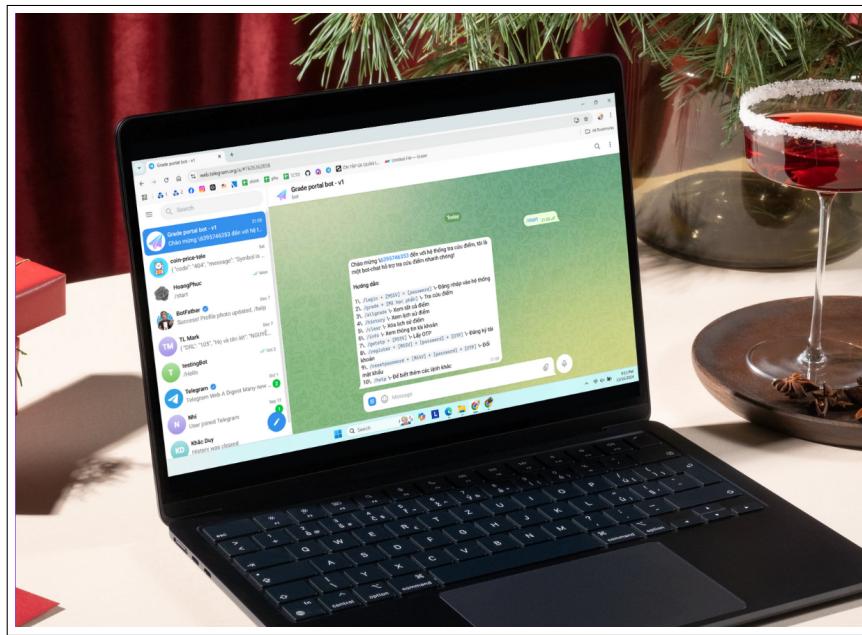
3 Thiết kế hệ thống

3.1 Mockup

Dưới đây là một số giao diện giao tiếp với người dùng sau khi đã triển khai hệ thống bot.

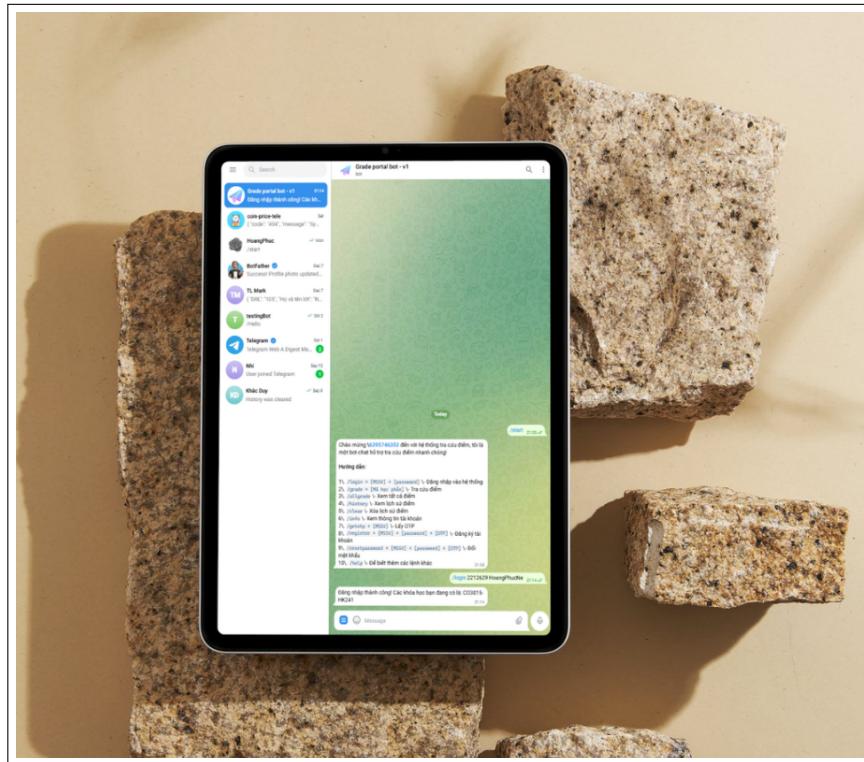


Hình 3.1: Mockup giao diện giao tiếp trên điện thoại.



