

ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN

ĐỒ ÁN TỐT NGHIỆP

NGÀNH: CÔNG NGHỆ THÔNG TIN
CHUYÊN NGÀNH: CÔNG NGHỆ PHẦN MỀM

ĐỀ TÀI
XÂY DỰNG ỨNG DỤNG TRÊN
ANDROID QUẢN LÝ VÀ XỬ LÝ FILE
PDF TÍCH HỢP AI

Người hướng dẫn: ThS. TRẦN HỒ THỦY TIÊN

Sinh viên thực hiện: NGUYỄN ĐỨC HUỠNH

Số thẻ sinh viên: 102210211

Lớp: 21TCLC_DT3

Đà Nẵng, 06/2025

ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN

ĐỒ ÁN TỐT NGHIỆP

NGÀNH: CÔNG NGHỆ THÔNG TIN
CHUYÊN NGÀNH: CÔNG NGHỆ PHẦN MỀM

ĐỀ TÀI
XÂY DỰNG ỨNG DỤNG TRÊN
ANDROID QUẢN LÝ VÀ XỬ LÝ FILE
PDF TÍCH HỢP AI

Người hướng dẫn: ThS. TRẦN HỒ THỦY TIÊN

Sinh viên thực hiện: NGUYỄN ĐỨC HUỠNH

Số thẻ sinh viên: 102210211

Lớp: 21TCLC_DT3

Đà Nẵng, 06/2025

TÓM TẮT

Tên đề tài: Xây dựng ứng dụng trên Android quản lý và xử lý file PDF tích hợp AI.

Sinh viên thực hiện: Nguyễn Đức Huỳnh

Số thẻ SV: 102210211

Lớp: 21TCLC_DT3

Tóm tắt đề tài: Trong thời đại công nghệ phát triển mạnh mẽ, nhu cầu làm việc và học tập với tài liệu điện tử ngày càng trở nên phổ biến. Đặc biệt, các định dạng tài liệu như PDF, Word, Excel hay PowerPoint thường xuyên được sử dụng trong công việc hành chính, giáo dục và nghiên cứu. Tuy nhiên, việc quản lý và xử lý hiệu quả các loại tệp này trên thiết bị di động vẫn còn gặp nhiều hạn chế, từ việc đọc nội dung, chỉnh sửa, đến trích xuất thông tin hoặc chuyển đổi ngôn ngữ.

Trước thực tế đó, cùng với sự tiến bộ vượt bậc của trí tuệ nhân tạo (AI), đề án “AI PDF” được xây dựng với mục tiêu phát triển một ứng dụng di động thông minh hỗ trợ quản lý và xử lý toàn diện các loại tài liệu số, đặc biệt là file PDF. Ứng dụng cung cấp các chức năng nổi bật như xem và chỉnh sửa nội dung PDF, tóm tắt văn bản, dịch ngôn ngữ tự động, nhận diện ký tự trong hình ảnh (OCR), giúp người dùng tiếp cận và thao tác với tài liệu một cách nhanh chóng, tiện lợi và hiệu quả.

Không dừng lại ở đó, ứng dụng còn hỗ trợ đọc và tương tác với nhiều định dạng tài liệu phổ biến khác như Word, Excel, PowerPoint, Epub, TXT, XML, HTML,... đáp ứng nhu cầu làm việc đa dạng trên thiết bị di động. Đặc biệt, hệ thống còn tích hợp tính năng lưu trữ và chia sẻ tài liệu thông qua nền tảng đám mây riêng của ứng dụng, giúp người dùng dễ dàng truy cập, đồng bộ và chia sẻ tài liệu mọi lúc mọi nơi một cách an toàn và bảo mật.

Sản phẩm không chỉ góp phần nâng cao hiệu quả làm việc và học tập của người dùng mà còn thể hiện tiềm năng ứng dụng mạnh mẽ của AI trong lĩnh vực xử lý văn bản và quản lý thông tin.

NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Họ tên sinh viên: Nguyễn Đức Huỳnh

Số thẻ sinh viên: 102210211

Lớp: 21TCLC_DT3 Khoa: Công nghệ thông tin Ngành: Công nghệ phần mềm

- Tên đề tài đồ án:* Xây dựng ứng dụng trên Android quản lý và xử lý file PDF tích hợp AI
- Đề tài thuộc diện:* Có ký kết thỏa thuận sở hữu trí tuệ đối với kết quả thực hiện
- Các số liệu và dữ liệu ban đầu:* Không có
- Nội dung các phần thuyết minh và tính toán:*

Nội dung của bài thuyết minh gồm:

Mở đầu: Giới thiệu về tổng quan đề tài, mục đích, ý nghĩa và yêu cầu đặt ra của đề tài, phương pháp thực hiện và bố cục của đồ án

Cơ sở lý thuyết: Trình bày những cơ sở lý thuyết được áp dụng trong đề tài

Triển khai và đánh giá kết quả: Trình bày cách triển khai, cài đặt vận hành hệ thống và đánh giá kết quả đạt được

Kết luận: Rút ra kết quả, những bài học đạt được qua quá trình xây dựng hệ thống, những điểm hạn chế và phương hướng phát triển trong tương lai.

- Các bản vẽ, đồ thị (ghi rõ các loại và kích thước bản vẽ):*

- Sơ đồ ca sử dụng
- Sơ đồ tuần tự
- Bảng thiết kế cơ sở dữ liệu

- Họ tên người hướng dẫn:* ThS. Trần Hồ Thủy Tiên

7. Ngày giao nhiệm vụ đồ án: /..... /202...

8. Ngày hoàn thành đồ án: /..... /202...

Đà Nẵng, ngày tháng ... năm 2025

Trưởng Bộ môn CNPM

Người hướng dẫn

LỜI NÓI ĐẦU

Trong kỷ nguyên số hóa, công nghệ thông tin đã và đang giữ vai trò then chốt trong sự phát triển của hầu hết các lĩnh vực trong đời sống xã hội. Việc ứng dụng công nghệ thông tin vào thực tiễn không chỉ góp phần nâng cao chất lượng cuộc sống mà còn mở ra những bước tiến đột phá, đưa con người đến gần hơn với một xã hội hiện đại, văn minh và hiệu quả.

Thực hiện đề án tốt nghiệp trong học kỳ này là cơ hội quý báu để em vận dụng những kiến thức đã học, đồng thời tìm hiểu, nghiên cứu và hiện thực hóa một ý tưởng công nghệ mang tính ứng dụng cao. Đây không chỉ là thử thách để em rèn luyện kỹ năng chuyên môn và tư duy giải quyết vấn đề mà còn là nền tảng quan trọng để em định hướng phát triển bản thân trong hành trình nghề nghiệp sắp tới.

Trong suốt quá trình học tập, nghiên cứu và triển khai đề tài, em đã nhận được sự hỗ trợ và hướng dẫn tận tình từ các thầy cô giảng viên trong khoa. Em đặc biệt trân trọng và biết ơn sự quan tâm, giảng dạy của quý thầy cô – những người đã truyền đạt cho em những kiến thức nền tảng quý giá trong suốt thời gian học tập tại Trường Đại học Bách Khoa – Đại học Đà Nẵng.

Em xin gửi lời cảm ơn chân thành và sâu sắc đến cô **ThS. Trần Hồ Thủy Tiên** – người đã tận tình hướng dẫn, định hướng và hỗ trợ em trong suốt quá trình thực hiện đề tài tốt nghiệp. Những nhận xét, góp ý và kiến thức thầy truyền đạt không chỉ giúp em hoàn thiện đề án một cách tốt nhất mà còn là hành trang vô cùng quý giá để em tự tin bước vào môi trường làm việc thực tế sau khi tốt nghiệp.

Em xin trân trọng cảm ơn!

CAM ĐOAN

Tôi xin cam đoan:

1. Báo cáo đề án tốt nghiệp với tên đề tài “ Xây dựng ứng dụng trên Android quản lý và xử lý file PDF tích hợp AI ”” là công trình nghiên cứu của chính cá nhân tôi dưới sự hướng dẫn trực tiếp của giảng viên ThS. Trần Hồ Thủy Tiên
2. Tôi đã tự đọc nghiên cứu, dịch tài liệu và tổng hợp các kiến thức đã làm nên báo cáo này và đảm bảo không sao chép ở bất cứ đâu.
3. Những lý thuyết trong luận văn đều được sử dụng tài liệu như tôi đã tham khảo ở phần tài liệu tham khảo đã có trong báo cáo.

Nếu có vi phạm, tôi xin chịu hoàn toàn trách nhiệm.

Sinh viên thực hiện

MỤC LỤC

TÓM TẮT

NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

LỜI NÓI ĐẦU	i
CAM ĐOAN	ii
MỤC LỤC.....	iii
DANH SÁCH HÌNH VẼ.....	vii
DANH SÁCH BẢNG	x
CHƯƠNG 1. CƠ SỞ LÝ THUYẾT.....	3
1.1 Kotlin, JepakCompose và Java, XML	3
1.1.1: Kotlin	3
1.1.2: Jetpack Compose	3
1.1.3: Java	4
1.1.4: XML	5
1.2 NodeJS và Express.js.....	5
1.2.1: NodeJS	5
1.2.2: Express.js.	6
1.3 Supabase PostgreSQL.	7
1.4 Room Database.....	8
1.5 Firebase.....	9
1.6 RESTful API	10
1.7 Xác thực và bảo mật người dùng bằng JWT	11
1.8 Dependency Injection (Hilt Dagger)	12
1.9 Network và Cloud ở mobile	13
1.9.1: Retrofit.....	13

1.9.2: OkHttp	14
1.10 Xử lý PDF	14
1.10.1: Thư viện Itext	14
1.10.2: Thư viện MuPDF	15
1.11 Xử lý AI Google ML Kit.....	15
1.12 Quảng cáo và thanh toán	16
1.12.1: Google Admob	16
1.12.2: Google Play Billing	16
1.13 Mô hình và kiến trúc phát triển ứng dụng	17
1.13.1: Mô hình kiến trúc MVVM.....	17
1.13.2: Mô hình kiến trúc Clean Architecture	18
1.14 Bảo mật và tối ưu hóa mã nguồn với ProGuard.....	19
1.15 Kết luận	20
CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG.....	21
2.1 Phân tích yêu cầu.....	21
2.1.1: Yêu cầu chức năng.....	21
2.1.2: Yêu cầu phi chức năng	22
2.2 Thiết kế hệ thống	25
2.2.1. Tổng quan	26
2.2.2. Biểu đồ ca sử dụng.....	28
2.2.3. Sơ đồ tuần tự.....	63
2.2.4. Cơ sở dữ liệu.....	73
2.2.5. Thiết kế kiến trúc hệ thống	75
CHƯƠNG 3. TRIỂN KHAI HỆ THỐNG VÀ ĐÁNH GIÁ	79
3.1 Môi trường và kỹ thuật phát triển.....	79
3.1.1. Thiết bị sử dụng	79
3.1.2. Công cụ và phân mềm phát triển	79

3.1.3. Các thư viện SDK tích hợp.....	80
3.2 Triển khai Mobile app	81
3.2.1. Triển khai tích hợp thư viện native (JNI) trong ứng dụng	81
3.2.2. Triển khai ProGuard trong ứng dụng.....	83
3.2.3. Triển khai quản lý phiên bản và Flavor trong ứng dụng	85
3.2.3. Triển khai quản lý ngôn ngữ trong ứng dụng	87
3.2.4. Triển khai tích hợp Firebase và Analytics trong ứng dụng	90
3.2.5. Triển khai tích hợp Google Admob trong ứng dụng	93
3.2.5. Triển khai tích hợp Google Play Billing trong ứng dụng.....	94
3.3 Triển khai Server NodeJS.....	96
3.4 Triển khai và kết quả của các chức năng chính.....	100
3.3.1. Quản lý tệp.....	100
3.3.2. Xem và chỉnh sửa file PDF.....	101
3.3.3. Xem file Office, TXT, Office	101
3.3.4 .Chức năng Scan to PDF	103
3.3.5 .Chức năng Translate PDF.....	104
3.3.6 .Chức năng Speech PDF	105
3.3.7 .Chức năng Summary PDF	106
3.3.8 .Chức năng HandWriting Recognition	107
3.3.8 .Chức năng Merge, Split PDF.....	108
3.3.9 .Chức năng Lock/Unlock PDF	109
3.3.10 .Chức năng Cloud Upload PDF	110
3.3.11 .Chức năng Share file Upload PDF	111
3.5 Kết luận	112
KẾT LUẬN	113
1. Kết luận	113
2. Hạn chế.....	113

3. Hướng phát triển.....	114
TÀI LIỆU THAM KHẢO	115

DANH SÁCH HÌNH VẼ

Hình 1.1: Kotlin	3
Hình 1.2: Jetpack Compose.....	4
Hình 1.3: Java for Android.....	5
Hình 1.4: NodeJS	6
Hình 1.5: Express.js	7
Hình 1.6: Supabase.....	8
Hình 1.7: Room Database	9
Hình 1.8: Firebase	10
Hình 1.9: Mô Hình RESTful API	11
Hình 1.10: JWT	12
Hình 1.11: Mô tả hoạt động của Retrofit	14
Hình 1.12: Google ML Kit.....	16
Hình 1.13: Mô hình kiến trúc MVVM.....	18
Hình 1.14: Mô hình Clean Architecture.....	19
Hình 2.1: Các công nghệ sử dụng	26
Hình 2.2: Sơ đồ Usecase tổng quan	28
Hình 2.3: Sơ đồ Usecase quản lý File	29
Hình 2.4: Sơ đồ Usecase xem và chỉnh sửa File PDF	34
Hình 2.5: Sơ đồ Usecase xem file Office, EPUB, Txt.....	41
Hình 2.6: Sơ đồ Usecase của các chức năng AI.....	46
Hình 2.7: Sơ đồ Usecase của các chức năng PDF Tools	52
Hình 2.8: Sơ đồ Usecase của các chức năng PDF Conversion Tools.....	56
Hình 2.9: Sơ đồ Usecase của các chức năng Cloud & Sync.....	59
Hình 2.10: Sơ đồ tuần tự quản lý File	63
Hình 2.11: Sơ đồ tuần tự xem file PDF cơ bản.....	64
Hình 2.12: Sơ đồ tuần tự chỉnh sửa file PDF	65

Hình 2.13: Sơ đồ tuần tự xem file EPUB.....	66
Hình 2.14: Sơ đồ tuần tự xem file Office.....	67
Hình 2.15: Sơ đồ tuần tự chức năng Scan to PDF và Handwriting Recognition.	68
Hình 2.16: Sơ đồ tuần tự chức năng Text to Speech, Translate, Summarize	69
Hình 2.17: Sơ đồ tuần tự chức năng Merge và Split PDF	70
Hình 2.18: Sơ đồ tuần tự chức năng Lock/Unlock, print PDF	71
Hình 2.19: Sơ đồ tuần tự chức năng Cloud & Sync.....	72
Hình 2.20: Hình ảnh mô tả cơ sở dữ liệu	73
Hình 2.21: Hình minh họa cho Storage Policies	75
Hình 2.22: Hình cấu trúc file thực tế.....	77
Hình 3.1: Tổ chức thư mục để triển khai tích hợp thư viện native (JNI).....	82
Hình 3.2: Cấu hình Gradle để thực hiện chỉ định ABI	82
Hình 3.3: Tạo file để thực hiện sử dụng thư viện	82
Hình 3.4: Cấu hình Gradle để sử dụng ProGuard	83
Hình 3.5: Tùy chỉnh file ProGuard giữ lại các lớp và thông tin	84
Hình 3.6: Tùy chỉnh file ProGuard loại trừ cảnh báo	84
Hình 3.7: Cấu hình cho phiên bản và Flavor	86
Hình 3.8: Cấu hình cho Signing cho từng Flavor	86
Hình 3.9: Cấu hình ngôn ngữ trong ứng dụng	87
Hình 3.10: Cấu hình để thực hiện tạo tài nguyên ngôn ngữ tự động	88
Hình 3.11: Script để thực hiện dịch ngôn ngữ nhanh bằng Google Translate	88
Hình 3.12: Script để thực hiện dịch ngôn ngữ nhanh bằng Google App Script ..	89
Hình 3.13: Firebase Analytics thực tế được triển khai	90
Hình 3.14: Triển khai Timber	91
Hình 3.15: Firebase Crashlytics thực tế được triển khai.....	92
Hình 3.16: Firebase Remote Config được triển khai	92
Hình 3.17: Firebase Remote Config thực tế được triển khai	93

Hình 3.18: Cấu trúc Admob được triển khai.....	94
Hình 3.19: Cấu trúc Billing được triển khai.....	95
Hình 3.20: Billing được triển khai	95
Hình 3.21: Billing Update được triển khai.....	96
Hình 3.22: Cấu trúc server NodeJS.....	97
Hình 3.23: Tổng quan sử dụng Supabase cho quản lý cơ sở dữ liệu.....	98
Hình 3.24: Sử dụng Postman cho việc kiểm tra API	98
Hình 3.25: Sử dụng Render để deploy Server lên môi trường Product	99
Hình 3.26: Giao diện app quản lý tệp	100
Hình 3.27: Giao diện xem và chỉnh sửa PDF.....	101
Hình 3.28: Giao diện xem và chỉnh sửa PDF.....	102
Hình 3.29: Chức năng Scan to PDF	103
Hình 3.30: Chức năng Translate PDF	104
Hình 3.31: Chức năng Speech PDF	105
Hình 3.32: Chức năng Summary PDF	106
Hình 3.33: Chức năng Handwriting Recognition PDF	107
Hình 3.34: Chức năng Handwriting Recognition PDF	108
Hình 3.35: Chức năng Lock/Unlock PDF	109
Hình 3.36: Chức năng Cloud Upload PDF	110
Hình 3.37: Chức năng Share File.....	111

DANH SÁCH BẢNG

Bảng 2.1: Đặc tả mở File	29
Bảng 2.2: Đặc tả tìm kiếm File	30
Bảng 2.3: Đặc tả sắp xếp, lọc file.....	31
Bảng 2.4: Đặc tả đổi tên/ xóa/ đánh dấu tệp	32
Bảng 2.5: Đặc tả xem tệp recent/ bookmark.....	32
Bảng 2.6: Đặc tả chia sẻ tệp PDF.....	33
Bảng 2.7: Đặc tả mở và xem file PDF	35
Bảng 2.8: Đặc tả chuyển trang khi xem file.....	35
Bảng 2.9: Đặc tả phóng to thu nhỏ khi xem file	36
Bảng 2.10: Đặc tả các chế độ xem khi xem file.....	36
Bảng 2.11: Đặc tả các chế độ xem khi xem file.....	37
Bảng 2.12: Đặc tả các chế độ xem khi xem file.....	38
Bảng 2.13: Đặc tả các chế độ chỉnh sửa và copy ký tự ở file PDF	38
Bảng 2.14: Đặc tả các chế độ thêm ký tự/ chữ ký nổi ở file PDF.....	39
Bảng 2.15: Đặc tả chế độ in file.....	40
Bảng 2.16: Đặc tả chế độ xem file Word.....	42
Bảng 2.17: Đặc tả chế độ xem file Excel.....	42
Bảng 2.18: Đặc tả chế độ xem file PowerPoint	43
Bảng 2.19: Đặc tả chế độ xem file Text, HTML, XML	44
Bảng 2.20: Đặc tả chế độ xem file EPUB.....	44
Bảng 2.21: Đặc tả chức năng AI Scan to PDF	47
Bảng 2.22: Đặc tả chức năng AI Speech PDF	48
Bảng 2.23: Đặc tả chức năng AI Translate PDF.....	49
Bảng 2.24: Đặc tả chức năng AI Summarize PDF.....	50

Bảng 2.25: Đặc tả chức năng AI PDF HandWriting recognition	51
Bảng 2.26 Đặc tả chức năng Merge PDFs	53
Bảng 2.27: Đặc tả chức năng Split PDF	53
Bảng 2.28: Đặc tả chức năng Lock/Unlock PDF	54
Bảng 2.29: Đặc tả chức năng Image to Text.....	57
Bảng 2.30: Đặc tả chức năng Word to PDF	57
Bảng 2.31 Đặc tả chức năng Browse Cloud Files.....	59
Bảng 2.32 Đặc tả chức năng Upload Files	60
Bảng 2.33 Đặc tả chức năng Download Files	61
Bảng 2.34 Đặc tả chức năng Share File	62

DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT

Chữ viết tắt	Tên đầy đủ
API	Application Programming Interface
ML Kit	Machine Learning Kit
REST	Representational State Transfer
APK	Android Package Kit
AAB	Android App Bundle
OCR	Optical Character Recognition
SDK	Software Development Kit
JNI	Java Native Interface
AI	Artificial Intelligence

MỞ ĐẦU

1. Mục đích thực hiện đề tài

Trong thời đại công nghệ phát triển nhanh chóng, việc quản lý và xử lý tài liệu điện tử trở thành một phần thiết yếu trong công việc và học tập. Đặc biệt, định dạng PDF đang được sử dụng rộng rãi nhờ tính ổn định và bảo mật cao. Tuy nhiên, các thao tác như chỉnh sửa, tóm tắt, dịch thuật hay trích xuất thông tin từ tài liệu PDF trên thiết bị di động vẫn còn nhiều hạn chế. Đề tài này được thực hiện nhằm xây dựng một ứng dụng di động tích hợp trí tuệ nhân tạo (AI), hỗ trợ người dùng tương tác hiệu quả với tài liệu PDF và nhiều định dạng khác, từ đó nâng cao hiệu suất làm việc và học tập.

2. Mục tiêu đề tài

- Phát triển ứng dụng di động có khả năng quản lý, đọc và chỉnh sửa tài liệu PDF.
- Tính năng nâng cao tích hợp cho phép người dùng thao tác, và sử dụng ứng dụng để giải quyết các nhu cầu thực tế khi xử lý tài liệu, như: Merge nhiều file PDF lại làm một File PDF, chia 1 file PDF thành 2 file PDF, đặt mật khẩu bảo vệ cho file PDF, Image to Text, ...
- Tích hợp các tính năng AI như tóm tắt nội dung, dịch ngôn ngữ và nhận diện ký tự quang học (OCR).
- Hỗ trợ người dùng đọc và quản lý nhiều định dạng tài liệu khác như Word, Excel, PowerPoint, Epub, TXT, XML, HTML,...
- Cung cấp chức năng tải lên, lưu trữ và chia sẻ tài liệu qua nền tảng đám mây tích hợp trong ứng dụng.
- Giao diện thân thiện, dễ sử dụng và tương thích với nhiều thiết bị Android.

3. Phạm vi và đối tượng nghiên cứu

- Phạm vi nghiên cứu: Tập trung vào nền tảng Android, xử lý tài liệu dạng PDF và các định dạng văn bản phổ biến khác. Đề tài chủ yếu áp dụng các nhu cầu cần thiết và thực tế khi sử dụng file PDF, ứng dụng kết hợp các tính năng nâng cao và các giải pháp AI hiện có để triển khai các chức năng như tóm tắt văn bản, dịch ngôn ngữ, OCR.
- Đối tượng nghiên cứu: Người dùng phổ thông có nhu cầu đọc, chỉnh sửa, quản lý tài liệu số trên thiết bị di động – bao gồm sinh viên, nhân viên văn phòng, giáo viên, nhà nghiên cứu,...

4. Phương pháp nghiên cứu

- Phân tích tài liệu: Nghiên cứu các yêu cầu thực tế trong việc xử lý tài liệu số và khảo sát các ứng dụng tương tự trên thị trường.
- Phân tích – thiết kế hệ thống: Sử dụng phương pháp hướng đối tượng để phân tích yêu cầu và thiết kế kiến trúc ứng dụng.
- Thử nghiệm và đánh giá: Triển khai ứng dụng mẫu và tiến hành kiểm thử để đánh giá hiệu quả sử dụng.

5. Cấu trúc đề án tốt nghiệp

Nội dung đề án được trình bày thành 3 chương như sau:

- Chương 1: Cơ sở lý thuyết và công nghệ áp dụng: Tổng hợp các kiến thức liên quan đến xử lý tài liệu, trí tuệ nhân tạo, OCR, và các công nghệ nền tảng cho ứng dụng
- Chương 2: Phân tích và thiết kế hệ thống: Phân tích yêu cầu, chức năng chính của hệ thống, các loại sơ đồ như sơ đồ nguyên lý hoạt động, sơ đồ ca sử dụng, sơ đồ tuần tự, bên cạnh đó còn là phân tích thiết kế mô hình quan hệ của cơ sở dữ liệu.
- Chương 3: Triển khai, xây dựng và đánh giá ứng dụng
- Định hướng phát triển trong tương lai.

CHƯƠNG 1. CƠ SỞ LÝ THUYẾT

1.1 Kotlin, JepackCompose và Java, XML

1.1.1: Kotlin

Kotlin là một ngôn ngữ lập trình hiện đại, được phát triển bởi JetBrains và chính thức được Google công nhận là ngôn ngữ chính thức cho phát triển ứng dụng Android từ năm 2017. Kotlin được thiết kế nhằm khắc phục những hạn chế của Java, mang lại cú pháp ngắn gọn, rõ ràng, đồng thời tăng tính an toàn và hiệu suất cho lập trình viên.

Một số ưu điểm nổi bật của Kotlin có thể kể đến như:

- Tính tương thích cao với Java, cho phép sử dụng đồng thời cả hai ngôn ngữ trong cùng một dự án.
- Cú pháp ngắn gọn và dễ đọc, giúp giảm thiểu lỗi lập trình và tăng năng suất phát triển.
- Tính năng null-safety, giúp hạn chế các lỗi NullPointerException – một trong những lỗi phổ biến trong lập trình Android.
- Hỗ trợ lập trình hàm (functional programming), giúp biểu đạt logic một cách linh hoạt và mạnh mẽ hơn.



Hình 1.1: Kotlin

1.1.2: Jetpack Compose

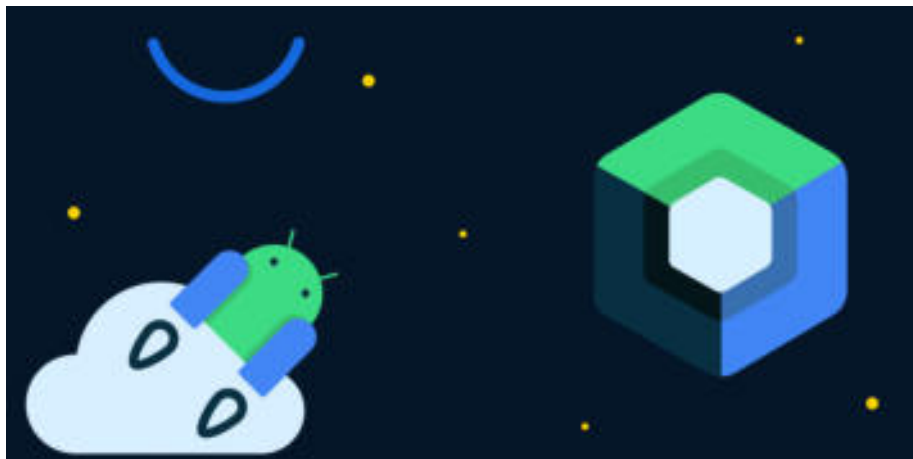
Jetpack Compose là một toolkit mới do Google phát triển dành cho việc xây dựng giao diện người dùng (UI) trên Android, ra mắt chính thức vào năm 2020. Đây là

công nghệ thay thế cho phương pháp xây dựng UI truyền thống sử dụng XML, mang lại cách tiếp cận hoàn toàn mới thông qua lập trình khai báo (declarative UI). Jetpack Compose cho phép lập trình viên mô tả UI bằng cách sử dụng các hàm Kotlin, giúp giao diện trở nên dễ viết, dễ bảo trì và phản ứng linh hoạt với các thay đổi trạng thái. Điều này giúp rút ngắn thời gian phát triển ứng dụng và giảm độ phức tạp trong quản lý giao diện người dùng.

Một số lợi ích nổi bật của Jetpack Compose bao gồm:

- Tính mô-đun và tái sử dụng cao, nhờ khả năng tạo các Composable function riêng biệt.
- Tích hợp chặt chẽ với kiến trúc MVVM và LiveData/StateFlow, giúp quản lý trạng thái hiệu quả.
- Hỗ trợ trực quan hóa giao diện trong thời gian thực (Preview), tăng cường trải nghiệm lập trình.
- Tương thích tốt với các thư viện Android hiện tại trong hệ sinh thái Jetpack.

Với sự kết hợp giữa Kotlin và Jetpack Compose, lập trình viên có thể xây dựng ứng dụng Android hiện đại, mượt mà và dễ bảo trì hơn so với các phương pháp truyền thống. Trong phạm vi đề án này, việc ứng dụng Kotlin cùng Jetpack Compose là một lựa chọn phù hợp nhằm đảm bảo hiệu quả phát triển và tính mở rộng của sản phẩm trong tương lai.



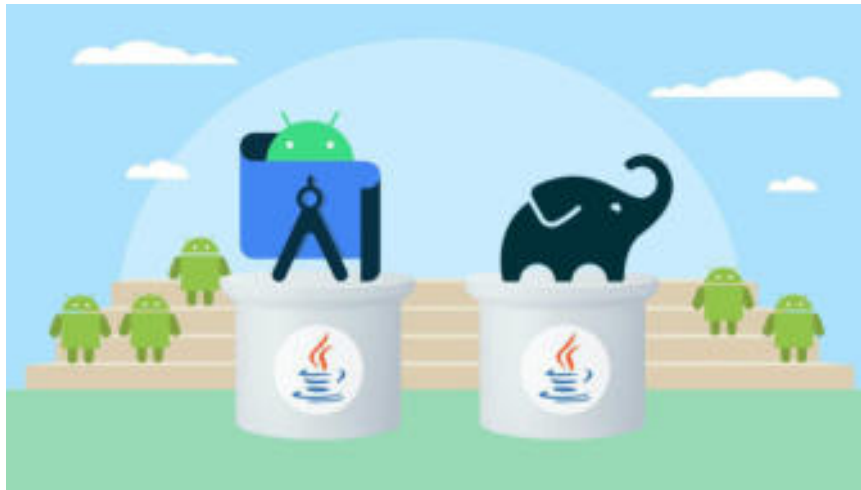
Hình 1.1: Jetpack Compose

1.1.3: Java

Java là một trong những ngôn ngữ lập trình nền tảng được Google lựa chọn làm ngôn ngữ chính cho Android từ những phiên bản đầu tiên. Với đặc điểm là một ngôn ngữ hướng đối tượng mạnh mẽ, cú pháp rõ ràng, và hỗ trợ đa nền tảng thông

qua máy ảo Java (JVM), Java mang đến sự ổn định, dễ bảo trì và mở rộng khi xây dựng các ứng dụng quy mô lớn.

Trong phát triển Android, Java được dùng để xử lý logic nghiệp vụ, tương tác với hệ điều hành, xử lý sự kiện, quản lý vòng đời của Activity/Fragment, kết nối cơ sở dữ liệu, xử lý API và nhiều thao tác khác. Nhờ vào hệ sinh thái phong phú và cộng đồng lớn mạnh, Java vẫn giữ vai trò chủ đạo trong nhiều dự án Android hiện nay, dù đã dần nhường chỗ cho Kotlin ở các dự án mới.



Hình 1.2: Java for Android

1.1.4: XML

XML (eXtensible Markup Language) là ngôn ngữ đánh dấu mở rộng, được Android sử dụng để mô tả cấu trúc và bố cục giao diện người dùng. Các thành phần UI như TextView, Button, ImageView, LinearLayout, v.v. được khai báo trực tiếp trong các tệp XML, giúp tách biệt rõ ràng phần hiển thị và phần xử lý logic (Java/Kotlin).

Việc sử dụng XML trong Android mang lại một số lợi ích như:

- Tổ chức giao diện trực quan: Cấu trúc dạng cây của XML giúp dễ dàng hình dung và thiết kế UI.
- Tách biệt rõ ràng giữa giao diện và logic: Giúp việc quản lý, chỉnh sửa và bảo trì mã trở nên hiệu quả hơn.
- Tương thích với công cụ thiết kế (Layout Editor) của Android Studio, giúp lập trình viên có thể kéo thả, xem trước giao diện theo thời gian thực.

1.2 NodeJS và Express.js

1.2.1: NodeJS

Node.js là một nền tảng phía máy chủ (server-side platform) được xây dựng trên

nền tảng V8 JavaScript Engine của Google. Node.js cho phép phát triển các ứng dụng mạng có khả năng mở rộng cao bằng cách sử dụng JavaScript – một ngôn ngữ vốn trước đây chỉ hoạt động ở phía trình duyệt (client-side). Với khả năng xử lý bất đồng bộ (asynchronous) và mô hình hướng sự kiện (event-driven), Node.js đã trở thành một trong những nền tảng phổ biến nhất để phát triển ứng dụng web, API và hệ thống thời gian thực.

Một số đặc điểm nổi bật của Node.js gồm:

- Hiệu suất cao: Node.js sử dụng mô hình non-blocking I/O, cho phép xử lý hàng ngàn kết nối đồng thời mà không cần tạo luồng mới cho mỗi yêu cầu, giúp tối ưu tài nguyên và cải thiện tốc độ xử lý.
- Thống nhất ngôn ngữ lập trình: Với Node.js, lập trình viên có thể sử dụng JavaScript ở cả phía máy khách (client) và phía máy chủ (server), giúp tăng tính nhất quán và dễ dàng trong quản lý mã nguồn.
- Hệ sinh thái phong phú: Node.js được hỗ trợ bởi trình quản lý gói npm (Node Package Manager) với hàng trăm nghìn thư viện mã nguồn mở, giúp đẩy nhanh quá trình phát triển phần mềm.
- Khả năng mở rộng tốt: Node.js phù hợp với các hệ thống yêu cầu xử lý khối lượng lớn dữ liệu theo thời gian thực như ứng dụng chat, theo dõi vị trí, xử lý file lớn, và đặc biệt là API backend cho ứng dụng di động.



Hình 1.4: NodeJS

1.2.2: Express.js.

Express.js là một framework nhẹ và linh hoạt dành cho Node.js, được thiết kế để

xây dựng các ứng dụng web và dịch vụ API một cách nhanh chóng và hiệu quả. Là một phần cốt lõi trong bộ MEN stack (MongoDB, Express.js, Node.js), Express giúp đơn giản hóa quá trình phát triển server bằng cách cung cấp tập hợp các công cụ và chức năng mạnh mẽ cho việc xử lý yêu cầu HTTP, định tuyến (routing), quản lý middleware, và tích hợp với cơ sở dữ liệu.

Một số đặc điểm nổi bật của Express.js bao gồm:

- Cấu trúc đơn giản, dễ học: Express cung cấp một giao diện rõ ràng, trực quan giúp lập trình viên có thể khởi tạo máy chủ và xây dựng API một cách nhanh chóng.
- Quản lý routing hiệu quả: Cho phép định nghĩa các endpoint RESTful một cách rõ ràng, dễ mở rộng và dễ kiểm soát logic xử lý.
- Middleware mạnh mẽ: Hệ thống middleware của Express hỗ trợ tách biệt logic xử lý từng bước như xác thực, ghi log, xử lý lỗi, giúp mã nguồn dễ bảo trì và mở rộng.
- Tích hợp dễ dàng với cơ sở dữ liệu và thư viện bên ngoài: Express hoạt động mượt mà với các hệ quản trị cơ sở dữ liệu như MongoDB, PostgreSQL, MySQL và nhiều thư viện khác của hệ sinh thái npm.



