

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



BÁO CÁO BÀI TẬP LỚN

SEAWHISPER

ỨNG DỤNG CHIA SẺ ẨN DANH
& CHỮA LÀNH TÂM HỒN

Môn học: Phát triển Ứng dụng Di động

Học kỳ: 2025–2026 (HK251)

Giảng viên: Hoàng Lê Hải Thanh

Sinh viên thực hiện MSSV

Trần Minh Khang	2211472
Hà Thái Toàn	2213524
Đoàn Nhật Tiến	2213449
Lê Minh Tuấn	2252868

TP. Hồ Chí Minh, năm 2026

Tóm tắt

SeaWhisper là ứng dụng di động chia sẻ ẩn danh theo mô hình “thả chai” (message in a bottle), nơi người dùng có thể gửi những tâm sự của mình ra “biển” và một người khác sẽ ngẫu nhiên “nhặt” được để đọc và phản hồi một cách nhẹ nhàng. Dự án hướng đến trải nghiệm chữa lành (healing), giảm độc hại (toxic) và khuyến khích tương tác tích cực thông qua cơ chế ẩn danh có kiểm duyệt.

Mục tiêu của hệ thống là tạo ra một không gian an toàn, không phán xét để người dùng giải tỏa cảm xúc; đồng thời áp dụng bộ lọc nội dung (AI Filter) nhằm hạn chế ngôn từ công kích, kích động, hoặc tiêu cực quá mức. Về mặt kỹ thuật, ứng dụng được xây dựng với React Native cho phía client, FastAPI cho phía server, Redis làm cơ sở dữ liệu lưu trữ nhanh và cơ chế ghép ngẫu nhiên (matching) để phân phối “chai” giữa người gửi và người nhận.

Báo cáo trình bày quy trình phát triển theo hướng lặp (iterative), các quyết định thiết kế MVP, kiến trúc hệ thống, thiết kế web service, chiến lược triển khai, kiểm thử và đánh giá trải nghiệm người dùng. Kết quả khảo sát ban đầu cho thấy nhu cầu cao đối với một nền tảng ẩn danh lành mạnh; các phân tích hành vi và phản hồi người dùng được sử dụng để đề xuất cải tiến cho các phiên bản tiếp theo, bao gồm tối ưu bộ lọc, đa dạng hóa cơ chế phản hồi cảm xúc và tăng cường tính an toàn cộng đồng.

Mục lục

Tóm tắt	1
1 Giới thiệu	6
1.1 Bối cảnh và vấn đề	6
1.2 Mục tiêu dự án	6
1.3 Phạm vi và đối tượng hướng tới	6
1.4 Giá trị thực tiễn và tác động	7
1.5 Cấu trúc báo cáo	7
2 Phương pháp và quy trình thực hiện	7
2.1 Cách tiếp cận phát triển	7
2.2 Quy trình làm việc nhóm	7
2.3 Công cụ và nền tảng sử dụng	8
2.4 Quyết định phạm vi MVP	8
2.5 Dánh giá phương pháp	8
3 Tính năng và chức năng hệ thống	9
3.1 Tổng quan các tính năng	9
3.2 Gửi chai (Send Bottle)	9
3.3 Nhận chai ngẫu nhiên (Receive Bottle)	9
3.4 Phản hồi (Chat/Reply)	9
3.5 Lưu trữ và xem lại chai	9
3.6 Giới hạn của MVP	10
4 Triển khai hệ thống	10
4.1 Tổng quan	10
4.2 Frontend với React Native Expo	10
4.3 Điều hướng và quản lý trạng thái	10
4.4 Backend với FastAPI	11
4.5 Redis làm kho lưu trữ MVP	11
4.6 Cơ chế phân phối ngẫu nhiên	11
4.7 Giới hạn triển khai	11
5 Trải nghiệm người dùng (UX) và luồng thao tác MVP	11
5.1 Tổng quan thiết kế trải nghiệm	11
5.2 Luồng 1: Xác thực người dùng (Đăng ký / Đăng nhập)	12
5.3 Luồng 2: Trang chủ và điều hướng	12
5.4 Luồng 3: Thả chai (Send Bottle)	13
5.5 Luồng 4: Nhận chai và xem nội dung	15
5.6 Luồng 5: Trò chuyện và phản hồi	16

5.7	Luồng 6: Lưu trữ chai	18
5.8	Luồng 7: Quản lý tài khoản và cài đặt	18
5.9	Luồng 8: Giới thiệu và hỗ trợ	20
6	Kiến trúc hệ thống	21
6.1	Tổng quan kiến trúc	21
6.2	Sơ đồ triển khai tổng thể và luồng dữ liệu	21
6.2.1	Luồng dữ liệu chính	21
6.2.2	Quy ước giao tiếp	22
6.3	Phân rã thành phần hệ thống	22
6.3.1	Frontend (React Native Expo)	22
6.3.2	Backend (FastAPI)	22
6.3.3	Database (Redis)	23
6.4	Use Case Diagram và phạm vi chức năng	23
6.5	Thiết kế dữ liệu (ERD) và mô hình lưu trữ	24
6.5.1	Mô tả thực thể	26
6.5.2	Lý do chọn Redis cho MVP	26
6.6	Bảo mật và tính ẩn danh ở mức kiến trúc	26
6.7	Dánh giá kiến trúc và khả năng mở rộng	27
7	Thiết kế Web Service (API)	27
7.1	Tổng quan	27
7.2	Quy ước thiết kế API	27
7.3	Nhóm API xác thực người dùng (auth)	27
7.3.1	POST /auth/register – Đăng ký tài khoản	28
7.3.2	POST /auth/login – Đăng nhập	28
7.4	Nhóm API thả chai và nhặt chai (sea-bottle)	28
7.4.1	POST /sea/bottle – Tạo chai mới	28
7.4.2	GET /sea/bottle – Lấy chai ngẫu nhiên	29
7.5	Nhóm API quản lý chai đã lưu (api-store-bottle)	29
7.5.1	POST /api/store-bottle – Lưu chai	29
7.5.2	GET /api/store-bottle – Lấy danh sách chai đã lưu	29
7.5.3	GET /api/store-bottle/{stored_bottle_id} – Lấy chi tiết chai đã lưu	30
7.5.4	DELETE /api/store-bottle/{stored_bottle_id} – Xóa chai đã lưu	30
7.6	Nhóm API hệ thống	30
7.6.1	GET /health – Health check	30
7.6.2	GET /api/me – Lấy thông tin người dùng hiện tại	30
7.7	Tài liệu API trực tuyến (Swagger)	30

8 Chiến lược triển khai (Deployment Strategy)	31
8.1 Tổng quan	31
8.2 Triển khai Backend	31
8.3 Expose Backend bằng Ngrok	31
8.4 Triển khai tài liệu API (Swagger)	31
8.5 Triển khai Landing Page	32
8.6 Quản lý mã nguồn và cộng tác nhóm	32
8.7 Dánh giá chiến lược triển khai	33
9 Testing Coverage Report	33
9.1 Mục tiêu kiểm thử	33
9.2 Công cụ kiểm thử và đánh giá chất lượng	33
9.3 Phân tích chất lượng mã nguồn với SonarCloud	34
9.4 Kết quả phân tích Frontend	34
9.5 Kết quả phân tích Backend	35
9.6 Kiểm thử API Backend	35
9.7 Kiểm thử luồng người dùng (User Flow)	35
9.8 Dánh giá tổng thể và hạn chế	36
10 Khảo sát sự hài lòng của người dùng và phân tích hành vi người dùng	36
10.1 Mục tiêu khảo sát	36
10.2 Phương pháp và đối tượng khảo sát	36
10.3 Tổng quan mẫu khảo sát	37
10.4 Kết quả đánh giá mức độ hài lòng	37
10.4.1 Mức độ hài lòng tổng thể	37
10.4.2 Dánh giá giao diện người dùng	37
10.4.3 Dánh giá chức năng thả chai và nhặt chai	37
10.4.4 Mức độ phù hợp trong chia sẻ cảm xúc cá nhân	37
10.5 Ý định sử dụng trong tương lai	38
10.6 Phân tích hành vi người dùng (User Behavior Analytics)	38
10.7 Tổng hợp phản hồi và đề xuất cải thiện	38
10.8 Dữ liệu khảo sát	38
11 Dánh giá dự án và những điểm cần cải thiện	39
11.1 Dánh giá mức độ đáp ứng mục tiêu dự án	39
11.2 Dánh giá về trải nghiệm người dùng (UX/UI)	39
11.3 Dánh giá về chức năng và hiệu năng hệ thống	39
11.4 Dánh giá về chất lượng mã nguồn	40
11.5 Hạn chế của dự án	40
11.6 Định hướng cải tiến và phát triển trong tương lai	40
11.7 Tổng kết	41

12 Kết Luận	41
Tài liệu tham khảo	42

1 Giới thiệu

1.1 Bối cảnh và vấn đề

Trong bối cảnh mạng xã hội phát triển mạnh, nhu cầu chia sẻ cảm xúc cá nhân của người trẻ ngày càng tăng. Tuy nhiên, việc chia sẻ dưới danh tính thật thường đi kèm rủi ro bị phán xét, bị soi xét đời tư, hoặc nhận phản hồi tiêu cực. Các nền tảng *ẩn danh* giúp giảm rào cản khi bộc lộ cảm xúc, nhưng đồng thời cũng dễ trở thành môi trường độc hại do thiếu cơ chế định hướng tương tác và kiểm duyệt phù hợp [1].

Với các ứng dụng *ẩn danh* phổ biến, người dùng có thể gặp tình huống: nội dung bị công kích, phản hồi mang tính chê giễu, hoặc xuất hiện hành vi bắt nạt trên không gian số. Các yếu tố này có thể tác động tiêu cực đến sức khỏe tinh thần, đặc biệt với nhóm người dùng trẻ hoặc đang ở trạng thái cảm xúc nhạy cảm [2]. Do đó, bài toán đặt ra không chỉ là *ẩn danh để để nói*, mà còn là *thiết kế trải nghiệm để an toàn*.

1.2 Mục tiêu dự án

SeaWhisper được xây dựng nhằm tạo ra một không gian *ẩn danh* nhưng an toàn, hướng đến trải nghiệm chữa lành (healing-oriented). Các mục tiêu chính gồm:

- Cung cấp cơ chế chia sẻ *ẩn danh* theo mô hình “thả chai – nhặt chai” (message in a bottle), giảm áp lực khi chia sẻ.
- Khuyến khích tương tác đồng cảm, hạn chế tranh cãi/công kích bằng thiết kế luồng phản hồi đơn giản và có định hướng.
- Đảm bảo hệ thống dễ triển khai ở mức MVP nhưng vẫn có khả năng mở rộng cho các phiên bản sau.

1.3 Phạm vi và đối tượng hướng tới

Dự án tập trung vào nhóm người dùng trẻ (đặc biệt Gen Z), sinh viên, và những người có nhu cầu giải tỏa tâm trạng, tìm kiếm sự đồng cảm nhưng không muốn lộ danh tính. Trong phạm vi MVP, dự án ưu tiên các chức năng cốt lõi:

- Đăng ký/đăng nhập (tối thiểu để hỗ trợ lưu trữ và quản lý dữ liệu cá nhân).
- Thả chai (tạo thông điệp).
- Nhặt chai ngẫu nhiên và xem nội dung.
- Phản hồi đơn giản và lưu chai vào “bộ sưu tập”.

1.4 Giá trị thực tiễn và tác động

SeaWhisper hướng tới mô hình “C2C cảm xúc”: mỗi người vừa là người chia sẻ vừa là người lắng nghe. Nếu triển khai thực tế, ứng dụng có thể:

- Giảm áp lực tâm lý bằng việc cung cấp một kênh giải bày ít ràng buộc hơn so với mạng xã hội danh tính thật [3].
- Hạn chế độc hại thường thấy trong nền tảng ẩn danh bằng cách điều tiết cách tương tác và cơ chế phản hồi.
- Khuyến khích hành vi đồng cảm, tạo cảm giác “được lắng nghe” trong cộng đồng người dùng.

1.5 Cấu trúc báo cáo

Phần còn lại của báo cáo được tổ chức như sau:

- Phương pháp và quy trình: mô tả cách tiếp cận lặp, tổ chức nhóm, công cụ.
- Tính năng: trình bày MVP và giới hạn.
- Triển khai và kiến trúc: mô tả công nghệ, dữ liệu, thiết kế API.
- Đánh giá: kiểm thử, khảo sát hài lòng và định hướng cải tiến.