Hình 3.2: Mockup giao diện giao tiếp trên Laptop.

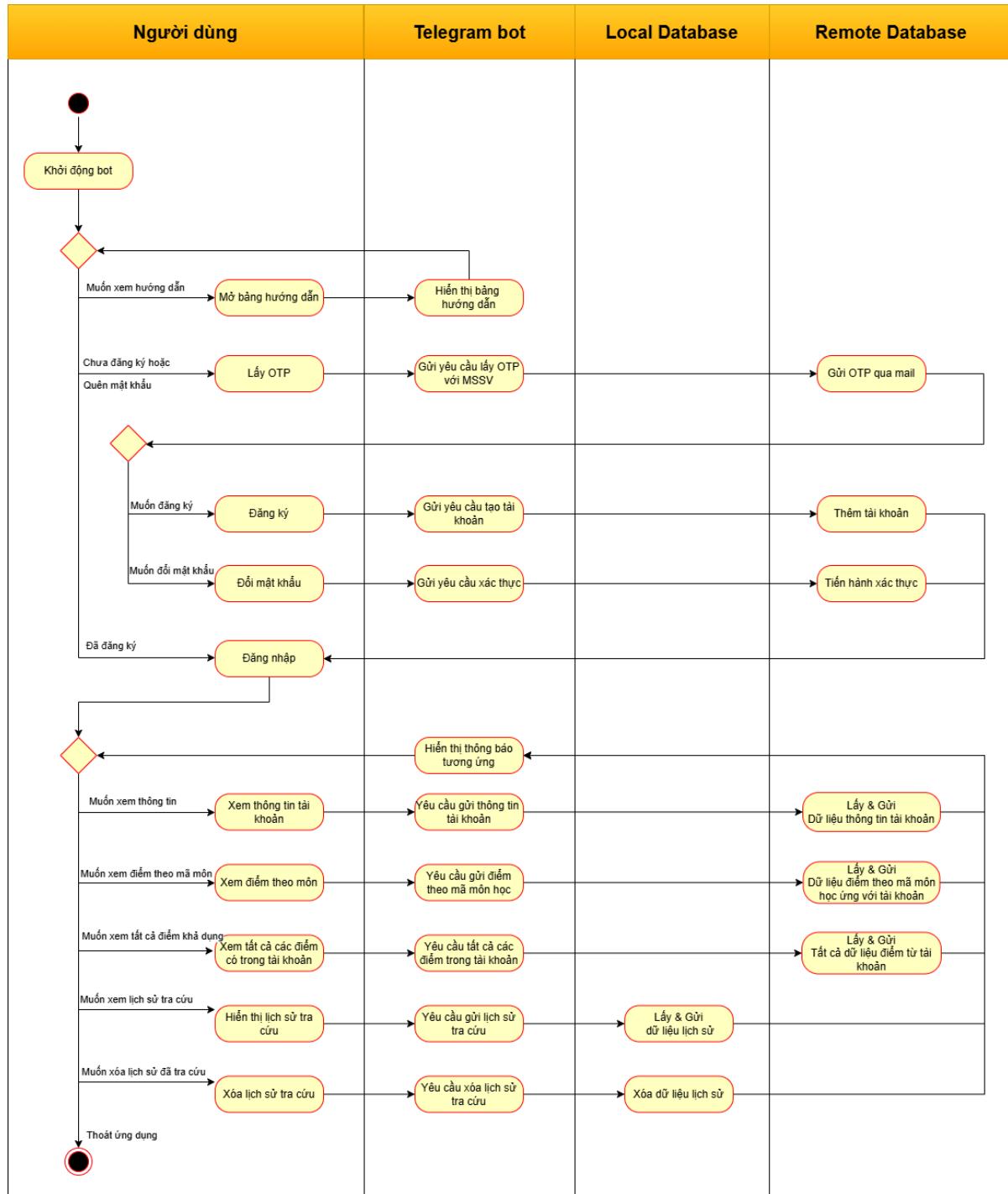
3. Thiết kế hệ thống



Hình 3.3: Mockup giao diện giao tiếp trên máy tính bảng.