Hình 1.5: Express.js

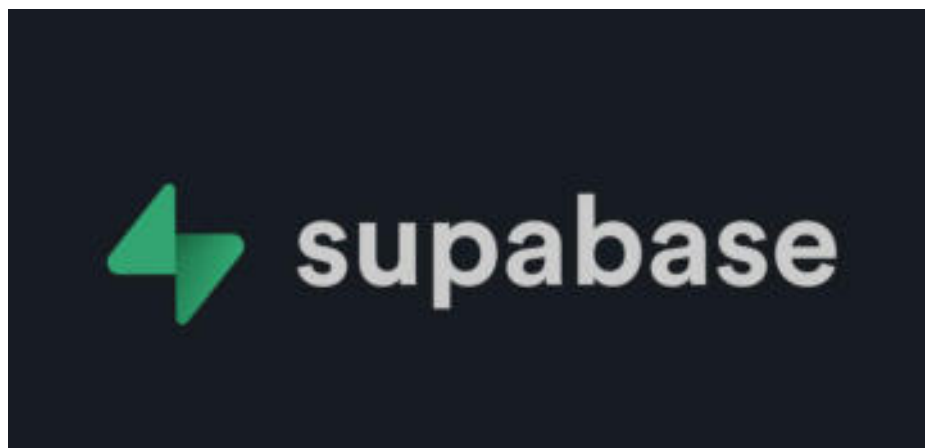
1.3 Supabase PostgreSQL.

Supabase là một nền tảng Backend-as-a-Service (BaaS) mã nguồn mở, được xây dựng dựa trên PostgreSQL nhằm hỗ trợ phát triển ứng dụng hiện đại một cách nhanh chóng, hiệu quả và bảo mật. Với khẩu hiệu “An open source Firebase alternative”, Supabase cung cấp bộ công cụ hoàn chỉnh cho backend như: cơ sở dữ liệu thời gian thực, xác thực người dùng, lưu trữ tệp, RESTful và GraphQL API tự động sinh ra từ cấu trúc dữ liệu.

Trong Supabase, PostgreSQL là thành phần cốt lõi của hệ thống lưu trữ. Nhờ vào sự kết hợp giữa sức mạnh của PostgreSQL và những tính năng mở rộng do Supabase cung cấp, nền tảng này cho phép các lập trình viên thao tác cơ sở dữ liệu thông qua giao diện trực quan, bảng điều khiển mạnh mẽ, hoặc thông qua truy vấn trực tiếp bằng SQL, mà vẫn duy trì được tính linh hoạt và bảo mật cao.

Một số đặc điểm nổi bật của Supabase PostgreSQL bao gồm:

- **Realtime Database:** Supabase cung cấp cơ chế theo dõi thay đổi dữ liệu theo thời gian thực dựa trên tính năng PostgreSQL Realtime, hỗ trợ xây dựng các ứng dụng tương tác trực tiếp mà không cần cấu hình phức tạp.
- **Tự động sinh API:** Mỗi bảng trong PostgreSQL sẽ tự động được ánh xạ thành một endpoint RESTful và GraphQL thông qua dịch vụ PostgREST, giúp tiết kiệm thời gian phát triển backend.
- **Xác thực và phân quyền:** Supabase tích hợp hệ thống xác thực người dùng, đồng thời cho phép quản lý phân quyền truy cập đến từng bảng hoặc hàng trong cơ sở dữ liệu thông qua chính sách bảo mật Row-Level Security (RLS).
- **Khả năng mở rộng và tích hợp:** Hỗ trợ lưu trữ file, sao lưu định kỳ, tích hợp dễ dàng với frontend và mobile app thông qua SDK chính thức.



Hình 1.6: Supabase

1.4 Room Database

RoomDB là một thư viện thuộc Android Jetpack, đóng vai trò là lớp trừu tượng (abstraction layer) giúp thao tác với SQLite trở nên đơn giản, an toàn và hiệu quả

hơn trong ứng dụng Android. Thay vì phải viết trực tiếp các câu lệnh SQL, Room cho phép lập trình viên khai báo các entity, DAO (Data Access Object) và sử dụng annotation để thao tác dữ liệu theo cách hướng đối tượng.

Một số ưu điểm nổi bật của RoomDB bao gồm:

- Tích hợp chặt chẽ với kiến trúc Android hiện đại: Room hỗ trợ LiveData, Flow, ViewModel, và coroutines, giúp xử lý dữ liệu hiệu quả và phản ứng với thay đổi trạng thái UI.
- Kiểm tra câu lệnh SQL tại thời điểm biên dịch: Giúp phát hiện lỗi sớm, đảm bảo tính đúng đắn của truy vấn.
- Cấu trúc rõ ràng, dễ bảo trì: Việc chia tách các thành phần Entity, DAO và Database giúp mã nguồn trở nên rõ ràng, dễ hiểu và dễ kiểm soát.
- Hỗ trợ migration dữ liệu: Cho phép nâng cấp cấu trúc cơ sở dữ liệu một cách an toàn và linh hoạt.



Hình 1.7: Room Database

1.5 Firebase

Firebase là một nền tảng phát triển ứng dụng di động và web được Google cung cấp, với mục tiêu hỗ trợ các lập trình viên xây dựng ứng dụng một cách nhanh chóng, hiệu quả và có thể mở rộng. Firebase không chỉ cung cấp các dịch vụ backend mạnh mẽ như lưu trữ thời gian thực, xác thực người dùng, và lưu trữ file, mà còn cung cấp công cụ phân tích, theo dõi lỗi và tối ưu hóa hiệu suất người dùng – tất cả trong một nền tảng tích hợp.

Firestore hoạt động dựa trên kiến trúc Backend-as-a-Service (BaaS), cho phép lập trình viên tập trung nhiều hơn vào logic nghiệp vụ và giao diện người dùng thay vì phải triển khai và duy trì hệ thống máy chủ phức tạp.

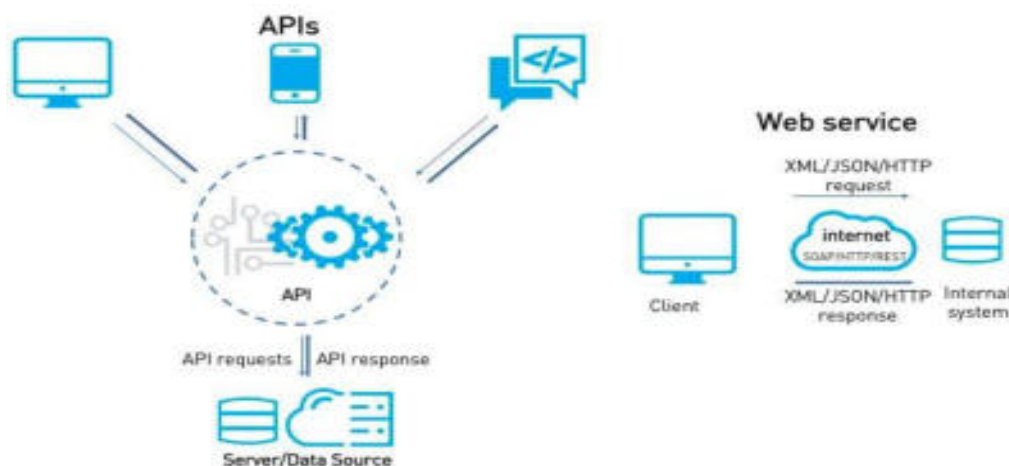


Hình 1.8: Firebase

1.6 RESTful API

REST là từ viết tắt của REpresentational State Transfer và là một kiến trúc cho các hệ thống hypermedia phân tán.

RESTful API là một tiêu chuẩn được sử dụng trong việc thiết kế API cho các ứng dụng web (Web thiết kế dịch vụ) để tạo thuận lợi cho việc quản lý tài nguyên. Nó tập trung vào tài nguyên hệ thống (văn bản tệp, hình ảnh, âm thanh, video hoặc dữ liệu động...), bao gồm các trạng thái tài nguyên được định dạng và truyền qua HTTP. REST định nghĩa một tập hợp các ràng buộc về cách kiến trúc của quy mô Internet hệ thống siêu phương tiện phân tán, chẳng hạn như Web, sẽ hoạt động. Kiến trúc REST nhấn mạnh khả năng mở rộng của các tương tác giữa các thành phần, giao diện thống nhất, triển khai độc lập các thành phần và tạo ra một kiến trúc phân lớp để tạo điều kiện thuận lợi cho các thành phần bộ nhớ đệm để giảm độ trễ do người dùng cảm nhận, thực thi bảo mật và đóng gói các hệ thống kế thừa.



Hình 1.9: Mô Hình RESTful API

1.7 Xác thực và bảo mật người dùng bằng JWT

Trong các hệ thống hiện đại, đặc biệt là các ứng dụng có kiến trúc phân tán như mô hình client-server hoặc microservices, việc bảo mật thông tin người dùng và quản lý phiên làm việc (session) đóng vai trò then chốt trong việc đảm bảo an toàn và tính riêng tư của dữ liệu. Một trong những phương pháp xác thực hiệu quả, phổ biến và dễ tích hợp hiện nay chính là JWT – JSON Web Token.

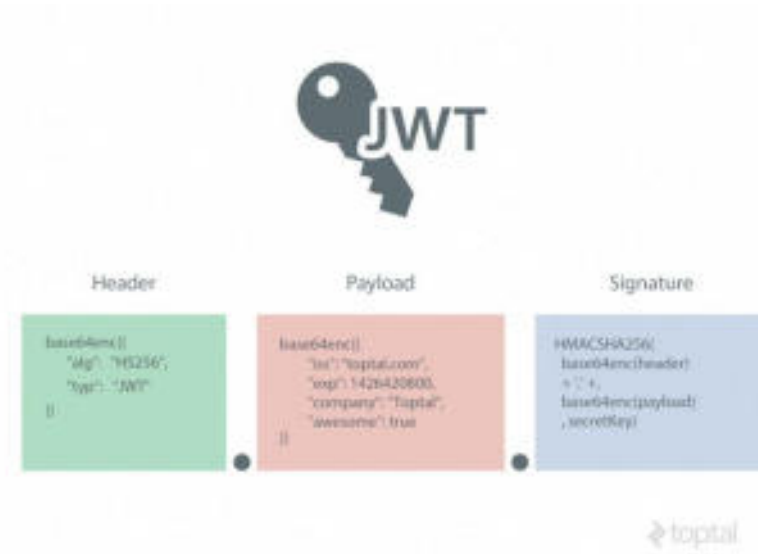
JWT là một chuẩn mở (RFC 7519) định dạng một chuỗi mã hóa nhỏ gọn, dùng để truyền tải thông tin giữa các bên một cách an toàn dưới dạng đối tượng JSON. Một token JWT gồm ba phần chính: Header, Payload và Signature. Trong đó:

- Header chứa thông tin về thuật toán mã hóa được sử dụng (ví dụ: HS256).
- Payload mang dữ liệu (claims) như user_id, thời gian hết hạn (exp), và các thông tin tùy chỉnh khác.
- Signature được tạo bằng cách mã hóa Header và Payload với một khóa bí mật, đảm bảo rằng token không bị chỉnh sửa trong quá trình truyền tải.

Ưu điểm nổi bật của JWT:

- Tính bảo mật cao: Dữ liệu trong token được ký bằng khóa bí mật hoặc khóa bất đối xứng (RSA), đảm bảo token không bị giả mạo.
- Gọn nhẹ và độc lập: Vì token được mã hóa dưới dạng chuỗi, không cần lưu session ở phía máy chủ, giảm tải đáng kể cho backend.

- Dễ tích hợp đa nền tảng: JWT có thể sử dụng linh hoạt giữa web, mobile và các API nội bộ.



Hình 1.10: JWT

1.8 Dependency Injection (Hilt Dagger)

Hilt là một thư viện hỗ trợ Dependency Injection (DI) chính thức do Google phát triển dành riêng cho nền tảng Android. Hilt được xây dựng trên nền tảng Dagger, một thư viện DI mạnh mẽ và phổ biến, với mục tiêu đơn giản hóa quá trình thiết lập và quản lý phụ thuộc trong các ứng dụng Android hiện đại.

- Khái niệm Dependency Injection (DI)

Dependency Injection là một kỹ thuật lập trình trong đó các đối tượng không tự khởi tạo các phụ thuộc mà nhận chúng từ bên ngoài. Điều này giúp tách biệt giữa các thành phần trong hệ thống, từ đó tăng khả năng tái sử dụng mã nguồn, kiểm thử dễ dàng hơn, và duy trì tính mở rộng linh hoạt trong kiến trúc phần mềm.

- Vai trò của Hilt trong Android

Trước khi Hilt ra đời, Dagger được sử dụng rộng rãi nhưng đòi hỏi nhiều cấu hình phức tạp, đặc biệt đối với vòng đời đặc thù của Android (Activity, Fragment, ViewModel, Service, v.v.). Hilt được thiết kế để giải quyết vấn đề đó bằng cách cung cấp:

Cấu trúc DI tích hợp chặt chẽ với vòng đời Android.

Tự động tạo mã (code generation) thông qua annotation, giảm thiểu boilerplate code.

Khả năng tích hợp trực tiếp với ViewModel, WorkManager, Navigation,... giúp dễ dàng chia sẻ phụ thuộc giữa các thành phần.

Hỗ trợ kiểm thử đơn vị và kiểm thử giao diện hiệu quả hơn thông qua injection phụ thuộc có thể cấu hình.

1.9 Network và Cloud ở mobile

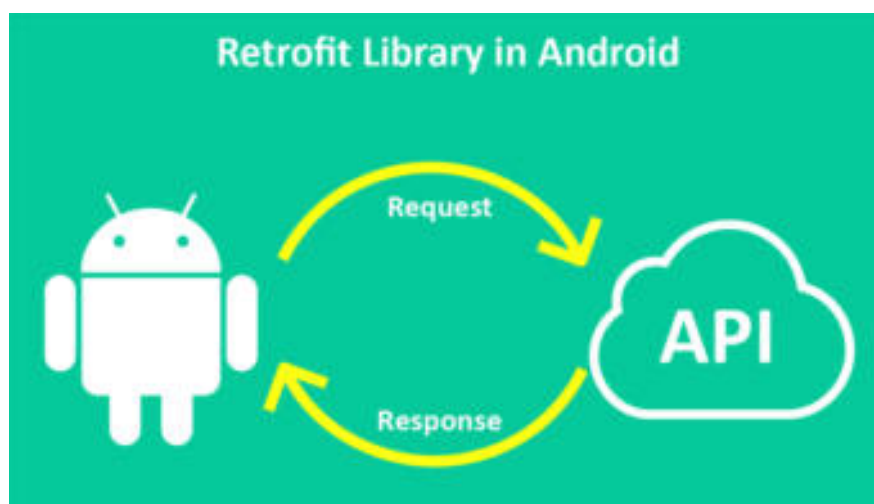
1.9.1: Retrofit

Retrofit là một thư viện HTTP client mạnh mẽ do Square phát triển, được sử dụng phổ biến trong các ứng dụng Android để thực hiện các tác vụ mạng như gửi yêu cầu (request) và nhận phản hồi (response) từ máy chủ thông qua các API dạng RESTful.

Điểm nổi bật của Retrofit là khả năng trừu tượng hóa quá trình gọi API: thay vì phải xử lý trực tiếp các yêu cầu mạng phức tạp, Retrofit cho phép nhà phát triển định nghĩa các endpoint dưới dạng interface Java/Kotlin, kết hợp với các annotation như @GET, @POST, @Body, @Query,... giúp mã nguồn trở nên đơn giản, dễ đọc và dễ bảo trì.

Ngoài ra, Retrofit hỗ trợ:

- Chuyển đổi dữ liệu tự động giữa JSON và các đối tượng Kotlin/Java thông qua thư viện chuyển đổi như Gson, Moshi.
- Tích hợp dễ dàng với Coroutine, RxJava, LiveData, giúp xử lý bất đồng bộ và luồng dữ liệu hiệu quả.
- Tùy biến linh hoạt thông qua interceptor, converter, hoặc client riêng.



Hình 1.3: Mô tả hoạt động của Retrofit

1.9.2: *OkHttp*

OkHttp là một thư viện HTTP client cấp thấp, hiệu năng cao, do Square phát triển, cũng là thành phần cốt lõi bên dưới Retrofit. Trong khi Retrofit cung cấp giao diện cấp cao cho việc định nghĩa và xử lý API, thì OkHttp chịu trách nhiệm thực thi trực tiếp các kết nối HTTP, quản lý kết nối socket, xử lý cache, nén dữ liệu, theo dõi tiến trình gửi/nhận dữ liệu và nhiều tác vụ mạng khác.

Một số đặc điểm nổi bật của OkHttp:

- Hỗ trợ HTTP/2, SPDY, và WebSocket.
- Tự động quản lý connection pooling và retry khi gặp lỗi.
- Hỗ trợ Interceptor cho phép ghi log, thêm header, kiểm soát request / response.
- Có thể hoạt động độc lập hoặc tích hợp với Retrofit làm tầng nền.

1.10 Xử lý PDF

1.10.1: *Thư viện Itext*

iText là một thư viện mã nguồn mở mạnh mẽ và phổ biến, được sử dụng rộng rãi trong các ứng dụng Java và Android để tạo mới, chỉnh sửa, phân tích và chuyển đổi các tài liệu PDF. Với khả năng làm việc trực tiếp trên cấu trúc PDF ở cấp độ thấp, iText giúp các nhà phát triển linh hoạt thao tác với các nội dung trong tài liệu như văn bản, hình ảnh, biểu mẫu, chữ ký số, chú thích, siêu dữ liệu,...

Đặc điểm nổi bật của iText:

- Tạo file PDF động từ dữ liệu có sẵn.
- Chỉnh sửa nội dung và cấu trúc của PDF: thêm/xóa/sửa đoạn văn bản, hình ảnh, watermark, v.v.
- Xử lý biểu mẫu (PDF forms) và chữ ký số (digital signature).
- Tách, gộp và sắp xếp lại các trang PDF một cách dễ dàng.
- Phân tích (parse) nội dung và trích xuất dữ liệu từ các file PDF hiện có.

1.10.2: Thư viện MuPDF

MuPDF là một thư viện mã nguồn mở nhỏ gọn và hiệu năng cao, được phát triển bởi Artifex Software, chuyên dùng để hiển thị và xử lý các định dạng tài liệu như PDF, XPS, EPUB, CBZ, và SVG. Thư viện này nổi bật với khả năng hiển thị chính xác bố cục, định dạng, hình ảnh và font chữ, đảm bảo người dùng có thể xem được các tài liệu phức tạp với chất lượng tương tự như trên máy tính.

Đặc điểm nổi bật của MuPDF:

- Hiển thị tài liệu PDF nhanh và nhẹ, tối ưu cho thiết bị di động.
- Hỗ trợ đầy đủ các tiêu chuẩn PDF như hình ảnh, vector, văn bản, font nhúng,...
- Có khả năng tương tác với tài liệu: tìm kiếm, chọn văn bản, điền form, chú thích,...
- Hỗ trợ chuyển đổi tài liệu sang ảnh bitmap hoặc các định dạng khác, phục vụ nhu cầu trích xuất nội dung.
- Có thể tích hợp vào ứng dụng Android thông qua API C hoặc bản build sẵn (JNI).

1.11 Xử lý AI Google ML Kit

Google ML Kit là một bộ công cụ cung cấp các API dựa trên trí tuệ nhân tạo (AI) và học máy (Machine Learning) do Google phát triển, dành riêng cho nền tảng di động Android và iOS. Thư viện này giúp các nhà phát triển dễ dàng tích hợp các tính năng AI vào ứng dụng mà không cần kiến thức chuyên sâu về học máy hoặc xây dựng mô hình từ đầu.

ML Kit cung cấp cả hai hình thức triển khai: on-device (trên thiết bị) và cloud-based (trên đám mây). Việc hỗ trợ on-device mang lại lợi thế về hiệu suất, khả năng hoạt động offline và tăng cường bảo mật dữ liệu người dùng.



Hình 1.4: Google ML Kit

1.12 Quảng cáo và thanh toán

1.12.1: *Google Admob*

Google AdMob là một nền tảng quảng cáo di động do Google phát triển, cho phép các nhà phát triển ứng dụng tạo doanh thu từ việc hiển thị quảng cáo trong ứng dụng của mình. AdMob cung cấp đa dạng hình thức quảng cáo như banner, quảng cáo toàn màn hình (interstitial), quảng cáo có thưởng (rewarded ads) và quảng cáo gốc (native ads), giúp người phát triển dễ dàng tích hợp các định dạng phù hợp với trải nghiệm người dùng.

Các đặc điểm nổi bật:

- Dễ dàng tích hợp vào ứng dụng Android/iOS.
- Tối ưu hóa doanh thu dựa trên hệ thống đấu giá thông minh (real-time bidding).
- Hỗ trợ đa dạng định dạng quảng cáo, linh hoạt với từng loại ứng dụng.
- Theo dõi và phân tích hiệu quả quảng cáo thông qua hệ thống báo cáo chi tiết.

1.12.2: *Google Play Billing*

Google Play Billing là dịch vụ chính thức của Google dành cho các nhà phát triển ứng dụng Android muốn cung cấp các sản phẩm kỹ thuật số có trả phí, bao gồm mua hàng một lần (in-app products), đăng ký thuê bao (subscriptions), hoặc mở khóa tính năng.

Các đặc điểm nổi bật:

- Tích hợp liền mạch với Google Play Store, giúp người dùng thanh toán dễ dàng và bảo mật.
- Hỗ trợ đa dạng loại hình thanh toán: thanh toán một lần, đăng ký định kỳ, dùng thử miễn phí,...
- Hệ thống bảo mật cao, bảo vệ dữ liệu người dùng và quyền truy cập nội dung.
- Quản lý đơn hàng, hoàn tiền và báo cáo doanh thu thông qua Google Play Console.

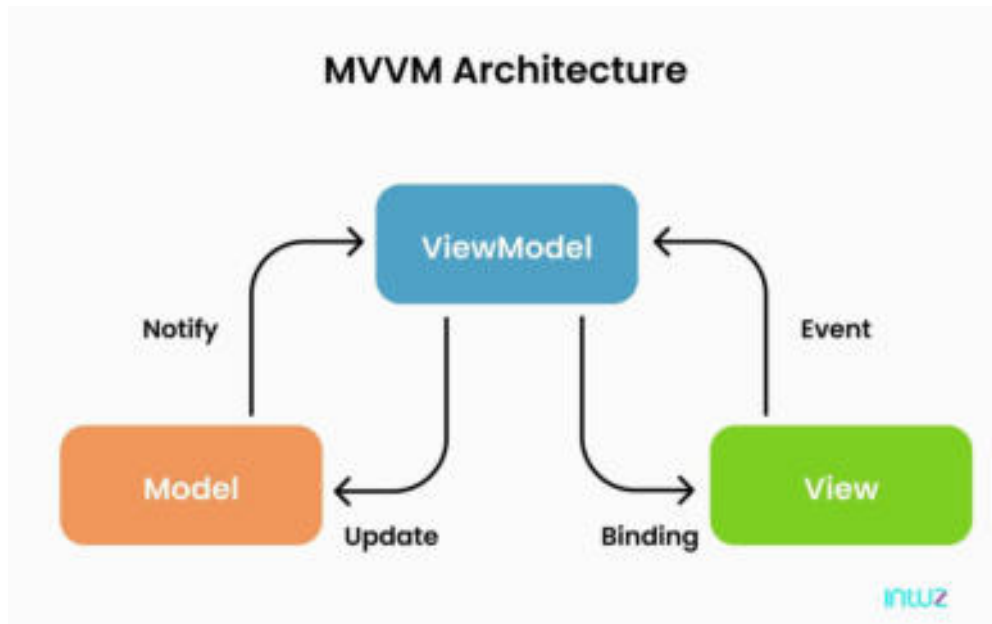
1.13 Mô hình và kiến trúc phát triển ứng dụng

1.13.1: Mô hình kiến trúc MVVM

MVVM là một mô hình kiến trúc phổ biến trong phát triển ứng dụng Android, đặc biệt khi kết hợp với Jetpack Compose hoặc XML truyền thống. MVVM giúp tách rời phần giao diện (UI) khỏi logic xử lý và dữ liệu, qua đó giảm sự phụ thuộc giữa các thành phần, tăng tính kiểm thử và tái sử dụng.

- Model: Chứa dữ liệu và logic xử lý nghiệp vụ.
- View: Giao diện người dùng (UI), chỉ đảm nhận việc hiển thị dữ liệu và nhận tương tác.
- ViewModel: Cầu nối trung gian giữa View và Model. ViewModel lưu trữ dữ liệu UI và xử lý logic trung gian, sử dụng các thành phần như LiveData, StateFlow, hoặc MutableState trong Compose để cập nhật giao diện theo thời gian thực.

Việc sử dụng MVVM giúp đảm bảo dữ liệu phản ứng theo thời gian thực, giữ cho UI đơn giản, dễ kiểm soát và ít lỗi.



Hình 1.5: Mô hình kiến trúc MVVM

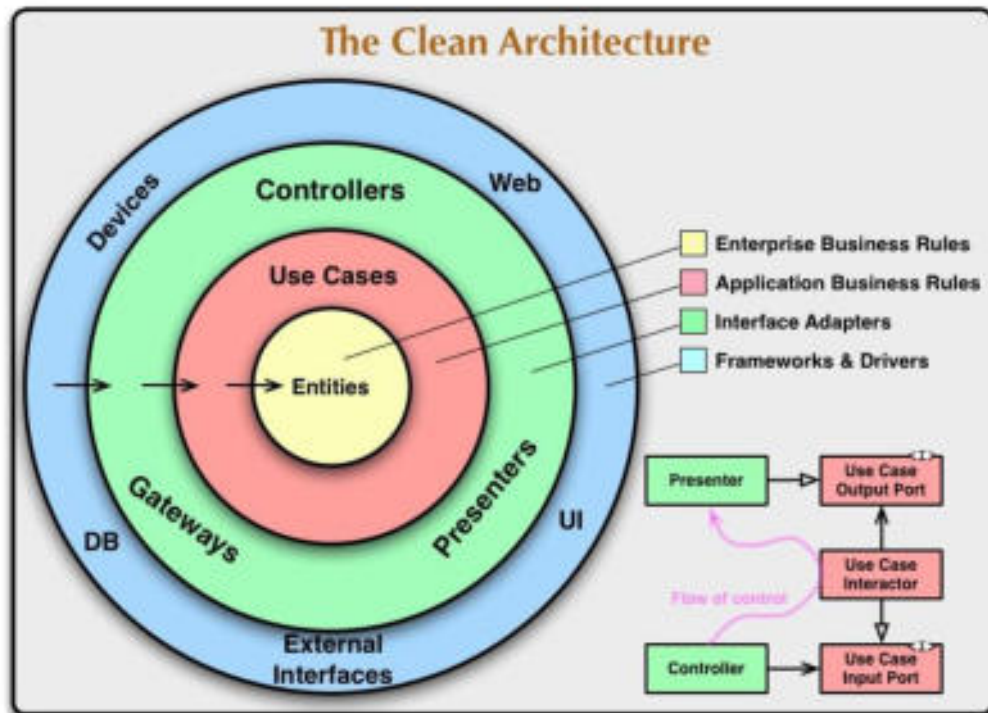
1.13.2: Mô hình kiến trúc Clean Architecture

Clean Architecture (Kiến trúc sạch) là một mô hình thiết kế phần mềm được đề xuất bởi Robert C. Martin (Uncle Bob), giúp phân tách dự án thành các lớp theo chiều ngang rõ ràng, đảm bảo sự độc lập giữa các tầng. Kiến trúc này chia hệ thống thành các lớp:

- Domain Layer (lõi nghiệp vụ): Chứa các use case và entity – độc lập với bất kỳ thư viện hay nền tảng nào.
- Data Layer: Chứa các repository cụ thể, thao tác với cơ sở dữ liệu hoặc API từ xa.
- Presentation Layer: Chứa UI và ViewModel, là nơi dữ liệu được hiển thị và tương tác với người dùng.

Clean Architecture giúp dự án:

- Dễ dàng mở rộng và thay đổi công nghệ ở từng tầng
- Tăng khả năng kiểm thử (unit test, UI test).
- Giảm sự phụ thuộc chéo giữa các thành phần, nhờ vào kỹ thuật Dependency Injection (sử dụng Hilt).



Hình 1.6: Mô hình Clean Architecture

1.14 Bảo mật và tối ưu hóa mã nguồn với ProGuard

ProGuard là một công cụ quan trọng trong quá trình build ứng dụng Android, có nhiệm vụ thu gọn (minify), tối ưu hóa (optimize), làm rối (obfuscate) và loại bỏ (shrink) các phần mã không sử dụng trong ứng dụng. Đây là một bước cần thiết trong quy trình phát hành (release) nhằm giảm kích thước tệp APK/AAB, tăng hiệu suất và bảo vệ mã nguồn khỏi bị dịch ngược (reverse engineering).

Các chức năng chính của ProGuard:

- Shrink (loại bỏ): Xóa bỏ các lớp, phương thức và biến không sử dụng để giảm kích thước tệp.
- Optimize (tối ưu hóa): Tối ưu hóa mã bytecode, loại bỏ các lệnh dư thừa hoặc không cần thiết.
- Obfuscate (làm rối tên): Đổi tên lớp, phương thức và biến thành những chuỗi ngắn, khó đọc (ví dụ: a.b.c()), giúp mã khó bị phân tích.
- Preverify: Kiểm tra tính hợp lệ sau khi mã đã được làm rối và tối ưu.

Lợi ích khi sử dụng ProGuard:

- Giảm kích thước ứng dụng: Nhờ loại bỏ những phần không dùng đến.

- Tăng độ bảo mật: Làm cho việc đọc và dịch ngược mã nguồn trở nên khó khăn hơn, bảo vệ các thuật toán nhạy cảm.
- Cải thiện hiệu suất: Loại bỏ các đoạn mã dư thừa giúp ứng dụng chạy hiệu quả hơn.

Trong quá trình xây dựng và đóng gói ứng dụng, ProGuard được cấu hình để hoạt động trong bản phát hành (release). Các thư viện và lớp quan trọng, chẳng hạn như lớp liên quan đến ViewModel, Hilt, hoặc các thư viện ML, thường sẽ được giữ lại (keep) bằng cách sử dụng các quy tắc cấu hình cụ thể trong tập tin

1.15 Kết luận

Trong chương này, đã trình bày tổng quan về đề tài, chương này đã trình bày tổng quan về các công nghệ, ngôn ngữ lập trình, thư viện và công cụ hỗ trợ được áp dụng cũng như các mô hình kiến trúc như MVVM và Clean Architecture nhằm đảm bảo tính bền vững, mở rộng và dễ bảo trì cho ứng dụng.

Việc nghiên cứu và tìm hiểu lý thuyết về các công nghệ đóng vai trò quan trọng và hỗ trợ đưa ra lựa chọn các framework, thư viện phù hợp nhất để sử dụng đối với đề án.

CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

2.1 Phân tích yêu cầu

2.1.1: Yêu cầu chức năng

Quản lý tệp

Ứng dụng cho phép hiển thị và quản lý danh sách tệp theo từng loại như PDF, Word, Excel, PowerPoint và văn bản thuần. Hỗ trợ lọc, tìm kiếm, sắp xếp theo tên, định dạng, thời gian, kích thước. Cho phép đánh dấu yêu thích, theo dõi lịch sử xem và lưu trữ tệp offline, đồng bộ với cloud.

Chức năng xem và chỉnh sửa tệp PDF :

Hỗ trợ xem tệp PDF với các thao tác zoom, cuộn, điều hướng trang và chuyển đổi chế độ sáng/tối. Cho phép chỉnh sửa như thêm, xóa, sắp xếp trang; chèn chữ ký, chú thích, ghi chú, đánh dấu; tách và gộp tệp PDF.

Chức năng xem tệp Office

Cho phép hiển thị các định dạng Word (DOC, DOCX...), Excel (XLS, XLSX...), PowerPoint (PPT, PPTX...) và văn bản thuần (TXT) với nhiều bảng mã khác nhau. Đảm bảo hiển thị đúng bố cục và nội dung tài liệu.

Chức năng chuyển đổi tệp

Ứng dụng hỗ trợ chuyển đổi các tài liệu Word sang định dạng PDF, đảm bảo giữ nguyên bố cục và nội dung ban đầu. Đồng thời, người dùng có thể chuyển đổi hình ảnh từ thư viện hoặc chụp mới thành tệp PDF, giúp số hóa tài liệu nhanh chóng và thuận tiện cho việc lưu trữ, chia sẻ.

Tính năng AI và OCR

Ứng dụng tích hợp AI để hỗ trợ nhận dạng và xử lý văn bản. Cho phép trích xuất nội dung từ PDF, nhận dạng chữ viết tay và văn bản trong ảnh (OCR). Hỗ trợ dịch văn bản giữa nhiều ngôn ngữ, tóm tắt nội dung tài liệu và chuyển văn bản thành giọng nói (Text-to-Speech). Ngoài ra, người dùng có thể quét tài liệu bằng camera, tự động nhận diện biên và chỉnh sửa, sau đó chuyển đổi thành tệp PDF.

Đồng bộ và lưu trữ Cloud

Ứng dụng hỗ trợ lưu trữ và đồng bộ tệp với nền tảng đám mây, cho phép truy cập từ nhiều thiết bị khác nhau. Đảm bảo an toàn và bảo mật dữ liệu người dùng trong quá trình đồng bộ hóa.

Quản lý người dùng

Ứng dụng cho phép người dùng đăng ký, đăng nhập bằng email và mật khẩu, hỗ trợ khôi phục mật khẩu khi cần thiết. Người dùng có thể xem và chỉnh sửa hồ sơ cá nhân, đồng thời quản lý các cài đặt liên quan đến tài khoản và ứng dụng.

Tính năng Premium và thanh toán

Cung cấp gói Premium giúp loại bỏ quảng cáo, mở khóa các chức năng nâng cao và tăng giới hạn lưu trữ. Hỗ trợ thanh toán trong ứng dụng thông qua Google Play Billing, đồng thời quản lý thông tin đăng ký và lịch sử giao dịch của người dùng.

Công cụ PDF nâng cao

Ứng dụng hỗ trợ thêm mật khẩu để bảo vệ tệp PDF, chia sẻ tài liệu qua các ứng dụng khác và thực hiện thao tác in trực tiếp tài liệu, giúp nâng cao tính bảo mật và tính linh hoạt trong sử dụng.

2.1.2: Yêu cầu phi chức năng

Hiệu suất và tốc độ phản hồi

Ứng dụng khởi động nhanh với thời gian dưới 3 giây. Việc mở và hiển thị tệp được tối ưu hóa để đảm bảo tệp có kích thước trung bình được hiển thị trong vòng không quá 3 giây. Tương tác cuộn trang phải diễn ra mượt mà, không xảy ra hiện tượng giật, lag.

Xử lý tệp lớn

Ứng dụng có khả năng xử lý các tệp PDF hoặc Office dung lượng lớn (> 50MB), đồng thời sử dụng bộ nhớ một cách hiệu quả. Cơ chế **lazy loading** được áp dụng nhằm giảm thiểu tiêu thụ tài nguyên hệ thống khi tải và hiển thị dữ liệu.

Hỗ trợ đa nhiệm

Hệ thống cho phép xử lý nhiều tác vụ đồng thời mà không ảnh hưởng đến hiệu năng chung. Việc sử dụng **coroutines** và quản lý luồng xử lý (threading) hợp lý

giúp tránh việc chặn luồng giao diện người dùng (UI thread) trong các tác vụ nặng như xử lý AI, tải file, hoặc trích xuất dữ liệu.

Giao diện người dùng

Giao diện được xây dựng theo nguyên tắc **Material Design 3**, mang lại cảm giác hiện đại, rõ ràng và thân thiện. Thiết kế trực quan giúp người dùng dễ dàng làm quen, kể cả với những người ít kinh nghiệm sử dụng thiết bị số. Ứng dụng hỗ trợ đa ngôn ngữ tổng cộng 22 ngôn ngữ.