2 Phương pháp và quy trình thực hiện

2.1 Cách tiếp cận phát triển

Dự án SeaWhisper được phát triển theo hướng lặp (*iterative development*). Thay vì cố gắng hoàn thiện đầy đủ ngay từ đầu, nhóm ưu tiên xây dựng nhanh một phiên bản khả dụng (MVP) để: (1) kiểm chứng ý tưởng “thả chai – nhặt chai”, (2) đánh giá trải nghiệm người dùng, và (3) giảm rủi ro kỹ thuật trong thời gian giới hạn học kỳ [4].

Với cách tiếp cận này, mỗi vòng lặp gồm: đặc tả yêu cầu nhỏ → thiết kế UI/flow → triển khai → kiểm thử thủ công → cải tiến.

2.2 Quy trình làm việc nhóm

Nhóm tổ chức công việc theo các bước:

- Phân tích yêu cầu và chốt phạm vi MVP.
- Thiết kế UI trên Figma, thống nhất luồng thao tác.
- Chia task Frontend/Backend theo module (Auth, Bottle, StoredBottle, UI screens).

- Tích hợp API và hoàn thiện luồng end-to-end.
- Kiểm thử bằng Swagger UI/Postman và chạy thử trên thiết bị thật.

Mã nguồn và cộng tác nhóm được quản lý qua GitHub (branching, pull request, review), giúp theo dõi lịch sử thay đổi và phân chia trách nhiệm rõ ràng [5].

2.3 Công cụ và nền tảng sử dụng

Các công cụ chính:

- GitHub: quản lý mã nguồn, issues, pull requests [6].
- Figma: thiết kế UI và prototype.
- React Native + Expo: phát triển ứng dụng di động đa nền tảng [7, 8].
- FastAPI: triển khai REST API, tự sinh tài liệu OpenAPI/Swagger [9, 10].
- Redis: lưu trữ dữ liệu nhanh cho MVP [11].
- Swagger UI/Postman: kiểm thử API thủ công [12, 13].

2.4 Quyết định phạm vi MVP

Do giới hạn thời gian, nhóm tập trung vào các chức năng thiết yếu:

- Auth cơ bản (đăng ký/đăng nhập) để phục vụ lưu trữ “bộ sưu tập”.
- Tạo bottle và lấy bottle ngẫu nhiên.
- Lưu bottle, xem danh sách đã lưu, xóa bottle.

Các hạng mục nâng cao như phân tích cảm xúc tự động, lọc nội dung nâng cao, cá nhân hóa feed được đưa vào định hướng tương lai.

2.5 Đánh giá phương pháp

Cách làm tập trung MVP giúp nhóm:

- Hoàn thành sản phẩm end-to-end trong học kỳ.
- Giảm độ phức tạp thiết kế dữ liệu và hạ tầng.
- Có cơ sở đo lường/đánh giá (survey) để đề xuất cải tiến có căn cứ.

3 Tính năng và chức năng hệ thống

3.1 Tổng quan các tính năng

SeaWhisper được thiết kế xoay quanh trải nghiệm chia sẻ ẩn danh mang tính chia sẻ. Trong MVP, hệ thống tập trung vào các luồng cốt lõi: thả chai, nhận chai, phản hồi và lưu trữ. Thiết kế này bám sát ý tưởng “message in a bottle”, tạo yếu tố ngẫu nhiên và kết nối gián tiếp giữa người dùng.

3.2 Gửi chai (Send Bottle)

Người dùng có thể tạo một “chai” bằng cách nhập nội dung tâm sự và gửi lên hệ thống. Về mặt trải nghiệm, màn hình gửi được tối giản để người dùng tập trung vào việc viết.

Về mặt hệ thống:

- Client gửi request tạo bottle.
- Backend validate dữ liệu (rỗng/quá dài).
- Bottle được lưu vào Redis với định danh và timestamp.

3.3 Nhận chai ngẫu nhiên (Receive Bottle)

Khi người dùng nhặt chai, hệ thống trả về ngẫu nhiên một bottle đang khả dụng. Tính “ngẫu nhiên có kiểm soát” giúp tăng cảm giác khám phá, đồng thời tránh lặp chai nhiều lần trong cùng ngữ cảnh xử lý.

Trường hợp không có bottle phù hợp, backend trả về 204 No Content để client hiển thị trạng thái “biển đang vắng”.

3.4 Phản hồi (Chat/Reply)

Sau khi đọc nội dung, người dùng có thể phản hồi bằng tin nhắn ngắn trong giao diện chat tối giản. Trong MVP, cơ chế phản hồi được thiết kế để hạn chế toxic bằng việc:

- Giới hạn độ dài phản hồi (nếu có).
- UI tập trung vào động viên/đồng cảm.

3.5 Lưu trữ và xem lại chai

Người dùng có thể lưu bottle vào “balo lưu trữ” và xem danh sách đã lưu theo trang. Mục tiêu là tăng tính gắn bó: người dùng có thể quay lại đọc các thông điệp có ý nghĩa.

3.6 Giới hạn của MVP

MVP chưa triển khai:

- Kiểm duyệt nội dung nâng cao (moderation) ở mức production.
- Analytics tự động (Firebase/GA).
- Cá nhân hóa theo chủ đề/hành vi.

Các mục này được trình bày ở phần định hướng cải tiến.

4 Triển khai hệ thống

4.1 Tổng quan

SeaWhisper triển khai theo mô hình client–server. Client là ứng dụng di động React Native (Expo), backend là FastAPI cung cấp REST API, và Redis làm nơi lưu trữ chính ở mức MVP. Thiết kế này giúp tách biệt giao diện và nghiệp vụ, thuận tiện mở rộng và bảo trì.

4.2 Frontend với React Native Expo

Frontend được phát triển bằng React Native và Expo để rút ngắn thời gian cấu hình môi trường và tăng tốc thử nghiệm trên thiết bị thật [7, 8]. Ứng dụng tổ chức theo hướng component-based:

- Screen components: Home, Send, Receive/Detail, Chat, Stored list, Account/Setting, Introduce/Support.
- Shared components: button, modal, input, loading overlay.

Giao tiếp backend qua HTTP, dữ liệu JSON. Token (nếu có) được lưu ở storage phía client và đính kèm header Authorization cho các API cần quyền.

4.3 Điều hướng và quản lý trạng thái

Điều hướng được thiết kế ưu tiên luồng chính (thả/nhặt). State ở mức MVP có thể quản lý bằng state cục bộ theo screen và context cho session (user/token/theme). Các trạng thái quan trọng:

- Loading / error khi gọi API.
- Session: token, user id.
- Theme: light/night cho các màn hình hỗ trợ.

4.4 Backend với FastAPI

Backend dùng FastAPI vì hỗ trợ kiểu dữ liệu rõ ràng, tốc độ phát triển nhanh và tự động sinh OpenAPI/Swagger [9, 10]. Backend tách lớp:

- Router: định nghĩa endpoint.
- Service: xử lý nghiệp vụ (auth, bottle, stored bottle).
- Data access: thao tác Redis.
- Schema/DTO: chuẩn hóa request/response.

4.5 Redis làm kho lưu trữ MVP

Redis phù hợp cho MVP vì truy xuất nhanh và hỗ trợ cấu trúc dữ liệu đơn giản [11]. Các key có thể tổ chức theo namespace:

- user:id
- bottle:id
- stored:user_id (list/set các stored bottle id)

4.6 Cơ chế phân phối ngẫu nhiên

Khi nhặt chai, backend chọn một bottle khả dụng (chưa bị đánh dấu đã phát) theo cơ chế đơn giản. Sau khi trả về, hệ thống cập nhật trạng thái để hạn chế phân phối trùng lặp trong cùng ngữ cảnh xử lý. Ở phiên bản mở rộng, cơ chế random có thể chuyển sang hàng đợi, ưu tiên theo thời gian, hoặc theo tag/chủ đề.

4.7 Giới hạn triển khai

Do tập trung MVP, hệ thống chưa tối ưu:

- Quan sát/giám sát (monitoring) và logging chuẩn production.
- Test tự động đầy đủ.
- Cơ chế chống spam và moderation hoàn chỉnh.

5 Trải nghiệm người dùng (UX) và luồng thao tác MVP

5.1 Tổng quan thiết kế trải nghiệm

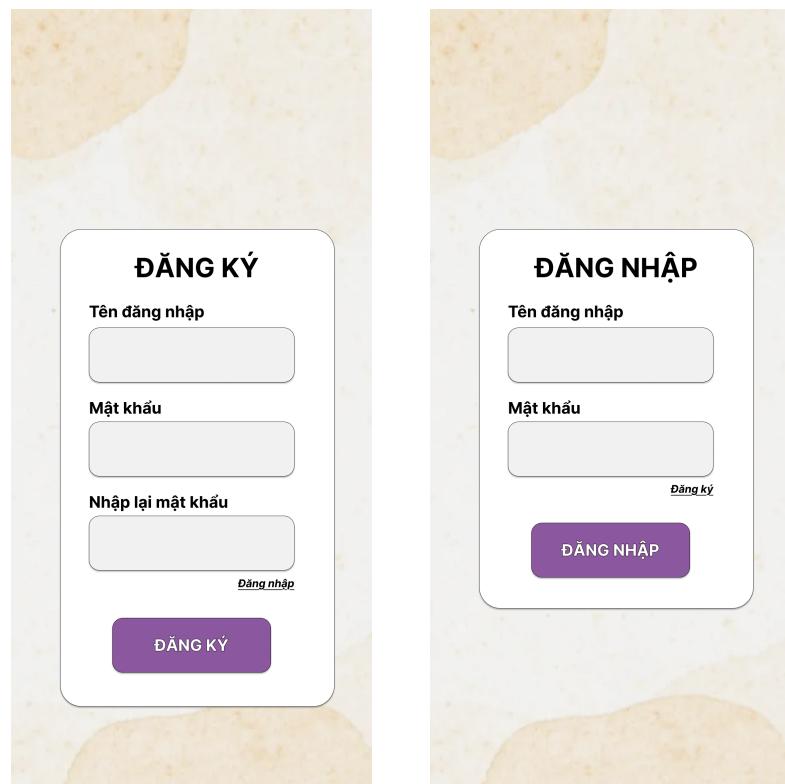
SeaWhisper được thiết kế theo định hướng tối giản và mang tính chữa lành, nhằm giúp người dùng dễ dàng chia sẻ cảm xúc mà không bị áp lực hay phân tâm. Các luồng thao tác

trong MVP được rút gọn, tập trung vào hai hành động chính là *thả chai* và *nhặt chai*, đồng thời cung cấp các chức năng phụ như trò chuyện, lưu trữ và quản lý tài khoản.

Ứng dụng hỗ trợ hai chế độ giao diện sáng và tối cho hầu hết các màn hình chức năng, ngoại trừ các màn hình xác thực ban đầu.

5.2 Luồng 1: Xác thực người dùng (Đăng ký / Đăng nhập)

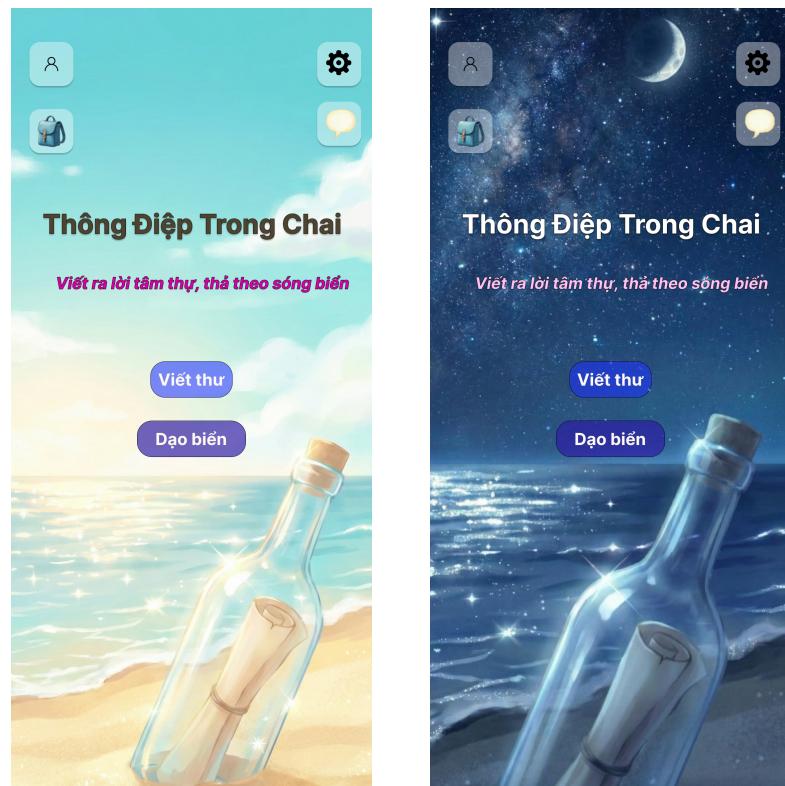
Luồng xác thực cho phép người dùng tạo và sử dụng tài khoản để truy cập các chức năng của ứng dụng. Trong MVP, màn hình đăng ký và đăng nhập được thiết kế đơn giản, rõ ràng và chỉ sử dụng một giao diện chung (không tách riêng giao diện sáng/tối).