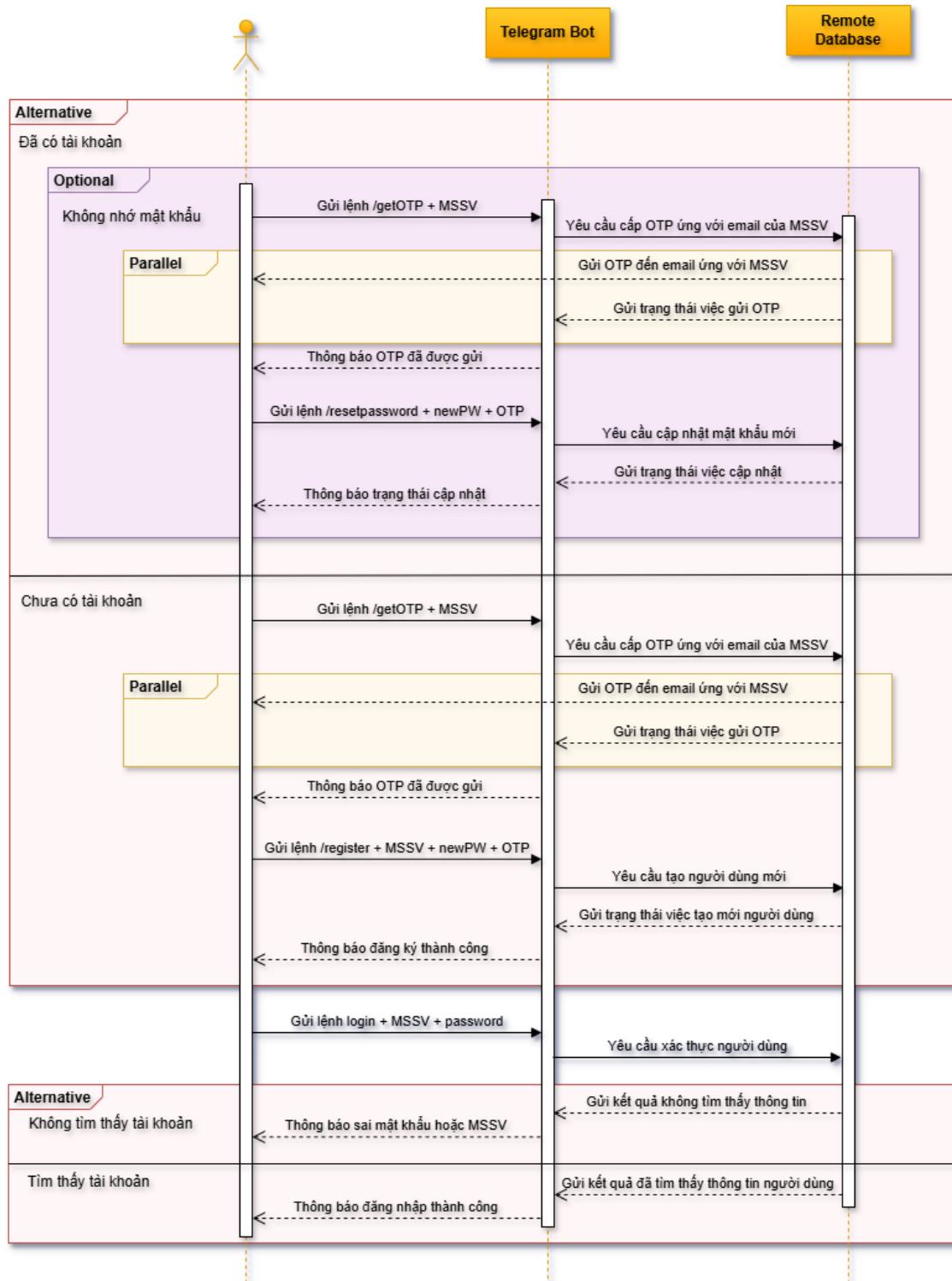
3. Thiết kế hệ thống

3.2 Sơ đồ Activity (hình 3.4)



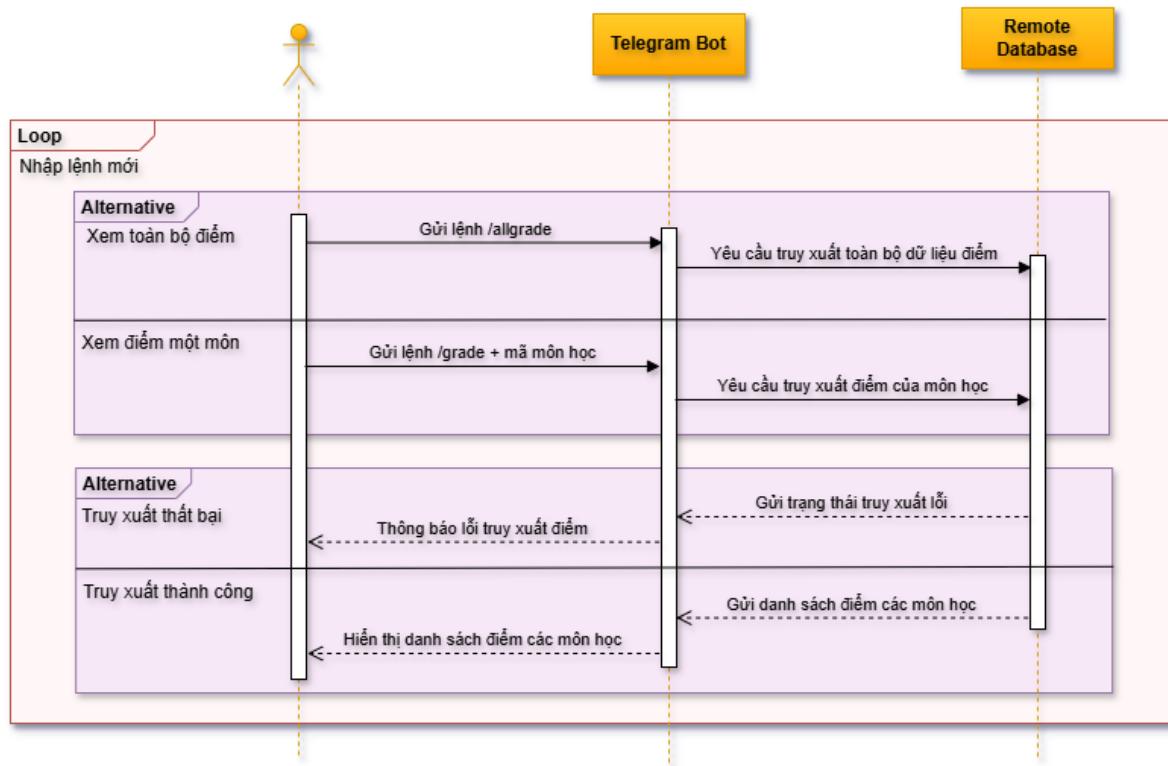
Hình 3.4: Sơ đồ Activity mô tả luồng hoạt động chính của hệ thống.

3.3 Sơ đồ Sequence (hình 3.5 và hình 3.6)



Hình 3.5: Sơ đồ Sequence mô tả chi tiết hoạt động hệ thống khi đăng ký, đăng nhập.

3. Thiết kế hệ thống



Hình 3.6: Sơ đồ Sequence mô tả chi tiết hoạt động hệ thống khi xem toàn bộ điểm và xem điểm một môn.