Trải nghiệm người dùng

Giao diện được tối ưu cho nhiều kích thước màn hình khác nhau (thiết kế responsive). Các thao tác đều có phản hồi trực quan, rõ ràng, giúp người dùng dễ dàng tương tác. Thời gian làm quen với ứng dụng được tối thiểu hóa.

Khả năng tiếp cận

Ứng dụng tuân thủ các tiêu chuẩn **WCAG 2.1** nhằm hỗ trợ người dùng khuyết tật. Hệ thống tương thích với trình đọc màn hình và đảm bảo độ tương phản màu sắc phù hợp cho người khiếm thị, từ đó nâng cao khả năng tiếp cận cho mọi đối tượng sử dụng.

Tính ổn định

Ứng dụng đạt tỷ lệ lỗi (crash) rất thấp, dưới 0.5% trên tổng số phiên hoạt động, góp phần đảm bảo hoạt động liên tục và mượt mà. Cơ chế xử lý lỗi được tích hợp nhằm tăng khả năng phục hồi, giúp người dùng không bị mất dữ liệu trong trường hợp xảy ra sự cố bất ngờ. Ứng dụng cũng tránh được tình trạng treo hoặc **ANR (Application Not Responding)**, đảm bảo phản hồi kịp thời với thao tác người dùng.

Tính nhất quán

Nội dung tập được hiển thị chính xác và nhất quán trên nhiều thiết bị và phiên bản hệ điều hành. Các yếu tố định dạng như văn bản, hình ảnh, đồ họa và bố cục tài liệu luôn được giữ nguyên trong quá trình hiển thị hoặc chuyển đổi. Kết quả xử lý (ví dụ như chuyển đổi tập) đảm bảo đồng nhất trên các nền tảng, tăng độ tin cậy cho người dùng khi làm việc với dữ liệu quan trọng.

Bảo vệ dữ liệu

Các thông tin nhạy cảm được mã hóa và lưu trữ an toàn trong cơ sở dữ liệu cục bộ. Hệ thống bảo mật nghiêm ngặt giúp bảo vệ thông tin đăng nhập và xác thực người dùng, đồng thời hỗ trợ xử lý an toàn đối với các tệp được đặt mật khẩu.

Quyền truy cập

Ứng dụng chỉ yêu cầu các quyền thực sự cần thiết cho chức năng hoạt động. Khi yêu cầu quyền, hệ thống luôn giải thích rõ ràng mục đích sử dụng, đồng thời cho phép người dùng tự do kiểm soát và điều chỉnh quyền truy cập vào dữ liệu cá nhân.

Truyền tải dữ liệu

Tất cả dữ liệu được truyền tải qua mạng đều sử dụng giao thức bảo mật **HTTPS**, giúp ngăn chặn các nguy cơ đánh cắp thông tin trong quá trình truyền. Ngoài ra, dữ liệu đồng bộ lên nền tảng đám mây cũng được mã hóa nhằm đảm bảo an toàn tuyệt đối cho người dùng.

Kiến trúc phần mềm

Áp dụng mô hình MVVM rõ ràng, đảm bảo phân tách trách nhiệm giữa các thành phần giao diện, xử lý logic và dữ liệu. Các tính năng được tổ chức theo dạng mô-đun, hỗ trợ việc mở rộng và bảo trì hệ thống một cách hiệu quả.

Tương thích nền tảng

Hỗ trợ các thiết bị Android từ phiên bản API 28 (Android 9.0) trở lên. Ứng dụng được tối ưu hóa để hoạt động mượt mà trên nhiều kích thước và độ phân giải màn hình, bao gồm cả điện thoại và máy tính bảng.

Tương thích phần cứng

Đảm bảo hiệu suất ổn định trên thiết bị có cấu hình trung bình ($RAM \geq 2GB$). Hệ thống được tối ưu để tương thích với nhiều kiến trúc CPU phổ biến như ARM và x86, đồng thời hỗ trợ tốt các loại thiết bị camera khác nhau phục vụ chức năng quét tài liệu.

Tương thích định dạng

Hệ thống xử lý tốt các định dạng Office từ phiên bản 1997 đến 2019, cùng với khả năng hỗ trợ đầy đủ các đặc tả đặc biệt của tệp PDF như biểu mẫu (form), lớp (layer), và tính năng bảo mật. Ngoài ra, ứng dụng còn hỗ trợ nhiều bảng mã văn bản để đảm bảo khả năng đọc và hiển thị chính xác nội dung.

Hoạt động Offline

Ứng dụng đảm bảo khả năng hoạt động ổn định ngay cả khi không có kết nối internet, cung cấp đầy đủ các chức năng cốt lõi, cho phép lưu trữ cục bộ các tệp

Quảng cáo và Thanh toán

Ứng dụng tích hợp quảng cáo và thanh toán theo chuẩn Google Play, thiết kế quảng cáo thân thiện, không gây gián đoạn trải nghiệm. Hạn chế số lượng quảng cáo trong mỗi phiên sử dụng để tránh gây khó chịu cho người dùng. Sử dụng hệ thống Google Play Billing để xử lý thanh toán an toàn, đáng tin cậy. Thông tin giao dịch được lưu trữ và mã hóa theo chuẩn bảo mật hiện hành.

Tuân thủ chính sách cửa hàng ứng dụng

Tuân thủ mọi yêu cầu và tiêu chuẩn kỹ thuật của Google Play Store. Luôn cập nhật theo các thay đổi về chính sách hoặc phiên bản Android mới.

Minh bạch quyền dữ liệu

Giải thích rõ ràng về quyền truy cập và mục đích sử dụng dữ liệu người dùng. Cam kết bảo mật và không thu thập dữ liệu ngoài phạm vi cho phép.

2.2 Thiết kế hệ thống

2.2.1. Tổng quan

Sơ đồ dưới đây thể hiện các công nghệ sử dụng tương ứng với thành phần của hệ thống theo từng lớp.

Mobile	
Backend	
Database & Storage	

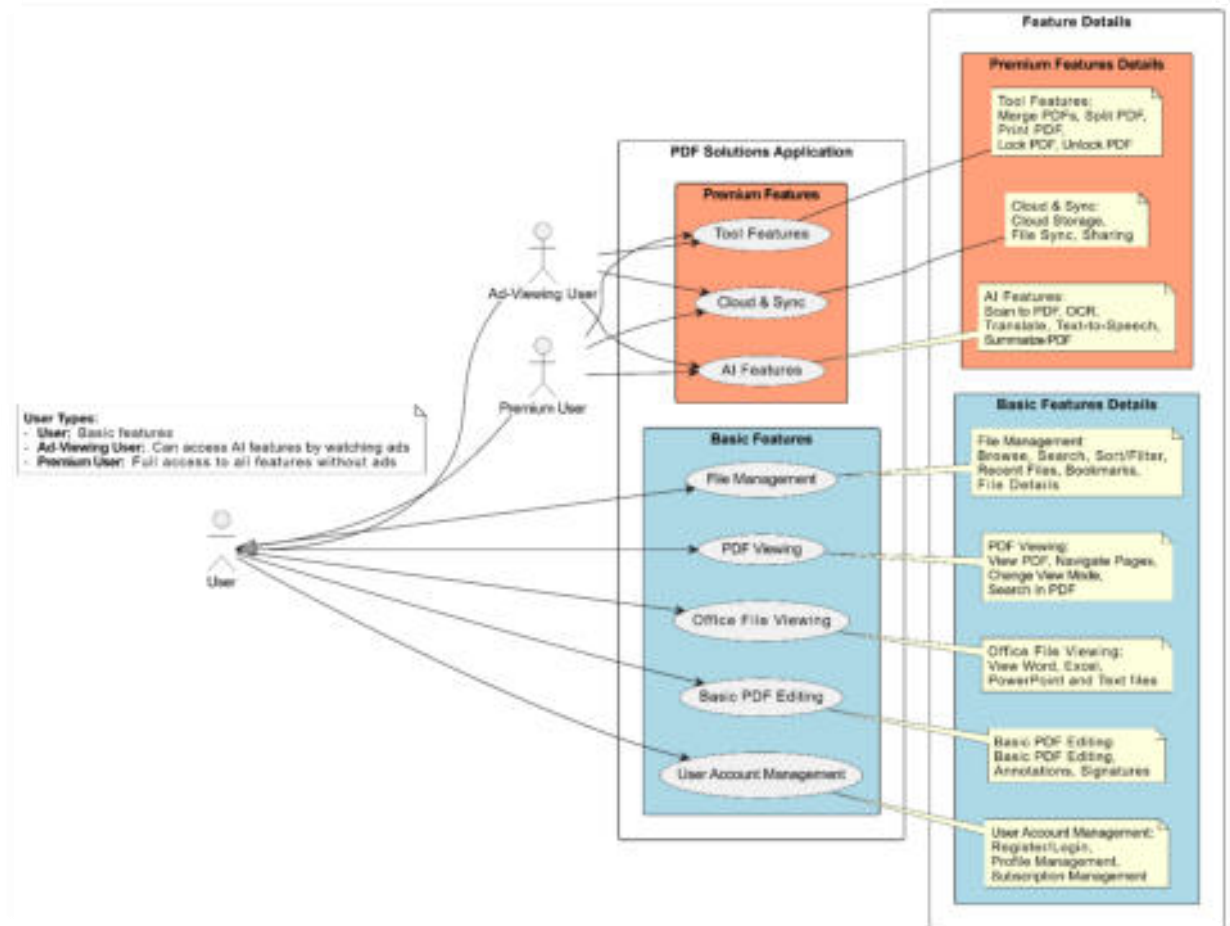
Hình 2.1: Các công nghệ sử dụng

Ứng dụng được phát triển nhằm hỗ trợ người dùng quản lý, xem và chỉnh sửa các tệp văn bản như PDF, Word, Excel... ngay trên thiết bị di động một cách tiện lợi và thông minh. Bên cạnh việc phục vụ cho mục tiêu nghiên cứu và hoàn thiện đồ án tốt nghiệp, ứng dụng còn được định hướng theo hướng sản phẩm thực tế có tiềm năng thương mại hóa. Với các tính năng cao cấp như AI, OCR, đồng bộ cloud và thanh toán in-app, quảng cáo cho người dùng miễn phí, ứng dụng hướng đến khả năng tạo doanh thu bền vững từ người dùng. Đây là bước khởi đầu cho việc phát triển một sản phẩm công nghệ hữu ích, hiện đại và có tính ứng dụng cao. Hệ thống bao gồm các thành phần chính như sau:

- **Ứng dụng di động:** Ứng dụng di động được phát triển bằng Kotlin - ngôn ngữ lập trình hiện đại được Google khuyến nghị cho Android, kết hợp với Java để đảm bảo tương thích với các thư viện legacy, ứng dụng được viết theo ngôn ngữ native để đảm bảo hiệu năng xử lý mang hiệu suất cao nhất. Giao diện người dùng được xây dựng bằng toolkit Jetpack Compose và XML, tuân thủ nguyên tắc tách biệt logic và giao diện. Ứng dụng được xây dựng theo định hướng dễ bảo trì và phát triển, đồng thời tích hợp thêm quảng cáo và thanh toán qua App với mục tiêu đem lại giá trị thực, lợi nhuận trong tương lai.
- **Máy chủ backend:** Hệ thống backend được xây dựng trên nền tảng Node.js, tận dụng mô hình non-blocking I/O để xử lý đồng thời nhiều kết nối. Express framework được áp dụng để phát triển các REST API với cú pháp đơn giản nhưng mạnh mẽ, hỗ trợ middleware linh hoạt cho việc xác thực, ghi log và xử lý lỗi. Kiến trúc này cho phép hệ thống dễ dàng mở rộng và tối ưu hiệu suất khi lượng người dùng tăng.
- **Cơ sở dữ liệu:** Giải pháp lưu trữ dữ liệu tập trung vào Supabase - nền tảng backend-as-a-service mã nguồn mở, cung cấp cơ sở dữ liệu PostgreSQL với khả năng real-time và hệ thống xác thực tích hợp. Bên cạnh đó, hệ thống còn sử dụng cơ sở dữ liệu SQL truyền thống để đảm bảo tính toàn vẹn dữ liệu, hỗ trợ các truy vấn phức tạp và duy trì hiệu suất ổn định trong điều kiện tải cao.

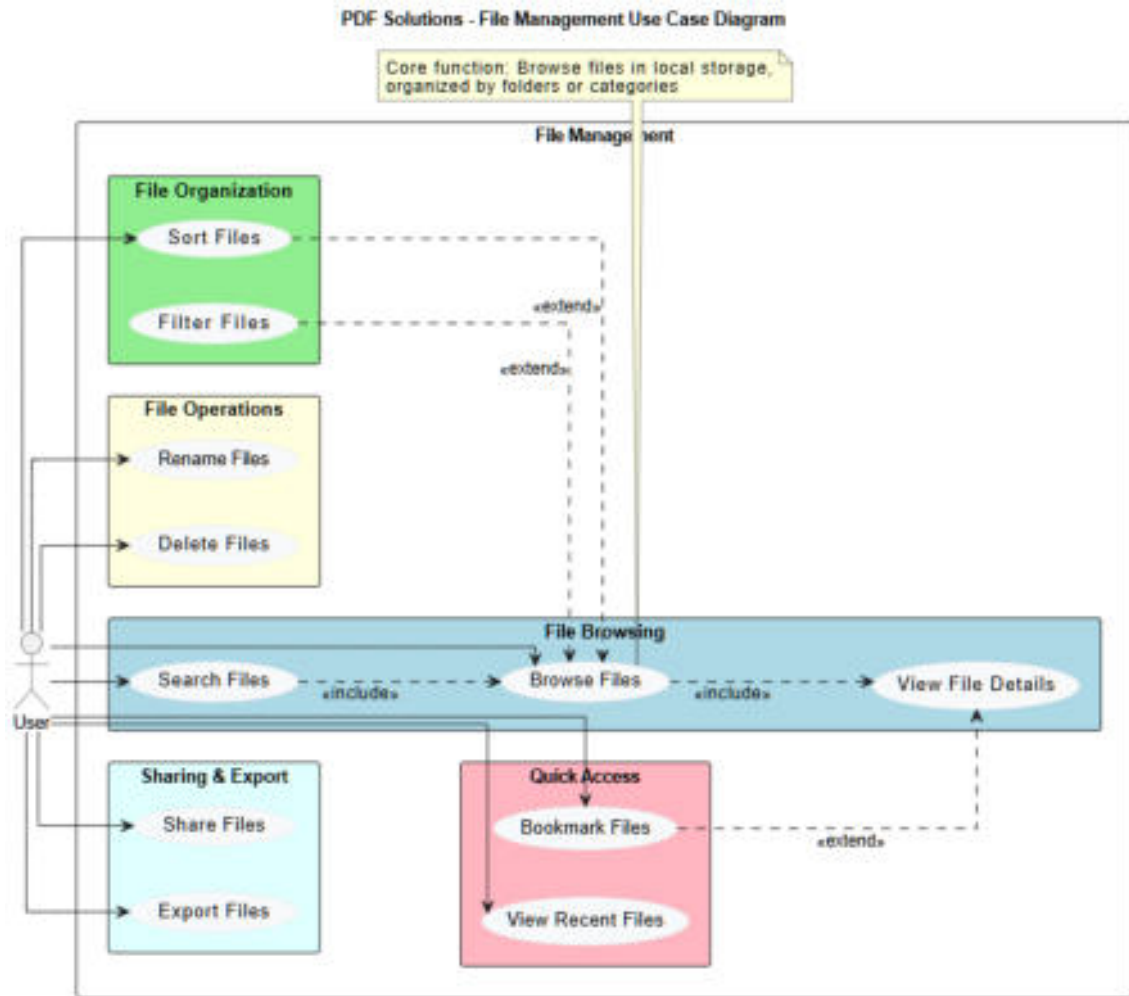
2.2.2. Biểu đồ ca sử dụng

a) Sơ đồ ca sử dụng và đặc tả tổng quát



Hình 2.2: Sơ đồ Usecase tổng quan

b) Sơ đồ ca sử dụng và đặc tả quản lý File



Hình 2.3: Sơ đồ Usecase quản lý File

Bảng 2.1: Đặc tả mở File

Mã ca sử dụng	UC-01
Tên ca sử dụng	Browse Files
Tác nhân	Người dùng
Mô tả	Cho phép người dùng duyệt qua các tệp trong bộ nhớ của thiết bị, được tổ chức theo thư mục hoặc danh mục
Điều kiện trước	Người dùng đã cài đặt và mở ứng dụng

Điều kiện sau	Người dùng đã cấp quyền xem toàn bộ file trong thiết bị, danh sách tệp được hiển thị cho người dùng		
Kịch bản		Hành động của tác nhân	Hành động của hệ thống
	1	Mở ứng dụng	Hiển thị màn hình Home và hiển thị danh sách tệp
	2	Chọn tệp để mở	Mở tệp được chọn
	3		Lưu trạng thái của tệp vào tab recent (đã đánh dấu, chi tiết thông tin tệp)

Bảng 2.2: Đặc tả tìm kiếm File

Mã ca sử dụng	UC-02		
Tên ca sử dụng	Search Files		
Tác nhân	Người dùng		
Mô tả	Cho phép người dùng tìm kiếm tệp theo tên hoặc nội dung		
Điều kiện trước	Người dùng đã mở ứng dụng và đang ở màn hình duyệt tệp		
Điều kiện sau	Hiển thị kết quả tìm kiếm phù hợp với từ khóa		
Kịch bản		Hành động của tác nhân	Hành động của hệ thống
	1	Mở ứng dụng	Hiển thị màn hình Home và hiển thị danh sách tệp

	2	Chọn biểu tượng tìm kiếm	Hiển thị kết quả tìm kiếm
	3	Thực hiện các thao tác quản lý (tìm kiếm, lọc, sắp xếp)	Thực hiện các thao tác quản lý theo yêu cầu
	4	Đánh dấu tệp yêu thích hoặc xem chi tiết	Lưu trạng thái của tệp (đã đánh dấu, chi tiết thông tin tệp)

Bảng 2.3: Đặc tả sắp xếp, lọc file

Mã ca sử dụng	UC-03		
Tên ca sử dụng	Sort and filter file		
Tác nhân	Người dùng		
Mô tả	Cho phép người dùng sắp xếp danh sách tệp theo các tiêu chí khác nhau		
Điều kiện trước	Người dùng đã mở ứng dụng và đang ở màn hình duyệt tệp		
Điều kiện sau	Danh sách tệp được sắp xếp theo tiêu chí đã chọn		
Kịch bản		Hành động của tác nhân	Hành động của hệ thống
	1	Mở ứng dụng	Hiển thị màn hình Home và hiển thị danh sách tệp
	2	Chọn biểu tượng sort/ filer	Hiển thị bảng tùy chọn các tiêu chí để sort/ filter

	3	Chọn tiêu chí	Áp dụng tiêu chí đã chọn, đóng màn hình bảng tùy chọn và thực hiện sắp xếp theo tiêu chí và lưu trạng thái
--	---	---------------	--

Bảng 2.4: Đặc tả đổi tên/ xóa/ đánh dấu tệp

Mã ca sử dụng	UC-04		
Tên ca sử dụng	Rename/ Delete/ Bookmark File		
Tác nhân	Người dùng		
Mô tả	Cho phép người dùng đổi tên/ xóa / đánh dấu tệp đã được chọn		
Điều kiện trước	Người dùng đã mở ứng dụng và đang ở màn hình duyệt tệp		
Điều kiện sau	Hiển thị kết quả tương ứng		
Kịch bản		Hành động của tác nhân	Hành động của hệ thống
	1	Mở ứng dụng	Hiển thị màn hình Home và hiển thị danh sách tệp
	2	Chọn biểu tượng tùy chỉnh	Hiển thị bottom sheet tùy chọn
	3	Thực hiện các thao tác đổi tên/ xóa/ đánh dấu tệp	Thực hiện các thao tác quản lý theo yêu cầu

Bảng 2.5: Đặc tả xem tệp recent/ bookmark

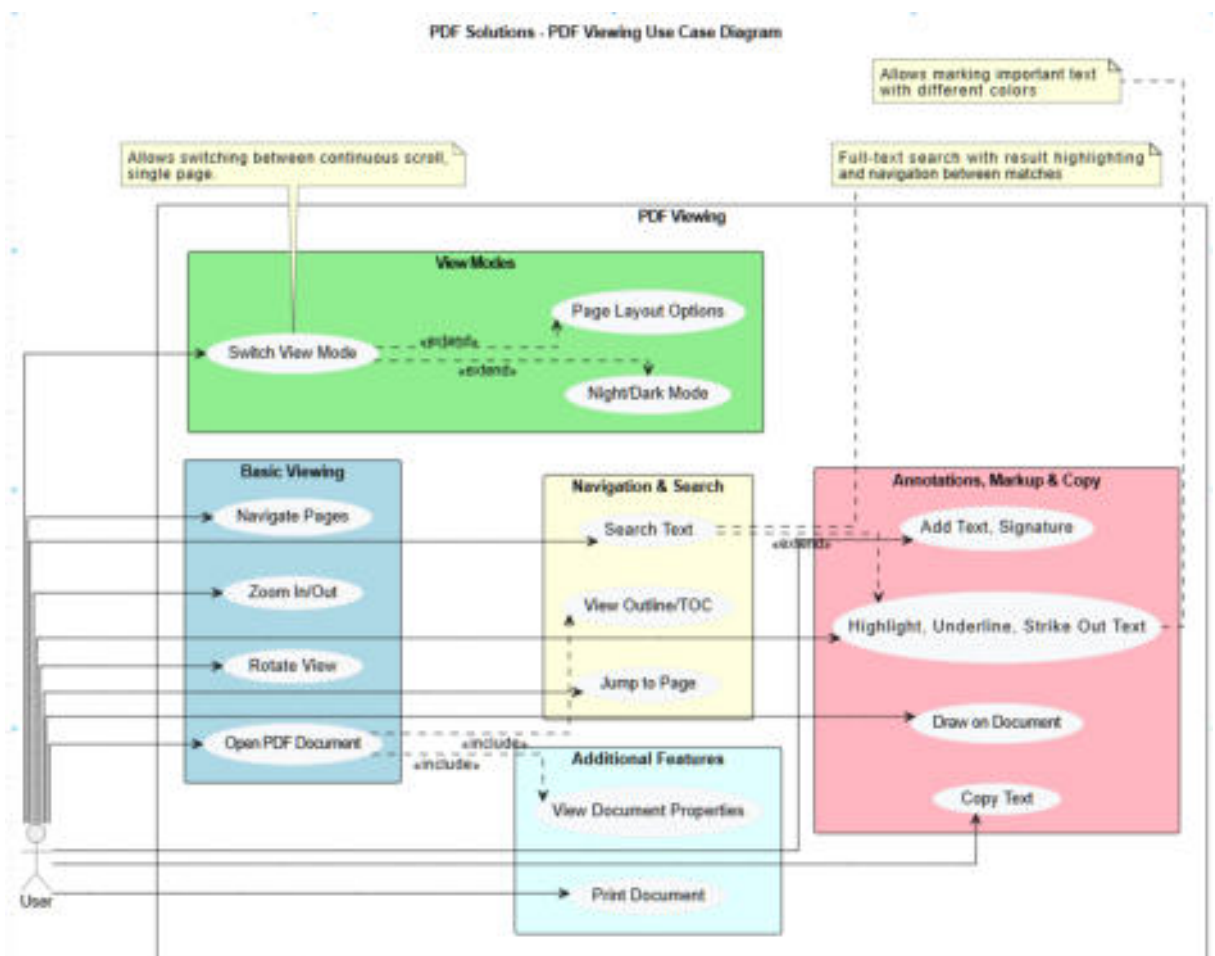
Mã ca sử dụng	UC-05		
Tên ca sử dụng	View Recent/ Bookmark File		
Tác nhân	Người dùng		
Mô tả	Cho phép người dùng xem tệp vừa được tương tác/ đã đánh dấu		
Điều kiện trước	Người dùng đã mở ứng dụng và đang ở màn hình duyệt tệp		
Điều kiện sau	Hiển thị kết quả tương ứng		
Kịch bản		Hành động của tác nhân	Hành động của hệ thống
	1	Mở ứng dụng	Hiển thị màn hình Home và hiển thị danh sách tệp
	2	Chọn tab tương ứng Recent/ Bookmark	Hiển thị danh sách vừa được tương tác / đã được đánh dấu

Bảng 2.6: Đặc tả chia sẻ tệp PDF

Mã ca sử dụng	UC-06		
Tên ca sử dụng	Share File		
Tác nhân	Người dùng		
Mô tả	Cho phép người dùng chia sẻ tệp		
Điều kiện trước	Người dùng đã mở ứng dụng và đang ở màn hình duyệt tệp		
Điều kiện sau	Chia sẻ tệp thành công		
Kịch bản		Hành động của tác nhân	Hành động của hệ thống

	1	Mở ứng dụng	Hiển thị màn hình Home và hiển thị danh sách tệp
	2	Chọn biểu tượng tùy chỉnh	Hiển thị bottom sheet tùy chọn
	3	Chọn biểu tượng chia sẻ và chọn cách thức chia sẻ	Chia sẻ tệp

b) Sơ đồ ca sử dụng và đặc tả xem và chỉnh sửa file PDF



Hình 2.4: Sơ đồ Usecase xem và chỉnh sửa File PDF

Bảng 2.7: Đặc tả mở và xem file PDF

Mã ca sử dụng	UC-07		
Tên ca sử dụng	Open PDF Document		
Tác nhân	Người dùng		
Mô tả	Cho phép người dùng mở và hiển thị tệp PDF		
Điều kiện trước	Người dùng đã chọn một tệp PDF để mở		
Điều kiện sau	Tệp PDF được mở và hiển thị trên màn hình		
Kịch bản		Hành động của tác nhân	Hành động của hệ thống
	1	Chọn tệp PDF từ danh sách tệp	Hiển thị nội dung file PDF trên màn hình

Bảng 2.8: Đặc tả chuyển trang khi xem file

Mã ca sử dụng	UC-08		
Tên ca sử dụng	Navigate Pages		
Tác nhân	Người dùng		
Mô tả	Cho phép người dùng điều hướng giữa các trang trong tệp PDF		
Điều kiện trước	Người dùng đã chọn một tệp PDF để mở		
Điều kiện sau	Tệp PDF được mở và hiển thị trên màn hình, file PDF lớn hơn 1 trang		
Kịch bản		Hành động của tác nhân	Hành động của hệ thống

	1	Vuốt màn hình để chuyển trang (vuốt xuống/ lên nếu đang ở chế độ dọc, vuốt trái phải nếu đang ở chế độ ngang) hoặc sử dụng thanh cuộn, màn hình preview để điều hướng	Hiển thị trang mới
--	---	---	--------------------

Bảng 2.9: Đặc tả phóng to thu nhỏ khi xem file

Mã ca sử dụng	UC-09		
Tên ca sử dụng	Zoom In/Out		
Tác nhân	Người dùng		
Mô tả	Cho phép người dùng phóng to, thu nhỏ nội dung PDF		
Điều kiện trước	Người dùng đã chọn một tệp PDF để mở		
Điều kiện sau	Tệp PDF được mở và hiển thị trên màn hình		
Kịch bản		Hành động của tác nhân	Hành động của hệ thống
	1	Sử dụng thao tác đa điểm để thực hiện phóng to thu nhỏ	Hiển thị nội dung theo thao tác, phóng to hoặc thu nhỏ

Bảng 2.10: Đặc tả các chế độ xem khi xem file

Mã ca sử dụng	UC-10		
Tên ca sử dụng	Rotate View/ Change View Mode/ Dark/Light Mode		
Tác nhân	Người dùng		

Mô tả	Cho phép người dùng chuyển đổi chế độ xem / trượt chuyển trang / hiển thị màu sắc PDF		
Điều kiện trước	Người dùng đã chọn một tệp PDF để mở		
Điều kiện sau	Tệp PDF được mở và hiển thị trên màn hình		
Kịch bản		Hành động của tác nhân	Hành động của hệ thống
	1	Nhấn vào nút xoay/ chế độ đọc	Hiển thị PDF được xoay ngang 90 độ/ chuyển đổi chế độ chuyển trang / chuyển đổi chế độ sáng tối xem PDF

Bảng 2.11: Đặc tả các chế độ xem khi xem file

Mã ca sử dụng	UC-11		
Tên ca sử dụng	Jump to Page		
Tác nhân	Người dùng		
Mô tả	Cho phép người dùng chuyển tới trang bằng cách điền số trang tùy chọn		
Điều kiện trước	Người dùng đã chọn một tệp PDF để mở, số trang điền vào phải nhỏ hơn hoặc bằng số trang của PDF đã mở.		
Điều kiện sau	Tệp PDF được mở và hiển thị trên màn hình		
Kịch bản		Hành động của tác nhân	Hành động của hệ thống
	1	Nhấn vào nút tùy chọn có chú thích go to page ở mục mở rộng	Hiển thị giao diện cho phép nhập số trang

	2	Nhập số trang muốn xem và chọn nút xác nhận	Di chuyển đến số trang đã nhập
--	---	---	--------------------------------

Bảng 2.12: Đặc tả các chế độ xem khi xem file

Mã ca sử dụng	UC-12		
Tên ca sử dụng	Search Text		
Tác nhân	Người dùng		
Mô tả	Cho phép người dùng tìm kiếm ký tự trong file PDF, đánh dấu bằng màu vàng và di chuyển đến trang chứa ký tự đó.		
Điều kiện trước	Người dùng đã chọn một tệp PDF để mở.		
Điều kiện sau	Tệp PDF được mở và hiển thị trên màn hình		
Kịch bản		Hành động của tác nhân	Hành động của hệ thống
	1	Nhấn vào nút tùy chọn có biểu tượng tìm kiếm	Hiển thị giao diện cho phép nhập ký tự
	2	Nhập ký tự muốn tìm kiếm và chọn nút xác nhận	Di chuyển đến trang ký tự đã tìm kiếm nếu có và đánh dấu ký tự bằng màu vàng

Bảng 2.13: Đặc tả các chế độ chỉnh sửa và copy ký tự ở file PDF

Mã ca sử dụng	UC-13		
Tên ca sử dụng	Highlight, Underline, Strike Out Text, Copy		
Tác nhân	Người dùng		

Mô tả	Cho phép người dùng đánh dấu, gạch chân, gạch ngang hoặc copy văn bản trong PDF		
Điều kiện trước	Người dùng đã chọn một tệp PDF để mở.		
Điều kiện sau	Tệp PDF được mở và hiển thị trên màn hình		
Kịch bản		Hành động của tác nhân	Hành động của hệ thống
	1	Nhấn vào nút tùy chọn có biểu tượng chỉnh sửa, và thực hiện chọn chức năng theo ý muốn	Hiển thị giao diện cho phép chọn tính năng muốn sử dụng
	2	Thực hiện chọn ký tự muốn đánh dấu, gạch chân, gạch ngang, copy	Thay đổi theo chỉnh sửa của người dùng hoặc Copy text nếu người dùng chọn chế độ copy

Bảng 2.14: Đặc tả các chế độ thêm ký tự/ chữ ký nổi ở file PDF

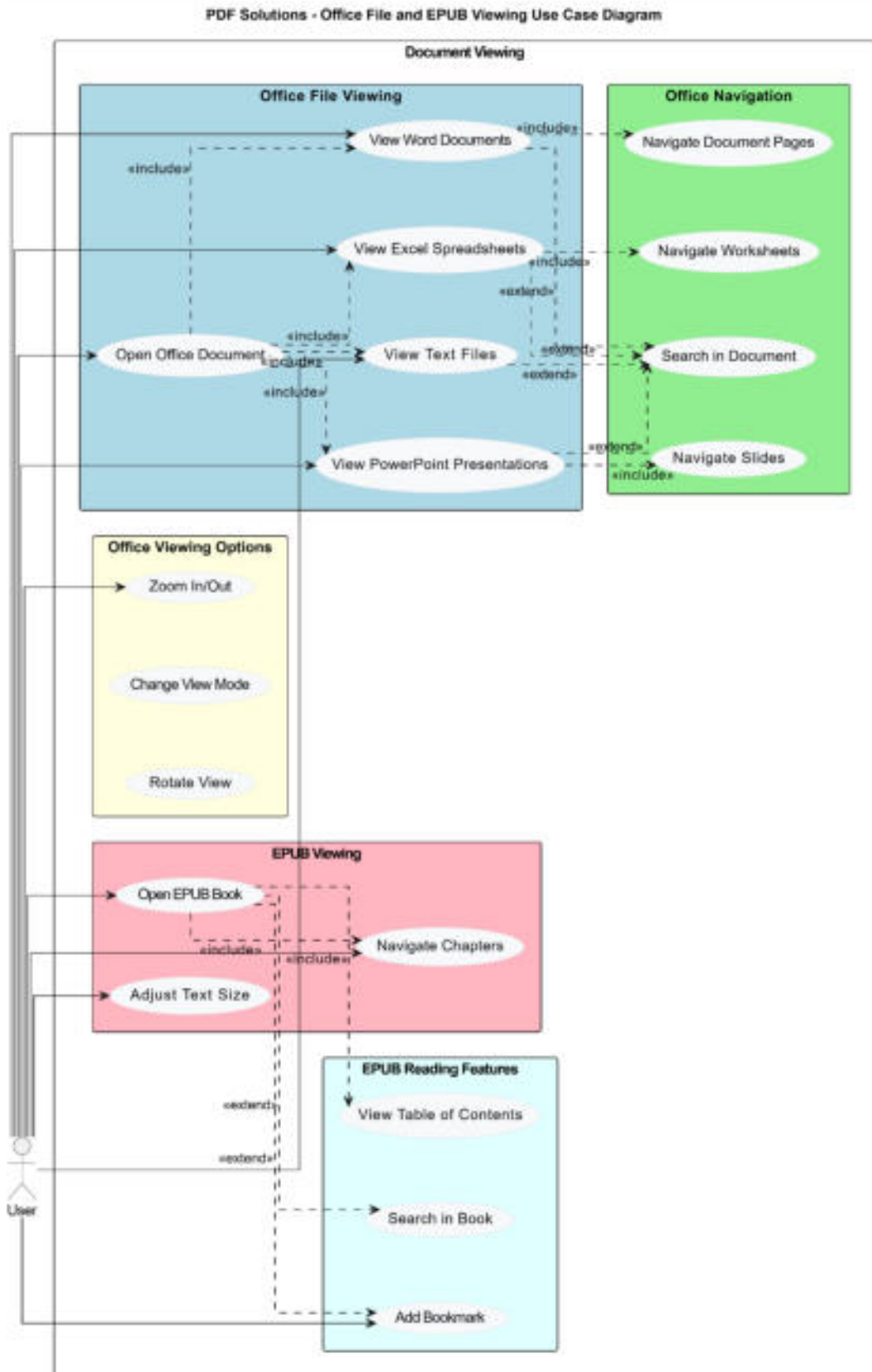
Mã ca sử dụng	UC-14		
Tên ca sử dụng	Add Text, Signature		
Tác nhân	Người dùng		
Mô tả	Cho phép người dùng thêm chữ nổi, chữ ký nổi		
Điều kiện trước	Người dùng đã chọn một tệp PDF để mở.		
Điều kiện sau	Tệp PDF được mở và hiển thị trên màn hình		
Kịch bản		Hành động của tác nhân	Hành động của hệ thống

	1	Nhấn vào nút tùy chọn có biểu tượng chỉnh sửa, và thực hiện chọn chức năng theo ý muốn	Hiển thị giao diện cho phép chọn tính năng muốn sử dụng
	2	Thực hiện chọn nhập ký tự hoặc ký lên giao diện sai đó nhấn nút xác nhận	Hiển thị chữ nổi/ chữ ký nổi lên màn hình và cho phép di chuyển đến khi nhấn nút xác nhận lưu