Hình 1: Màn hình đăng ký và đăng nhập

5.3 Luồng 2: Trang chủ và điều hướng

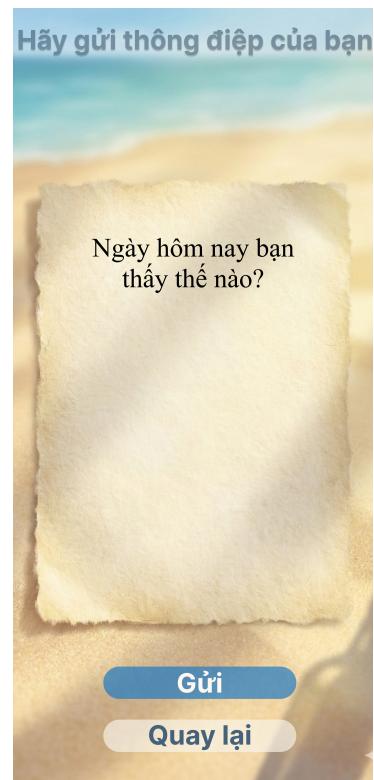
Sau khi đăng nhập, người dùng được đưa tới trang chủ, nơi hiển thị các hành động chính của ứng dụng. Từ đây, người dùng có thể điều hướng tới các chức năng như nhặt chai, thả chai, quản lý tài khoản và cài đặt.



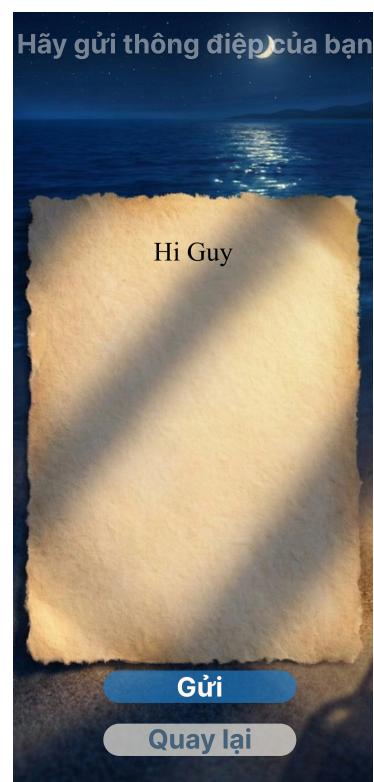
Hình 2: Trang chủ ở chế độ sáng và tối

5.4 Luồng 3: Thả chai (Send Bottle)

Luồng thả chai cho phép người dùng nhập nội dung tâm sự và gửi ra hệ thống. Sau khi xác nhận gửi, ứng dụng hiển thị trạng thái đang gửi để tạo cảm giác chai đang “trôi” trên biển.



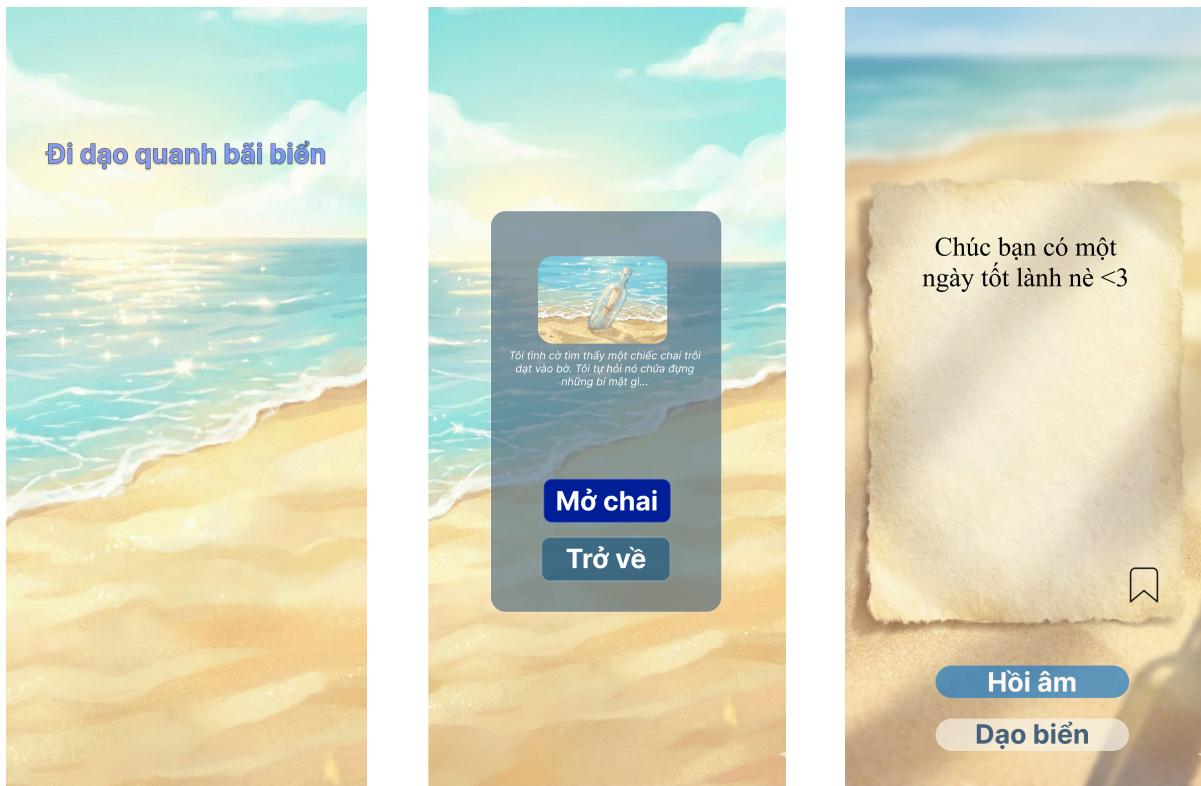
Hình 3: Luồng thả chai ở chế độ sáng



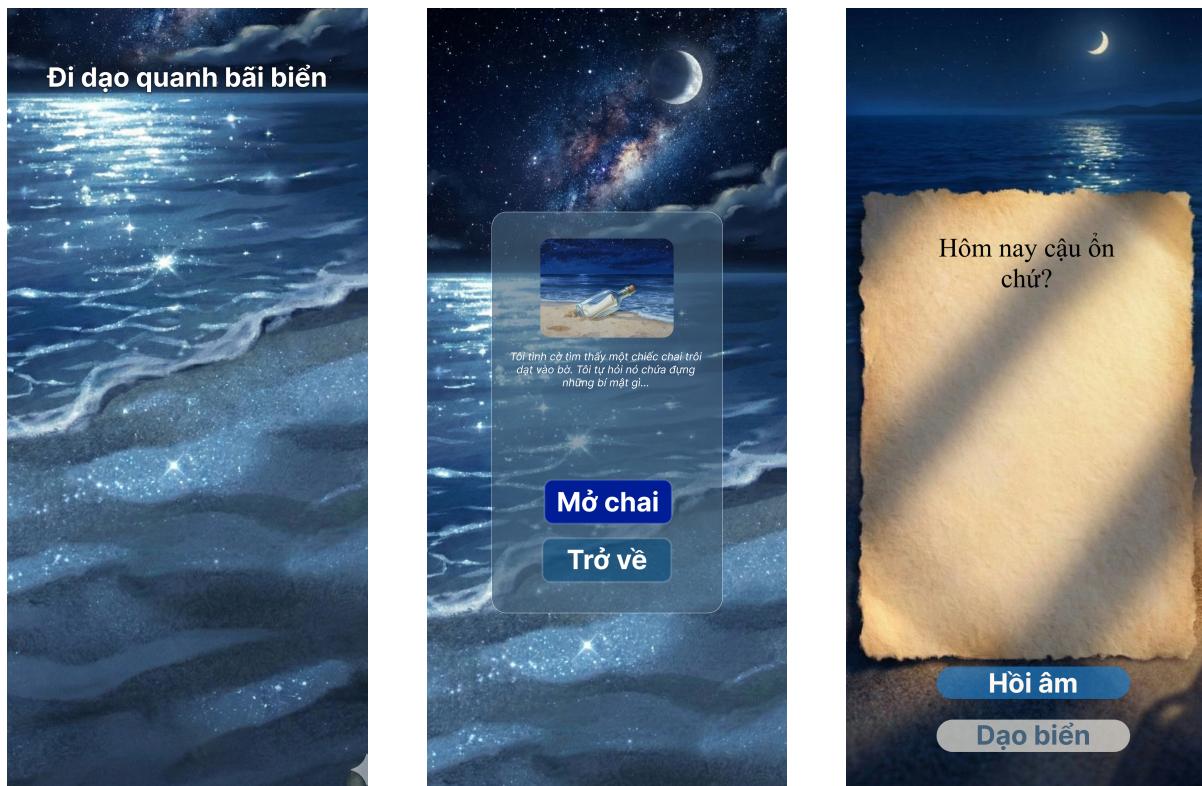
Hình 4: Luồng thả chai ở chế độ tối

5.5 Luồng 4: Nhặt chai và xem nội dung

Khi người dùng chọn nhặt chai, ứng dụng hiển thị trạng thái chờ, sau đó thông báo mở chai và nội dung chi tiết của chai được nhận.



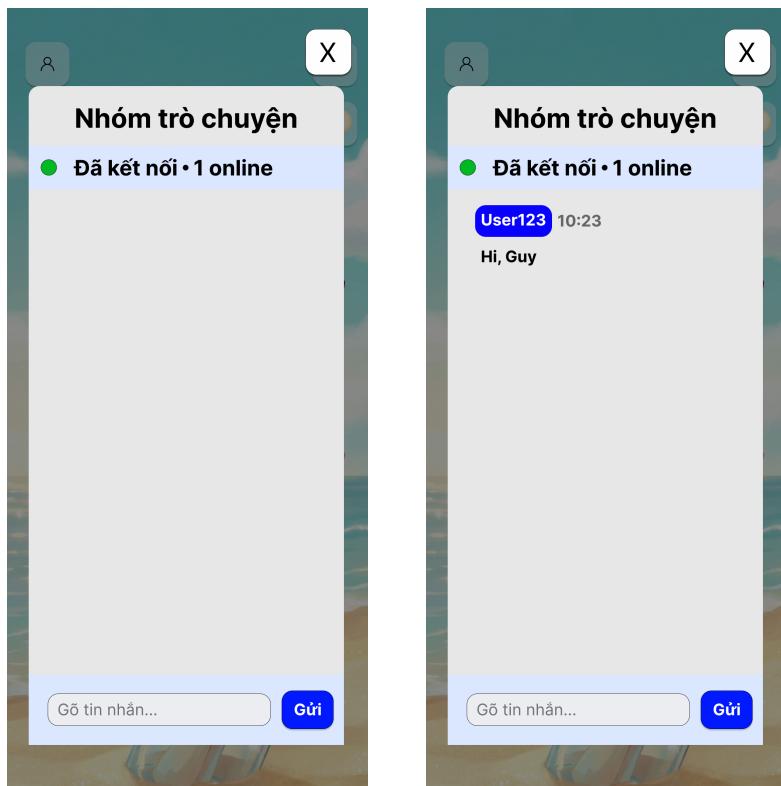
Hình 5: Luồng nhặt chai và xem nội dung (chế độ sáng)



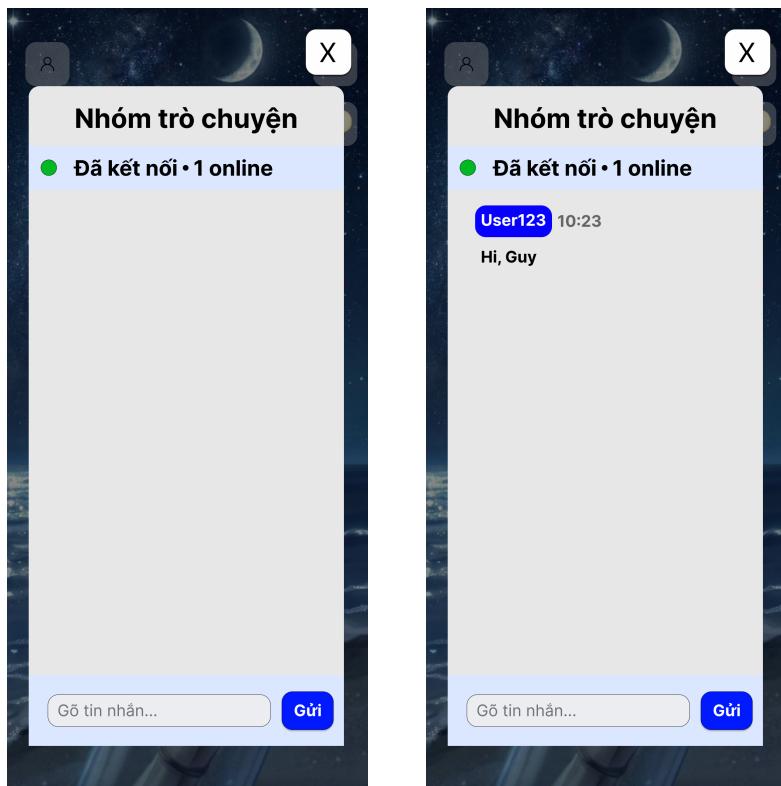
Hình 6: Luồng nhặt chai và xem nội dung (chế độ tối)

5.6 Luồng 5: Trò chuyện và phản hồi

Sau khi đọc chai, người dùng có thể phản hồi thông qua giao diện trò chuyện đơn giản. Chức năng này tạo tương tác hai chiều nhưng vẫn giữ mức độ nhẹ nhàng.