4 Các loại tài liệu

4.1 Báo cáo họp

Tổng hợp meeting minutes của từng tuần: [Tổng kết họp hàng tuần](#)

4.2 Cấu trúc thư mục

```
GRADE-PORTAL-TELE/
+-- .github/
|   +-- workflows/
|       +-- docker-image.yml
+-- report/
|   +-- Genneral_Report.md
+-- source/
|   +-- config/
|       |   +-- config.go
|       |   +-- database.go
|       +-- internal/
|           +-- bot/
|               |   +-- bot.go
|               |   +-- commands.go
|               |   +-- middleware.go
|               +-- handlers/
|                   |   +-- auth_handler.go
|                   |   +-- bot_handle.go
|                   |   +-- grade_handler.go
|               +-- models/
|                   |   +-- models_API.go
|                   |   +-- models_DBTele.go
|                   |   +-- user.go
|               +-- services/
|                   +-- auth_service.go
|                   +-- error_handling.go
|                   +-- grade_service.go
|                   +-- history_service.go
+-- .air.toml
+-- .dockerignore
+-- .env
```

4. Các loại tài liệu

```
+-- .gitignore  
+-- docker-compose.yml  
+-- Dockerfile  
+-- entrypoint.sh  
+-- go.mod  
+-- go.sum  
+-- main.go  
+-- Makefile  
+-- README.md
```

Mô tả

1. bot/bot.go

- Khởi tạo bot Telegram, kết nối cơ sở dữ liệu, lắng nghe và xử lý các cập nhật từ người dùng.

2. bot/commands.go

- Xử lý các lệnh được nhập từ người dùng bằng cách gọi các hàm trong thư mục handlers.

3. handlers/

- Thư mục handlers/ chứa các hàm để nhận các lệnh từ người dùng, gọi hàm trong thư mục services/ để xử lý yêu cầu, sau đó gửi phản hồi lại cho người dùng.

4. services/

- Thư mục services/ chứa các hàm để thực hiện nghiệp vụ, tương tác với cơ sở dữ liệu và xử lý logic ứng dụng, từ đó trả về kết quả cho các hàm trong thư mục handlers/

5. models/

- Thư mục models dùng để liên kết với cơ sở dữ liệu, từ đó định nghĩa cấu trúc dữ liệu chính để sử dụng xuyên suốt ứng dụng.

6. Dockerfile

- Thiết kế và cấu hình môi trường container để triển khai ứng dụng.

4.3 Command Guide

Xem bảng 4.1.

Bảng 4.1: Bảng tổng hợp các câu lệnh trong bot Telegram.

18

STT	Lệnh	Chức năng	Mô tả	Bot hiển thị
1	/login + [MSSV] + [password]	Đăng nhập vào hệ thống.	Cho phép sinh viên đăng nhập bằng mã số sinh viên (MSSV) và mật khẩu đã đăng ký trước đó.	Nếu đăng nhập thành công: Hiển thị chào mừng. Nếu thất bại: báo lỗi.
2	/grade + [Mã học phần]	Tra cứu điểm theo mã học phần.	Người dùng nhập mã học phần (courseID) để nhận được điểm chi tiết cho môn học tương ứng.	Nếu thành công: hiển thị điểm chi tiết môn học. Nếu lỗi: báo lỗi.
3	/allgrade	Xem tất cả điểm của các môn đã học.	Hiển thị bảng điểm chi tiết của tất cả học phần sinh viên đã hoàn thành.	Nếu thành công: hiển thị danh sách điểm chi tiết (nhiều môn). Nếu lỗi: báo lỗi.
4	/history	Xem lịch sử điểm đã tra cứu.	Hiển thị các môn học mà người dùng đã tra cứu gần đây nhất.	Nếu có lịch sử: hiển thị lịch sử đã tra cứu. Nếu không có lịch sử: báo không có lịch sử.
5.	/clear	Xóa lịch sử tra cứu	Xóa toàn bộ lịch sử tra cứu của người dùng.	Nếu xóa thành công: “Lịch sử tra cứu đã được xóa”. Nếu xảy ra lỗi: “Error”.
6	/info	Xem thông tin phiên đăng nhập	Hiển thị thông tin cá nhân của sinh viên (họ tên, lớp, MSSV, email, ...).	Nếu thành công: hiển thị thông tin cá nhân của sinh viên. Nếu lỗi: báo lỗi.
7	/register [MSSV Password OTP]	Đăng ký tạo phiên đăng nhập	Đăng ký tài khoản sinh viên trong hệ thống.	Nếu thành công: thông báo thành công. Nếu lỗi: báo lỗi.