Bảng 2.15: Đặc tả chế độ in file

Mã ca sử dụng	UC-15		
Tên ca sử dụng	Print Document		
Tác nhân	Người dùng		
Mô tả	Cho phép người dùng in tệp PDF		
Điều kiện trước	Người dùng đã chọn một tệp PDF, đã kết nối máy in		
Điều kiện sau	In thành công		
Kịch bản		Hành động của tác nhân	Hành động của hệ thống
	1	Nhấn vào nút tùy chọn có biểu tượng in.	Hiển thị giao diện cho phép tùy chọn các thuộc tính khi in.
	2	Thực hiện tùy chỉnh các thuộc tính để in	
	3	Nhấn nút xác nhận	Thực hiện chức năng in

c) Sơ đồ ca sử dụng và đặc tả xem Office, EPUB, Txt File



Hình 2.5: Sơ đồ Usecase xem file Office, EPUB, Txt

Bảng 2.16: Đặc tả chế độ xem file Word

Mã ca sử dụng	UC-16		
Tên ca sử dụng	View Word Documents		
Tác nhân	Người dùng		
Mô tả	Cho phép người dùng xem nội dung tài liệu Word (.doc, .docx)		
Điều kiện trước	Người dùng đã mở một tài liệu Word		
Điều kiện sau	Nội dung tài liệu Word được hiển thị đúng định dạng		
Kịch bản		Hành động của tác nhân	Hành động của hệ thống
	1	Chọn tài liệu Word từ danh sách tệp	Phân tích và tải nội dung tài liệu Word
	2	Xem nội dung tài liệu	Hiển thị văn bản, hình ảnh, bảng và các định dạng khác
	3	Cuộn để xem các trang khác nhau	Di chuyển qua các trang tiếp theo

Bảng 2.17: Đặc tả chế độ xem file Excel

Mã ca sử dụng	UC-17		
Tên ca sử dụng	View Excel Spreadsheets		
Tác nhân	Người dùng		
Mô tả	Cho phép người dùng xem nội dung bảng tính Excel (.xls, .xlsx)		
Điều kiện trước	Người dùng đã mở một tài liệu Excel		

Điều kiện sau	Nội dung bảng tính Excel được hiển thị đúng định dạng		
Kịch bản		Hành động của tác nhân	Hành động của hệ thống
	1	Chọn tài liệu Excel từ danh sách tệp	Phân tích và tải nội dung tài liệu Excel
	2	Xem nội dung bảng tính	Hiển thị dữ liệu trong các ô, công thức, biểu đồ
	3	Chuyển đổi giữa các worksheet	Hiển thị nội dung và danh sách các worksheet có trong tài liệu

Bảng 2.18: Đặc tả chế độ xem file PowerPoint

Mã ca sử dụng	UC-18		
Tên ca sử dụng	View PowerPoint Presentations		
Tác nhân	Người dùng		
Mô tả	Cho phép người dùng xem nội dung bài thuyết trình PowerPoint (.ppt, .pptx)		
Điều kiện trước	Người dùng đã mở một tài liệu PowerPoint		
Điều kiện sau	Nội dung bài thuyết trình được hiển thị đúng định dạng		
Kịch bản		Hành động của tác nhân	Hành động của hệ thống
	1	Chọn tài liệu PowerPoint từ danh sách tệp	Phân tích và tải nội dung tài liệu PowerPoint
	2	Xem nội dung các slides	Hiển thị các slide với đầy đủ văn bản, hình ảnh, đồ họa

	3	Trượt để xem các slide khác nhau	Di chuyển qua các slide tiếp theo
--	---	----------------------------------	-----------------------------------

Bảng 2.19: Đặc tả chế độ xem file Text, HTML, XML

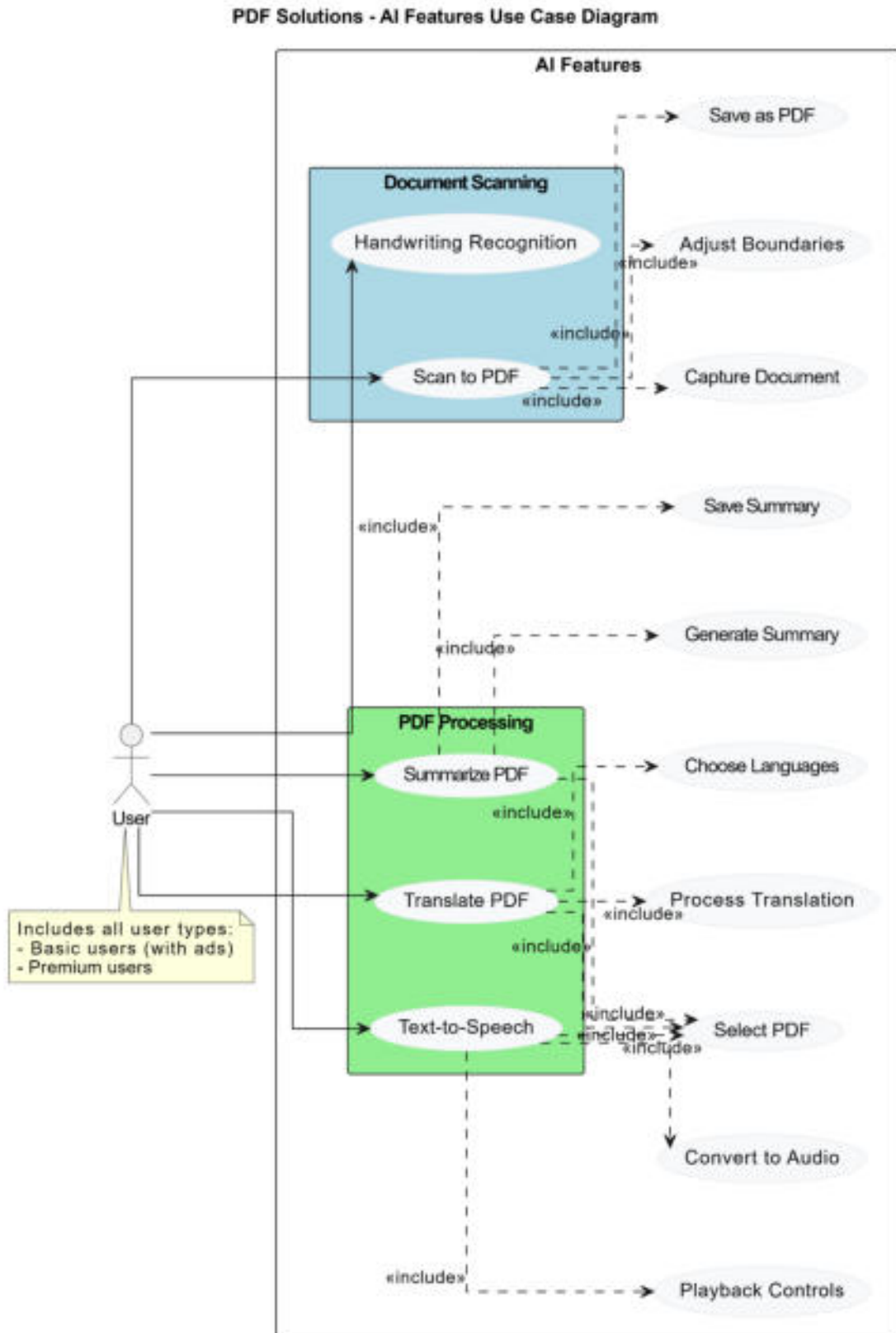
Mã ca sử dụng	UC-19		
Tên ca sử dụng	View Text, HTML, XML Files		
Tác nhân	Người dùng		
Mô tả	Cho phép người dùng xem nội dung tệp văn bản thuần		
Điều kiện trước	Người dùng đã mở một tệp văn bản		
Điều kiện sau	Nội dung tệp văn bản được hiển thị		
Kịch bản		Hành động của tác nhân	Hành động của hệ thống
	1	Chọn tệp từ danh sách tệp	Phân tích và tải nội dung tài liệu
	2	Xem nội dung văn bản	Hiển thị các văn bản với định dạng chuẩn
	3	Trượt để xem toàn bộ nội dung	Tự động ngắt dòng nếu cần thiết

Bảng 2.20: Đặc tả chế độ xem file EPUB

Mã ca sử dụng	UC-20
Tên ca sử dụng	View EPUB Book

Tác nhân	Người dùng		
Mô tả	Cho phép người dùng mở, xem và tương tác với sách điện tử định dạng EPUB		
Điều kiện trước	Người dùng đã cài đặt ứng dụng và có sách EPUB trên thiết bị		
Điều kiện sau	Sách EPUB được mở và hiển thị cho người dùng		
Kịch bản		Hành động của tác nhân	Hành động của hệ thống
	1	Chọn tệp từ danh sách tệp	Phân tích và tải nội dung sách EPUB
	2	Xem nội dung văn bản	Tải và hiển thị nội dung sách với định dạng phù hợp
	3	Trượt hoặc nhấn phím điều hướng để chuyển trang	Chuyển trang

d) Sơ đồ ca sử dụng và đặc tả của các chức năng AI



Hình 2.6: Sơ đồ Usecase của các chức năng AI

Bảng 2.21: Đặc tả chức năng AI Scan to PDF

Mã ca sử dụng	UC-21		
Tên ca sử dụng	Scan to PDF		
Tác nhân	Người dùng		
Mô tả	Cho phép người dùng quét và tự động nhận dạng tài liệu bằng camera và chuyển đổi thành file PDF		
Điều kiện trước	Người dùng đã cài đặt ứng dụng và cấp quyền truy cập camera		
Điều kiện sau	Tài liệu đã được quét và lưu dưới dạng PDF		
		Hành động của tác nhân	Hành động của hệ thống
	1	Chọn tính năng "Scan to PDF"	Kích hoạt camera
	2	Căn chỉnh camera với tài liệu	Tự động phát hiện cạnh tài liệu, có 2 chức năng tự động chụp khi camera ổn định hoặc chụp thủ công
Kịch bản chính	3	Chụp tài liệu	Chụp và xử lý hình ảnh
	4	Điều chỉnh đường biên của tài liệu nếu cần	Chuyển đổi hình ảnh thành PDF
	5	Lưu tài liệu thành PDF	Lưu tệp vào bộ nhớ thiết bị
	1	Nếu điều kiện ánh sáng kém, camera rung lắc	Cảnh báo người dùng và đề xuất sử dụng đèn flash, gợi ý người dùng giữ ổn định camera

Kịch bản thay thế	2	Nếu không phát hiện được cạnh tài liệu	Cho phép người dùng điều chỉnh thủ công
--------------------------	---	--	---

Bảng 2.22: Đặc tả chức năng AI Speech PDF

Mã ca sử dụng	UC-22		
Tên ca sử dụng	Speech PDF		
Tác nhân	Người dùng		
Mô tả	Cho phép người dùng chuyển đổi nội dung văn bản PDF thành âm thanh		
Điều kiện trước	Người dùng đã mở một tài liệu PDF chứa văn bản có thể trích xuất		
Điều kiện sau	Nội dung văn bản được đọc thành giọng nói với đa ngôn ngữ		
Kịch bản chính		Hành động của tác nhân	Hành động của hệ thống
	1	Chọn tính năng Speech PDF	Mở màn hình chọn file
	2	Chọn tài liệu PDF	Trích xuất văn bản, ví trí văn bản từ PDF
	3	Điều chỉnh tốc độ đọc, ngôn ngữ đọc	Hiển thị các giao diện với tùy chọn trên màn hình, hành động này có thể thay đổi trong khi đọc
	4	Điều chỉnh đường biên của tài liệu nếu cần	Chuyển đổi hình ảnh thành PDF

	5	Bắt đầu phát âm thanh	Phát âm thanh và đánh dấu văn bản đang được đọc bằng highlight màu vàng, trong quá trình này có thể pause/resume và tiếp tục quá trình đọc ngay tại vị trí đã dừng
Kịch bản thay thế	1	Nếu PDF không chứa văn bản có thể trích xuất	Hiển thị giao diện cảnh báo người dùng

Bảng 2.23: Đặc tả chức năng AI Translate PDF

Mã ca sử dụng	UC-23		
Tên ca sử dụng	Translate PDF		
Tác nhân	Người dùng		
Mô tả	Cho phép người dùng dịch nội dung văn bản PDF sang ngôn ngữ khác.		
Điều kiện trước	Người dùng đã mở một tài liệu PDF chứa văn bản hoặc PDF có hình ảnh chứa văn bản.		
Điều kiện sau	Nội dung văn bản được dịch sang ngôn ngữ mục tiêu		
Kịch bản chính		Hành động của tác nhân	Hành động của hệ thống
	1	Chọn tính năng Translate PDF	Mở màn hình chọn file
	2	Chọn tài liệu PDF	Hiển thị màn hình chọn ngôn ngữ đầu vào và ngôn ngữ đích

	3	Chọn ngôn ngữ đầu vào và ngôn ngữ đích	Thực hiện tải xuống gói ngôn ngữ
	4	Nhấn nút tiếp tục	Chuyển đổi sang màn hình thể hiện quá trình dịch file, sau khi xong quá trình thì hiển thị màn hình thành công và tiến hành lưu file dưới dạng PDF

Bảng 2.24: Đặc tả chức năng AI Summarize PDF

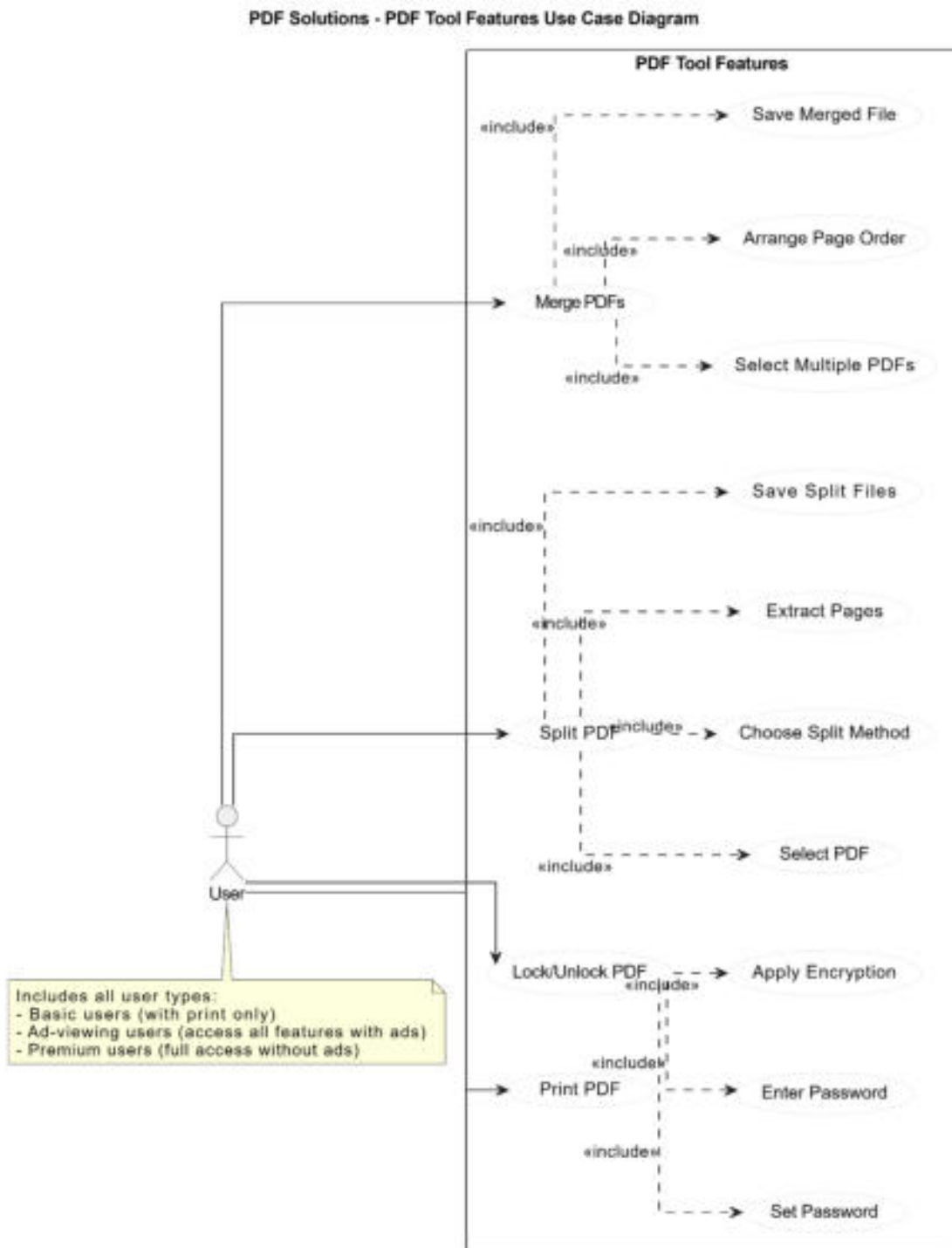
Mã ca sử dụng	UC-24		
Tên ca sử dụng	Summarize PDF		
Tác nhân	Người dùng		
Mô tả	Cho phép người dùng tóm tắt nội dung file PDF		
Điều kiện trước	Người dùng đã mở một tài liệu PDF chứa văn bản		
Điều kiện sau	Nội dung văn bản được tóm tắt và dịch sang ngôn ngữ mục tiêu		
Kịch bản chính		Hành động của tác nhân	Hành động của hệ thống
	1	Chọn tính năng Summarize PDF	Mở màn hình chọn file
	2	Chọn tài liệu PDF	Hiển thị màn hình chọn ngôn ngữ đích
	3	Chọn ngôn ngữ đích và nhấn nút continue	Thực hiện quá trình tóm tắt file PDF, sau khi hoàn thành

		quá trình thì tạo mới một file PDF chứa nội dung tóm tắt
--	--	--

Bảng 2.25: Đặc tả chức năng AI PDF HandWriting recognition

Mã ca sử dụng	UC-25		
Tên ca sử dụng	PDF HandWriting recognition		
Tác nhân	Người dùng		
Mô tả	Cho phép người dùng chuyển từ chữ viết tay sang văn bản số hóa		
Điều kiện trước	Người dùng đã quét tài liệu chứa chữ viết tay hoặc mở hình ảnh chứa chữ viết tay.		
Điều kiện sau	Chữ viết tay được chuyển đổi thành văn bản số hóa		
Kịch bản chính		Hành động của tác nhân	Hành động của hệ thống
	1	Chọn tính năng HandWriting recognition	Mở màn hình chọn file ảnh và nút quét tài liệu
	2	Chọn danh sách hình ảnh hoặc quét tài liệu để lấy danh sách ảnh, các ảnh có sẽ tự động xác định biên của tài liệu hoặc tùy chỉnh biên thủ công, sau đó nhấn nút done	Hiển thị quá trình xử lý, sau khi xử lý thành công thì di chuyển đến màn hình thành công và tạo mới một file PDF chứa văn bản số hóa

e) Sơ đồ ca sử dụng và đặc tả của các chức năng PDF Tools



Hình 2.7: Sơ đồ Usecase của các chức năng PDF Tools

Bảng 2.26 Đặc tả chức năng Merge PDFs

Mã ca sử dụng	UC-26		
Tên ca sử dụng	Merge PDFs		
Tác nhân	Người dùng		
Mô tả	Cho phép người dùng kết hợp nhiều tệp PDF thành một tệp PDF duy nhất		
Điều kiện trước	Người dùng đã cài đặt ứng dụng và có ít nhất hai tệp PDF để kết hợp.		
Điều kiện sau	Các tệp PDF được kết hợp thành một tệp duy nhất		
Kịch bản chính		Hành động của tác nhân	Hành động của hệ thống
	1	Chọn tính năng "Merge PDFs"	Hiển thị giao diện chọn tệp
	2	Chọn các tệp PDF cần kết hợp	Hiển thị giao diện các file đã chọn
	3	Sắp xếp thứ tự các tệp	Cho phép sắp xếp thứ tự
	4	Xác nhận và lưu tệp kết quả	Kết hợp các file theo thứ tự đã chọn, lưu file PDF mới đã kết hợp

Bảng 2.27: Đặc tả chức năng Split PDF

Mã ca sử dụng	UC-27
Tên ca sử dụng	Split PDF

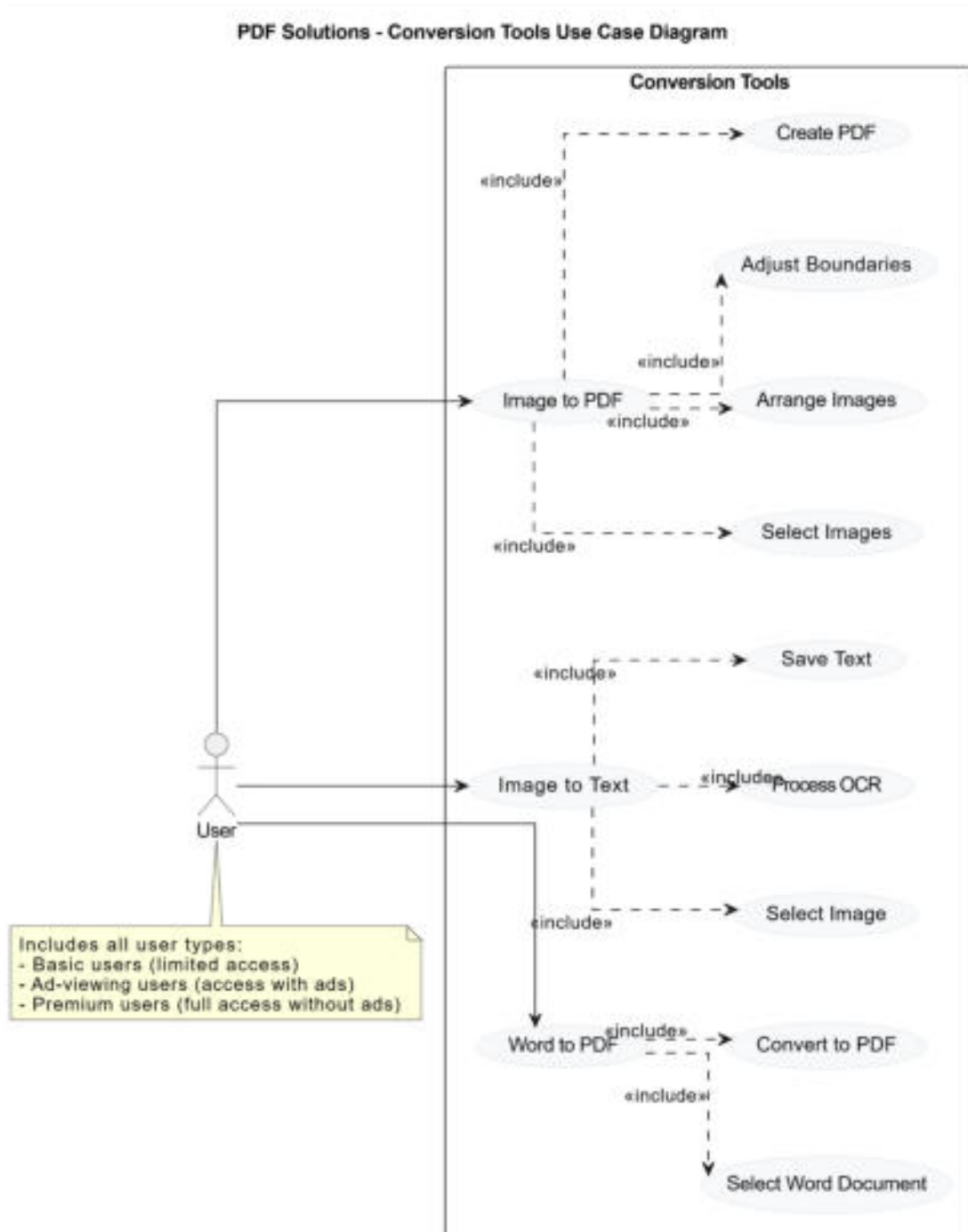
Tác nhân	Người dùng		
Mô tả	Cho phép người dùng phân tách 1 PDF làm nhiều tệp PDF		
Điều kiện trước	Người dùng đã cài đặt ứng dụng và có tệp PDF cần tách		
Điều kiện sau	Tệp PDF được tách thành các tệp nhỏ hơn theo yêu cầu		
Kịch bản chính		Hành động của tác nhân	Hành động của hệ thống
	1	Chọn tính năng "Split PDF"	Hiện thị giao diện chọn tệp
	2	Chọn tệp PDF cần tách	Hiện thị quá trình xử lý, sau khi xử lý thành công thì di chuyển đến màn hình thành công và tạo mới một file PDF chứa văn bản số hóa
	3	Chọn trang và chọn số file cần tách	Hiện thị giao diện các trang đã chọn và số file cần tách đã chọn
	4	Xác nhận và lưu các tệp kết quả	Tách tệp theo tùy chọn đã chọn, thực hiện lưu các tệp đã tách.

Bảng 2.28: Đặc tả chức năng Lock/Unlock PDF

Mã ca sử dụng	UC-28
Tên ca sử dụng	Lock/Unlock PDF
Tác nhân	Người dùng

Mô tả	Cho phép người dùng thêm hoặc gỡ bỏ bảo vệ bằng mật khẩu cho tệp PDF		
Điều kiện trước	Người dùng đã cài đặt ứng dụng và có tệp PDF cần bảo vệ hoặc gỡ bỏ bảo vệ		
Điều kiện sau	Tệp PDF được bảo vệ bằng mật khẩu hoặc được gỡ bỏ bảo vệ		
Kịch bản chính		Hành động của tác nhân	Hành động của hệ thống
	1	Chọn tính năng "Lock/Unlock PDF"	Hiển thị giao diện chọn tệp
	2	Chọn tệp PDF	Phát hiện trạng thái bảo vệ của tệp, hiển thị tùy chọn tương ứng khóa hoặc mở khóa
	3	Nhập mật khẩu để thực hiện khóa hoặc mở khóa PDF	Áp dụng hoặc gỡ bỏ bảo vệ theo yêu cầu
	4	Xác nhận và lưu tệp	Lưu tệp

f) Sơ đồ ca sử dụng và đặc tả của các chức năng PDF Conversion Tools



Hình 2.8: Sơ đồ Usecase của các chức năng PDF Conversion Tools

Bảng 2.29: Đặc tả chức năng Image to Text

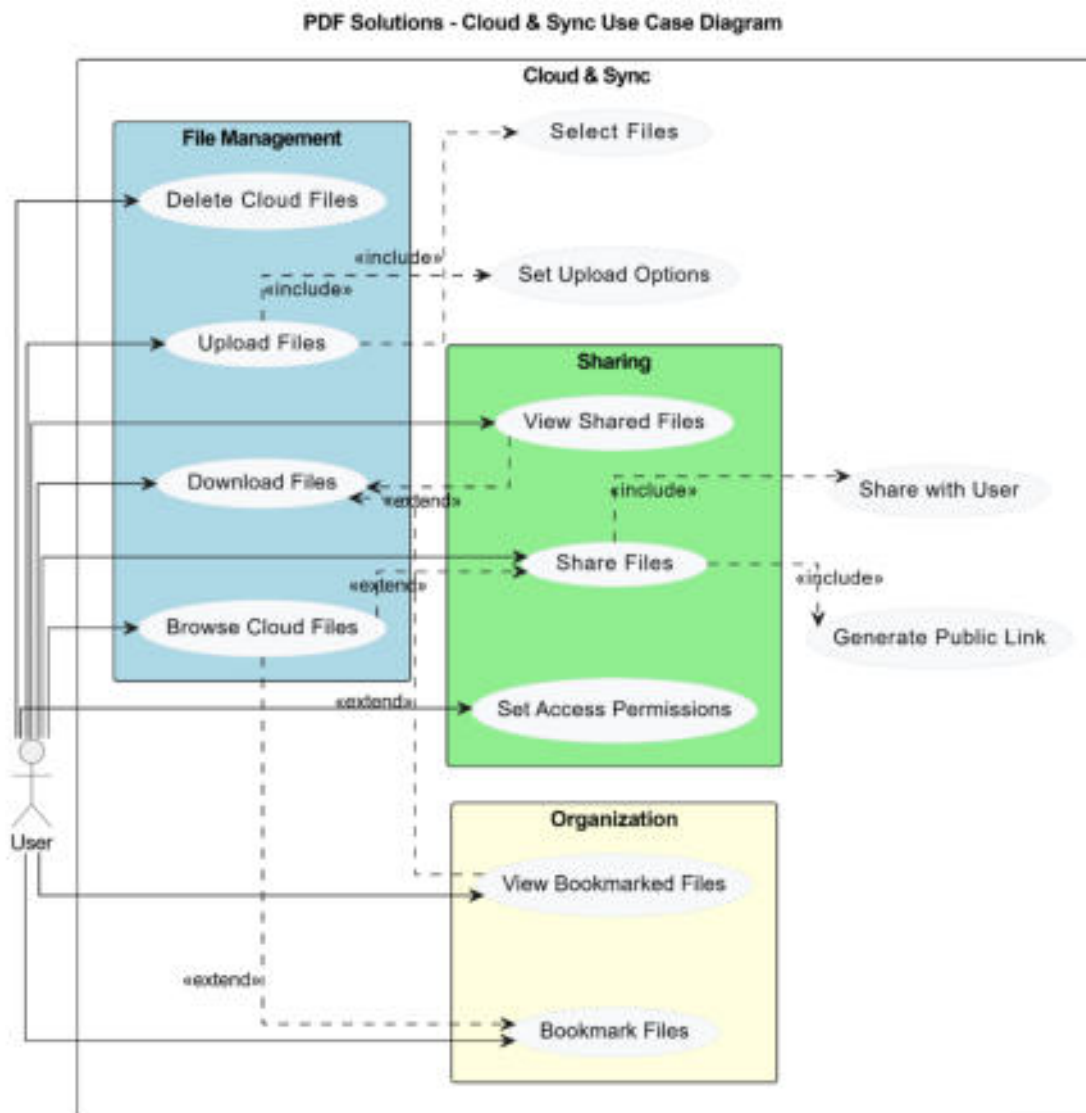
Mã ca sử dụng	UC-29		
Tên ca sử dụng	Image to Text		
Tác nhân	Người dùng		
Mô tả	Cho phép người dùng trích xuất văn bản từ hình ảnh sử dụng công nghệ OCR		
Điều kiện trước	Người dùng đã cài đặt ứng dụng và có hình ảnh chứa văn bản		
Điều kiện sau	Văn bản được trích xuất từ hình ảnh		
Kịch bản chính		Hành động của tác nhân	Hành động của hệ thống
	1	Chọn tính năng "Image to Text"	Hiển thị giao diện chọn hình ảnh
	2	Chọn hình ảnh chứa văn bản hoặc sử dụng camera để chụp	Hiển thị hình ảnh được chọn
	3	Xác nhận trích xuất văn bản	Xử lý hình ảnh bằng công nghệ OCR
	4	Xác nhận và lưu tệp	Lưu văn bản vào file PDF mới

Bảng 2.30: Đặc tả chức năng Word to PDF

Mã ca sử dụng	UC-30
Tên ca sử dụng	Word to PDF
Tác nhân	Người dùng

Mô tả	Cho phép người dùng chuyển đổi tài liệu Word thành tệp PDF		
Điều kiện trước	Người dùng đã cài đặt ứng dụng và có tài liệu Word cần chuyển đổi		
Điều kiện sau	Tài liệu Word được chuyển đổi thành tệp PDF		
Kịch bản chính		Hành động của tác nhân	Hành động của hệ thống
	1	Chọn tính năng "Word to PDF"	Hiện thị giao diện chọn tài liệu
	2	Chọn tài liệu Word cần được chuyển đổi	Thực hiện chuyển đổi tài liệu Word thành PDF, duy trì định dạng, phông chữ và bố cục
	3	Xác nhận chuyển đổi	Lưu tệp PDF

f) Sơ đồ ca sử dụng và đặc tả của các chức năng Cloud & Sync



Hình 2.9: Sơ đồ Usecase của các chức năng Cloud & Sync

Bảng 2.31 Đặc tả chức năng Browse Cloud Files

Mã ca sử dụng	UC-31
Tên ca sử dụng	Browse Cloud Files
Tác nhân	Người dùng

Mô tả	Cho phép người dùng xem danh sách các tệp PDF đã lưu trữ trên đám mây		
Điều kiện trước	Người dùng đã đăng nhập vào hệ thống		
Điều kiện sau	Người dùng lấy được danh sách file đã tải lên		
Kịch bản chính		Hành động của tác nhân	Hành động của hệ thống
	1	Người dùng truy cập màn hình Cloud	Hiển thị giao diện Cloud
	2	Người dùng chọn "My Files"	Thực hiện gọi API và hiển thị danh sách file người dùng đã tải lên

Bảng 2.32 Đặc tả chức năng Upload Files

Mã ca sử dụng	UC-32		
Tên ca sử dụng	Upload Files		
Tác nhân	Người dùng		
Mô tả	Cho phép người dùng chọn tệp từ thiết bị và tải lên đám mây		
Điều kiện trước	Người dùng đã đăng nhập vào hệ thống		
Điều kiện sau	Người dùng lấy được danh sách file đã tải lên		
Kịch bản chính		Hành động của tác nhân	Hành động của hệ thống
	1	Người dùng truy cập màn hình Cloud	Hiển thị giao diện Cloud
	2	Người dùng chọn "Upload File"	Hệ thống hiển thị giao diện cho phép chọn file trong thiết bị

	3	Người dùng chọn tệp cần tải lên, ấn xác nhận	Hiển thị giao diện nhập thông tin
	4	Người dùng nhập tiêu đề, mô tả và chọn chế độ chia sẻ và xác nhận tải lên	Gọi API để tải file lên Cloud