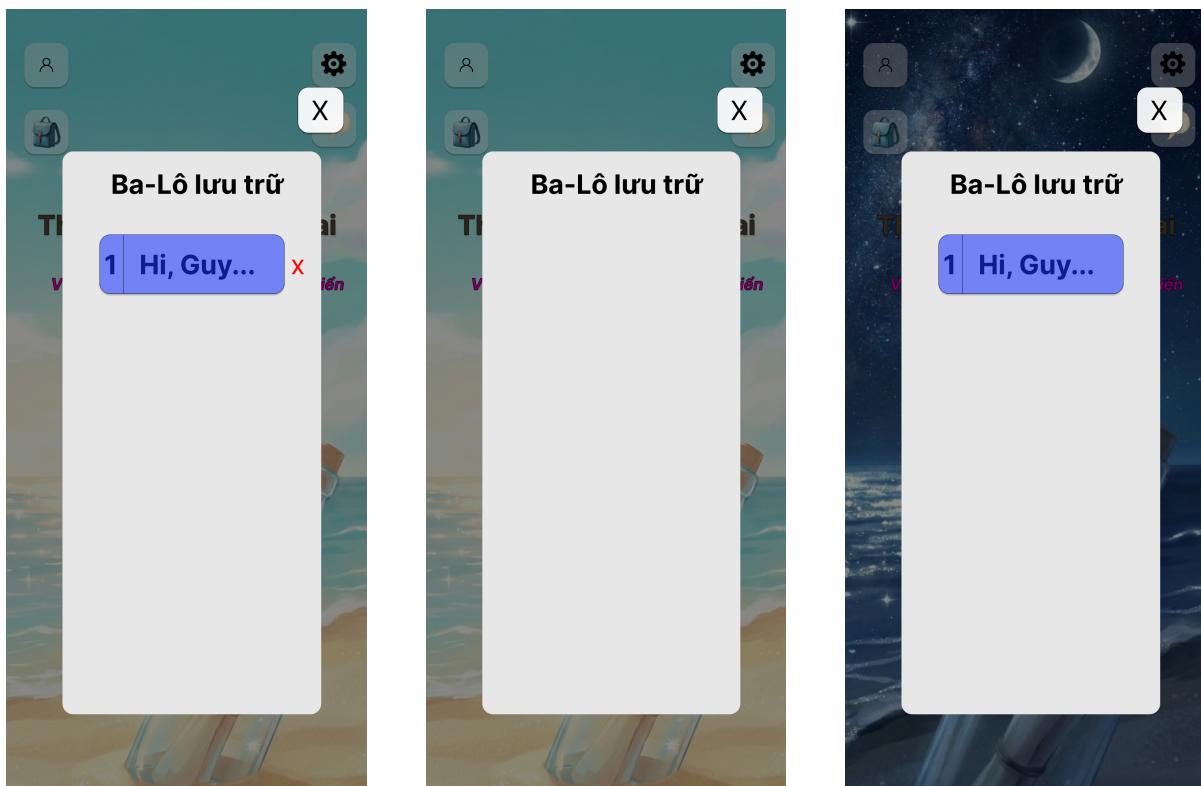
Hình 7: Giao diện trò chuyện (chế độ sáng)



Hình 8: Giao diện trò chuyện (chế độ tối)

5.7 Luồng 6: Lưu trữ chai

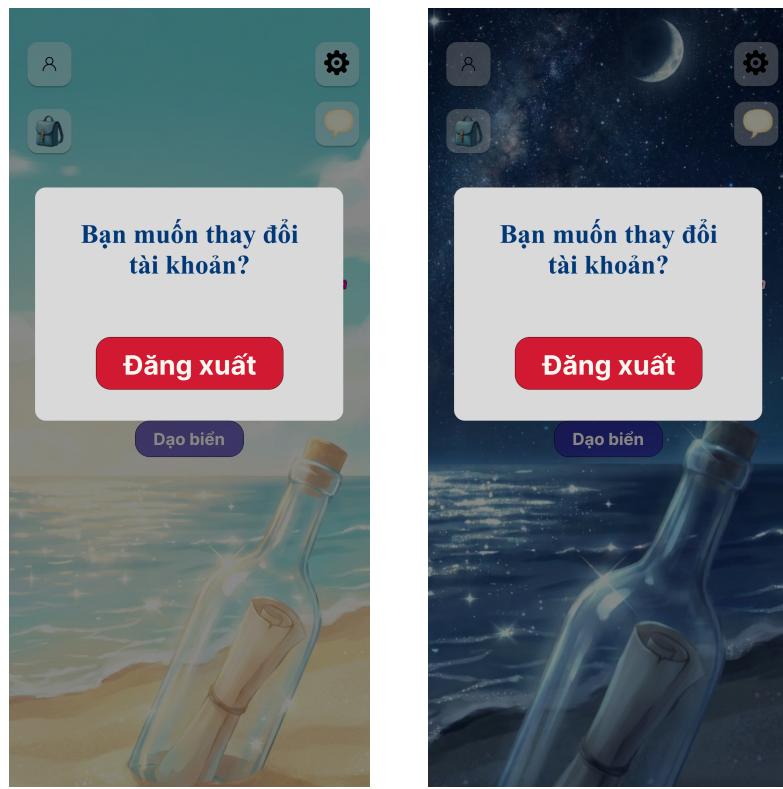
Người dùng có thể lưu lại các chai đã nhận để xem lại sau. Ứng dụng cung cấp cả trạng thái danh sách có dữ liệu và danh sách rỗng.



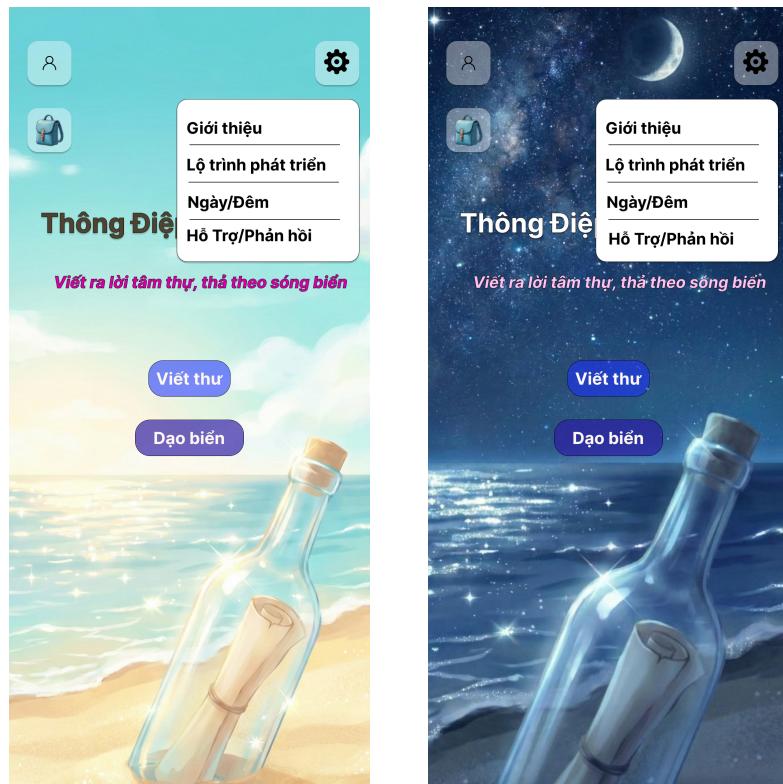
Hình 9: Chức năng lưu trữ chai ở chế độ sáng và tối

5.8 Luồng 7: Quản lý tài khoản và cài đặt

Ứng dụng cho phép người dùng quản lý thông tin tài khoản và các thiết lập cơ bản, bao gồm chuyển đổi giao diện sáng/tối.



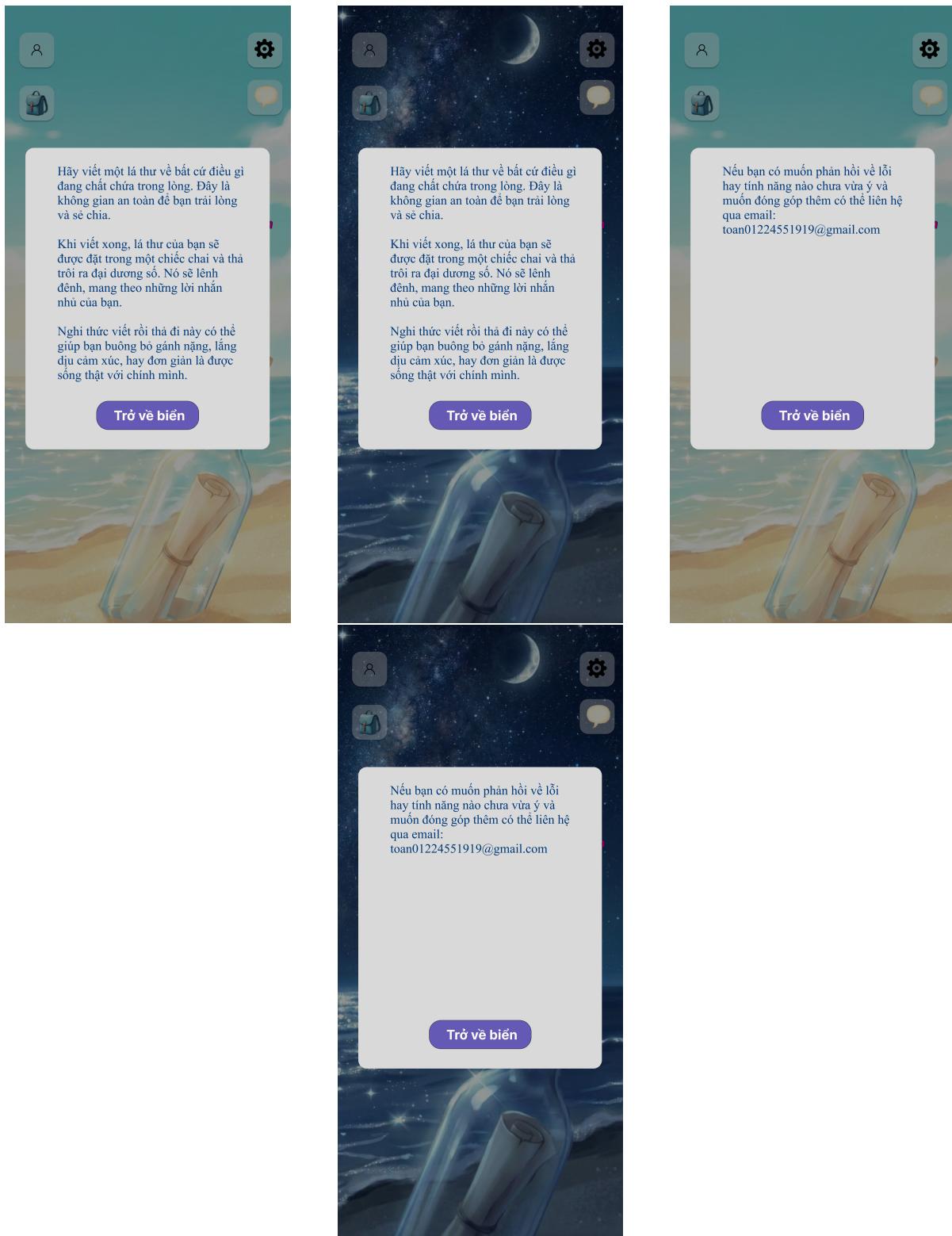
Hình 10: Màn hình tài khoản ở chế độ sáng và tối



Hình 11: Màn hình cài đặt ở chế độ sáng và tối

5.9 Luồng 8: Giới thiệu và hỗ trợ

Ứng dụng cung cấp các màn hình giới thiệu và hỗ trợ để giải thích ý tưởng hệ thống và cung cấp thông tin cần thiết cho người dùng.