Tiếp theo trang sau

4. Các loại tài liệu

STT	Lệnh	Chức năng	Mô tả	Bot hiển thị
8	/getotp [MSSV]	Lấy OTP để đăng ký hoặc đổi mật khẩu	Gửi mã OTP đến email sinh viên để hỗ trợ đăng ký hoặc lấy lại mật khẩu.	Nếu thành công: báo kết quả. Nếu lỗi: báo lỗi.

5. Docker Compose và Deployment

5 Docker Compose và Deployment

5.1 Quy trình triển khai

1. Thiết lập Docker Image

- Mục tiêu: Đóng gói ứng dụng bot Telegram vào một Docker Image trên máy local.
- Dockerfile: là một tệp văn bản chứa tập hợp các lệnh để tự động xây dựng một Docker Image.

```
FROM golang:1.23 AS builder

WORKDIR /app

# Copy go.mod and go.sum and download dependencies
COPY go.mod go.sum .
RUN go mod download

# Copy the source code and build the binary
COPY . .
RUN CGO_ENABLED=0 GOOS=linux go build -o main main.go

# Stage 2: Run
FROM alpine:latest

# Thiết lập thư mục làm việc
WORKDIR /root/

# Sao chép binary từ giai đoạn build
COPY --from=builder /app/main .

# Expose the port
EXPOSE 8080

# Start the application
CMD ["/app/main"]
```

Hình 5.1: Dockerfile

- Chạy lệnh build để tạo một docker image trên máy cá nhân.

5. Docker Compose và Deployment

```
docker build -t grade-portal-tele .
```

Hình 5.2: Lệnh tạo docker image

2. Đẩy Image vừa tạo lên Docker Hub

- Đặt tên cho image theo định dạng chuẩn.

```
docker tag grade-portal-tele phongthanphong/grade-portal-tele:latest
```

Hình 5.3: Lệnh đặt tên cho image

- Đẩy image lên Docker Hub.

```
docker push phongthanphong/grade-portal-tele:latest
```

Hình 5.4: Lệnh đẩy image lên Docker Hub

3. Xây dựng docker-compose giúp kéo image dự án về và chạy trên máy cá nhân

```
services:  
  app:  
    image: phongthanphong/grade-portal-tele:latest  
    env_file:  
      - .env  
    ports:  
      - "8080:8080"  
    container_name: grade-portal-tele  
    restart: unless-stopped
```

Hình 5.5: docker-compose.yml

- File docker-compose giúp kéo image grade-portal-tele phiên bản latest trên Docker Hub của user phongthanphong.
- Load các biến môi trường từ file .env vào container.
- Map cổng 8080 của máy host tới cổng 8080 của container.
- Đặt tên cho container là grade-portal-tele.
- Container sẽ tự động khởi động lại khi gặp lỗi trừ khi bạn dừng nó thủ công.

5. Docker Compose và Deployment

4. Xây dựng CI/CD với Github Actions

GitHub Actions được sử dụng để tự động hóa quy trình triển khai.

Workflow bao gồm hai giai đoạn:

- Xây dựng và đẩy Docker Image lên Docker Hub.
- Triển khai ứng dụng trên máy chủ.

5.2 Kết quả triển khai

1. Tự động hóa

- Workflow tự động kích hoạt khi có thay đổi trên nhánh main.
- Docker Image được tự động build và đẩy lên Docker Hub.

2. Triển khai nhanh chóng

- Image từ Docker Hub được kéo xuống và triển khai trên máy chủ.
- Quá trình deploy không gây downtime nhờ vào việc xóa container cũ và chạy container mới.

3. Tối ưu hóa bảo mật

- Các thông tin nhạy cảm như BOT_TOKEN, DATABASE_URL, và API_URL được bảo mật qua GitHub Secrets.