Bảng 2.33 Đặc tả chức năng Download Files

Mã ca sử dụng	UC-33		
Tên ca sử dụng	Upload Files		
Tác nhân	Người dùng		
Mô tả	Cho phép người dùng tải tệp từ Cloud xuống thiết bị		
Điều kiện trước	Người dùng đã đăng nhập vào hệ thống, người dùng tải file do chính bản thân tải lên Cloud, người dùng được chia sẻ link tải file public, hoặc được người upload chia sẻ chỉ định với người tải		
Điều kiện sau	Người dùng tải thành công file xuống thiết bị		
Kịch bản chính		Hành động của tác nhân	Hành động của hệ thống
	1	Người dùng truy cập màn hình Cloud	Hiển thị giao diện Cloud
	2	Người dùng chọn "My File, Shared hoặc Bookmark"	Hệ thống hiển thị giao diện danh sách file
	3	Người dùng chọn tệp cần tải xuống, đối với public link thì nhập link lên trang web, đối với file tự upload	Gọi API để tải file từ Cloud xuống thiết bị

	hoặc file được share chỉ định thì nhấn nút tải xuống
--	--

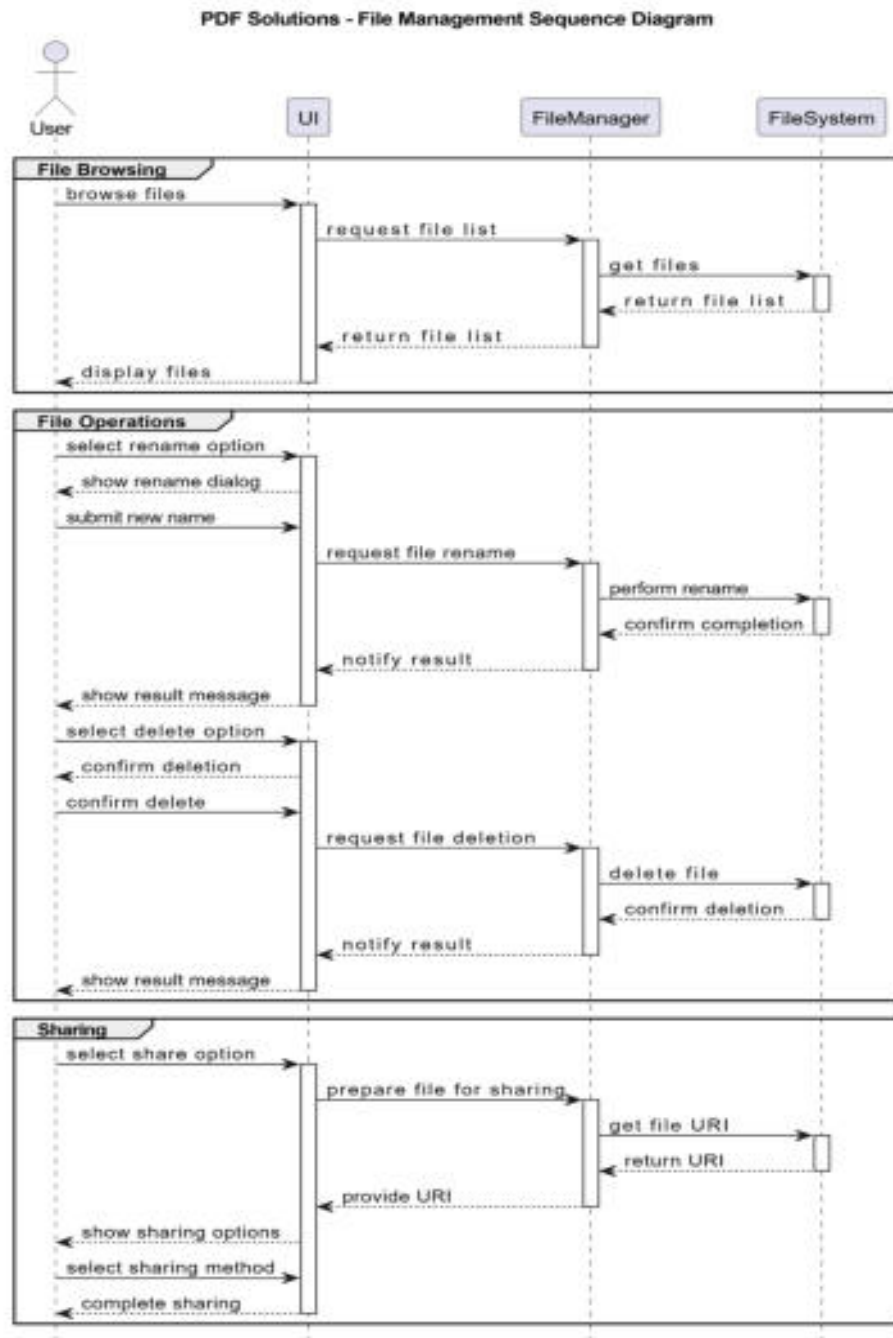
Bảng 2.34 Đặc tả chức năng Share File

Mã ca sử dụng	UC-34		
Tên ca sử dụng	Share File		
Tác nhân	Người dùng		
Mô tả	Cho phép người dùng chia sẻ tệp với người khác		
Điều kiện trước	Người dùng đã đăng nhập và có ít nhất một tệp trên đám mây		
Điều kiện sau	Người dùng chia sẻ thành công file		
Kịch bản chính		Hành động của tác nhân	Hành động của hệ thống
	1	Người dùng truy cập màn hình Cloud	Hiển thị giao diện Cloud
	2	Người dùng chọn "My File "	Hệ thống hiển thị giao diện danh sách file
	3	Người dùng chọn tệp cần chia sẻ, đổi trạng thái quyền thành public nếu muốn chia sẻ công khai bằng link, trạng thái user specific nếu muốn chia sẻ với người chỉ định	Hệ thống gọi API để cập nhật trạng thái chia sẻ của file
	5	Đối với public, người dùng nhấn nút generate link và thực hiện copy link rồi gửi cho người cần chia sẻ, đối với chế độ user specific thì người	Gọi API để tạo link public, hoặc cập nhật người được chia sẻ