Hình 12: Màn hình giới thiệu và hỗ trợ ở chế độ sáng và tối

6 Kiến trúc hệ thống

6.1 Tổng quan kiến trúc

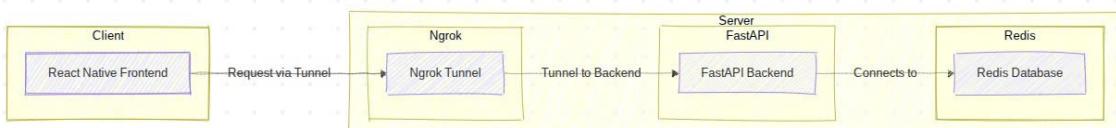
SeaWhisper được xây dựng theo mô hình client-server nhằm tách biệt rõ ràng giữa tầng giao diện (frontend) và tầng xử lý nghiệp vụ (backend). Ứng dụng di động (React Native Expo) đóng vai trò client, chịu trách nhiệm hiển thị giao diện, thu thập dữ liệu người dùng và gửi/nhận dữ liệu thông qua HTTP. Backend (FastAPI) là trung tâm xử lý nghiệp vụ: xác thực, quản lý chia, phân phối chia ngẫu nhiên, lưu trữ và truy vấn dữ liệu. Redis được sử dụng như cơ sở dữ liệu chính trong MVP nhờ tốc độ truy xuất cao và phù hợp với các thao tác kiểu hàng đợi/danh sách.

Thiết kế kiến trúc của dự án hướng đến các tiêu chí:

- **Đơn giản, dễ triển khai:** phù hợp phạm vi MVP và thời gian học kỳ.
- **Đảm bảo tính ẩn danh:** không hiển thị thông tin cá nhân khi chia sẻ nội dung.
- **Hiệu năng tốt:** phản hồi nhanh cho thao tác gửi/nhặt chiai.
- **Dễ mở rộng:** có thể bổ sung thêm dịch vụ (moderation, analytics) ở các phiên bản sau.

6.2 Sơ đồ triển khai tổng thể và luồng dữ liệu

Hình 13 minh họa sơ đồ triển khai tổng thể. Ứng dụng di động gọi đến backend qua API REST. Trong giai đoạn phát triển/demo, hệ thống có thể sử dụng cơ chế expose API (ví dụ qua ngrok) để thiết bị thật truy cập backend nội bộ. Backend xử lý request và đọc/ghi dữ liệu vào Redis.



Hình 13: Sơ đồ triển khai tổng thể hệ thống SeaWhisper

6.2.1 Luồng dữ liệu chính

Các luồng dữ liệu chính trong MVP gồm:

- **Đăng ký / Đăng nhập:** client gửi thông tin đăng ký/đăng nhập → backend xác thực → Redis lưu/đọc thông tin người dùng → trả về token (nếu có) và thông tin phiên.
- **Thả chiai:** client gửi nội dung chiai → backend tạo bản ghi Bottle → lưu Redis → trả về trạng thái thành công.

- **Nhặt chai:** client gửi yêu cầu lấy chai → backend chọn ngẫu nhiên chai chưa được đọc → trả về nội dung → cập nhật trạng thái để tránh phân phôi lặp.
- **Lưu chai:** client yêu cầu lưu → backend tạo StoredBottle gắn với user → Redis lưu mapping → trả về kết quả.

6.2.2 Quy ước giao tiếp

Client và backend giao tiếp bằng:

- **HTTP/HTTPS** theo RESTful API.
- Dữ liệu trao đổi ở định dạng **JSON**.
- Token (nếu có) được gửi qua header `Authorization: Bearer <token>`.

6.3 Phân rã thành phần hệ thống

6.3.1 Frontend (React Native Expo)

Frontend chịu trách nhiệm:

- Hiển thị UI theo luồng MVP (đăng nhập, trang chủ, thả chai, nhặt chai, lưu trữ, chat, cài đặt).
- Thực hiện kiểm tra dữ liệu đầu vào cơ bản (ví dụ: nội dung không rỗng).
- Gọi API và xử lý trạng thái tải (loading), lỗi (error) và kết quả (success).
- Quản lý trạng thái phiên (session) và điều hướng giữa các màn hình.

6.3.2 Backend (FastAPI)

Backend được tổ chức theo hướng tách lớp để dễ bảo trì:

- **Router/Controller:** định nghĩa endpoint, nhận request và trả response.
- **Service:** chứa logic nghiệp vụ (tạo chai, chọn chai ngẫu nhiên, lưu chai).
- **Repository/Data access:** thao tác đọc/ghi Redis.
- **Schema/DTO:** chuẩn hóa cấu trúc dữ liệu vào/ra (request/response).

Các nhóm chức năng chính:

- **Auth:** đăng ký/đăng nhập, tạo token và quản lý user.
- **Bottle:** tạo chai, lấy chai ngẫu nhiên, cập nhật trạng thái chai.
- **StoredBottle:** lưu chai, xem danh sách đã lưu, xóa chai đã lưu.

6.3.3 Database (Redis)

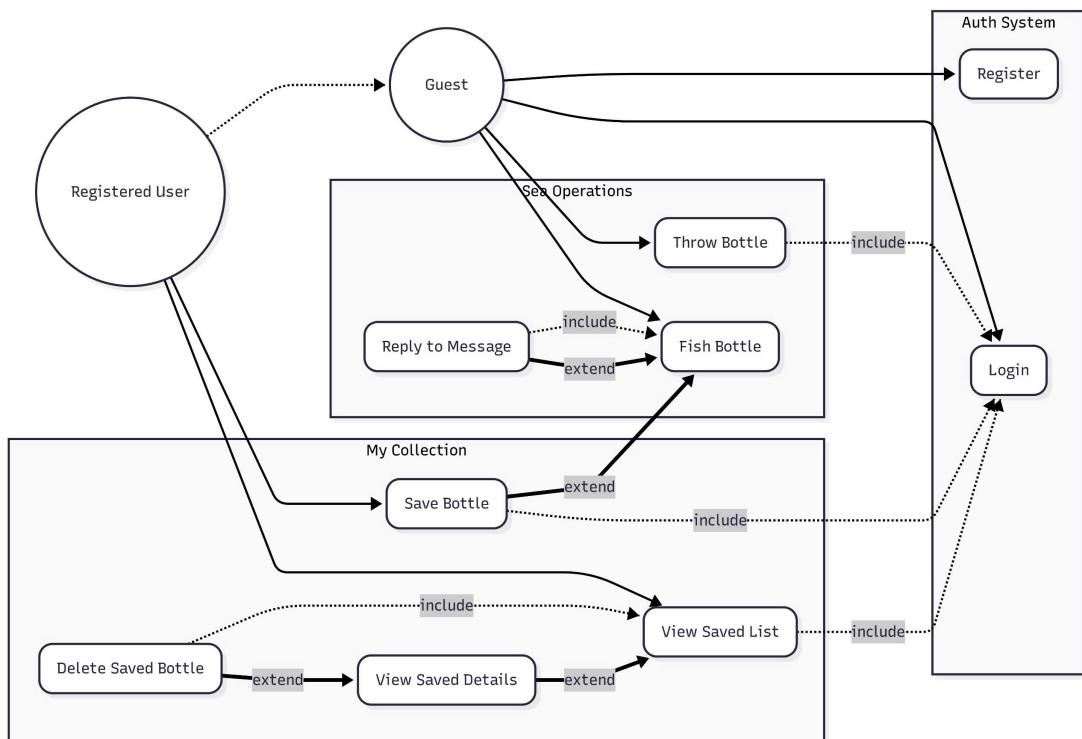
Redis đảm nhiệm vai trò lưu trữ dữ liệu tạm thời/nhanh trong MVP, hỗ trợ:

- Lưu user và thông tin xác thực.
- Lưu danh sách chai và trạng thái chai.
- Lưu danh sách chai đã lưu theo từng user (hỗ trợ phân trang mức cơ bản).

6.4 Use Case Diagram và phạm vi chức năng

Hình 14 mô tả các tác nhân và các chức năng chính. Trong hệ thống có thể xem hai nhóm tác nhân:

- **Guest (chưa đăng nhập):** có thể truy cập một số chức năng cơ bản tùy thiết kế, nhưng trong MVP tập trung cho người dùng có tài khoản để phục vụ lưu trữ.
- **Registered User (đã đăng nhập):** thực hiện đầy đủ thao tác thả chai, nhặt chai, chat, lưu/xem/xóa chai đã lưu.



Hình 14: Use Case Diagram của hệ thống SeaWhisper

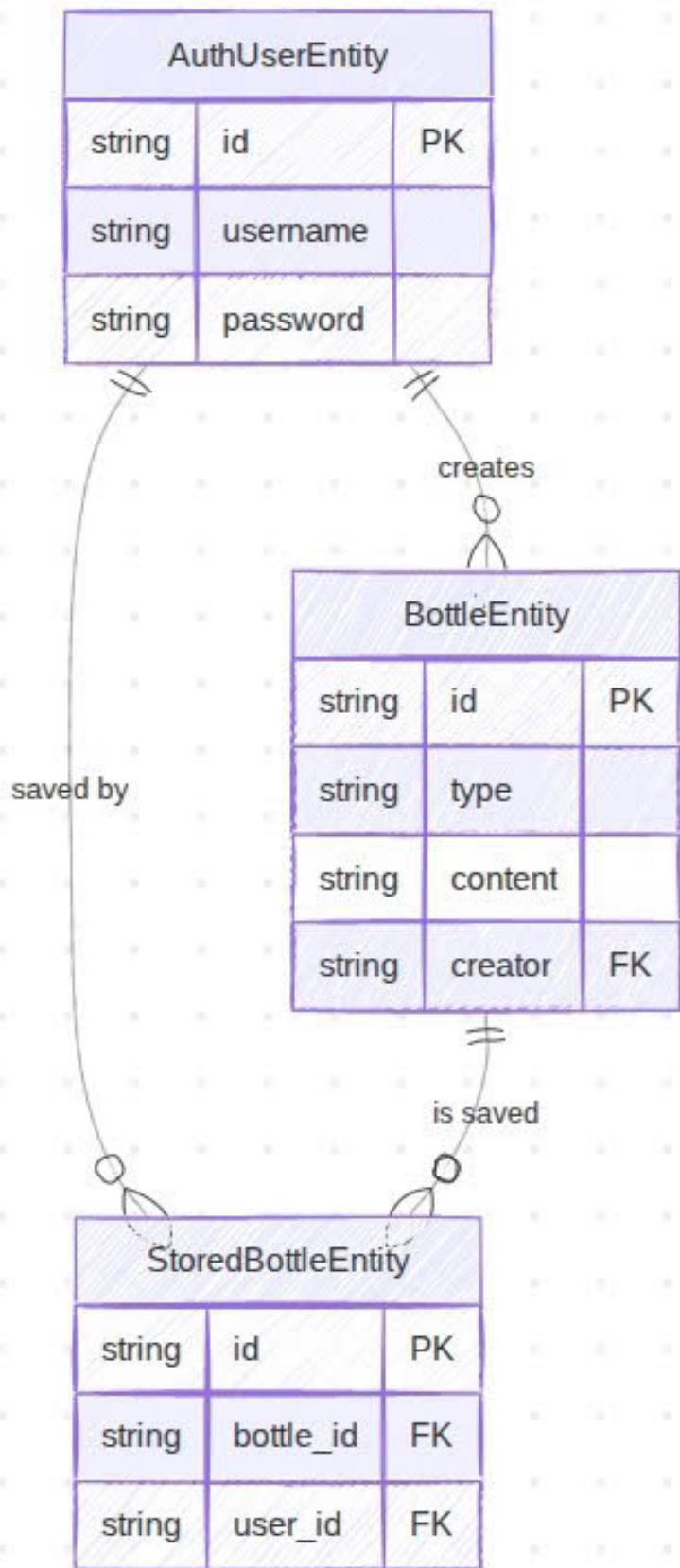
Từ use case, các yêu cầu chức năng MVP có thể tóm tắt:

- Đăng ký / đăng nhập.
- Thả chai (tạo nội dung).

- Nhặt chai (lấy ngẫu nhiên).
- Trò chuyện/ phản hồi.
- Lưu chai và quản lý danh sách đã lưu.