		upload nhập email của người được chia sẻ và nhấn nút send	
--	--	--	--

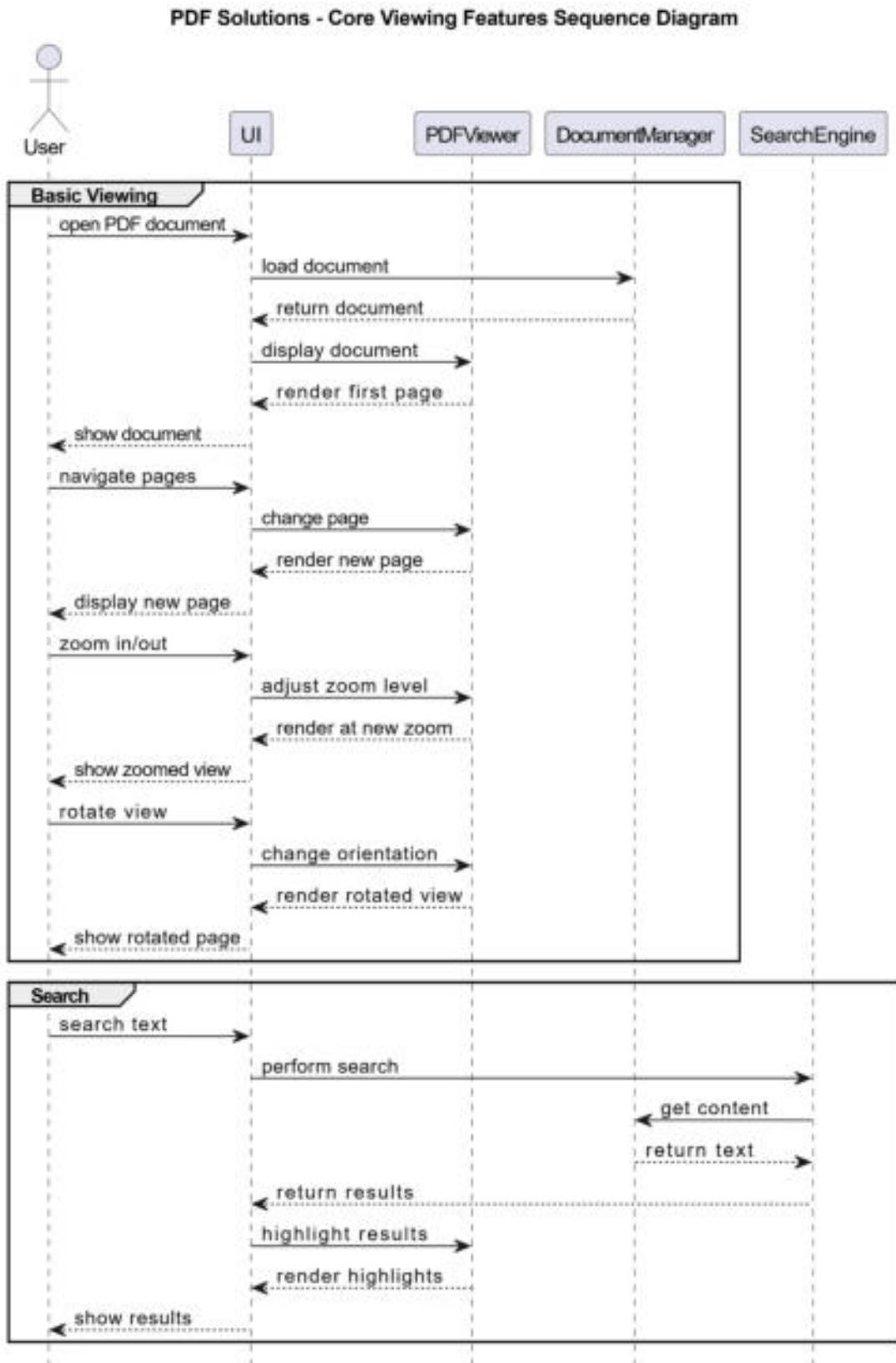
2.2.3. Sơ đồ tuần tự

a) Sơ đồ tuần tự quản lý File

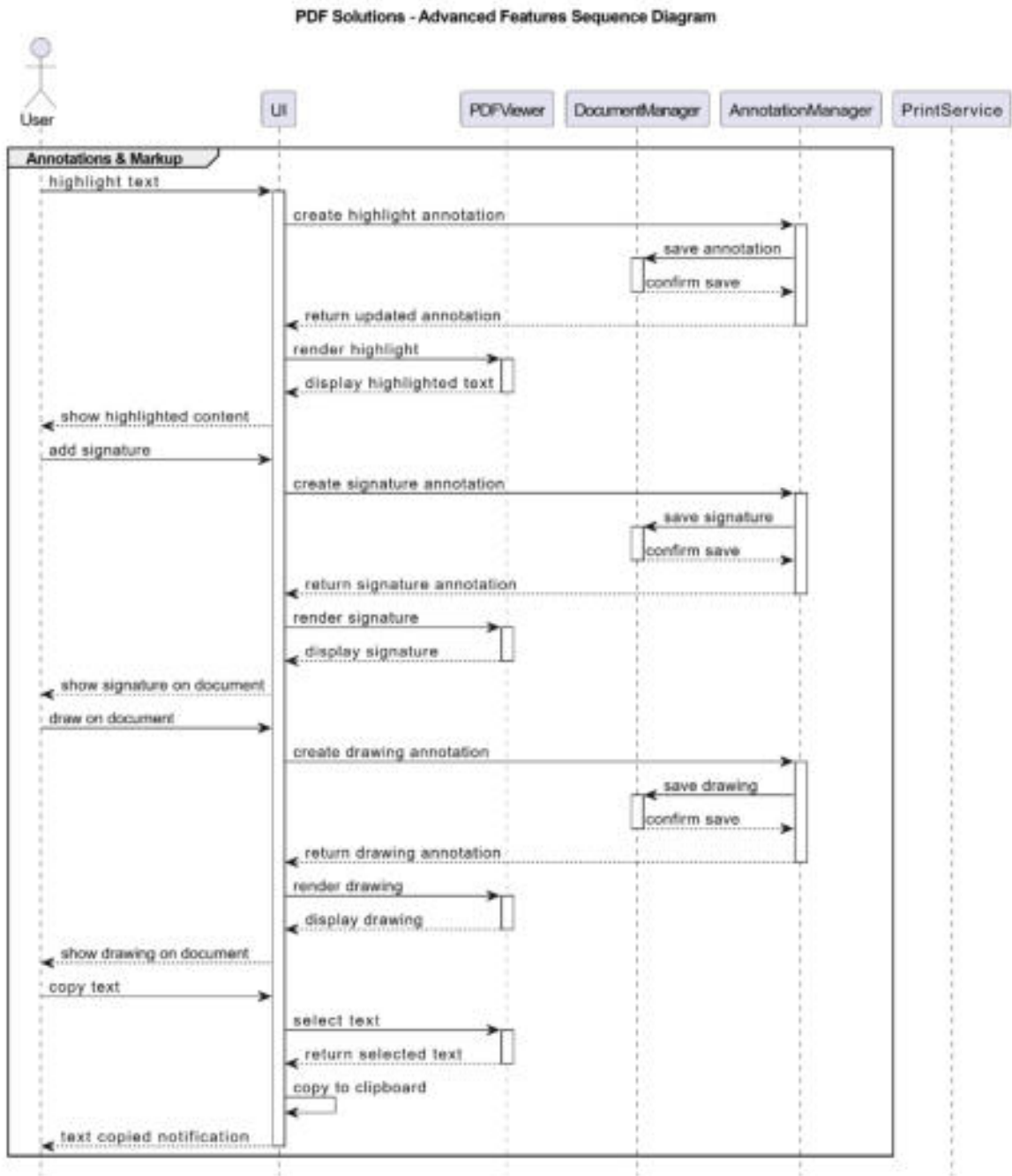


Hình 2.10: Sơ đồ tuần tự quản lý File

b) Sơ tuần tự xem và chỉnh sửa file PDF

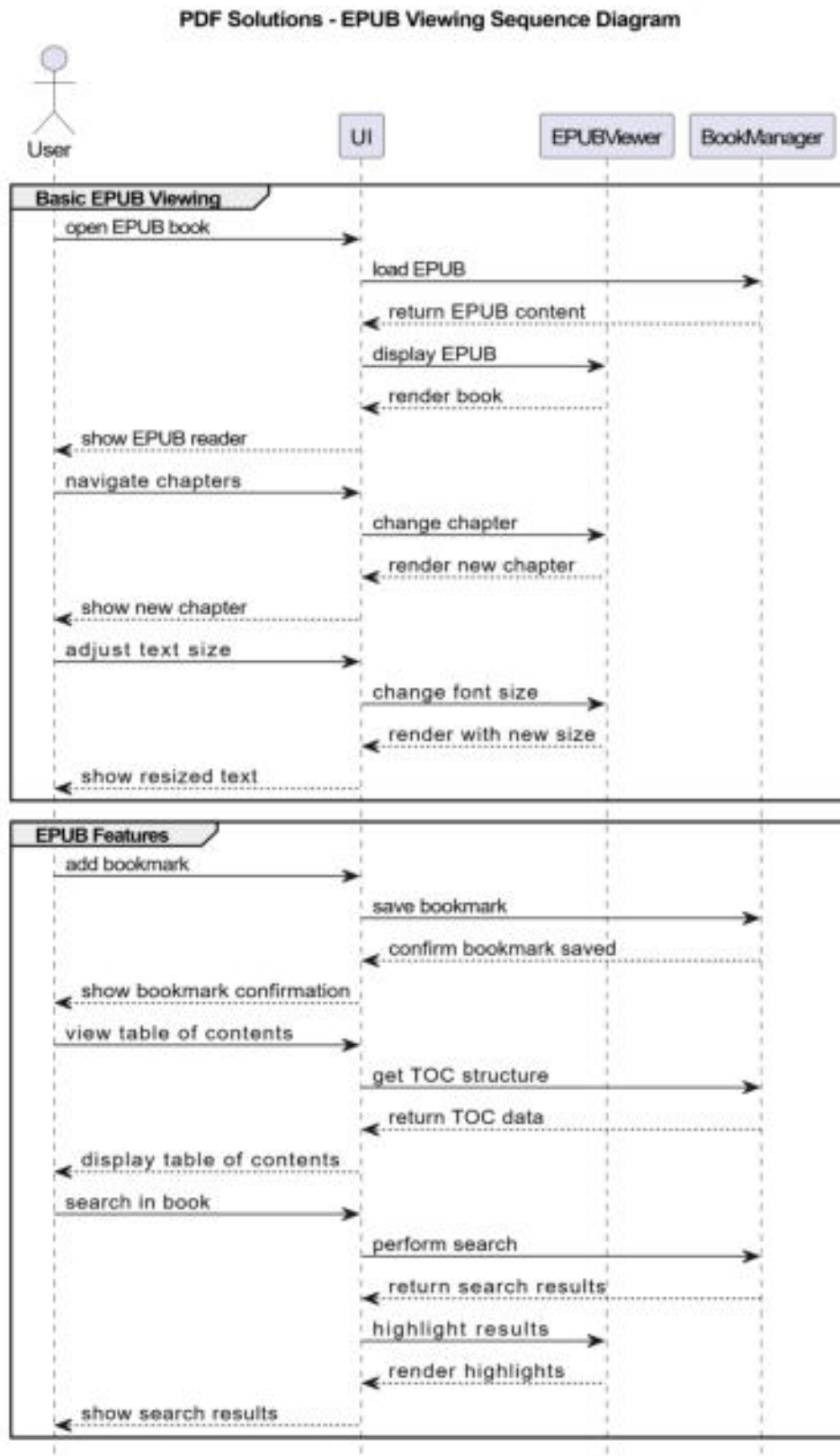


Hình 2.11: Sơ đồ tuần tự xem file PDF cơ bản

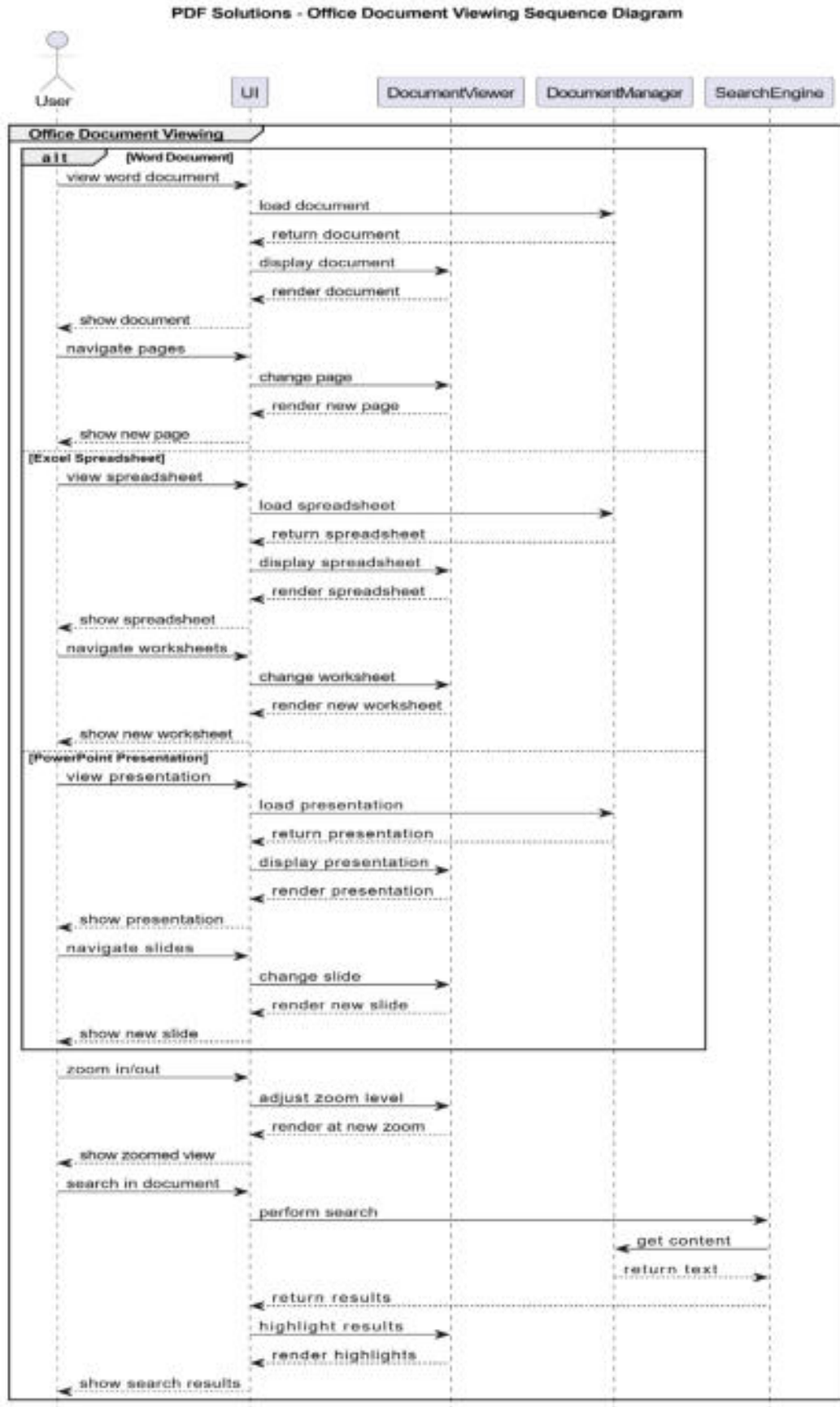


Hình 2.12: Sơ đồ tuần tự chỉnh sửa file PDF

c) Sơ tuần tự xem file Office, TXT, EPUB

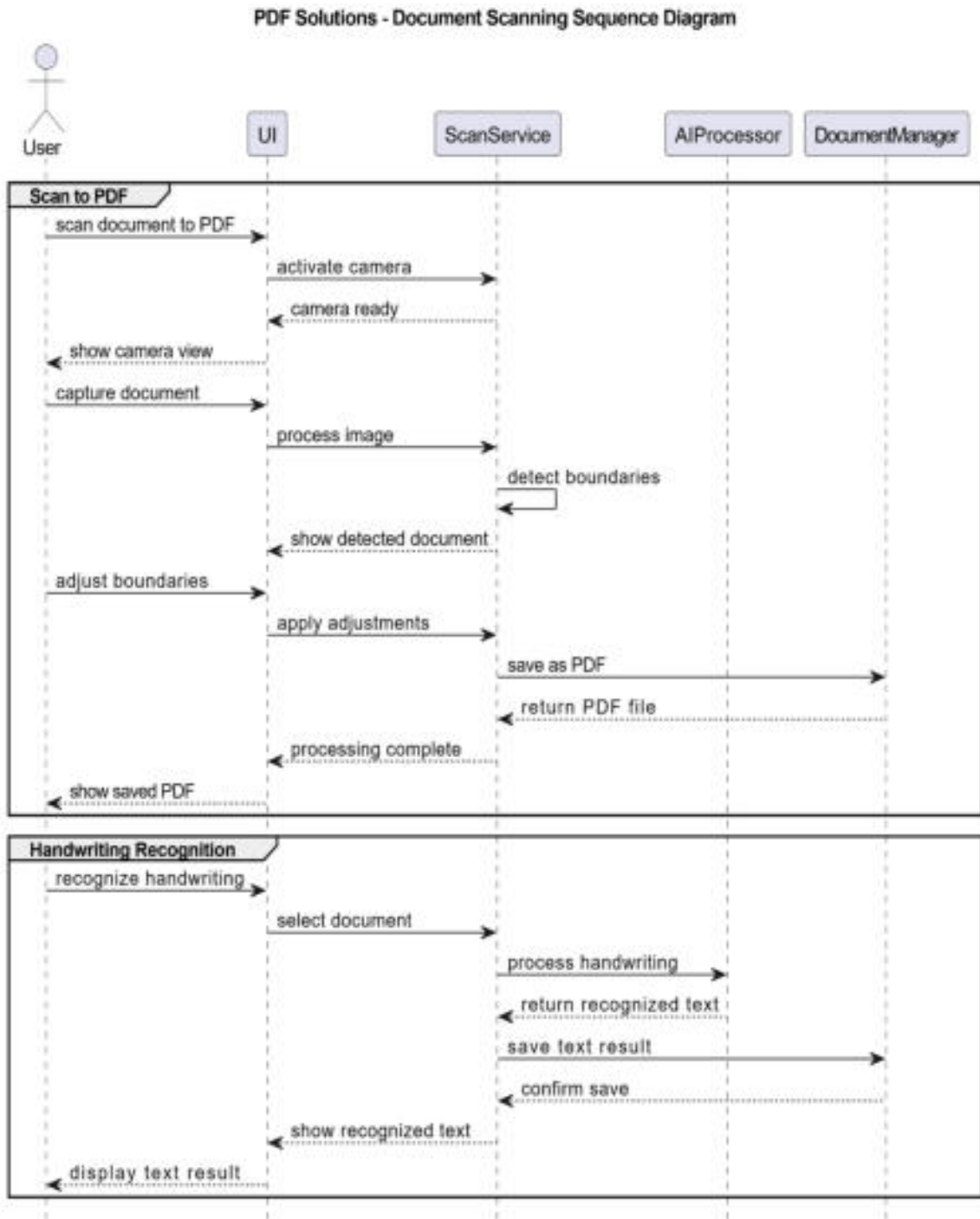


Hình 2.13: Sơ đồ tuần tự xem file EPUB

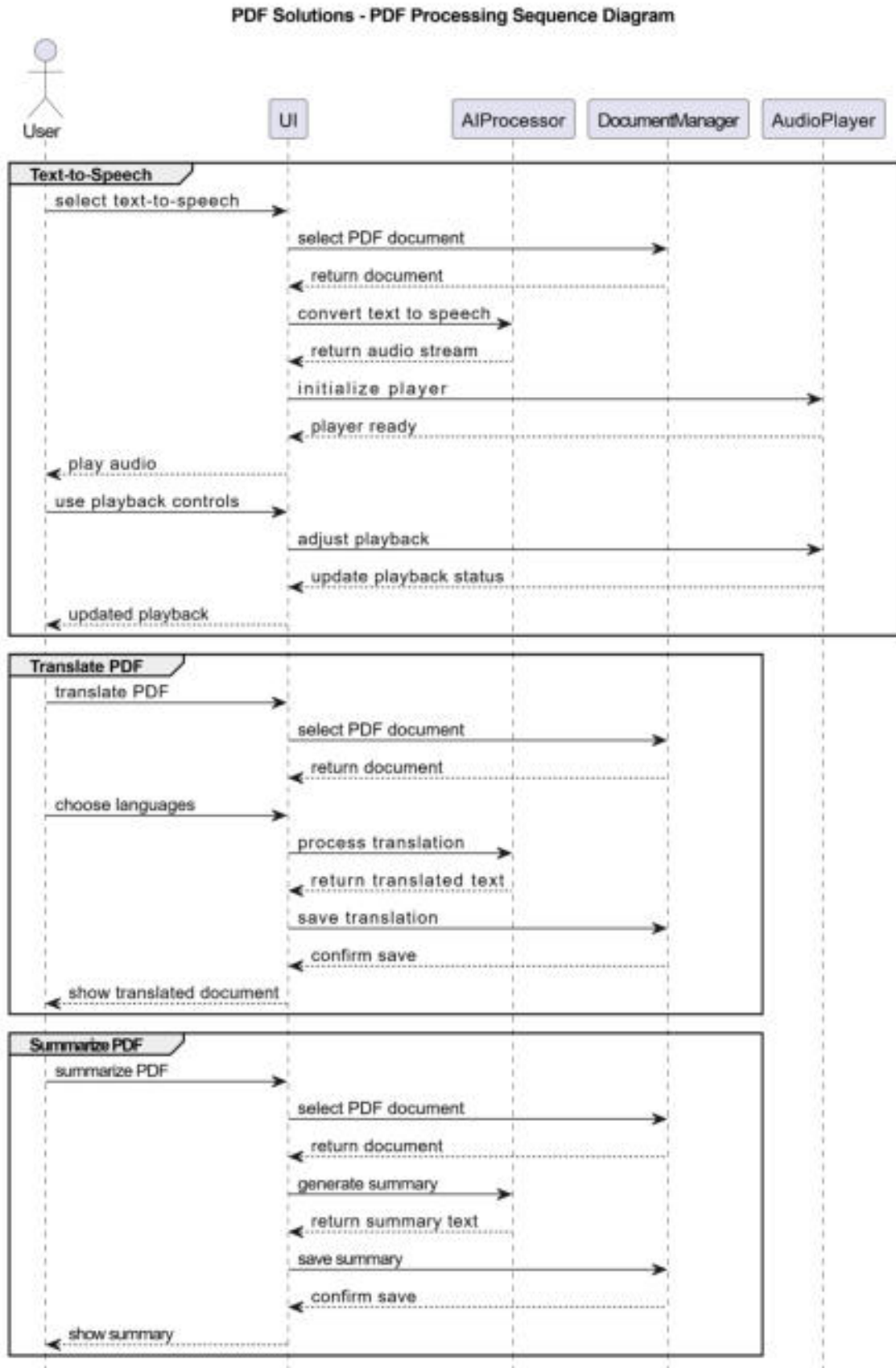


Hình 2.14: Sơ đồ tuần tự xem file Office

c) Sơ tuần tự các chức năng AI

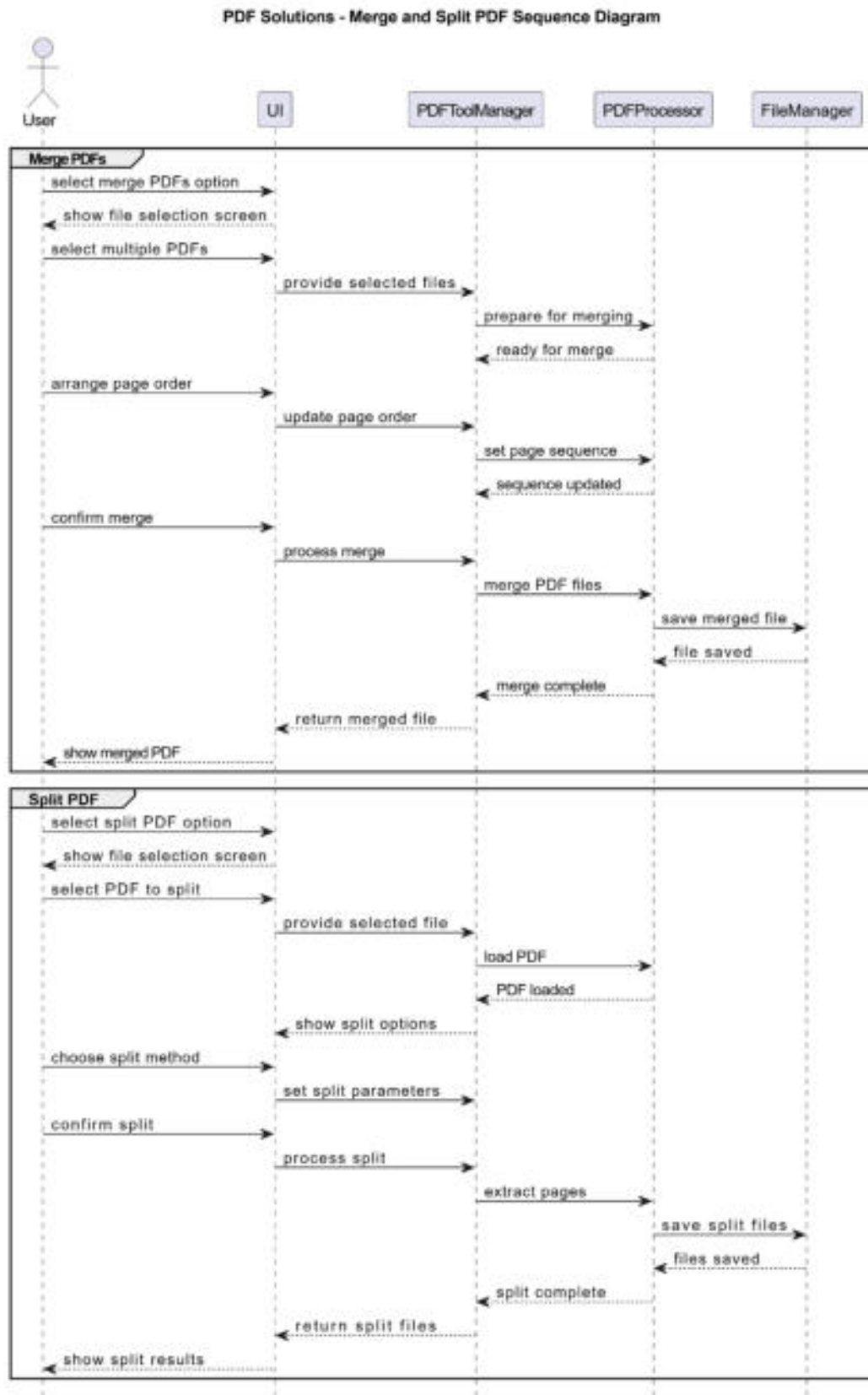


Hình 2.15: Sơ đồ tuần tự chức năng Scan to PDF và Handwriting Recognition

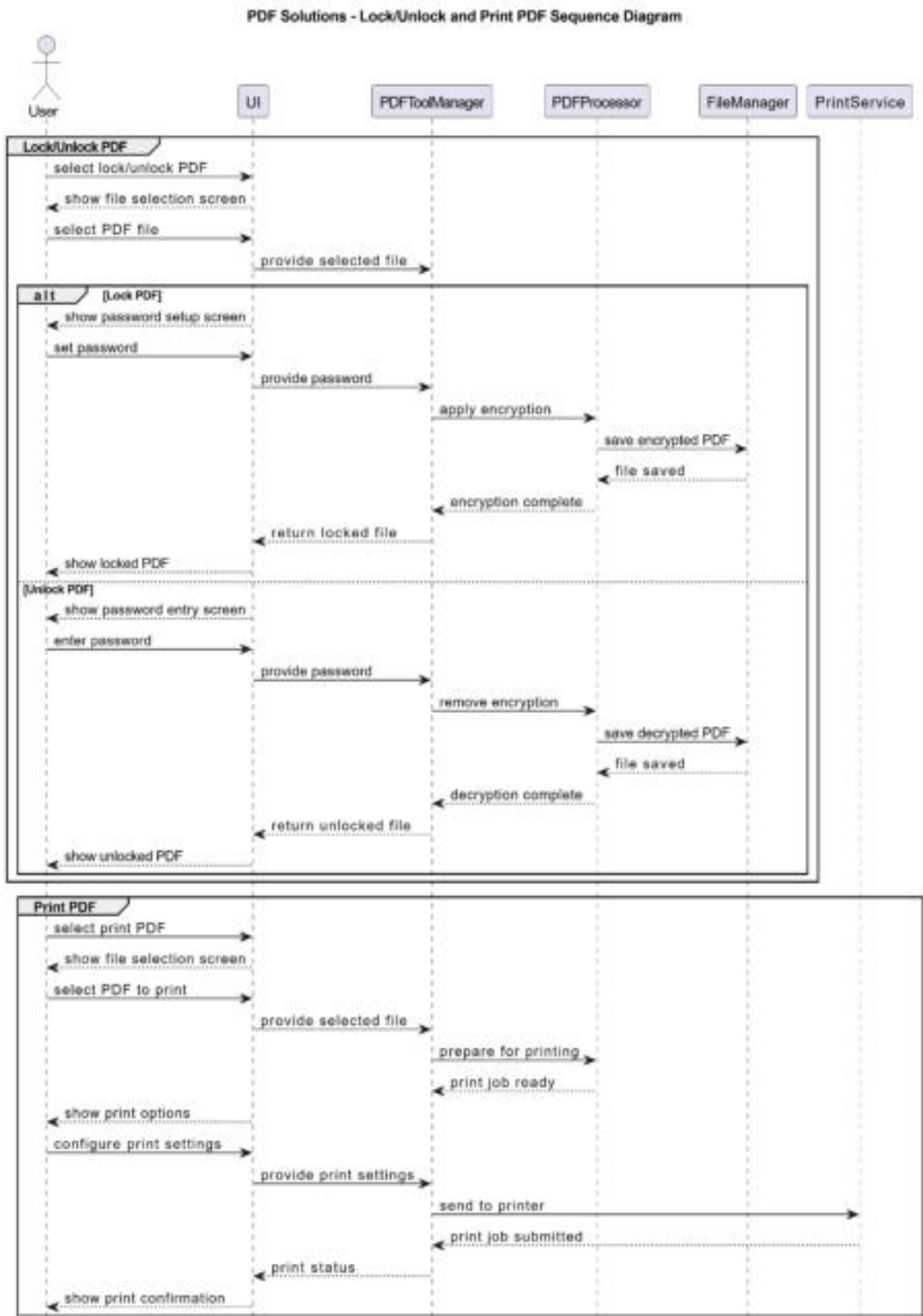


Hình 2.16: Sơ đồ tuần tự chức năng Text to Speech, Translate, Summarize

d) Sơ tuần tự các chức năng Tools PDF

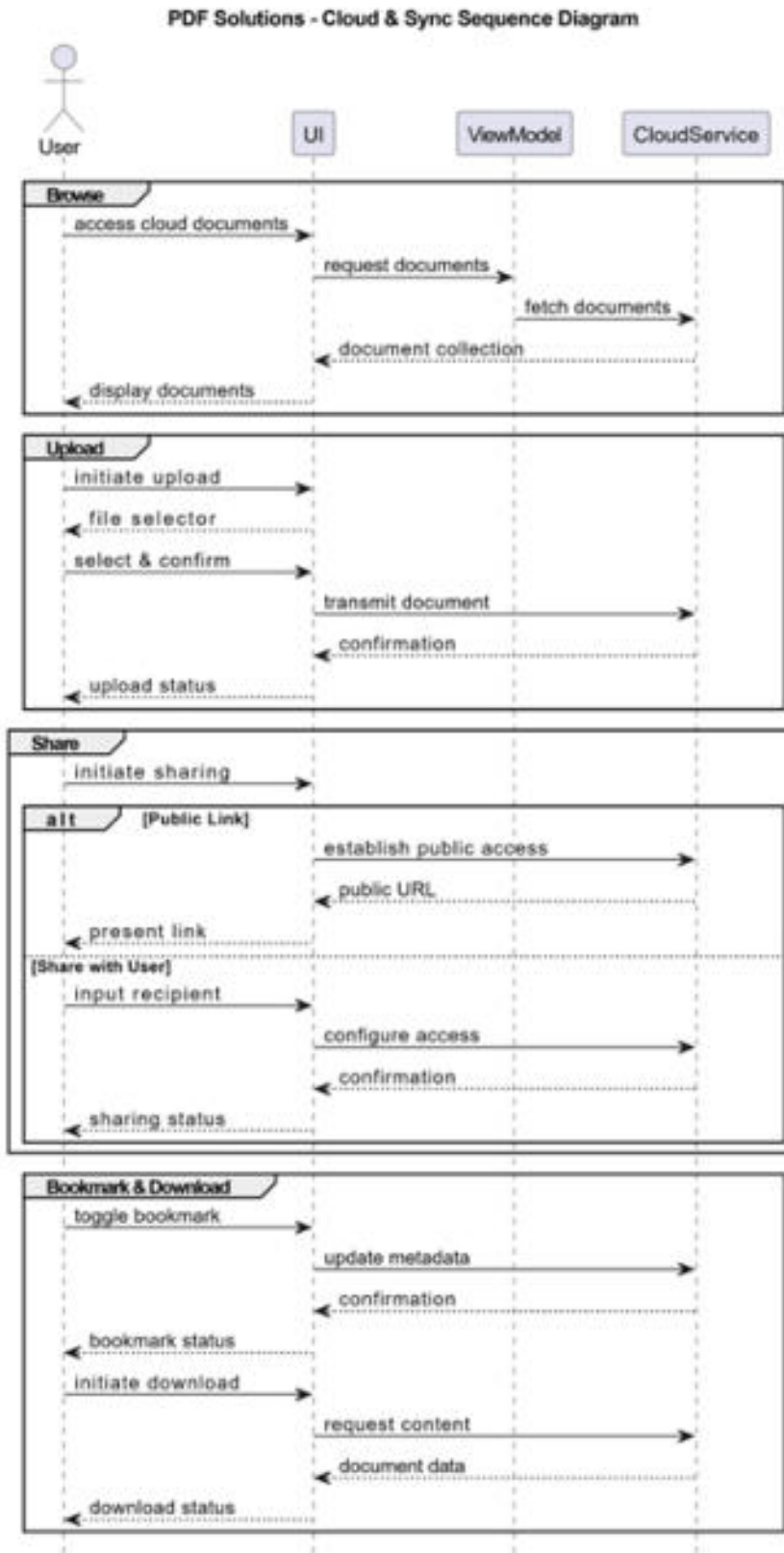


Hình 2.17: Sơ đồ tuần tự chức năng Merge và Split PDF



Hình 2.18: Sơ đồ tuần tự chức năng Lock/Unlock, print PDF

e) Sơ tuần tự các chức năng Cloud & Sync

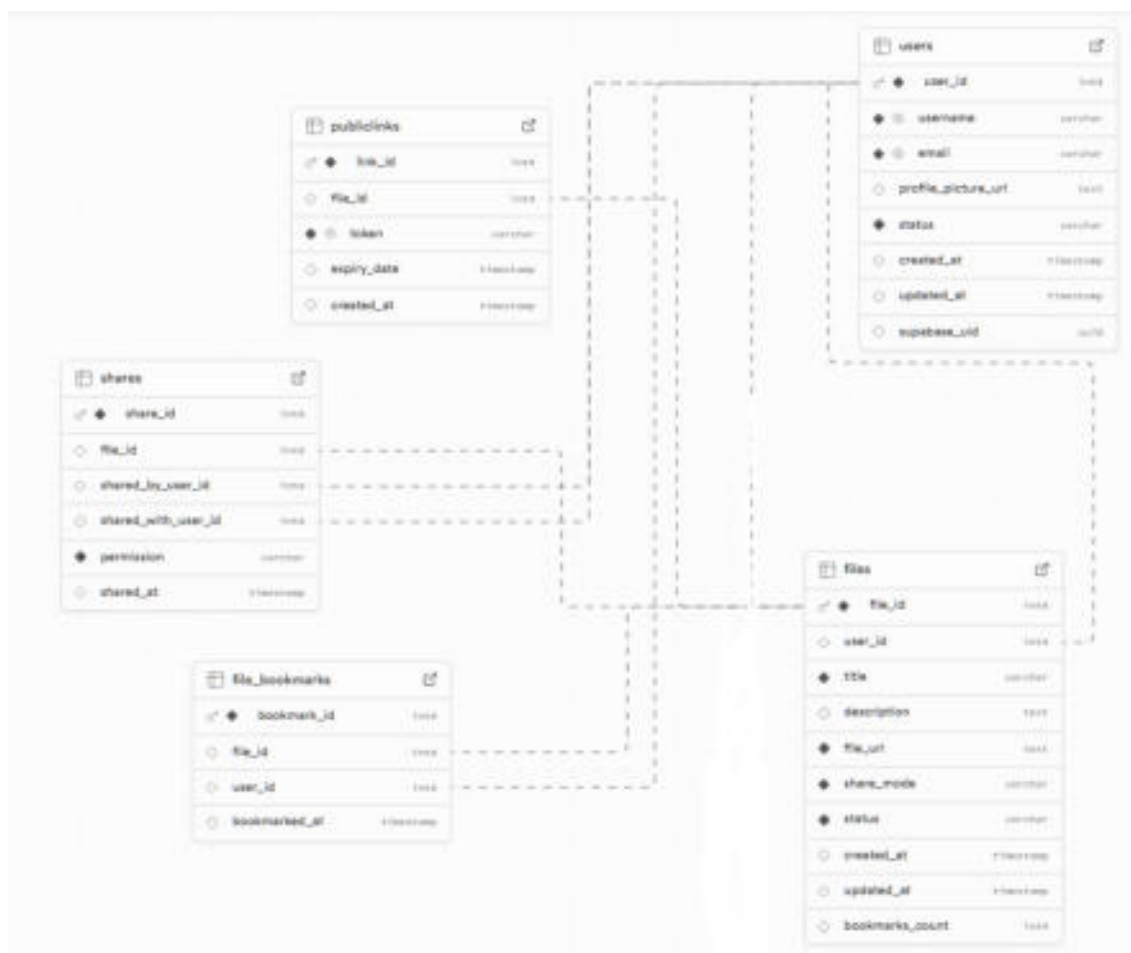


Hình 2.19: Sơ đồ tuần tự chức năng Cloud & Sync

2.2.4. Cơ sở dữ liệu

a) Cơ sở dữ liệu supabase

Cơ sở dữ liệu của ứng dụng quản lý tài liệu PDF được thiết kế với mục tiêu tạo nền tảng vững chắc cho việc lưu trữ, chia sẻ và tương tác với tài liệu PDF. Kiến trúc cơ sở dữ liệu tuân theo các nguyên tắc thiết kế quan hệ, đảm bảo tính toàn vẹn dữ liệu và hiệu suất truy vấn tối ưu. Hệ thống bao gồm 5 bảng chính như hình ảnh phía dưới, mỗi bảng đảm nhận một chức năng cụ thể trong hệ sinh thái quản lý tài liệu.



Hình 2.20: Hình ảnh mô tả cơ sở dữ liệu

Cơ sở dữ liệu sử dụng chỉ mục (Indexes) để có thể triển khai một hệ thống toàn diện và tối ưu hóa hiệu suất truy vấn, sử dụng Trigger để tự động cập nhật chỉ số ở các bảng

Mối quan hệ của các bảng:

Mối quan hệ Users – Files: Mối quan hệ một-nhiều giữa Users và Files thể hiện rằng một người dùng có thể sở hữu nhiều tài liệu, nhưng mỗi tài liệu chỉ thuộc về một người dùng. Mối quan hệ này được thiết lập thông qua khóa ngoại user_id trong bảng Files, với ràng buộc ON DELETE CASCADE đảm bảo khi một người dùng bị xóa, tất cả tài liệu của họ cũng sẽ bị xóa.

Mối quan hệ Files – Shares: Mối quan hệ một-nhiều giữa Files và Shares thể hiện rằng một tài liệu có thể được chia sẻ với nhiều người dùng khác nhau. Mối quan hệ này được thiết lập thông qua khóa ngoại file_id trong bảng Shares, với ràng buộc ON DELETE CASCADE đảm bảo khi một tài liệu bị xóa, tất cả bản ghi chia sẻ liên quan cũng sẽ bị xóa.

Mối quan hệ Files – PublicLinks: Mối quan hệ một-nhiều giữa Files và PublicLinks thể hiện rằng một tài liệu có thể có nhiều liên kết công khai khác nhau (ví dụ: với các thời hạn khác nhau). Mối quan hệ này được thiết lập thông qua khóa ngoại file_id trong bảng PublicLinks, với ràng buộc ON DELETE CASCADE đảm bảo khi một tài liệu bị xóa, tất cả liên kết công khai liên quan cũng sẽ bị xóa.

Mối quan hệ Users - File_Bookmarks - Files: Mối quan hệ nhiều-nhiều giữa Users và Files thông qua bảng trung gian File_Bookmarks thể hiện rằng một người dùng có thể đánh dấu nhiều tài liệu, và một tài liệu có thể được đánh dấu bởi nhiều người dùng. Mối quan hệ này được thiết lập thông qua các khóa ngoại user_id và file_id trong bảng File_Bookmarks, với các ràng buộc ON DELETE CASCADE đảm bảo tính toàn vẹn dữ liệu.

b) Thiết lập storage policies và bucket trên supabase

Hệ thống lưu trữ và chia sẻ tài liệu PDF được triển khai trên nền tảng Supabase sử dụng bucket pdffiles làm kho lưu trữ trung tâm. Kiến trúc bảo mật của hệ thống được thiết kế dựa trên mô hình kiểm soát truy cập chi tiết thông qua các chính sách (policies) được định nghĩa rõ ràng, đảm bảo tính toàn vẹn dữ liệu và quyền riêng tư của người dùng.



Hình 2.21: Hình minh họa cho Storage Policies

Các chính sách bảo mật được thiết lập trên bucket pdffiles bao gồm:

- **Chính sách truy cập công khai:** Cho phép truy cập không xác thực đến các tài liệu PDF được đánh dấu với chế độ chia sẻ `public_link`. Hệ thống xác thực điều kiện này bằng cách kiểm tra sự tồn tại của bản ghi trong bảng files với thuộc tính `share_mode` tương ứng.
- **Chính sách chia sẻ có kiểm soát:** Người dùng được chia sẻ tài liệu có thể xem tài liệu đó thông qua cơ chế liên kết nhiều bảng. Hệ thống xác thực mối quan hệ giữa người dùng hiện tại qua `user_id` với bảng shares và files để đảm bảo quyền truy cập hợp lệ.
- **Chính sách xóa tài liệu:** Chỉ chủ sở hữu tệp mới có quyền xóa tài liệu của họ. Điều này được thực hiện bằng cách xác thực mối quan hệ giữa người dùng hiện tại với tài liệu.
- **Chính sách tải lên tài liệu:** Mọi người dùng đã xác thực đều có quyền tải lên tài liệu PDF vào bucket pdffiles.
- **Chính sách xem tài liệu cá nhân:** Người dùng có quyền xem tất cả tài liệu do họ tải lên

2.2.5. Thiết kế kiến trúc hệ thống

a) Tổng quan kiến trúc MVVM và Clean Architecture trong ứng dụng

Để đảm bảo hệ thống phát triển có khả năng mở rộng, bảo trì dễ dàng và tối ưu hiệu suất, đề án lựa chọn áp dụng kiến trúc Clean Architecture kết hợp với mô hình MVVM (Model - View - ViewModel) – một kiến trúc phổ biến trong phát triển ứng dụng Android hiện đại.

Kiến trúc này cho phép phân tách rõ ràng các thành phần của ứng dụng thành nhiều lớp độc lập, giúp nâng cao khả năng kiểm thử, giảm thiểu sự phụ thuộc giữa các thành phần, đồng thời tăng khả năng tái sử dụng và phát triển linh hoạt trong tương lai.

Hệ thống được chia thành ba tầng chính:

Tầng Presentation (Giao diện người dùng)

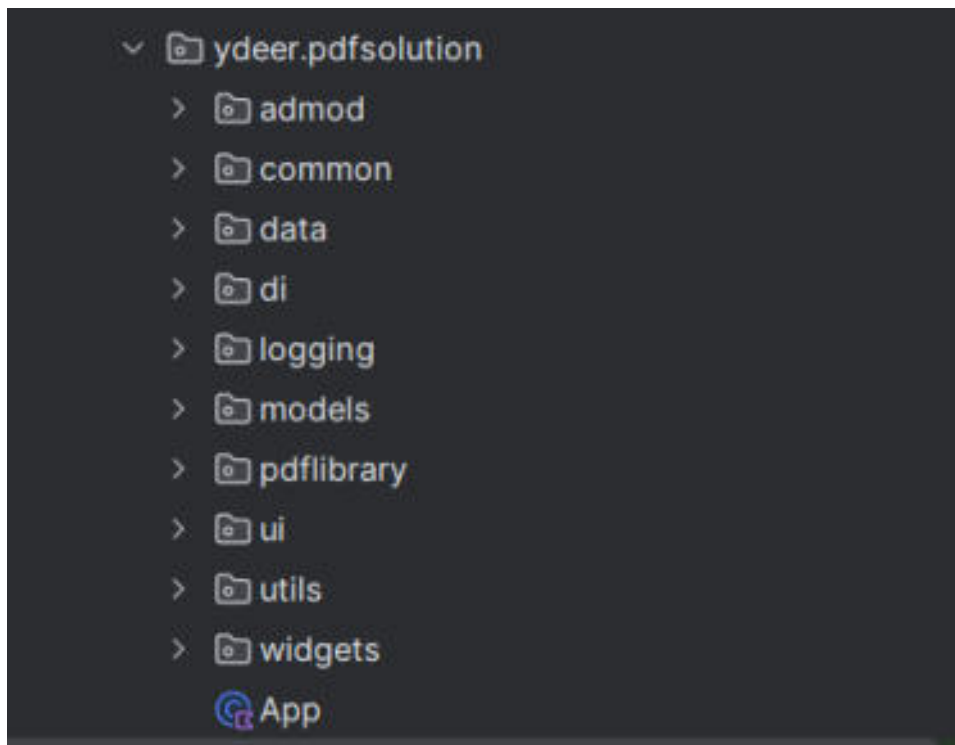
- Chịu trách nhiệm hiển thị dữ liệu và xử lý tương tác của người dùng.
- Sử dụng Jetpack Compose để xây dựng UI phản ứng theo trạng thái.
- ViewModel đóng vai trò trung gian giữa giao diện và tầng nghiệp vụ (Domain), giúp xử lý logic giao diện, quản lý trạng thái (State) và điều phối dữ liệu.
- ViewModel sử dụng StateFlow / LiveData để đảm bảo cập nhật giao diện theo thời gian thực.

Tầng Domain (Nghiệp vụ)

- Chứa các logic nghiệp vụ cốt lõi (business logic) độc lập với các thành phần Android.
- Bao gồm các UseCase đại diện cho từng hành động cụ thể (ví dụ: tải danh sách tệp, chuyển đổi tệp, đồng bộ dữ liệu...).
- Các thành phần ở tầng này chỉ tương tác với các interface (contract) được định nghĩa bởi tầng Data, giúp đảm bảo tính tách biệt và kiểm thử đơn giản.

Tầng Data (Xử lý và cung cấp dữ liệu)

- Chịu trách nhiệm lấy dữ liệu từ các nguồn khác nhau: API, cơ sở dữ liệu cục bộ (RoomDB), Supabase, Firebase...
- Bao gồm: Repository: cầu nối giữa tầng Domain và các nguồn dữ liệu, Datasource: triển khai chi tiết việc lấy dữ liệu từ local hoặc remote, Model chuyển đổi (Mapper): giúp chuyển đổi dữ liệu giữa các tầng (DTO ↔ Entity ↔ UI model)
- Sử dụng thư viện như Retrofit, Room, Firebase SDK, Supabase SDK để giao tiếp với các hệ thống bên ngoài.



Hình 2.22: Hình cấu trúc file thực tế

b) Luồng giữ liệu và giao tiếp giữa các tầng trong dự án

Tầng dữ liệu (Data Layer):

Repositories: Là điểm truy cập chính cho dữ liệu, triển khai các interface được định nghĩa ở tầng Domain.

- Điều phối dữ liệu từ nhiều nguồn (local, remote)
- Xử lý logic cache khi cần thiết
- Chuyển đổi dữ liệu từ dạng DTO sang các entities domain

Data Sources:

- Remote Data Source: Giao tiếp với API sử dụng Retrofit
- Local Data Source: Tương tác với cơ sở dữ liệu local (Room, SharedPreferences)
- File Data Source: Xử lý thao tác với file PDF và các định dạng khác

Tầng domain (Domain Layer):

Use Cases: Chứa logic nghiệp vụ cụ thể, mỗi use case thực hiện một nhiệm vụ riêng biệt:

- Chuyển đổi từ image sang PDF
- Xử lý chỉnh sửa PDF
- Tách và ghép PDF
- Bảo mật PDF (khóa/mở khóa)

Entities: Mô hình dữ liệu nghiệp vụ thuần túy, không phụ thuộc vào Android framework

- Định nghĩa các đối tượng như PDF, Tool, User

Repository Interfaces: Định nghĩa contract cho việc truy xuất dữ liệu

Tầng giao diện người dùng (Presentation Layer):

Composable:

- Hiển thị UI và thu thập tương tác người dùng
- Quan sát luồng dữ liệu từ ViewModel
- Điều hướng người dùng giữa các màn hình
- Thực hiện nhận các thay đổi dữ liệu để hiển thị UI phù hợp

CHƯƠNG 3. TRIỂN KHAI HỆ THỐNG VÀ ĐÁNH GIÁ

3.1 Môi trường và kỹ thuật phát triển

3.1.1. Thiết bị sử dụng

Để đảm bảo quá trình triển khai hệ thống diễn ra thuận lợi và hiệu quả, nhóm đã lựa chọn một môi trường phát triển hiện đại, phù hợp với yêu cầu kỹ thuật của ứng dụng và xu hướng phát triển phần mềm hiện nay. Môi trường phát triển bao gồm các yếu tố phần cứng, phần mềm, công cụ hỗ trợ và các nền tảng dịch vụ đám mây tích hợp.

Quá trình phát triển được thực hiện trên máy tính cá nhân với cấu hình đủ mạnh để đáp ứng yêu cầu biên dịch, chạy thử ứng dụng và giả lập thiết bị:

- **Laptop:** MSI Bravo, trang bị vi xử lý Ryzen 5 thế hệ 5, RAM 16GB, SSD 512GB, sử dụng hệ điều hành Windows 11 (64-bit).
- **Thiết bị kiểm thử thực tế:** Bao gồm các điện thoại Android Poco X3 NFC Android 12 (API 31) và AVD (Android Virtual Device) của Android Studio (API 28 – API 35). Điều này nhằm đảm bảo khả năng tương thích và kiểm thử trải nghiệm người dùng thực tế.

3.1.2. Công cụ và phân mềm phát triển

Android Studio Meerkat: Môi trường phát triển chính dành cho ứng dụng Android, hỗ trợ Kotlin và Jetpack Compose, tích hợp công cụ kiểm thử, giả lập và trình quản lý phụ thuộc Gradle.

Visual Studio Code: Được sử dụng để cài đặt môi trường cho Node.js (v18) và Express.js, xây dựng máy chủ backend phục vụ các API RESTful.

Postman: Công cụ kiểm thử API, giúp kiểm tra phản hồi từ server trong quá trình phát triển và gỡ lỗi.

GitHub: Dùng để quản lý mã nguồn, theo dõi lịch sử phát triển và làm việc nhóm hiệu quả, sử dụng Gitflow để quản lý mã nguồn hiệu quả.

Figma: Công cụ thiết kế giao diện người dùng, hỗ trợ tạo nguyên mẫu UI/UX và trình bày luồng tương tác.

Render: Nền tảng triển khai (deployment platform) dành cho backend, được sử dụng để host và vận hành server Node.js/Express một cách liên tục, có khả năng tự động build và deploy từ GitHub repository.

Supabase Studio: Giao diện quản trị dữ liệu và hệ thống backend được sử dụng để quản lý cơ sở dữ liệu PostgreSQL, xác thực người dùng, lưu trữ tệp và thực hiện đồng bộ cloud. Supabase cũng đóng vai trò như một dịch vụ cloud backend giúp triển khai cơ sở dữ liệu trực tuyến nhanh chóng và dễ dàng mở rộng.

Firestore là một nền tảng phát triển ứng dụng di động và web của Google, cung cấp nhiều dịch vụ, trong đó dự án này sử dụng 2 dịch vụ chính là Remote config cho phép ta thay đổi hành vi và giao diện của ứng dụng mà không cần phải phát hành bản cập nhật mới, Crashlytics cho phép xem số lượng crash và thông tin crash.

3.1.3. Các thư viện SDK tích hợp

Để tăng tốc quá trình phát triển và mở rộng chức năng của ứng dụng, em đã tích hợp nhiều thư viện và SDK mã nguồn mở, phổ biến và đáng tin cậy trong hệ sinh thái Android. Dưới đây là một số thành phần tiêu biểu:

Jetpack Compose: Bộ công cụ UI hiện đại từ Google, cho phép xây dựng giao diện bằng cách khai báo thành phần dưới dạng hàm. Compose hỗ trợ quản lý trạng thái hiệu quả và dễ dàng tùy biến giao diện theo điều kiện runtime.

Room Database: Thư viện quản lý cơ sở dữ liệu SQLite nội bộ theo hướng ORM (Object Relational Mapping), hỗ trợ annotation để truy vấn dữ liệu, đảm bảo an toàn type và hiệu suất cao trong thao tác dữ liệu offline.

Hilt (Dagger): Thư viện Dependency Injection chính thức của Android, dựa trên Dagger. Hilt giúp giảm boilerplate code, đơn giản hóa việc quản lý vòng đời của các dependency trong kiến trúc MVVM.

Retrofit: Thư viện HTTP client cấp cao được sử dụng để giao tiếp với backend. Retrofit hỗ trợ parsing JSON tự động bằng Gson/Moshi và dễ dàng khai báo các endpoint bằng annotation.

OkHttp: HTTP client cấp thấp được sử dụng cùng Retrofit, giúp kiểm soát chi tiết các request/response, hỗ trợ caching, logging và xử lý kết nối hiệu quả, kể cả trong điều kiện mạng không ổn định.

Firestore Authentication SDK: Hỗ trợ xác thực người dùng bằng email/password hoặc qua các phương thức khác (Google, Facebook). Tích hợp dễ dàng với backend và Supabase để quản lý người dùng thống nhất.

Google ML Kit: Bộ SDK xử lý ngôn ngữ và hình ảnh thông minh, được dùng để nhận dạng văn bản (OCR), dịch ngôn ngữ, nhận diện chữ viết tay và hỗ trợ Text-to-Speech. ML Kit hoạt động tốt cả online và offline.

iText 7: Thư viện Java mạnh mẽ để thao tác với tệp PDF. Cho phép thêm/xoá trang, chèn chữ ký, chú thích, tạo biểu mẫu và xử lý bảo mật PDF ngay trong thiết bị di động.

MuPDF: Bộ thư viện hiển thị PDF nhẹ và hiệu năng cao, hỗ trợ nhiều định dạng tài liệu. Được sử dụng để xem và thao tác trực tiếp trên tệp PDF với tốc độ tải trang nhanh và hỗ trợ annotation.

Google AdMob SDK: Được tích hợp để hiển thị quảng cáo trong ứng dụng, hỗ trợ định dạng banner, interstitial và rewarded ads. AdMob giúp tạo nguồn thu nhập cho ứng dụng dựa trên hành vi người dùng.

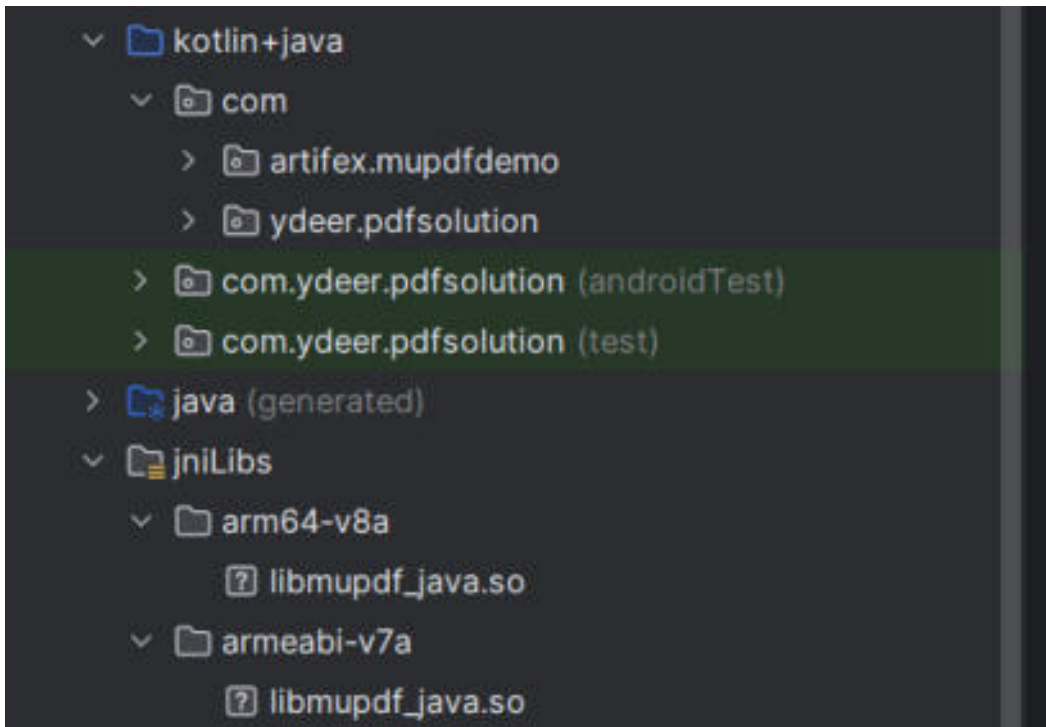
Google Play Billing Library: Cung cấp giải pháp thanh toán trong ứng dụng (in-app billing), cho phép người dùng mua gói Premium, mở khoá tính năng hoặc gia hạn gói dịch vụ định kỳ qua Google Play Store.

3.2 Triển khai Mobile app

3.2.1. Triển khai tích hợp thư viện native (JNI) trong ứng dụng

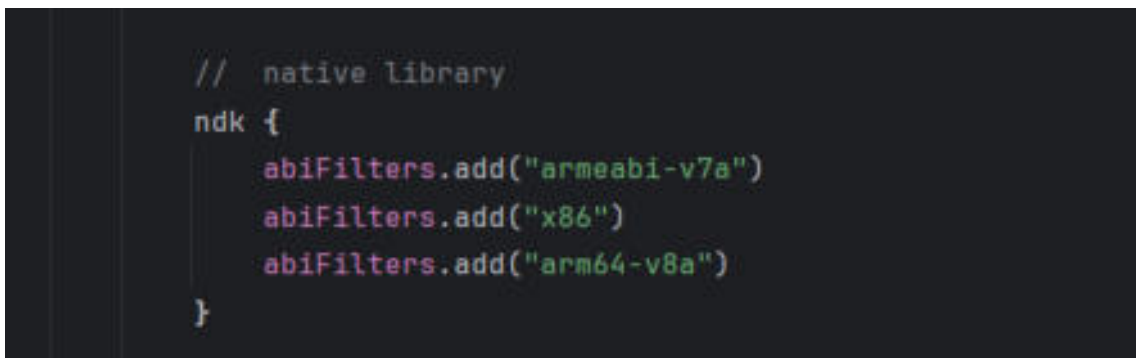
Trong quá trình phát triển hệ thống, một số thư viện xử lý tài liệu như **MuPDF** yêu cầu tích hợp các thành phần native dưới dạng thư viện .so (Shared Object) để thực hiện các tác vụ hiệu năng cao như render PDF, xử lý font hoặc mã hóa dữ liệu. Để đảm bảo khả năng tương thích đa nền tảng cũng như tăng hiệu suất xử lý, ứng dụng đã được cấu hình để hỗ trợ tích hợp thư viện native thông qua cơ chế **Java Native Interface (JNI)**.

Đầu tiên, các thư viện native được đặt trong thư mục `libs/` hoặc `src/main/jniLibs/`, phân chia theo từng kiến trúc CPU (ví dụ: `armeabi-v7a`, `arm64-v8a`, `x86`,...). Mỗi tệp .so tương ứng với một thư viện biên dịch cho từng kiến trúc phần cứng. Tiếp theo, cấu hình `sourceSets` trong tệp `build.gradle` được cập nhật để ứng dụng nhận diện thư mục chứa thư viện:



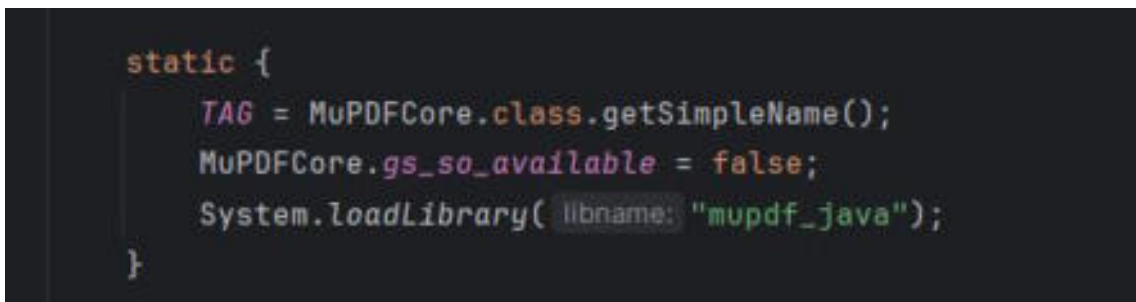
Hình 3.1: Tổ chức thư mục để triển khai tích hợp thư viện native (JNI)

Tiếp theo cần cấu hình Gradle để thực hiện chỉ định ABI Filters trong build.gradle.kts:



Hình 3.2: Cấu hình Gradle để thực hiện chỉ định ABI

Tiếp theo cần tạo một file Java để thực hiện tương tác với thư viện Native:



Hình 3.3: Tạo file để thực hiện sử dụng thư viện

Xong bước này chúng ta đã có thể triển khai thành công thư viện MuPDF vào ứng dụng.

3.2.2. Triển khai ProGuard trong ứng dụng

ProGuard là một công cụ tối ưu hóa mã nguồn được tích hợp trong Android Gradle Plugin, có vai trò rút gọn (minify), làm rối mã (obfuscate) và loại bỏ mã không sử dụng (shrink) nhằm giảm kích thước tệp APK cũng như tăng cường bảo mật mã nguồn ứng dụng. Trong quá trình triển khai hệ thống, việc cấu hình ProGuard được thực hiện nhằm bảo vệ mã nguồn, tránh việc bị dịch ngược (reverse engineering) và đồng thời đảm bảo hiệu suất cũng như khả năng hoạt động ổn định sau khi đóng gói ứng dụng.

```
buildTypes {
    getByName(name: "release") {
        isDebuggable = false
        isMinifyEnabled = true
        isShrinkResources = true
        productFlavors.getByName(name: "dev").signingConfig = signingConfigs.getByName(name: "devRelease")
        productFlavors.getByName(name: "prod").signingConfig = signingConfigs.getByName(name: "prodRelease")
        proguardFiles(
            getDefaultProguardFile(name: "proguard-android-optimize.txt"),
            "proguard-rules.pro"
        )
    }
}
```

Hình 3.4: Cấu hình Gradle để sử dụng ProGuard

Tiếp theo chúng ta cần tùy chỉnh file

Trong file proguard-rules.pro, các luật tùy chỉnh được khai báo để:

Bảo vệ các lớp bị phản chiếu (reflection), ví dụ thư viện Gson, Retrofit, giữ lại các thông tin Signature, Annotations cho Gson

```
-keepattributes Signature

# For using GSON @Expose annotation
-keepattributes *Annotation*

# Gson specific classes
-dontwarn sun.misc.**
#-keep class com.google.gson.stream.** { *; }

# Application classes that will be serialized/deserialized over Gson
-keep class com.google.gson.examples.android.model.** { <fields>; }

# Prevent proguard from stripping interface information from TypeAdapter,
# JsonSerializer, JsonDeserializer instances (so they can be used in @Json
-keep class * extends com.google.gson.TypeAdapter
-keep class * implements com.google.gson.TypeAdapterFactory
-keep class * implements com.google.gson.JsonSerializer
-keep class * implements com.google.gson.JsonDeserializer

# Prevent RB from leaving Data object members always null
-keepclassmembers,allowobfuscation class * {
    @com.google.gson.annotations.SerializedName <fields>;
}
}
```

Hình 3.5: Tùy chỉnh file ProGuard giữ lại các lớp và thông tin

Tiếp theo trong chỉnh sửa, chúng ta cần loại trừ các cảnh báo về những lớp Java AWT hay Java ImageIO không còn tồn tại trên Android

```
80     -dontwarn java.awt.Color
81     -dontwarn java.awt.Dimension
82     -dontwarn java.awt.Rectangle
83     -dontwarn java.awt.color.ColorSpace
84     -dontwarn java.awt.geom.AffineTransform
85     -dontwarn java.awt.geom.Dimension2D
86     -dontwarn java.awt.geom.Path2D
87     -dontwarn java.awt.geom.PathIterator
88     -dontwarn java.awt.geom.Point2D
```

Hình 3.6: Tùy chỉnh file ProGuard loại trừ cảnh báo

ProGuard được cấu hình với tác dụng và cách sử dụng như sau:

- Sử dụng `-keep` để giữ lại các lớp quan trọng không nên bị obfuscate
- Sử dụng `-keepattributes` để giữ lại metadata cần thiết (Signature, Annotations)
- Sử dụng `-dontwarn` để bỏ qua cảnh báo về các lớp không tồn tại trên Android

3.2.3. Triển khai quản lý phiên bản và Flavor trong ứng dụng

Việc quản lý phiên bản và flavor trong ứng dụng đóng vai trò quan trọng trong quá trình phát triển và phân phối sản phẩm. Nó cho phép phân tách các phiên bản thử nghiệm, phiên bản dành cho nhà phát triển hoặc người dùng cuối, từ đó hỗ trợ kiểm thử, triển khai và duy trì ứng dụng một cách hiệu quả, chuyên nghiệp.

Quản lý phiên bản (Versioning)

Ứng dụng được cấu hình với hai thông số chính trong file `build.gradle` để nhận diện phiên bản:

- **versionCode**: là số nguyên tăng dần, được Google Play sử dụng để xác định bản cập nhật.
- **versionName**: là chuỗi hiển thị cho người dùng, phản ánh rõ ràng thay đổi chức năng.

Quản lý Flavor:

Flavor cho phép tạo ra nhiều biến thể của ứng dụng từ cùng một mã nguồn. Trong dự án này, hai flavor chính được định nghĩa:

- **Flavor dev**: Dùng trong quá trình phát triển, có hậu tố `.dev` trong `applicationId` để tránh xung đột với bản chính thức. Phiên bản này được bật các log, các công cụ debug, sử dụng môi trường test.
- **Flavor production**: Là phiên bản chính thức, dành cho người dùng cuối. Được cấu hình tối ưu hóa hiệu suất, tắt log, sử dụng ProGuard và kết nối đến môi trường production thực tế

Lợi ích

Việc triển khai quản lý phiên bản và flavor giúp:

- Dễ dàng phát triển và kiểm thử song song trên các môi trường khác nhau.
- Hạn chế lỗi do nhầm lẫn giữa các bản thử nghiệm và bản chính thức.

- Tăng cường hiệu quả khi phối hợp làm việc nhóm hoặc tích hợp CI/CD.

```
productFlavors {
    create( name: "dev" ) {
        applicationIdSuffix = ".dev"
        versionCode = 10
        versionName = "1.0.10"
        resValue( type: "string", name: "app_name", value: "PDF Solutions Dev" )
        resValue( type: "string", name: "app_eedod_id", value: "ca-app-pub-1940256099942544-1347511713" )
    }

    create( name: "prod" ) {
        versionCode = 17
        versionName = "1.0.3"
        resValue( type: "string", name: "app_name", value: "PDF Solutions" )
        resValue( type: "string", name: "app_eedod_id", value: "ca-app-pub-7720541491635683-1540860614" )
    }
}

applicationVariants.all {
    outputs.forEach { output ->
        if (output is com.android.build.gradle.internal.api.BaseVariantOutputImpl) {
            output.outputFileName =
                "${PdfScanner}_${name}_${versionName}_${this.versionCode}.${output.outputFile.extension}"
        }
    }
}
```

Hình 3.7: Cấu hình cho phiên bản và Flavor

Tiếp theo cần cấu hình Signing cho từng Flavor

```
signingConfigs {
    create( name: "devRelease" ) {
        keyAlias = "pdf_solutions_dev"
        keyPassword = "PDFSolutions#Ydeer"
        storeFile = file( path: "../app/src/dev/pdf_solutions_dev.jks" )
        storePassword = "PDFSolutionsDev#Ydeer"
    }

    create( name: "prodRelease" ) {
        keyAlias = "pdf_solutions"
        keyPassword = "PDFSolutions#Ydeer"
        storeFile = file( path: "../app/src/prod/pdf_solutions_dev.jks" )
        storePassword = "PDFSolutions#Ydeer"
    }
}
```

Hình 3.8: Cấu hình cho Signing cho từng Flavor

Thông qua việc tổ chức và quản lý phiên bản chuyên nghiệp, đồ án không chỉ đáp ứng yêu cầu kỹ thuật, mà còn thể hiện tư duy sản phẩm rõ ràng, định hướng mở rộng và triển khai ứng dụng ra thị trường trong tương lai.

Thực hiện tạo AppScript để thực hiện chạy chức năng tự động, file xuất ra được lưu vào Google Drive cá nhân:

```
PDFSolutionLanguageAppScript

function translateAndSaveStrings() {
  const FOLDER_ID = 'PDF Solutions';

  const SPREADSHEET_NAME = 'LanguageTemplate';
  const SHEET_NAME = 'Data';
  const SOURCE_COLUMN_INDEX = D;
  const TARGET_LANGUAGES = [
    { code: "es", name: "Spanish" },
    { code: "fr", name: "French" },
    { code: "de", name: "German" },
    { code: "vi", name: "Vietnamese" },
    { code: "ja", name: "Japanese" },
    { code: "pl", name: "Polish" },
    { code: "pt", name: "Portuguese" },
    { code: "hr", name: "Croatian" },
    { code: "hi", name: "Hindi" },
    { code: "nl", name: "Dutch" },
    { code: "ko", name: "Korean" },
    { code: "in", name: "Indonesian" },
    { code: "it", name: "Italian" },
    { code: "no", name: "Norwegian" },
    { code: "ru", name: "Russian" },
    { code: "fi", name: "Finnish" },
    { code: "th", name: "Thai" },
    { code: "tr", name: "Turkish" },
    { code: "sv", name: "Swedish" },
    { code: "zh", name: "Chinese" },
    { code: "da", name: "Danish" }
  ];

  const spreadsheet = SpreadsheetApp.getActiveSpreadsheet();
```

Hình 3.12: Script để thực hiện dịch ngôn ngữ nhanh bằng Google App Script

Lợi ích của quy trình này

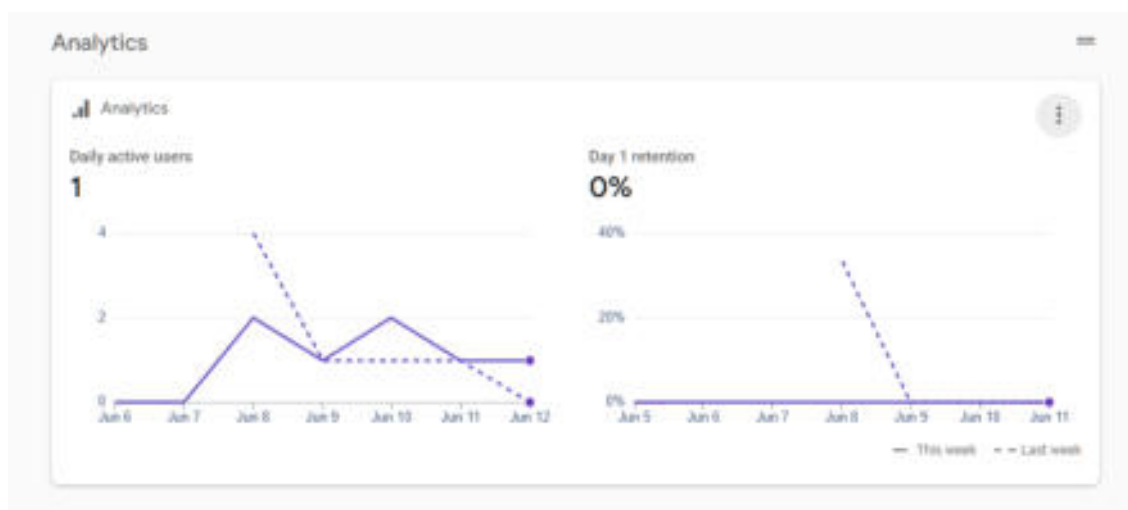
- **Nhanh chóng:** Dịch hàng trăm chuỗi chỉ trong vài phút.
- **Dễ quản lý:** Giao diện bảng tính trực quan, phù hợp cả với người không chuyên kỹ thuật.
- **Tự động hóa:** Hạn chế sai sót khi xử lý thủ công, giảm chi phí nhân lực.
- **Dễ mở rộng:** Có thể thêm ngôn ngữ mới chỉ bằng cách thêm một cột.

3.2.4. Triển khai tích hợp Firebase và Analytics trong ứng dụng

Để nâng cao khả năng theo dõi, phân tích hành vi người dùng và hỗ trợ xử lý lỗi hiệu quả, hệ thống đã được tích hợp đầy đủ các dịch vụ do Firebase cung cấp. Firebase là một nền tảng phát triển ứng dụng của Google, cung cấp nhiều dịch vụ thiết yếu giúp tối ưu hóa hiệu suất vận hành, cải thiện trải nghiệm người dùng và hỗ trợ ra quyết định dựa trên dữ liệu thực tế.

Firestore Analytics:

Firestore Analytics được sử dụng để theo dõi các hành vi chính của người dùng, bao gồm: số lần mở ứng dụng, thời gian sử dụng, sự kiện tùy chỉnh (như xem tệp, chia sẻ, chỉnh sửa, đăng nhập, đăng ký), giúp nhóm phát triển phân tích hiệu quả các tính năng và hành trình người dùng trong ứng dụng. Các sự kiện được gửi thông qua API `logEvent()` từ Firebase SDK.



Hình 3.13: Firebase Analytics thực tế được triển khai

Firestore Crashlytics:

Crashlytics hỗ trợ ghi nhận và báo cáo các sự cố hoặc lỗi phát sinh trong quá trình người dùng sử dụng ứng dụng. Mỗi khi xảy ra lỗi, báo cáo chi tiết bao gồm stack trace, thiết bị, trạng thái bộ nhớ, phiên bản app... sẽ được gửi về Firebase Console. Điều này giúp nhóm phát triển kịp thời khắc phục lỗi và cải thiện độ ổn định.

Đầu tiên để thực hiện ghi các thông tin thì em đã triển khai Timber để quản lý log tích hợp với Crashlytics:

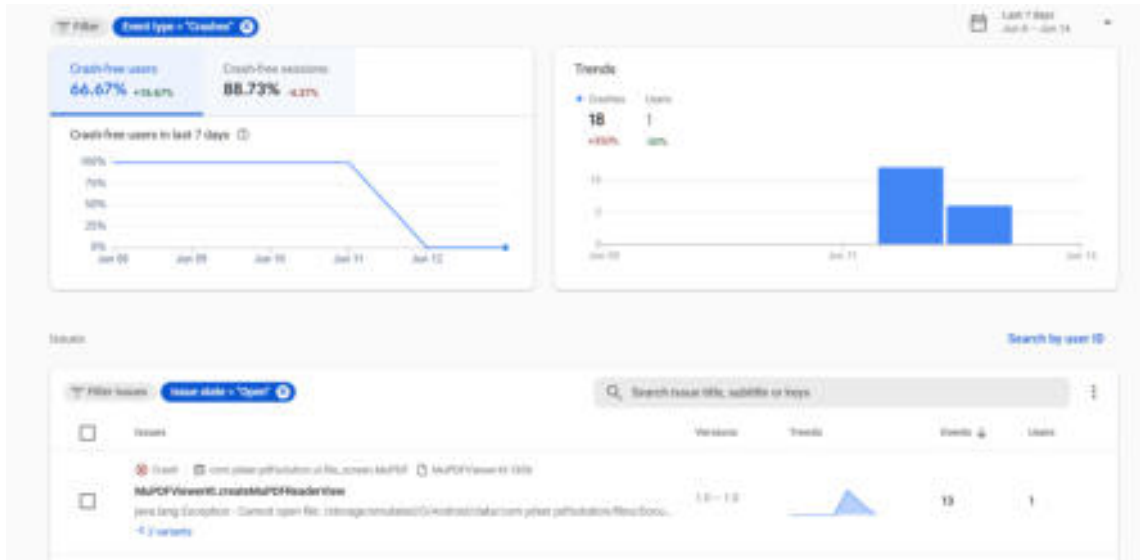
```
class logger {
    companion object {
        fun init() {
            if (BuildConfig.DEBUG) {
                Timber.plant(Timber.DebugTree())
            } else {
                Timber.plant(CrashlyticsTree())
            }
        }
    }
}

private class CrashlyticsTree : Timber.Tree() {
    override fun log(priority: Int, tag: String?, message: String, t: Throwable?) {
        if (priority >= Log.INFO) {
            FirebaseCrashlytics.getInstance().log("$tag: $message")
            t?.let { FirebaseCrashlytics.getInstance().recordException(it) }
        }
    }
}
```

Hình 3.14: Triển khai Timber

Tác dụng của Timber:

1. **Quản lý log hiệu quả:** Timber cung cấp cơ chế logging linh hoạt hơn Android Log
2. **Tự động thêm tag:** Không cần chỉ định tag cho mỗi log
3. **Phân biệt môi trường:** Sử dụng DebugTree trong môi trường debug và CrashlyticsTree trong môi trường production
4. **Tích hợp với Crashlytics:** Tự động gửi log lên Firebase Crashlytics trong môi trường production
5. **Lọc log theo mức độ:** Chỉ gửi log có mức độ từ INFO trở lên lên Crashlytics



Hình 3.15: Firebase Crashlytics thực tế được triển khai

Firestore: Firestore cho phép thay đổi các giá trị trong ứng dụng mà không cần cập nhật phiên bản mới. Ví dụ như thay đổi banner khuyến mãi, bật/tắt một số tính năng thử nghiệm, hoặc điều chỉnh thông điệp giao diện theo từng đối tượng người dùng. Firestore được đồng bộ khi người dùng mở app hoặc theo chu kỳ định sẵn.

Triển khai Firebase Remote Config trong ứng dụng:

```
@Singleton
class FirebaseRemoteConfigManager @Inject constructor(
    val appPrefs: AppPrefs
) {

    private val remoteConfig: FirebaseRemoteConfig = FirebaseRemoteConfig.getInstance()

    init {
        val configSettings: FirebaseRemoteConfigSettings = FirebaseRemoteConfigSettings.Builder()
            .setMinimumFetchIntervalInSeconds(0)
            .build()
        remoteConfig.setConfigSettingsAsync(configSettings)
        remoteConfig.setDefaultsAsync(R.xml.remote_config_defaults)
    }

    suspend fun setupRemoteConfig(): FirebaseRemoteConfig? {
        return suspendCancellableCoroutine { continuation ->
            try {
                Log.d( tag = "RemoteConfig", msg = "Starting Remote Config setup...")

                remoteConfig.fetchAndActivate()
                    .addOnSuccessListener { fetchedRemotely ->
```

Hình 3.16: Firebase Remote Config được triển khai



Hình 3.17: Firebase Remote Config thực tế được triển khai

3.2.5. Triển khai tích hợp Google Admob trong ứng dụng

Để hỗ trợ mô hình thương mại hóa và tạo nguồn doanh thu bền vững cho ứng dụng, hệ thống đã được tích hợp Google AdMob – nền tảng quảng cáo di động do Google phát triển, cho phép hiển thị quảng cáo trong ứng dụng một cách linh hoạt và hiệu quả.

Cấu hình và tích hợp AdMob SDK

AdMob SDK được tích hợp vào dự án thông qua Gradle với các thư viện chính như `com.google.android.gms:play-services-ads`. File `AndroidManifest.xml` được cấu hình các quyền liên quan đến kết nối mạng, đồng thời khai báo AdMob App ID theo yêu cầu từ Google. Tất cả cấu hình đều tuân thủ theo chính sách của Google Play để đảm bảo an toàn và minh bạch cho người dùng.

Các hình thức quảng cáo triển khai

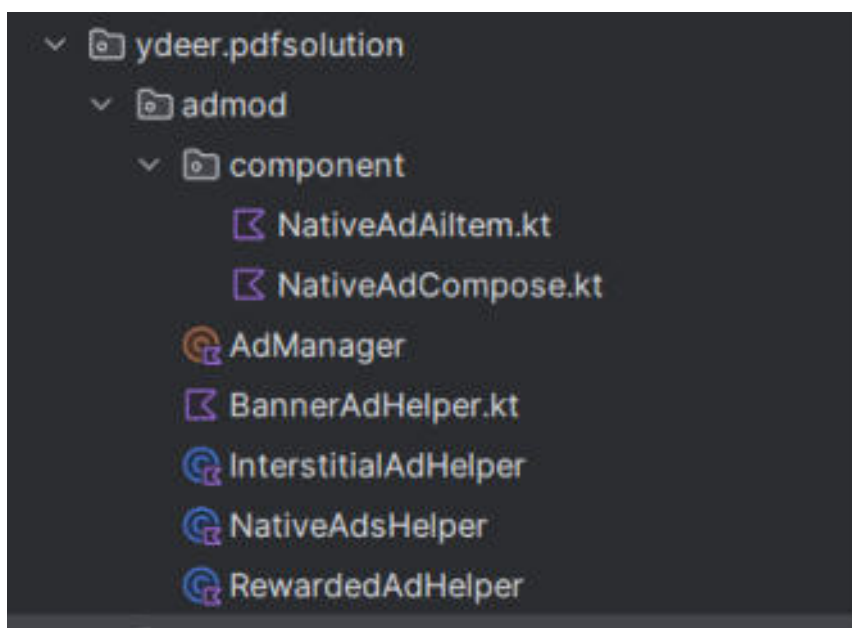
Ứng dụng triển khai các loại quảng cáo sau:

- **Banner Ads:** hiển thị ở các vị trí cố định (thường là dưới cùng màn hình) trong các màn hình ít tương tác để không gây gián đoạn.
- **Interstitial Ads** (quảng cáo toàn màn hình): chỉ xuất hiện ở các điểm chuyển màn hình (ví dụ: sau khi xem tệp hoặc trước khi mở tệp mới) để giảm thiểu ảnh hưởng đến trải nghiệm.
- **Rewarded Ads** (quảng cáo thưởng): được triển khai cho một số chức năng nâng cao, cho phép người dùng xem quảng cáo để mở khóa tính năng hoặc tăng giới hạn tạm thời mà không cần mua gói Premium.
- **Native Ads** (quảng cáo bản địa): tích hợp trực tiếp vào giao diện ứng dụng dưới dạng nội dung có phong cách và cách trình bày tương đồng với phần còn lại của ứng dụng. Ví dụ, trong danh sách tệp hoặc màn hình tính năng, quảng cáo native sẽ được trình bày như một phần tự nhiên của nội dung,

giúp nâng cao tỉ lệ tương tác trong khi vẫn đảm bảo trải nghiệm người dùng liền mạch và không gây khó chịu.

Quản lý tần suất và trải nghiệm người dùng

Tần suất hiển thị quảng cáo được kiểm soát nghiêm ngặt để không làm phiền người dùng. Các logic quản lý quảng cáo được viết riêng trong một module, kết hợp với SharedPreferences để ghi nhận lịch sử hiển thị, đảm bảo không hiển thị lặp lại trong thời gian ngắn. Ngoài ra, hệ thống cũng hỗ trợ ẩn toàn bộ quảng cáo khi người dùng nâng cấp lên gói Premium.



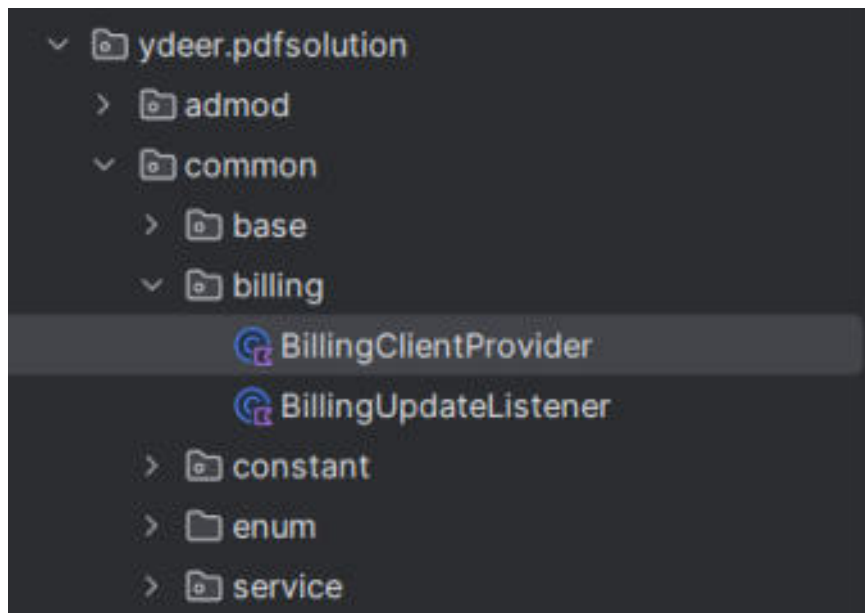
Hình 3.18: Cấu trúc Admob được triển khai

3.2.5. Triển khai tích hợp Google Play Billing trong ứng dụng

Để hỗ trợ mô hình kinh doanh Freemium, ứng dụng đã được tích hợp hệ thống thanh toán **Google Play Billing**, cho phép người dùng nâng cấp lên phiên bản **Premium** nhằm mở khóa các tính năng nâng cao, loại bỏ quảng cáo và mở rộng giới hạn lưu trữ.

Cấu hình và tích hợp Billing Library

Hệ thống sử dụng thư viện chính thức Google Play Billing Library (v6.0+) để đảm bảo tương thích và an toàn theo quy định của Google. Các SKU (product ID) cho các gói đăng ký như Premium Monthly, Premium Yearly được định nghĩa trực tiếp trên Google Play Console và đồng bộ với ứng dụng thông qua API.



Hình 3.19: Cấu trúc Billing được triển khai

Quy trình tích hợp bao gồm:

- Cấu hình BillingClient để kiểm tra trạng thái đăng ký.
- Gọi launchBillingFlow để kích hoạt giao diện mua hàng gốc của Google.
- Theo dõi các callback onPurchasesUpdated() để xử lý kết quả thanh toán và cập nhật trạng thái Premium cho người dùng.

```
1 package com.ydeer.pdfsolution.common.billing
2
3 import android.content.Context
4 import com.android.billingclient.api.BillingClient
5 import com.android.billingclient.api.PurchasesUpdatedListener
6
7 class BillingClientProvider(
8     context: Context,
9     updateListener: PurchasesUpdatedListener
10 ) {
11     val billingClient = BillingClient.newBuilder(context)
12         .enablePendingPurchases()
13         .setListener(updateListener)
14         .build()
15 }
16
```

Hình 3.20: Billing được triển khai

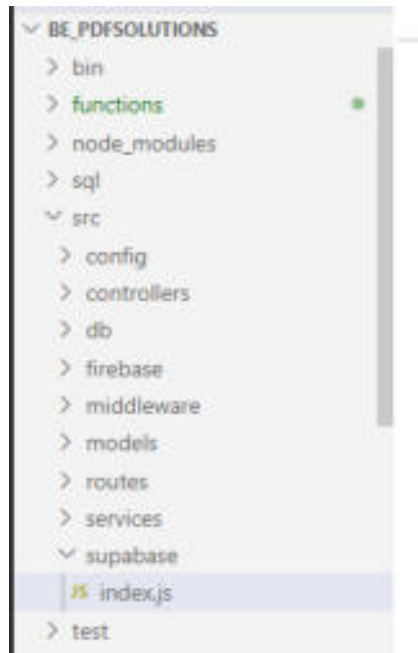
Xử lý đăng ký và kiểm tra trạng thái

Ứng dụng thực hiện lưu trạng thái đăng ký cục bộ thông qua SharedPreferences, kết hợp với truy vấn định kỳ từ BillingClient.queryPurchasesAsync() để kiểm tra tình trạng còn hiệu lực của gói Premium. Ngoài ra, ứng dụng cũng triển khai BillingReceiver để theo dõi các thay đổi từ Google Play như hủy gói, hết hạn hoặc chuyển đổi gói.

```
1 package com.ydeer.pdfsolution.common.billing
2
3 import android.util.Log
4 import com.android.billingclient.api.BillingResult
5 import com.android.billingclient.api.Purchase
6 import com.android.billingclient.api.PurchasesUpdatedListener
7 import kotlinx.coroutines.flow.MutableStateFlow
8 import kotlinx.coroutines.flow.asStateFlow
9 import javax.inject.Inject
10 import javax.inject.Singleton
11
12 @Singleton
13 class BillingUpdateListener @Inject constructor() : PurchasesUpdatedListener {
14
15     private val _purchaseUpdate = MutableStateFlow<List<Purchase>?>(null)
16     val purchaseUpdate = _purchaseUpdate.asStateFlow()
17
18     override fun onPurchasesUpdated(billingResult: BillingResult, purchases: MutableList<Purchase>) {
19         Log.d("TEST", "onPurchasesUpdated ===> BillingResult $billingResult $purchases")
20         val newList: MutableList<Purchase> = mutableListOf<Purchase>()
21         newList.addAll(billingResult.purchases ?: emptyList())
22         _purchaseUpdate.value = newList
23     }
24 }
```

Hình 3.21: Billing Update được triển khai

3.3 Triển khai Server NodeJS



Hình 3.22: Cấu trúc server NodeJS

Vai trò từng thư mục:

- controllers/: xử lý logic cho từng endpoint.
- routes/: định nghĩa các API và ánh xạ đến controller tương ứng.
- services/: thao tác trực tiếp với database và logic nghiệp vụ.
- middlewares/: xác thực, xử lý lỗi.
- config/: chứa cấu hình Supabase, biến môi trường, v.v.

Tiếp theo kết nối với Supabase, cấu hình supabaseUrl và supabaseKey trong biến môi trường .env để đảm bảo bảo mật cho các cấu hình quan trọng.

Tạo một dự án trên trang web chính thức của Supabase và thực hiện kết nối, lựa chọn sử dụng Supabase do Supabase cung cấp các dịch vụ như cơ sở dữ liệu PostgreSQL, xác thực người dùng (Authentication), lưu trữ tệp (Storage), API tự động (auto-generated RESTful API), và bảng điều khiển quản lý web—all-in-one, giúp tiết kiệm thời gian triển khai và đồng bộ

Để triển khai server Node.js lên Render, các bước thực hiện như sau:

Chuẩn bị repository: Mã nguồn server được đẩy lên GitHub, đảm bảo đầy đủ các tệp cấu hình như package.json, index.js, .env.example và các thư mục cần thiết.

Tạo dịch vụ Web Service trên Render:

Truy cập trang chủ <https://render.com> và đăng nhập.

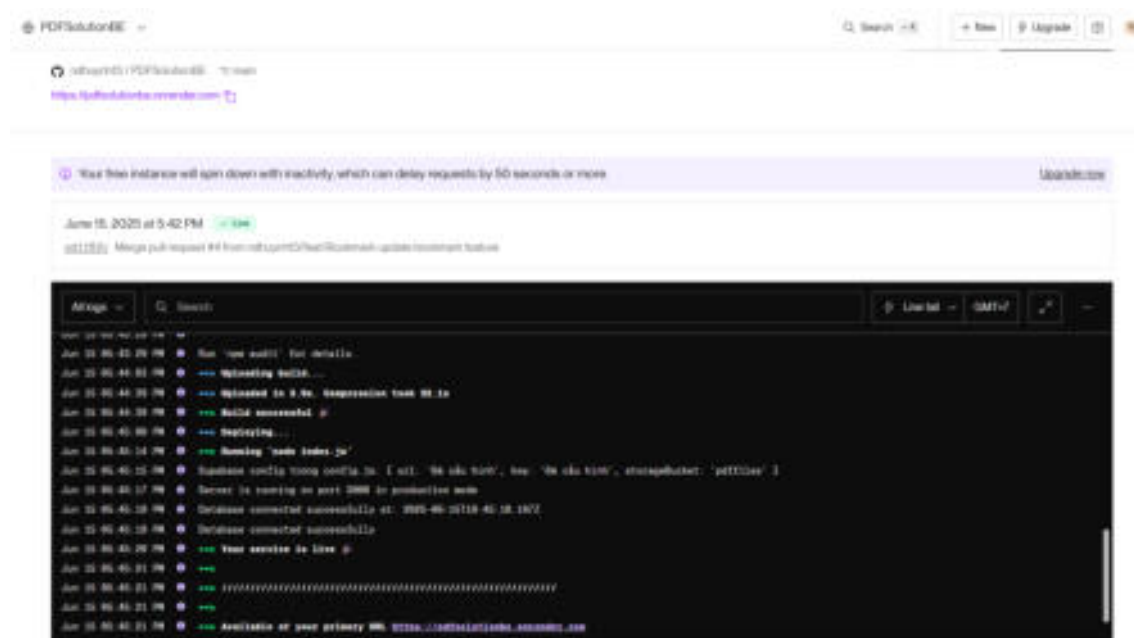
Tạo dịch vụ mới với lựa chọn **Web Service**, kết nối với repository GitHub chứa server.

Cấu hình thông tin cơ bản: tên dịch vụ, chi nhánh Git, runtime (Node.js), command để khởi chạy (npm start hoặc node index.js).

Thêm biến môi trường (Environment Variables) như URL Supabase, API key, JWT secret,... tương ứng với .env.

Deploy và theo dõi logs:

Render sẽ tự động build và deploy server. Sau khi hoàn tất, một URL public sẽ được cấp phát, ví dụ: <https://pdfsolutionbe.onrender.com>. Người dùng có thể truy cập hoặc test API ngay trên URL này. Logs được hiển thị trong bảng điều khiển giúp theo dõi hoạt động và xử lý lỗi.



Hình 3.25: Sử dụng Render để deploy Server lên môi trường Product

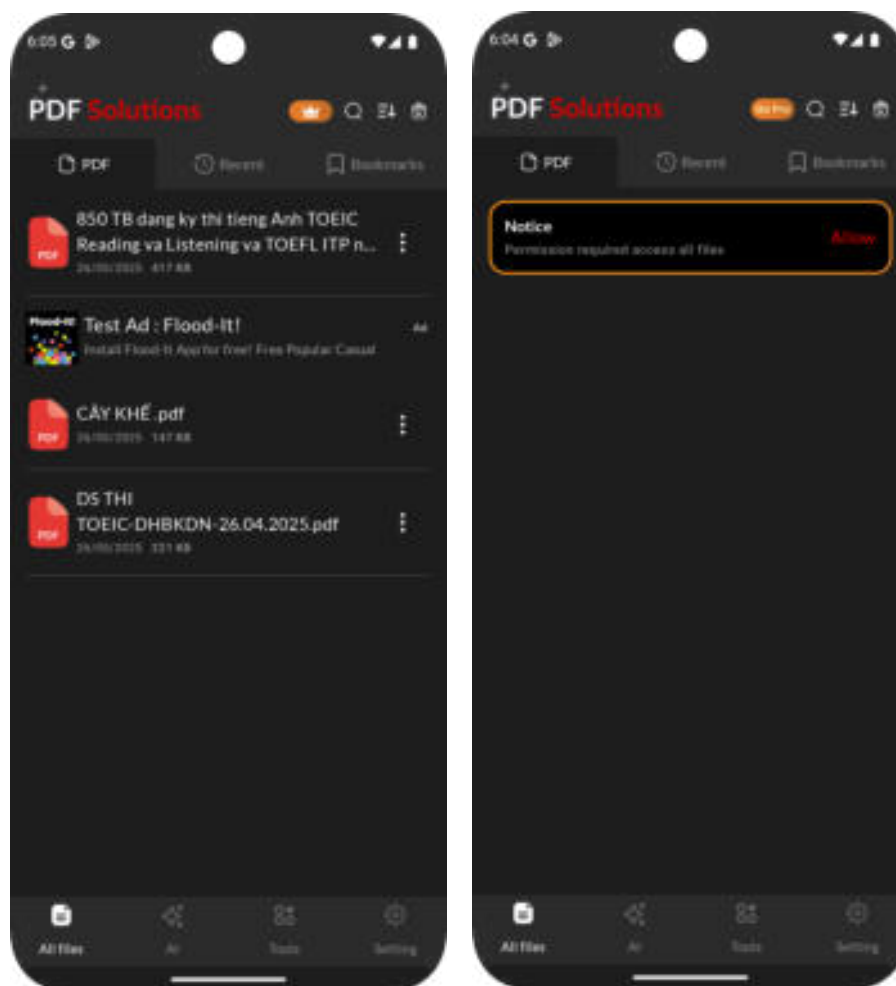
Việc sử dụng Render giúp đơn giản hóa quá trình triển khai, giảm thiểu chi phí ban đầu và phù hợp với các ứng dụng đang trong giai đoạn phát triển, nghiên cứu

hoặc thử nghiệm sản phẩm. Trong tương lai có thể cân nhắc sử dụng các gói cao hơn cho phát hành thực tế với lượng sử dụng cao hơn.

3.4 Triển khai và kết quả của các chức năng chính

3.3.1. Quản lý tệp

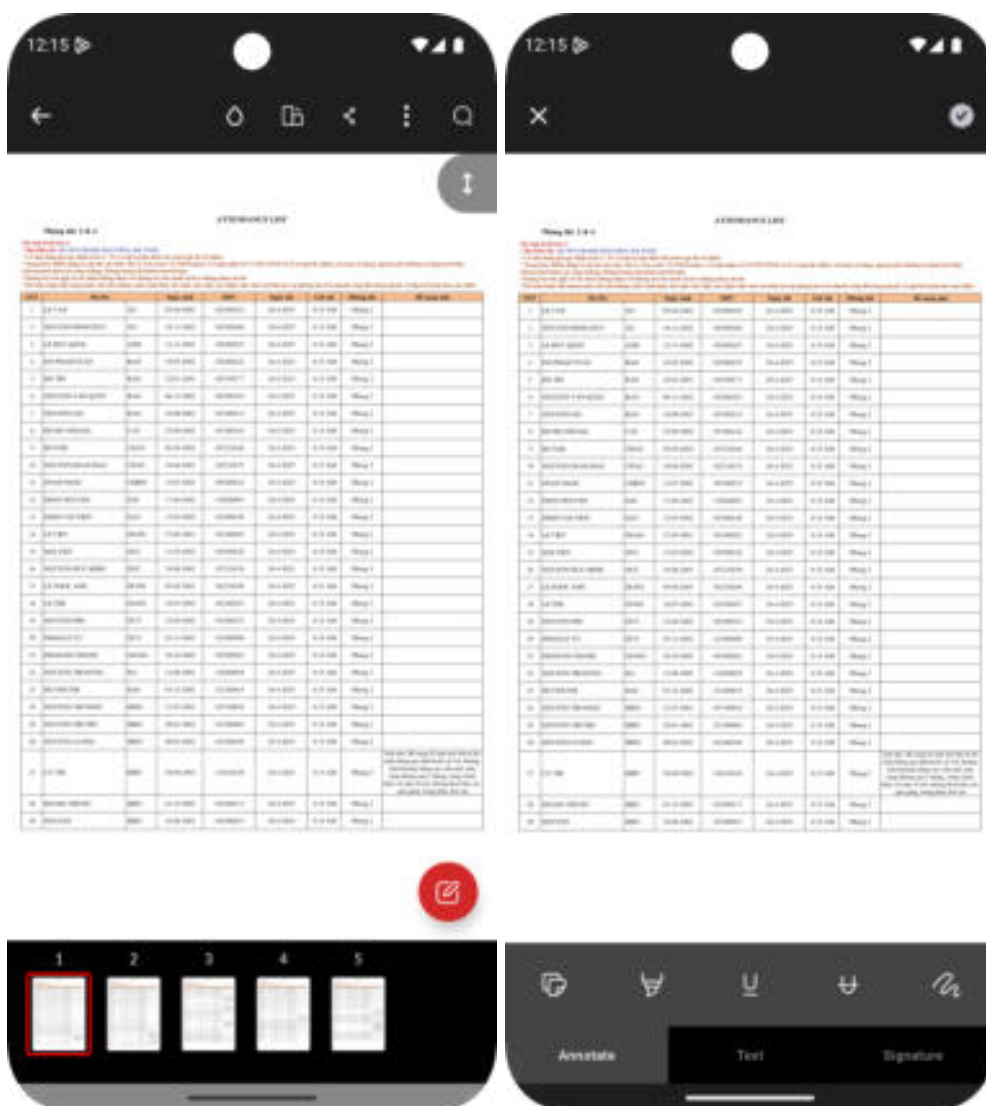
Chức năng quản lý tệp được thiết kế như một hệ thống tích hợp, cho phép người dùng thực hiện các thao tác cơ bản và nâng cao với tài liệu số. Kiến trúc của module này được xây dựng trên mô hình Repository Pattern kết hợp với Room Database, đảm bảo khả năng lưu trữ, truy xuất và thao tác với dữ liệu ngay cả khi không có kết nối internet. Người dùng có thể dễ dàng phân loại tài liệu theo nhiều tiêu chí khác nhau như định dạng (PDF, Word, Excel, PowerPoint,...), thời gian tạo, kích thước và tên tệp. Đặc biệt, hệ thống tìm kiếm được tối ưu hóa thông qua thuật toán indexing, cho phép truy xuất thông tin nhanh chóng ngay cả khi số lượng tệp lớn. Việc hiển thị các tệp được thiết kế theo nguyên tắc Material Design 3, cung cấp trải nghiệm trực quan và thân thiện với người dùng.