6.5 Thiết kế dữ liệu (ERD) và mô hình lưu trữ

Hình 15 minh họa ERD của hệ thống. Trong phạm vi MVP, dữ liệu tập trung vào ba thực thể chính: `AuthUser`, `Bottle` và `StoredBottle`. Mỗi quan hệ thể hiện rằng một người dùng có thể lưu nhiều chai, và mỗi chai có thể được lưu bởi nhiều người dùng (tùy thiết kế), do đó `StoredBottle` đóng vai trò bảng liên kết.



Hình 15: Sơ đồ ERD của hệ thống

6.5.1 Mô tả thực thể

- **AuthUser:**

- Lưu thông tin tài khoản phục vụ xác thực (ví dụ: username/email, mật khẩu đã hash, thời điểm tạo).
- Có thể kèm các metadata tối thiểu phục vụ trải nghiệm (ví dụ: cài đặt theme).

- **Bottle:**

- Lưu nội dung chai (text), thời điểm tạo, và trạng thái (ví dụ: chưa đọc/đã phân phối).
- Có thể lưu `creatorId` ở dạng định danh nội bộ (không hiển thị ra UI) để phục vụ kiểm soát hệ thống.

- **StoredBottle:**

- Lưu quan hệ giữa user và chai đã lưu (`userId`, `bottleId`), cùng thời điểm lưu.
- Phục vụ truy vấn danh sách chai đã lưu theo từng user.

6.5.2 Lý do chọn Redis cho MVP

Redis phù hợp trong MVP do:

- Truy xuất nhanh, phù hợp tương tác thời gian thực.
- Hỗ trợ cấu trúc dữ liệu danh sách/tập hợp, thuận tiện cho hàng đợi chai và danh sách lưu.
- Triển khai đơn giản, giảm thời gian thiết kế schema phức tạp.

Tuy nhiên, trong triển khai thực tế quy mô lớn, hệ thống có thể cần chuyển sang cơ sở dữ liệu bền vững hơn (PostgreSQL/MySQL) hoặc kết hợp Redis làm cache.

6.6 Bảo mật và tính ẩn danh ở mức kiến trúc

Trong phạm vi MVP, hệ thống đảm bảo tính ẩn danh ở mức hiển thị và luồng dữ liệu:

- Nội dung chai không hiển thị danh tính người gửi trên UI.
- Client chỉ trao đổi với backend bằng token/định danh nội bộ.
- Phân quyền cơ bản: các API lưu/xem/xóa chai đã lưu yêu cầu user hợp lệ.

6.7 Đánh giá kiến trúc và khả năng mở rộng

Kiến trúc hiện tại đáp ứng tốt yêu cầu MVP và có thể mở rộng theo các hướng:

- Tách backend thành các service độc lập (Auth, Bottle, StoredBottle) khi hệ thống lớn hơn.
- Bổ sung cơ chế logging/monitoring và phân tích hành vi người dùng.
- Thay đổi chiến lược phân phối chai (theo chủ đề, theo thời gian, theo mức độ tương tác).
- Triển khai cơ chế kiểm duyệt nội dung hoặc báo cáo vi phạm khi có thêm thời gian và tài nguyên.

7 Thiết kế Web Service (API)

7.1 Tổng quan

SeaWhisper sử dụng kiến trúc RESTful API để kết nối giữa ứng dụng di động (React Native Expo) và backend (FastAPI). Các API được thiết kế xoay quanh các chức năng cốt lõi của hệ thống, bao gồm xác thực người dùng, thả chai, nhặt chai và quản lý danh sách chai đã lưu.

Backend được xây dựng bằng FastAPI, hỗ trợ tự động sinh tài liệu API theo chuẩn OpenAPI và cung cấp giao diện Swagger để phục vụ phát triển, kiểm thử và đánh giá hệ thống.

7.2 Quy ước thiết kế API

Các quy ước chung được áp dụng cho toàn bộ hệ thống:

- Tuân thủ mô hình RESTful.
- Sử dụng các phương thức HTTP tiêu chuẩn: GET, POST, DELETE.
- Dữ liệu trao đổi sử dụng định dạng JSON.
- Các API yêu cầu xác thực sử dụng header `Authorization: Bearer <token>`.
- Trạng thái xử lý được biểu diễn thông qua mã trạng thái HTTP.

7.3 Nhóm API xác thực người dùng (auth)

Nhóm API này chịu trách nhiệm đăng ký và đăng nhập người dùng, cung cấp định danh để phục vụ các chức năng yêu cầu quyền truy cập.

7.3.1 POST /auth/register – Đăng ký tài khoản

Mô tả: Tạo tài khoản người dùng mới trong hệ thống.

Request body:

```
{  
    "username": "string",  
    "password": "string"  
}
```

Phản hồi:

- **201 Created:** Đăng ký thành công.
- **409 Conflict:** Tên người dùng đã tồn tại.

7.3.2 POST /auth/login – Đăng nhập

Mô tả: Xác thực thông tin đăng nhập và trả về token sử dụng cho các API yêu cầu xác thực.

Request body:

```
{  
    "username": "string",  
    "password": "string"  
}
```

Phản hồi:

- **200 OK:** Đăng nhập thành công, trả về token.
- **401 Unauthorized:** Thông tin đăng nhập không hợp lệ.

7.4 Nhóm API thả chai và nhặt chai (sea-bottle)

Đây là nhóm API cốt lõi, hiện thực hóa ý tưởng “message in a bottle” của hệ thống.

7.4.1 POST /sea/bottle – Tạo chai mới

Mô tả: Cho phép người dùng gửi nội dung tâm sự dưới dạng một chai mới.

Request body:

```
{  
    "content": "Nội dung tâm sự"  
}
```

Ghi chú: API này có thể hoạt động với hoặc không có token xác thực. Nếu có token, hệ thống lưu thông tin người tạo ở mức nội bộ nhưng không hiển thị ra giao diện.

Phản hồi:

- **201 Created:** Chai được tạo thành công.
- **400 Bad Request:** Nội dung không hợp lệ.

7.4.2 GET /sea/bottle – Lấy chai ngẫu nhiên

Mô tả: Trả về ngẫu nhiên một chai hiện có trong hệ thống.

Phản hồi:

- **200 OK:** Trả về nội dung chai.
- **204 No Content:** Không còn chai khả dụng.

Sau khi chai được trả về, backend đảm bảo không phân phối trùng lặp trong cùng một ngữ cảnh xử lý.

7.5 Nhóm API quản lý chai đã lưu (api-store-bottle)

Nhóm API này cho phép người dùng lưu, xem và quản lý các chai đã lưu trong bộ sưu tập cá nhân.

7.5.1 POST /api/store-bottle – Lưu chai

Mô tả: Lưu một chai vào danh sách lưu trữ của người dùng hiện tại.

Request body:

```
{  
    "bottle_id": "string"  
}
```

Phản hồi:

- **201 Created:** Lưu chai thành công.
- **409 Conflict:** Chai đã tồn tại trong danh sách lưu.

7.5.2 GET /api/store-bottle – Lấy danh sách chai đã lưu

Mô tả: Trả về danh sách các chai đã lưu của người dùng, hỗ trợ phân trang thông qua tham số truy vấn.

Query parameter:

- **page:** số trang (ví dụ: page=1)

Phản hồi:

- **200 OK:** Trả về danh sách chai đã lưu.

7.5.3 GET /api/store-bottle/{stored_bottle_id} – Lấy chi tiết chai đã lưu

Mô tả: Trả về thông tin chi tiết của một chai đã lưu cụ thể.

Phản hồi:

- **200 OK:** Trả về thông tin chi tiết chai.
- **404 Not Found:** Không tìm thấy chai đã lưu.

7.5.4 DELETE /api/store-bottle/{stored_bottle_id} – Xóa chai đã lưu

Mô tả: Xóa một chai khỏi danh sách lưu trữ của người dùng.

Phản hồi:

- **204 No Content:** Xóa thành công.
- **404 Not Found:** Không tìm thấy chai cần xóa.

7.6 Nhóm API hệ thống

7.6.1 GET /health – Health check

Mô tả: API dùng để kiểm tra trạng thái hoạt động của backend.

Phản hồi:

- **200 OK:** Backend hoạt động bình thường.

7.6.2 GET /api/me – Lấy thông tin người dùng hiện tại

Mô tả: Trả về thông tin cơ bản của người dùng hiện đang đăng nhập, dựa trên token xác thực.

Phản hồi:

- **200 OK:** Trả về thông tin người dùng.
- **401 Unauthorized:** Token không hợp lệ hoặc hết hạn.

7.7 Tài liệu API trực tuyến (Swagger)

Toàn bộ các API được mô tả trong chương này đều được triển khai thực tế và công bố thông qua Swagger UI tại địa chỉ:

<https://unsatiating-clustered-phoenix.ngrok-free.dev/swagger>

Swagger UI hỗ trợ xem chi tiết endpoint, tham số, cấu trúc request/response và thử nghiệm API trực tiếp, đảm bảo tính nhất quán giữa thiết kế và triển khai thực tế của hệ thống.

8 Chiến lược triển khai (Deployment Strategy)

8.1 Tổng quan

Trong phạm vi MVP, SeaWhisper được triển khai theo hướng đơn giản, linh hoạt và dễ kiểm soát, phù hợp với môi trường học tập và thử nghiệm. Chiến lược triển khai tập trung vào việc đảm bảo backend hoạt động ổn định, frontend có thể truy cập backend từ thiết bị thật, đồng thời cung cấp các điểm truy cập công khai để phục vụ đánh giá và demo.

Hệ thống được chia thành các thành phần triển khai chính: backend API, tài liệu API trực tuyến (Swagger), landing page giới thiệu dự án và hệ thống quản lý mã nguồn.

8.2 Triển khai Backend

Backend của SeaWhisper được xây dựng bằng FastAPI và chạy dưới dạng một dịch vụ độc lập. Trong giai đoạn phát triển và demo, backend được triển khai trên môi trường cục bộ (local environment), cho phép nhóm dễ dàng theo dõi log, debug và chỉnh sửa nhanh.

Để phục vụ việc truy cập từ thiết bị di động và bên ngoài mạng nội bộ, backend được expose ra Internet thông qua dịch vụ tạo tunnel.

8.3 Expose Backend bằng Ngrok

Ngrok được sử dụng để tạo tunnel từ backend cục bộ ra Internet, giúp client di động và giảng viên có thể truy cập API mà không cần triển khai lên hạ tầng cloud phức tạp.

Lợi ích của việc sử dụng ngrok:

- Không cần cấu hình máy chủ hoặc domain riêng.
- Cho phép truy cập backend từ thiết bị thật và môi trường bên ngoài.
- Phù hợp cho demo, kiểm thử và đánh giá MVP.

Qua ngrok, backend FastAPI được expose với một địa chỉ công khai và được sử dụng xuyên suốt trong quá trình phát triển frontend.

8.4 Triển khai tài liệu API (Swagger)

FastAPI tự động sinh tài liệu API theo chuẩn OpenAPI và cung cấp giao diện Swagger UI. Swagger được sử dụng như một công cụ trung tâm để:

- Kiểm tra trạng thái các endpoint.
- Thử nghiệm API độc lập với ứng dụng di động.
- Đổi chiếu giữa thiết kế API và triển khai thực tế.

Tài liệu API trực tuyến của hệ thống hiện đang được triển khai tại địa chỉ:

<https://unsatiating-clustered-phoenix.ngrok-free.dev/swagger>

Việc cung cấp Swagger online giúp tăng tính minh bạch và hỗ trợ quá trình đánh giá hệ thống.

8.5 Triển khai Landing Page

Bên cạnh ứng dụng di động, dự án SeaWhisper có một landing page công khai nhằm giới thiệu tổng quan ý tưởng, mục tiêu và các tính năng chính của hệ thống. Landing page đóng vai trò là điểm truy cập thông tin chính thức cho dự án.

Landing page được triển khai thông qua GitHub Pages và có thể truy cập tại:

<https://github.com/btlmobile/showcase>

Trang này hỗ trợ:

- Giới thiệu ý tưởng và định hướng của SeaWhisper.
- Trình bày các tính năng chính của ứng dụng.
- Cung cấp liên kết đến mã nguồn và tài liệu liên quan.

8.6 Quản lý mã nguồn và cộng tác nhóm

Toàn bộ mã nguồn của dự án được quản lý thông qua GitHub Organization. Việc sử dụng GitHub Organization cho phép nhóm phân chia rõ ràng các repository, quản lý quyền truy cập và theo dõi quá trình đóng góp của từng thành viên.

GitHub Organization của dự án có thể truy cập tại:

<https://github.com/btlmobile>

GitHub được sử dụng để:

- Lưu trữ và quản lý mã nguồn frontend và backend.
- Theo dõi lịch sử commit và quá trình phát triển.
- Hỗ trợ làm việc nhóm và phối hợp giữa các thành viên.

8.7 Đánh giá chiến lược triển khai

Chiến lược triển khai hiện tại đáp ứng tốt yêu cầu của phiên bản MVP:

- Đảm bảo backend hoạt động ổn định và dễ truy cập.
- Hỗ trợ kiểm thử và demo hệ thống một cách thuận tiện.
- Giảm chi phí và độ phức tạp trong giai đoạn đầu.

Trong các phiên bản tương lai, hệ thống có thể được mở rộng bằng cách triển khai backend lên các nền tảng cloud (AWS, Railway, Render) và bổ sung CI/CD để tự động hóa quá trình build và deploy.

9 Testing Coverage Report

9.1 Mục tiêu kiểm thử

Mục tiêu của hoạt động kiểm thử trong dự án SeaWhisper là:

- Đảm bảo các chức năng cốt lõi của ứng dụng hoạt động đúng theo thiết kế.
- Phát hiện sớm các lỗi liên quan đến logic xử lý, bảo mật và khả năng bảo trì mã nguồn.
- Đánh giá chất lượng mã nguồn trong phạm vi phát triển Minimum Viable Product (MVP).

Do giới hạn về thời gian và nguồn lực, nhóm tập trung kiểm thử ở mức độ hệ thống và phân tích chất lượng mã, thay vì triển khai đầy đủ các bộ kiểm thử tự động (unit test) cho toàn bộ module.

9.2 Công cụ kiểm thử và đánh giá chất lượng

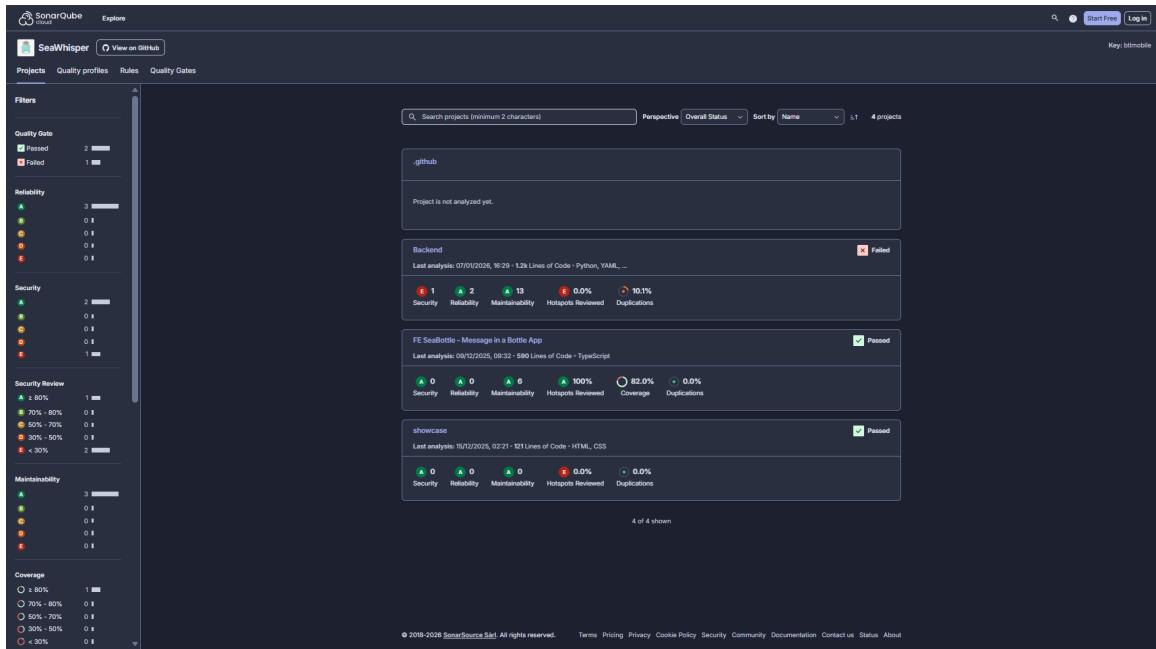
Nhóm sử dụng các công cụ sau trong quá trình kiểm thử và đánh giá:

- **SonarCloud:** Phân tích chất lượng mã nguồn, bao gồm bảo mật (Security), độ tin cậy (Reliability), khả năng bảo trì (Maintainability), độ trùng lặp mã (Duplications) và độ bao phủ kiểm thử (Coverage).
- **Swagger UI:** Kiểm thử thủ công (manual testing) các API backend thông qua giao diện tài liệu REST API.
- **Kiểm thử thủ công trên thiết bị:** Xác nhận luồng người dùng (user flow) trên ứng dụng mobile React Native Expo.

9.3 Phân tích chất lượng mã nguồn với SonarCloud

Dự án SeaWhisper sử dụng SonarCloud để phân tích chất lượng mã nguồn cho các repository thuộc GitHub Organization của nhóm. Dashboard tổng quan của các project có thể truy cập công khai tại:

<https://sonarcloud.io/organizations/btlmobile/projects>



Hình 16: Tổng quan phân tích chất lượng mã nguồn dự án SeaWhisper trên SonarCloud

Hình 16 thể hiện kết quả phân tích chất lượng mã nguồn của các thành phần chính trong hệ thống, bao gồm frontend, backend và landing page.

9.4 Kết quả phân tích Frontend

Phần frontend của ứng dụng SeaWhisper được phát triển bằng React Native Expo và TypeScript. Kết quả phân tích trên SonarCloud cho thấy:

- Frontend đạt **Quality Gate: Passed**.
- Không phát hiện vấn đề nghiêm trọng liên quan đến bảo mật và độ tin cậy.
- Độ trùng lặp mã ở mức rất thấp.
- Cấu trúc mã rõ ràng, dễ bảo trì trong phạm vi MVP.

Điều này cho thấy phần frontend đáp ứng tốt các tiêu chí chất lượng mã và phù hợp cho việc mở rộng trong các phiên bản tiếp theo.

9.5 Kết quả phân tích Backend

Backend của hệ thống được xây dựng bằng FastAPI và Redis. Kết quả phân tích SonarCloud cho thấy backend hiện tại chưa đạt Quality Gate do tồn tại một số vấn đề liên quan đến:

- Một số cảnh báo về bảo mật và code smell.
- Chưa triển khai đầy đủ các kiểm thử tự động dẫn đến độ bao phủ kiểm thử thấp.

Tuy nhiên, trong phạm vi MVP, nhóm ưu tiên hoàn thiện các chức năng nghiệp vụ chính như xác thực người dùng, tạo bottle, lấy bottle ngẫu nhiên và quản lý bottle đã lưu. Các vấn đề được SonarCloud chỉ ra được xem là cơ sở quan trọng cho việc cải thiện chất lượng mã nguồn trong các giai đoạn phát triển tiếp theo.

9.6 Kiểm thử API Backend

Các API backend được kiểm thử thủ công thông qua Swagger UI, bao gồm các nhóm chức năng chính:

- Xác thực người dùng (đăng ký, đăng nhập).
- Tạo bottle và lấy bottle ngẫu nhiên.
- Lưu bottle, xem danh sách bottle đã lưu và xóa bottle đã lưu.

Quá trình kiểm thử xác nhận rằng:

- Các API phản hồi đúng mã trạng thái HTTP.
- Dữ liệu trả về đúng theo thiết kế schema.
- Các trường hợp lỗi phổ biến được xử lý hợp lý (ví dụ: bottle không tồn tại, truy cập không hợp lệ).

9.7 Kiểm thử luồng người dùng (User Flow)

Ngoài kiểm thử API, nhóm tiến hành kiểm thử thủ công luồng người dùng trên ứng dụng mobile, bao gồm:

- Luồng đăng ký và đăng nhập.
- Luồng thả chai (Send Bottle).
- Luồng nhặt chai (Receive Bottle).
- Luồng lưu bottle và quản lý bộ sưu tập cá nhân.

Kết quả cho thấy các luồng chính hoạt động ổn định, đáp ứng được trải nghiệm người dùng theo mục tiêu của ứng dụng SeaWhisper.

9.8 Đánh giá tổng thể và hạn chế

Tổng thể, hoạt động kiểm thử của dự án đạt yêu cầu đối với một sản phẩm ở giai đoạn MVP. Tuy nhiên, vẫn tồn tại một số hạn chế:

- Chưa triển khai đầy đủ unit test và integration test cho backend.
- Độ bao phủ kiểm thử tự động còn thấp.

Các hạn chế này được ghi nhận và xem là định hướng cải tiến quan trọng cho các phiên bản tiếp theo, khi dự án được mở rộng về quy mô và tính năng.

10 Khảo sát sự hài lòng của người dùng và phân tích hành vi người dùng

10.1 Mục tiêu khảo sát

Khảo sát người dùng được thực hiện nhằm đánh giá mức độ hài lòng và cảm nhận ban đầu của người dùng đối với ứng dụng SeaWhisper ở giai đoạn Minimum Viable Product (MVP). Cụ thể, khảo sát hướng tới các mục tiêu sau:

- Đánh giá mức độ hài lòng tổng thể của người dùng đối với ứng dụng.
- Đánh giá trải nghiệm giao diện và các chức năng cốt lõi như thả chai và nhặt chai.
- Xác định mức độ phù hợp của ứng dụng trong việc hỗ trợ chia sẻ cảm xúc cá nhân.
- Thu thập ý kiến phản hồi và đề xuất cải thiện cho các phiên bản tiếp theo.

Kết quả khảo sát đóng vai trò là cơ sở quan trọng để đánh giá hiệu quả ban đầu của ý tưởng và định hướng phát triển trong tương lai.

10.2 Phương pháp và đối tượng khảo sát

Khảo sát được thực hiện thông qua nền tảng Google Forms với các đặc điểm sau:

- Đối tượng khảo sát: Sinh viên và người trẻ trong độ tuổi từ 18 đến 24.
- Hình thức câu hỏi: Kết hợp câu hỏi trắc nghiệm, thang đo Likert 5 mức và câu hỏi mở.
- Thời điểm thực hiện: Sau khi hoàn thành phiên bản MVP của ứng dụng SeaWhisper.

Thang đo Likert 5 mức được sử dụng để đánh giá mức độ hài lòng, với các mức từ 1 (rất không hài lòng) đến 5 (rất hài lòng). Phương pháp này giúp định lượng cảm nhận người dùng một cách rõ ràng và dễ phân tích.

10.3 Tổng quan mẫu khảo sát

Kết quả khảo sát cho thấy đa số người tham gia thuộc nhóm sinh viên và người trẻ, phù hợp với nhóm đối tượng mục tiêu của ứng dụng SeaWhisper. Về kinh nghiệm sử dụng ứng dụng ẩn danh, phần lớn người tham gia cho biết họ đã từng sử dụng các nền tảng tương tự như NGL hoặc Whisper, cho phép họ đưa ra đánh giá dựa trên trải nghiệm thực tế.

Điều này giúp đảm bảo tính khách quan và giá trị tham khảo của kết quả khảo sát.

10.4 Kết quả đánh giá mức độ hài lòng

10.4.1 Mức độ hài lòng tổng thể

Kết quả khảo sát cho thấy khoảng 65–70% người tham gia đánh giá mức độ hài lòng tổng thể của ứng dụng SeaWhisper ở mức hài lòng hoặc rất hài lòng. Một tỷ lệ nhỏ người dùng cho biết cảm nhận ở mức trung lập hoặc chưa hài lòng, phản ánh thực tế rằng ứng dụng vẫn đang trong giai đoạn MVP và chưa hoàn thiện đầy đủ các tính năng nâng cao.

Kết quả này cho thấy SeaWhisper đã đáp ứng tương đối tốt kỳ vọng ban đầu của người dùng về một nền tảng chia sẻ cảm xúc ẩn danh.

10.4.2 Đánh giá giao diện người dùng

Phần lớn người tham gia đánh giá tích cực về giao diện ứng dụng, cho rằng thiết kế mang phong cách biển và màu sắc dịu nhẹ giúp tạo cảm giác dễ chịu và thư giãn. Khoảng trên 70% người dùng lựa chọn mức hài lòng hoặc rất hài lòng đối với tiêu chí này.

Điều này cho thấy hướng thiết kế tập trung vào trải nghiệm cảm xúc (healing-oriented design) là phù hợp với mục tiêu của ứng dụng.

10.4.3 Đánh giá chức năng thả chai và nhặt chai

Hai chức năng cốt lõi của SeaWhisper là thả chai và nhặt chai nhận được phản hồi tích cực từ người dùng. Đa số người tham gia cho biết các chức năng này dễ sử dụng và mang lại trải nghiệm thú vị.

Tuy nhiên, vẫn tồn tại một tỷ lệ người dùng đánh giá ở mức trung lập hoặc chưa hài lòng, chủ yếu do ứng dụng hiện chưa có các cơ chế lọc nội dung nâng cao hoặc phản hồi phong phú. Điều này được xem là phù hợp với phạm vi MVP của dự án.

10.4.4 Mức độ phù hợp trong chia sẻ cảm xúc cá nhân

Khoảng 70% người tham gia cho rằng SeaWhisper là một nền tảng phù hợp để chia sẻ cảm xúc cá nhân. Kết quả này xác nhận rằng ý tưởng xây dựng một không gian chia sẻ ẩn danh mang tính chữa lành là có tiềm năng và đáp ứng nhu cầu thực tế của người dùng trẻ.

10.5 Ý định sử dụng trong tương lai

Khi được hỏi về ý định sử dụng SeaWhisper trong tương lai, phần lớn người tham gia cho biết họ sẵn sàng hoặc có thể tiếp tục sử dụng ứng dụng nếu được phát triển và hoàn thiện thêm. Tổng tỷ lệ người dùng lựa chọn “Có” hoặc “Có thể” chiếm trên 80%, trong khi một tỷ lệ nhỏ cho biết họ không có ý định sử dụng tiếp.

Kết quả này cho thấy ứng dụng có tiềm năng phát triển người dùng nếu được cải thiện thêm về tính năng và trải nghiệm.

10.6 Phân tích hành vi người dùng (User Behavior Analytics)

Do dự án đang ở giai đoạn MVP và chưa được triển khai rộng rãi, nhóm chưa tích hợp các công cụ phân tích hành vi người dùng tự động như Google Analytics hoặc Firebase Analytics. Thay vào đó, phân tích hành vi người dùng được thực hiện dựa trên các phương pháp định tính, bao gồm:

- Quan sát quá trình sử dụng ứng dụng trong các buổi kiểm thử nội bộ.
- Phản hồi trực tiếp từ người dùng tham gia khảo sát.
- Dánh giá khả năng hoàn thành các luồng chức năng chính.

Kết quả cho thấy người dùng có thể dễ dàng tiếp cận và hoàn thành các luồng chính như đăng nhập, thả chai, nhặt chai và lưu bottle. Các phản hồi định tính cho thấy người dùng quan tâm nhiều nhất đến trải nghiệm đọc thông điệp và cảm giác đồng cảm mà ứng dụng mang lại.

10.7 Tổng hợp phản hồi và đề xuất cải thiện

Phân tích các câu trả lời mở cho thấy một số đề xuất cải thiện phổ biến bao gồm:

- Bổ sung thêm các hình thức phản hồi cảm xúc đa dạng hơn.
- Cải thiện khả năng phân loại hoặc gợi ý nội dung.
- Hoàn thiện thêm giao diện và hiệu ứng tương tác.

Những đề xuất này được xem là định hướng quan trọng cho các phiên bản phát triển tiếp theo của SeaWhisper.

10.8 Dữ liệu khảo sát

Dữ liệu khảo sát chi tiết được tổng hợp và lưu trữ dưới dạng bảng tính Google Sheets nhằm phục vụ việc đối chiếu và kiểm tra khi cần thiết. Liên kết dữ liệu khảo sát được công bố tại:

<https://docs.google.com/spreadsheets/d/10ytnulUoMh47sXLI4K11ON-HdLEMNfOf2FuMkmmqxEdit>

Trong báo cáo này, nhóm chỉ trình bày các kết quả tổng hợp và phân tích chính nhằm đảm bảo tính súc tích và tập trung vào mục tiêu đánh giá trải nghiệm người dùng.

11 Đánh giá dự án và những điểm cần cải thiện

11.1 Đánh giá mức độ đáp ứng mục tiêu dự án

Dựa trên kết quả triển khai và phân tích ở các chương trước, có thể nhận định rằng dự án SeaWhisper đã đáp ứng tốt các mục tiêu ban đầu được đặt ra cho một sản phẩm ở giai đoạn Minimum Viable Product (MVP).

Cụ thể, ứng dụng đã:

- Cung cấp một nền tảng chia sẻ cảm xúc ẩn danh với các chức năng cốt lõi hoạt động ổn định.
- Xây dựng trải nghiệm người dùng hướng tới sự nhẹ nhàng và thư giãn thông qua thiết kế giao diện.
- Triển khai thành công hệ thống backend hỗ trợ xác thực người dùng, quản lý bottle và lưu trữ dữ liệu.

Các kết quả khảo sát người dùng và kiểm thử thực tế cho thấy ứng dụng đáp ứng tương đối tốt nhu cầu chia sẻ cảm xúc và khám phá nội dung ẩn danh của nhóm người dùng mục tiêu.

11.2 Đánh giá về trải nghiệm người dùng (UX/UI)

Từ kết quả khảo sát mức độ hài lòng, giao diện người dùng của SeaWhisper được đánh giá tích cực, đặc biệt về màu sắc, bố cục và phong cách thiết kế lấy cảm hứng từ biển. Thiết kế này góp phần tạo ra cảm giác an toàn và thư giãn cho người dùng khi tương tác với ứng dụng.

Các luồng người dùng chính như đăng nhập, thả chai, nhặt chai và lưu bottle được đánh giá là dễ tiếp cận và dễ sử dụng. Tuy nhiên, một số người dùng vẫn đánh giá ở mức trung lập hoặc chưa hài lòng, cho thấy vẫn còn dư địa để cải thiện trải nghiệm người dùng trong các phiên bản tiếp theo.

11.3 Đánh giá về chức năng và hiệu năng hệ thống

Về mặt chức năng, các tính năng cốt lõi của hệ thống đã được triển khai đầy đủ và hoạt động đúng theo thiết kế. Backend FastAPI kết hợp với Redis đáp ứng tốt yêu cầu xử lý dữ liệu nhanh và đơn giản, phù hợp với phạm vi MVP.

Tuy nhiên, do tập trung vào việc hoàn thiện chức năng nghiệp vụ, hệ thống chưa được tối ưu toàn diện về mặt hiệu năng và khả năng mở rộng. Một số vấn đề về chất lượng mã nguồn ở backend đã được SonarCloud chỉ ra, phản ánh nhu cầu cải thiện trong các giai đoạn phát triển tiếp theo.

11.4 Đánh giá về chất lượng mã nguồn

Phân tích chất lượng mã nguồn bằng SonarCloud cho thấy sự khác biệt rõ rệt giữa các thành phần của hệ thống. Phần frontend và landing page đạt Quality Gate, cho thấy mã nguồn được tổ chức tương đối tốt, dễ đọc và dễ bảo trì.

Ngược lại, phần backend vẫn tồn tại một số cảnh báo liên quan đến bảo mật và khả năng bảo trì. Điều này được xem là chấp nhận được trong phạm vi MVP, khi nhóm ưu tiên phát triển chức năng hơn là tối ưu mã nguồn ở mức độ sâu.

11.5 Hạn chế của dự án

Mặc dù đạt được các mục tiêu chính, dự án vẫn tồn tại một số hạn chế:

- Chưa triển khai đầy đủ các kiểm thử tự động như unit test và integration test.
- Chưa tích hợp công cụ phân tích hành vi người dùng tự động như Google Analytics hoặc Firebase Analytics.
- Một số tính năng nâng cao như lọc nội dung hoặc gợi ý phản hồi cảm xúc chưa được triển khai do giới hạn thời gian.

Các hạn chế này phản ánh đúng bối cảnh và phạm vi của một dự án MVP trong khuôn khổ môn học.

11.6 Định hướng cải tiến và phát triển trong tương lai

Dựa trên các đánh giá và phản hồi từ người dùng, một số hướng cải tiến cho SeaWhisper trong tương lai được đề xuất như sau:

- Bổ sung các cơ chế lọc và phân loại nội dung để nâng cao trải nghiệm người dùng.
- Triển khai các bộ kiểm thử tự động nhằm nâng cao độ tin cậy và chất lượng mã nguồn.
- Tích hợp công cụ phân tích hành vi người dùng để thu thập dữ liệu sử dụng thực tế.
- Mở rộng các hình thức tương tác và phản hồi cảm xúc giữa người dùng.

Những định hướng này sẽ giúp SeaWhisper phát triển từ một MVP thành một sản phẩm hoàn chỉnh và bền vững hơn trong tương lai.

11.7 Tổng kết

Tổng thể, SeaWhisper là một dự án đáp ứng tốt các yêu cầu của một ứng dụng mobile ở giai đoạn MVP. Dự án không chỉ thể hiện khả năng triển khai kỹ thuật mà còn cho thấy sự quan tâm đến trải nghiệm người dùng và yếu tố cảm xúc. Các hạn chế hiện tại được xem là cơ hội để tiếp tục cải tiến và phát triển trong các phiên bản tiếp theo.

12 Kết Luận

Trong khuôn khổ đồ án Mobile Application Development, nhóm đã thiết kế và triển khai ứng dụng SeaWhisper – một nền tảng chia sẻ cảm xúc ẩn danh hướng tới trải nghiệm nhẹ nhàng và chữa lành cho người dùng trẻ.

Dự án đã hoàn thành các mục tiêu chính ở giai đoạn Minimum Viable Product (MVP), bao gồm xây dựng giao diện mobile thân thiện, triển khai các chức năng cốt lõi như thả chai, nhặt chai và quản lý bottle đã lưu, cũng như phát triển hệ thống backend hỗ trợ xác thực người dùng và xử lý dữ liệu. Các kết quả kiểm thử và khảo sát người dùng cho thấy ứng dụng đáp ứng tương đối tốt kỳ vọng ban đầu về trải nghiệm và ý tưởng sử dụng.

Qua quá trình thực hiện đồ án, nhóm không chỉ áp dụng các kiến thức về phát triển ứng dụng mobile và thiết kế hệ thống backend, mà còn có cơ hội tiếp cận các công cụ hiện đại như Swagger, SonarCloud và GitHub Organization trong quy trình phát triển phần mềm. Những hạn chế còn tồn tại, đặc biệt về kiểm thử tự động và phân tích hành vi người dùng, được xem là các hướng cải tiến tiềm năng trong tương lai.

Tổng thể, SeaWhisper là một sản phẩm khả thi ở mức MVP và có tiềm năng phát triển thêm nếu được đầu tư mở rộng về tính năng, chất lượng mã nguồn và trải nghiệm người dùng. Dự án đã đáp ứng đầy đủ yêu cầu của đồ án và tạo nền tảng cho các nghiên cứu hoặc triển khai tiếp theo trong lĩnh vực ứng dụng chia sẻ cảm xúc và tương tác ẩn danh.

Tài liệu

- [1] Noam Lapidot-Lefler **and** Azy Barak. ?The effects of anonymity on online behavior? **in***Computers in Human Behavior*: 28.2 (2012), **pages** 434–443.
- [2] Michele P. Hamm, Amanda S. Newton **and** Alyssa Chisholm. ?Online harassment and mental health outcomes? **in***JAMA Pediatrics*: 169.8 (2015), **pages** 770–777.
- [3] Sabine Trepte **and** Leonard Reinecke. ?Self-disclosure and well-being in online contexts? **in***Social Media + Society*: 2.2 (2016), **pages** 1–12.
- [4] Eric Ries. *The Lean Startup*. Crown Business, 2011.
- [5] *Understanding the GitHub flow*. 2026. URL: <https://docs.github.com/en/get-started/quickstart/github-flow> (**urlseen** 08/01/2026).
- [6] *Github Documentation*. 2026. URL: <https://docs.github.com/> (**urlseen** 08/01/2026).
- [7] *React Native Documentation*. 2026. URL: <https://reactnative.dev/docs/getting-started> (**urlseen** 08/01/2026).
- [8] *Expo Documentation*. 2026. URL: <https://docs.expo.dev/> (**urlseen** 08/01/2026).
- [9] *FastAPI Documentation*. 2026. URL: <https://fastapi.tiangolo.com/> (**urlseen** 08/01/2026).
- [10] *OpenAPI Specification*. 2026. URL: <https://spec.openapis.org/oas/latest.html> (**urlseen** 08/01/2026).
- [11] *Redis Documentation*. 2026. URL: <https://redis.io/docs/latest/> (**urlseen** 08/01/2026).
- [12] *Swagger UI Documentation*. 2026. URL: <https://swagger.io/tools/swagger-ui/> (**urlseen** 08/01/2026).
- [13] *Postman Documentation*. 2026. URL: <https://learning.postman.com/docs/> (**urlseen** 08/01/2026).