Hình 3.26: Giao diện app quản lý tệp

3.3.2. Xem và chỉnh sửa file PDF

Chức năng xem và chỉnh sửa tệp PDF cho phép người dùng tương tác toàn diện với tệp PDF. Module này tích hợp các thư viện xử lý PDF mạnh mẽ, hỗ trợ đa dạng thao tác như thêm/xóa trang, chỉnh sửa nội dung văn bản, chèn hình ảnh và chữ ký số. Chức năng này được xây dựng bằng cách thêm thủ công bằng cách tạo module thư viện MuPDF nằm cùng cấp, xử lý native thông qua file .so



Hình 3.27: Giao diện xem và chỉnh sửa PDF

3.3.3. Xem file Office, TXT, Office

Chức năng xem và chỉnh sửa tệp PDF cho phép người dùng tương tác toàn diện với tệp PDF. Module này tích hợp các thư viện xử lý PDF mạnh mẽ, hỗ trợ đa dạng

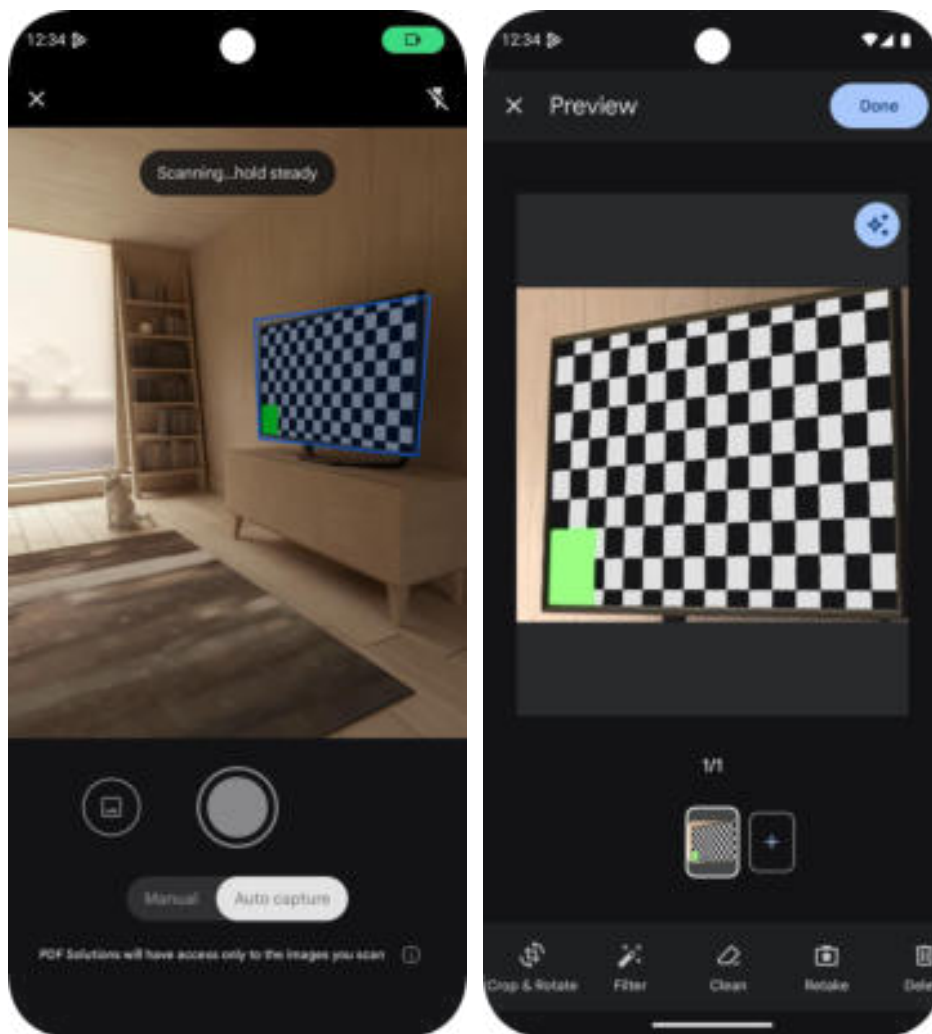
thao tác như thêm/xóa trang, chỉnh sửa nội dung văn bản, chèn hình ảnh và chữ ký số. Chức năng này được xây dựng bằng cách thêm thủ công bằng cách tạo module thư viện MuPDF nằm cùng cấp, xử lý native thông qua file .so



Hình 3.28: Giao diện xem và chỉnh sửa PDF

3.3.4 .Chức năng Scan to PDF

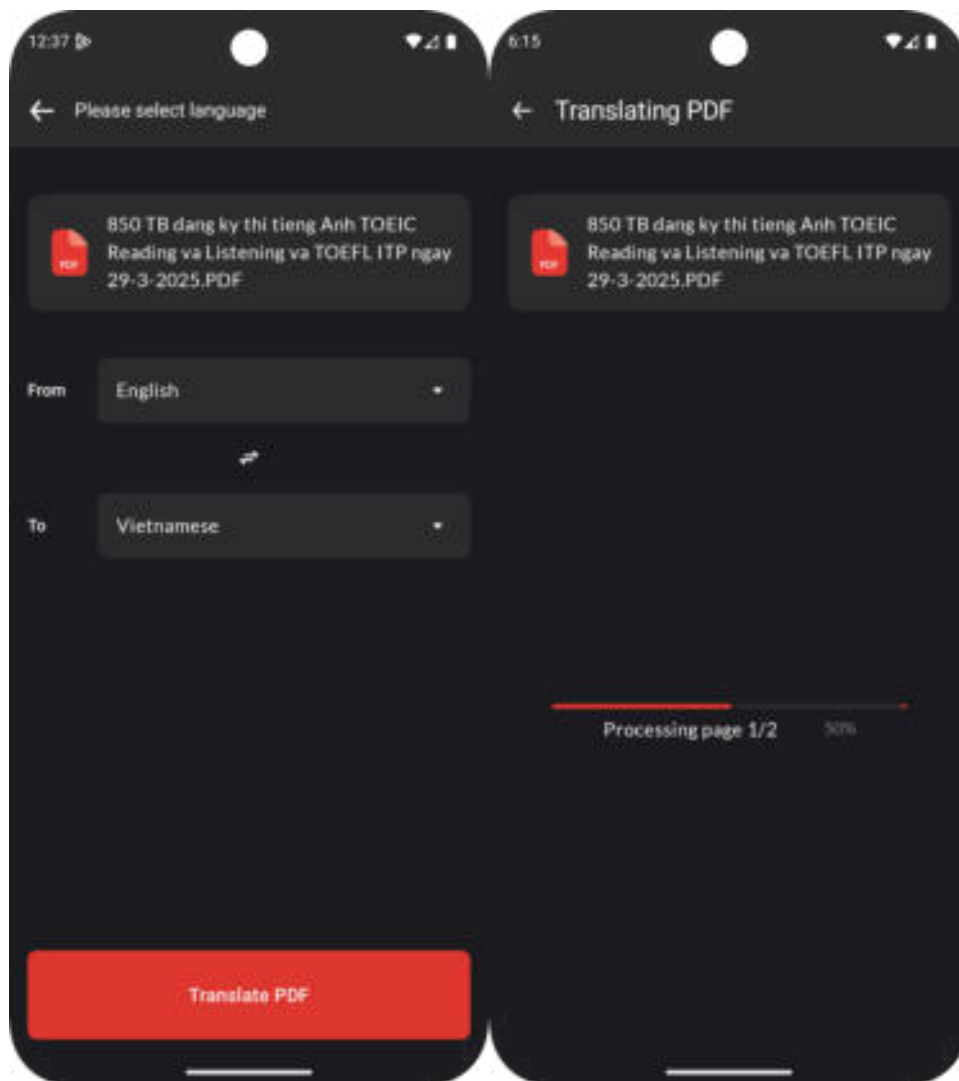
Chức năng Scan to PDF được triển khai như một giải pháp chuyển đổi tài liệu vật lý sang định dạng số một cách nhanh chóng và hiệu quả. Module này tích hợp hoàn chỉnh với camera của thiết bị thông qua các API Jetpack Compose mới nhất, cho phép người dùng chụp tài liệu với nhiều tùy chọn như tự động nhận diện biên, điều chỉnh độ sáng/tương phản và loại bỏ bóng đổ. Quá trình xử lý ảnh áp dụng các thuật toán xử lý hình ảnh tiên tiến để tăng cường chất lượng văn bản, tối ưu hóa màu sắc và giảm dung lượng tệp đầu ra. Đặc biệt, hệ thống hỗ trợ chế độ chụp nhiều trang liên tiếp với khả năng xem trước, sắp xếp lại thứ tự và chỉnh sửa từng trang trước khi chuyển đổi thành tệp PDF hoàn chỉnh. Kết quả quét được lưu trữ vào cơ sở dữ liệu OfficeFile, tích hợp liền mạch với các chức năng quản lý và chỉnh sửa PDF khác trong ứng dụng.



Hình 3.29: Chức năng Scan to PDF

3.3.5 .Chức năng Translate PDF

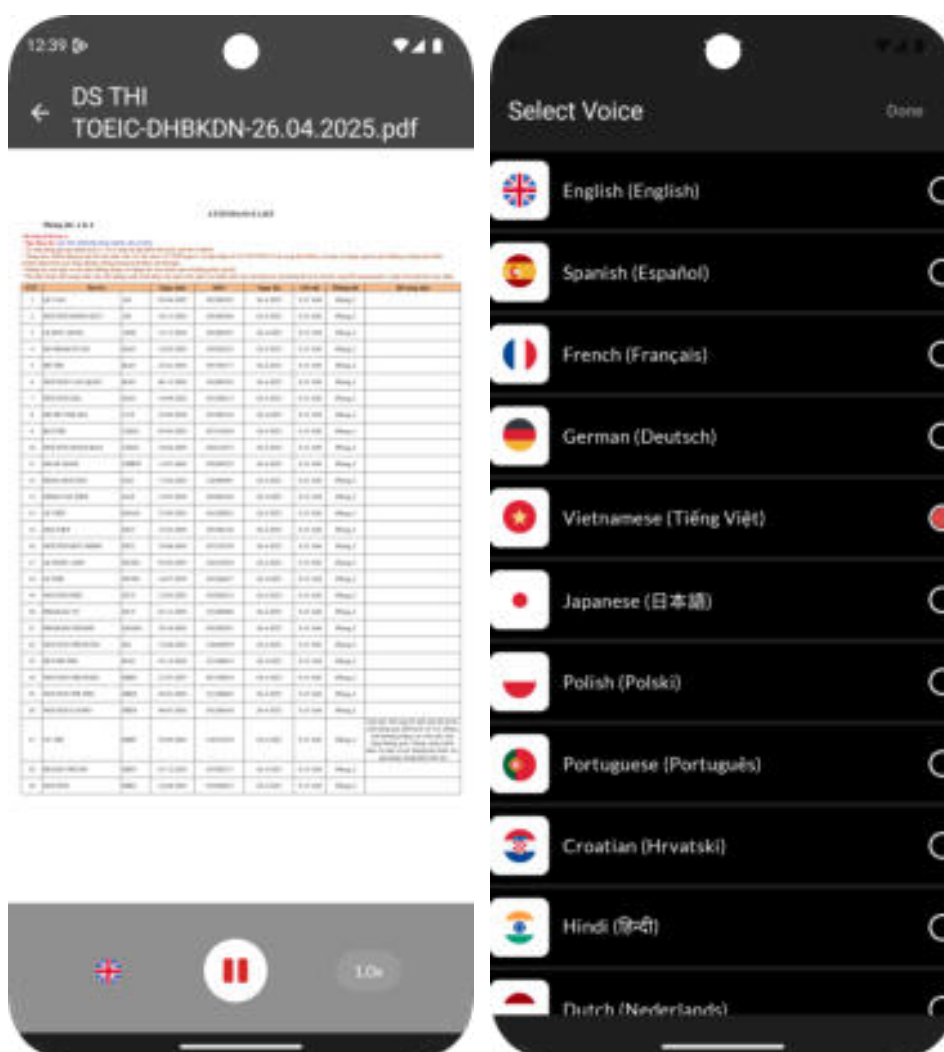
Chức năng Translate PDF được phát triển như một giải pháp tiên tiến nhằm vượt qua rào cản ngôn ngữ trong tài liệu số. Module này tích hợp các API dịch thuật hiện đại kết hợp với thuật toán trích xuất văn bản chuyên biệt PdfTextExtractor từ thư viện iText, cho phép xử lý chính xác nội dung PDF đa ngôn ngữ. Kiến trúc triển khai dựa trên mô hình Repository Pattern, giúp tách biệt logic xử lý dịch thuật và giao diện người dùng, đồng thời hỗ trợ đa dạng cặp ngôn ngữ với khả năng mở rộng linh hoạt. Giao diện được thiết kế trực quan với chế độ xem song song (văn bản gốc - văn bản dịch), hỗ trợ dịch toàn tài liệu hoặc chỉ các phần được chọn. Đặc biệt, chức năng cung cấp khả năng lưu kết quả dịch thành tệp PDF mới hoặc xuất ra định dạng văn bản thuần túy, đồng thời tích hợp cơ chế lưu vào bộ nhớ đệm để tối ưu hiệu suất khi xử lý tài liệu lớn và phức tạp.



Hình 3.30: Chức năng Translate PDF

3.3.6 .Chức năng Speech PDF

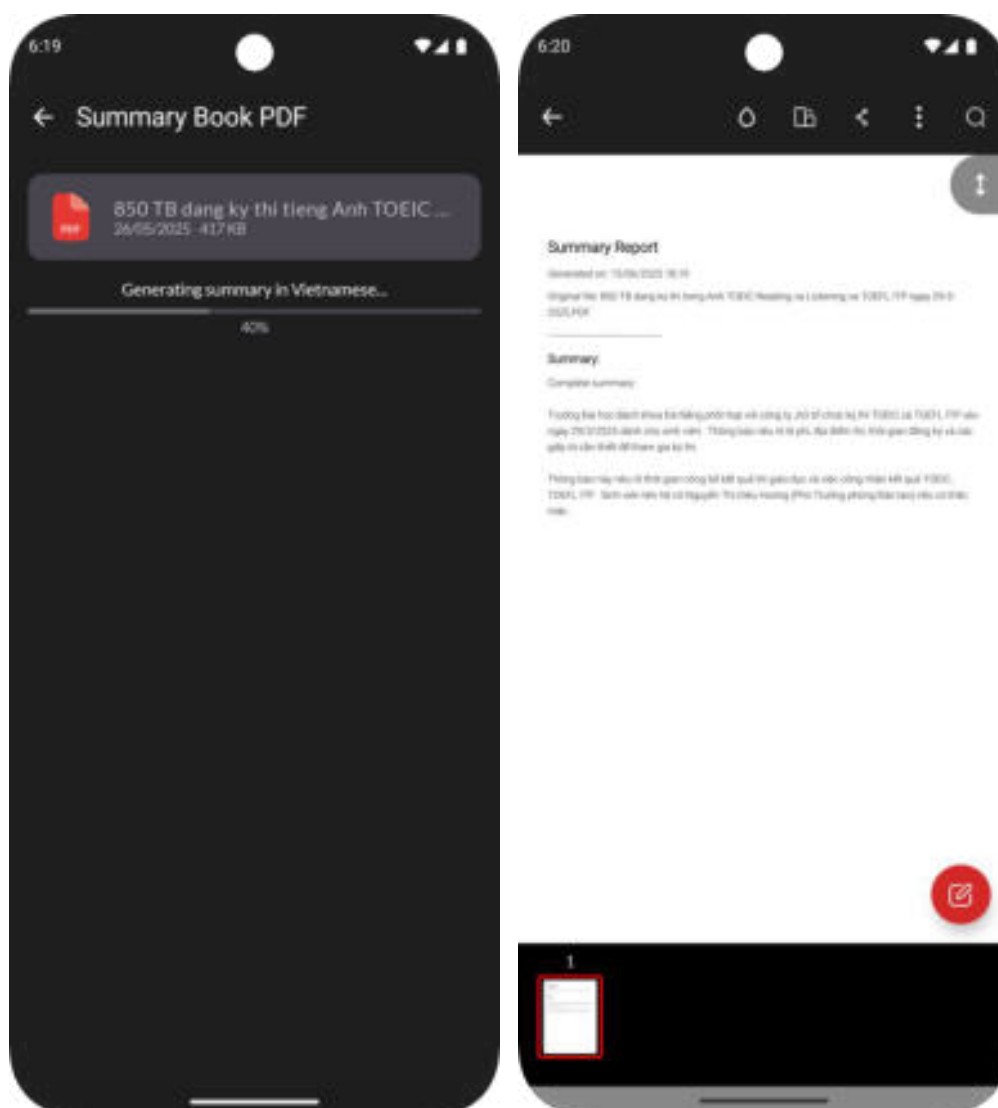
Chức năng Speech PDF được xây dựng như một giải pháp đa phương tiện tiên tiến nhằm chuyển đổi nội dung văn bản PDF thành dạng âm thanh. Module này tích hợp Text-to-Speech API của Android với PdfTextExtractor để trích xuất nội dung văn bản chính xác từ tài liệu PDF đa định dạng. Kiến trúc được thiết kế dựa trên mô hình MVVM kết hợp với coroutines, cho phép xử lý không đồng bộ hiệu quả, đảm bảo ứng dụng vận hành mượt mà ngay cả khi xử lý tài liệu lớn. Người dùng có thể điều chỉnh các tham số phát âm như tốc độ, ngôn ngữ phù hợp với nhu cầu cá nhân. Đặc biệt, chức năng hỗ trợ đánh dấu trực quan văn bản đang được đọc, cho phép tạm dừng/tiếp tục và di chuyển đến vị trí cụ thể trong tài liệu. Các trạng thái phát âm được quản lý qua ViewModel, giúp duy trì trải nghiệm người dùng nhất quán khi xoay màn hình hoặc chuyển đổi giữa các ứng dụng.



Hình 3.31: Chức năng Speech PDF

3.3.7. Chức năng Summary PDF

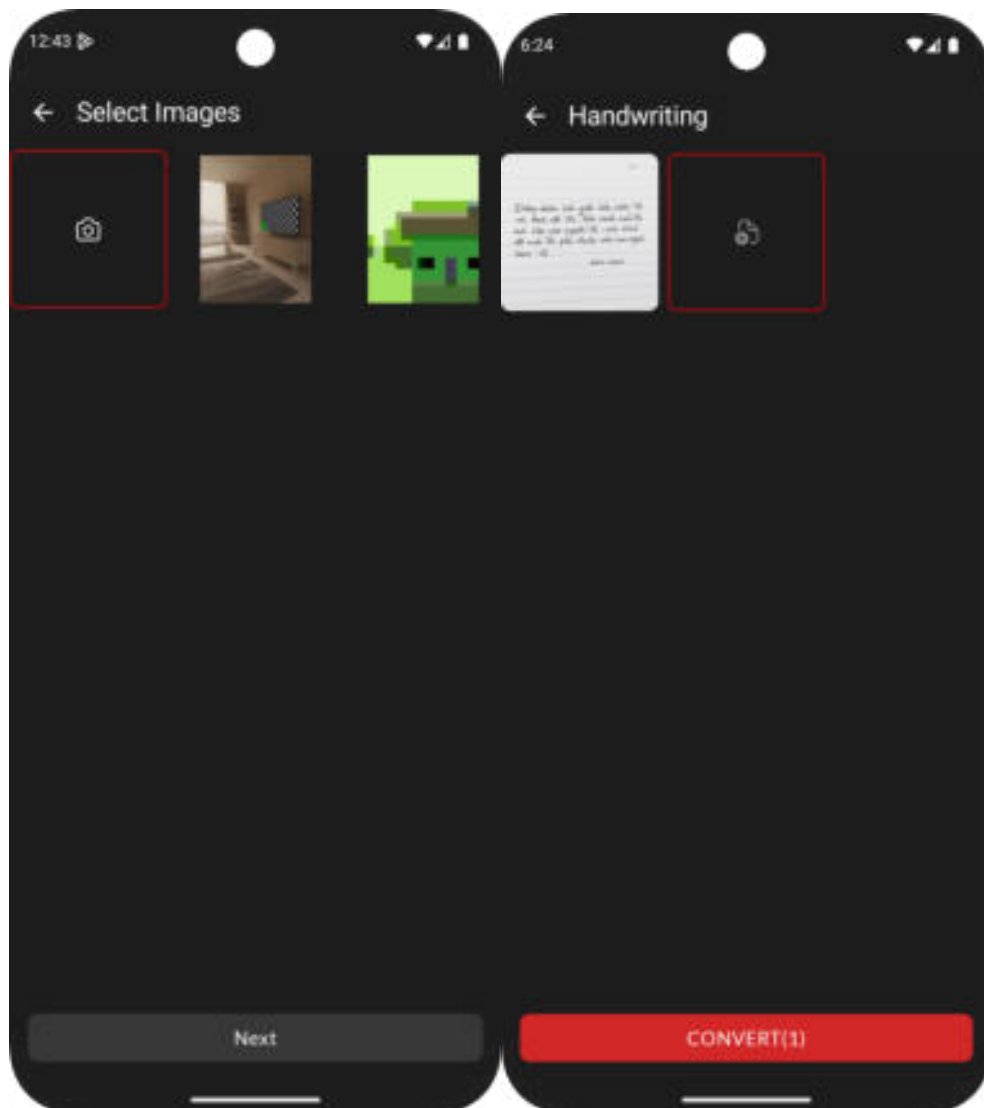
Chức năng Summary PDF được phát triển như một giải pháp ứng dụng trí tuệ nhân tạo để tạo bản tóm tắt tự động từ tài liệu PDF phức tạp. Module này kết hợp API Gemini của Google với PdfTextExtractor để trích xuất nội dung văn bản từ tài liệu, sau đó áp dụng các thuật toán AI để phân tích ngữ nghĩa và tạo bản tóm tắt ngắn gọn. Kiến trúc được thiết kế theo mô hình Repository Pattern, tách biệt hoàn toàn logic xử lý AI và giao diện người dùng, cho phép tích hợp linh hoạt với nhiều mô hình ngôn ngữ khác nhau, chức năng sử dụng kỹ thuật chunk để có thể tóm tắt tài liệu lớn mà không sợ bị vượt quá token của API cung cấp khả năng xuất kết quả tóm tắt thành tệp PDF mới hoặc văn bản thuần túy để dễ dàng chia sẻ và tham khảo.



Hình 3.32: Chức năng Summary PDF

3.3.8 .Chức năng HandWriting Recognition

Chức năng Handwriting Recognition được triển khai như một giải pháp chuyển đổi chữ viết tay thành văn bản số thông qua công nghệ thị giác máy tính và học sâu được cung cấp bởi Google ML Kit. Module này tích hợp các mô hình nhận diện chữ viết tay tiên tiến với khả năng xử lý đa ngôn ngữ, được tối ưu hóa cho thiết bị di động. Kiến trúc ứng dụng được thiết kế dựa trên mô hình MVVM kết hợp với Repository Pattern, cho phép xử lý không đồng bộ hiệu quả và tách biệt logic nhận diện khỏi giao diện người dùng. Quá trình xử lý ảnh bao gồm các bước tiền xử lý (normalization, segmentation), nhận diện và hậu xử lý (context correction).



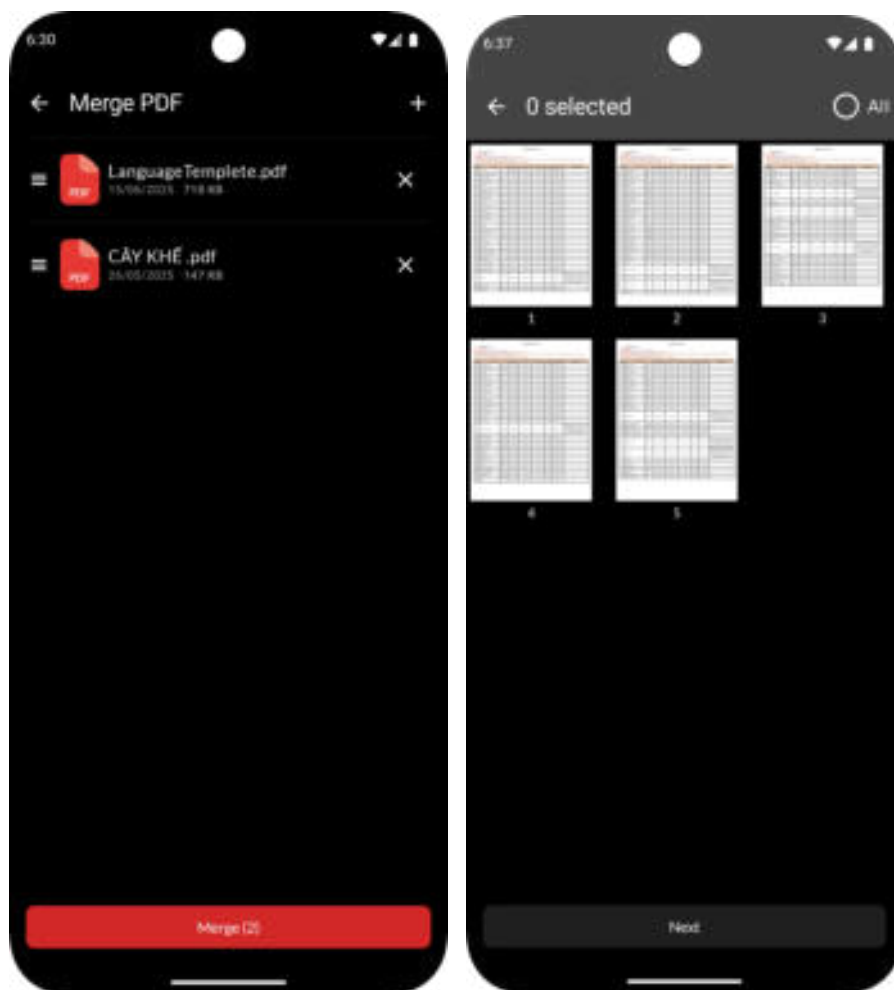
Hình 3.33: Chức năng Handwriting Recognition PDF

3.3.8 .Chức năng Merge, Split PDF

Chức năng Gộp và Tách PDF được xây dựng nhằm đáp ứng nhu cầu xử lý linh hoạt tài liệu số trong quá trình học tập, làm việc và lưu trữ. Đây là hai tính năng quan trọng cho phép người dùng dễ dàng kết hợp nhiều tệp PDF riêng biệt thành một tài liệu duy nhất (Merge), hoặc chia một tài liệu lớn thành nhiều phần nhỏ hơn (Split), phục vụ cho việc tổ chức nội dung hiệu quả và thuận tiện trong chia sẻ.

Ở chức năng **Gộp PDF**, hệ thống sử dụng các thư viện như iText và PdfBox để lần lượt đọc và xử lý các tệp đầu vào, sau đó hợp nhất các trang theo thứ tự mong muốn. Quá trình hợp nhất đảm bảo giữ nguyên định dạng gốc của văn bản, hình ảnh và bố cục trang, đồng thời cho phép xuất ra tệp PDF mới với tên tùy chỉnh.

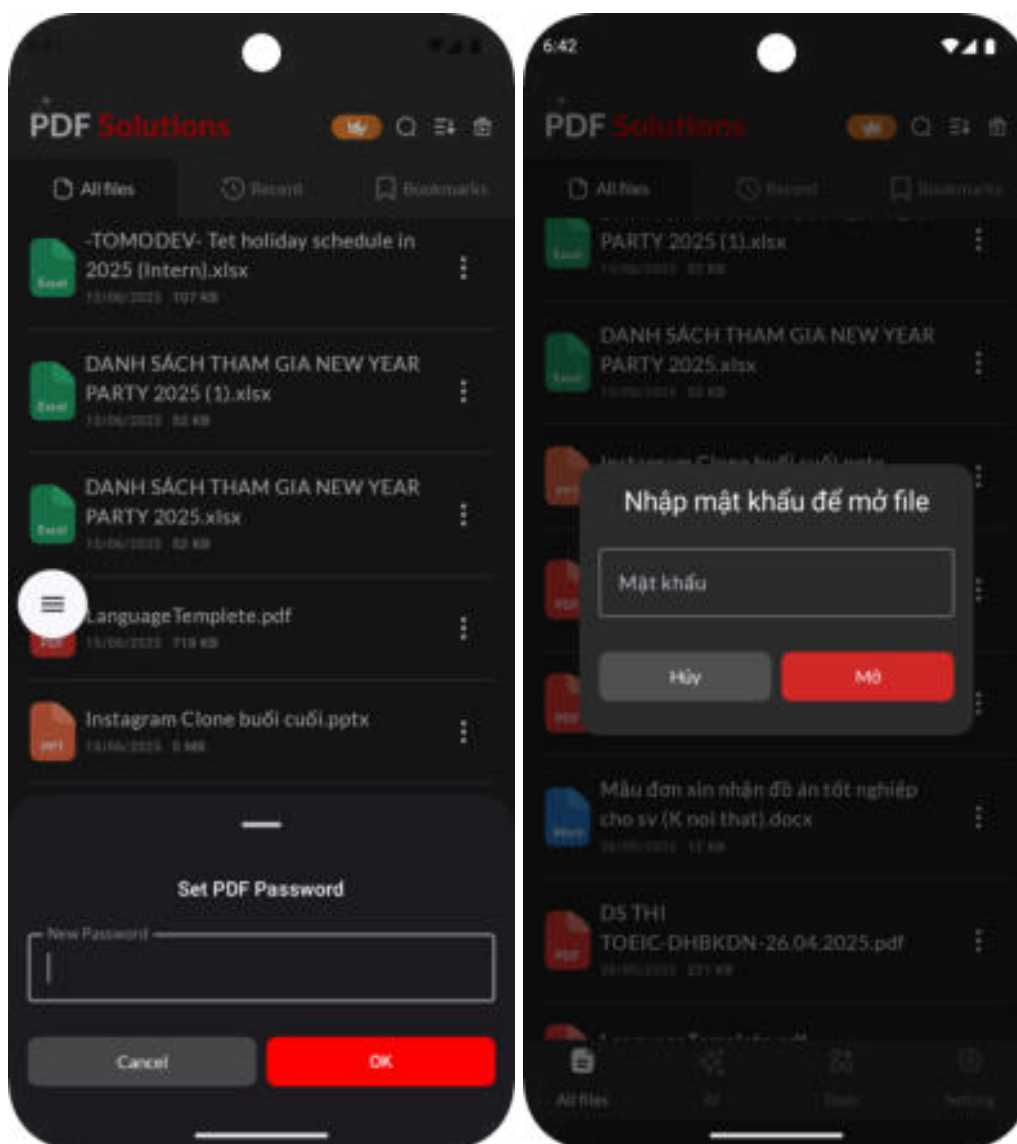
Đối với chức năng **Tách PDF**, người dùng có thể lựa chọn các phương thức tách như: tách theo số lượng trang cụ thể, theo từng khoảng trang, hoặc tách mỗi trang thành một tệp riêng biệt. Quá trình này được thực hiện một cách tối ưu và an toàn, bảo toàn toàn vẹn nội dung, đảm bảo định dạng không bị sai lệch.



Hình 3.34: Chức năng Handwriting Recognition PDF

3.3.9 .Chức năng Lock/Unlock PDF

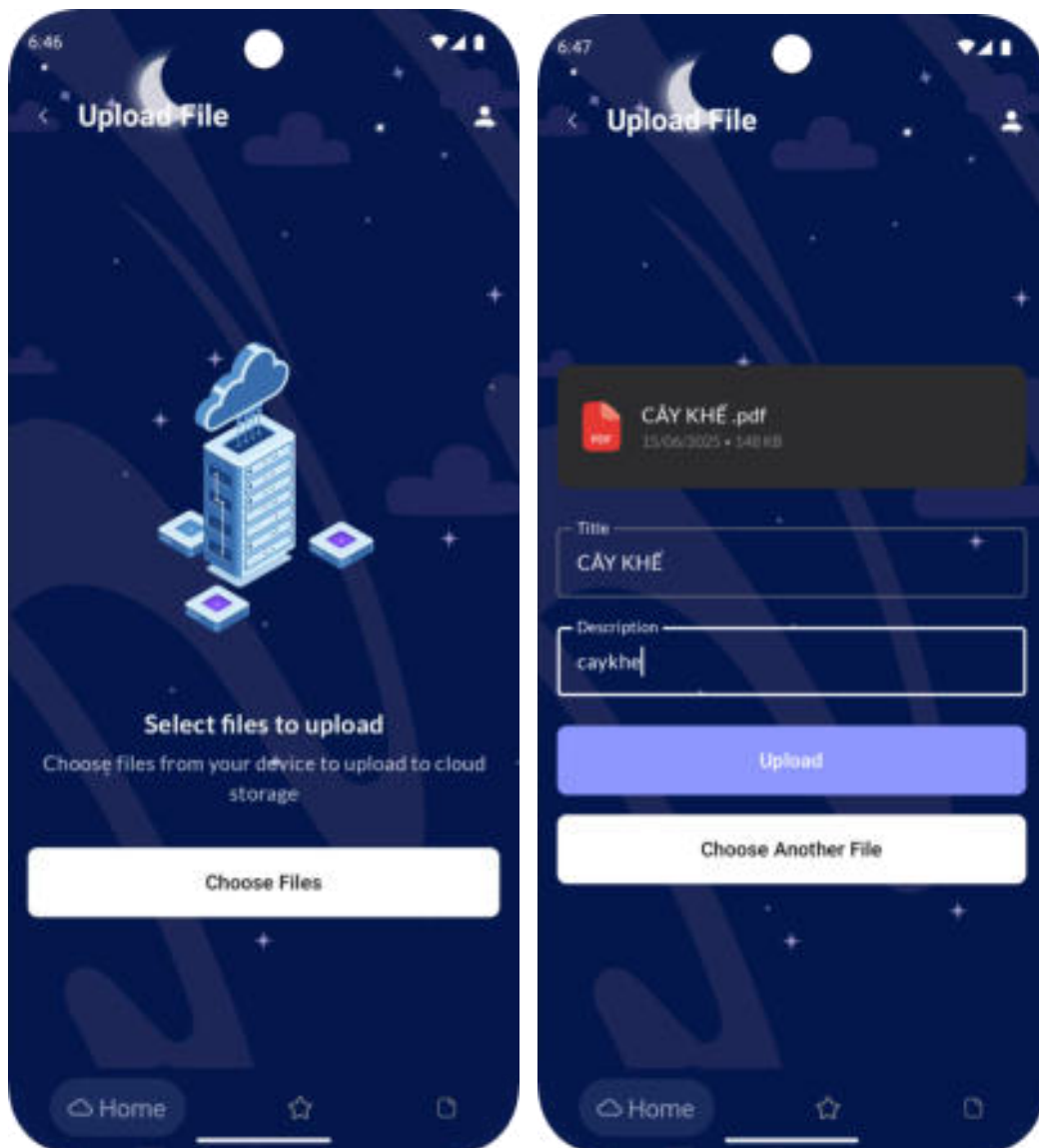
Chức năng Khóa/Mở khóa PDF giúp người dùng bảo vệ tài liệu bằng mật khẩu hoặc loại bỏ bảo mật khỏi tệp đã khóa. Khi khóa, hệ thống áp dụng mã hóa AES 256-bit để giới hạn quyền truy cập như chỉnh sửa, in ấn hay sao chép. Mở khóa yêu cầu nhập đúng mật khẩu để giải mã và phục hồi quyền truy cập đầy đủ. Chức năng sử dụng thư viện iText/PdfBox, đảm bảo bảo mật và giữ nguyên định dạng tài liệu. Giao diện được thiết kế trực quan, cho phép thao tác nhanh chóng. Hệ thống cũng xử lý lỗi nhập sai mật khẩu và cung cấp phản hồi rõ ràng. Tính năng tích hợp liền mạch với luồng quản lý tệp trong ứng dụng.



Hình 3.35: Chức năng Lock/Unlock PDF

3.3.10 .Chức năng Cloud Upload PDF

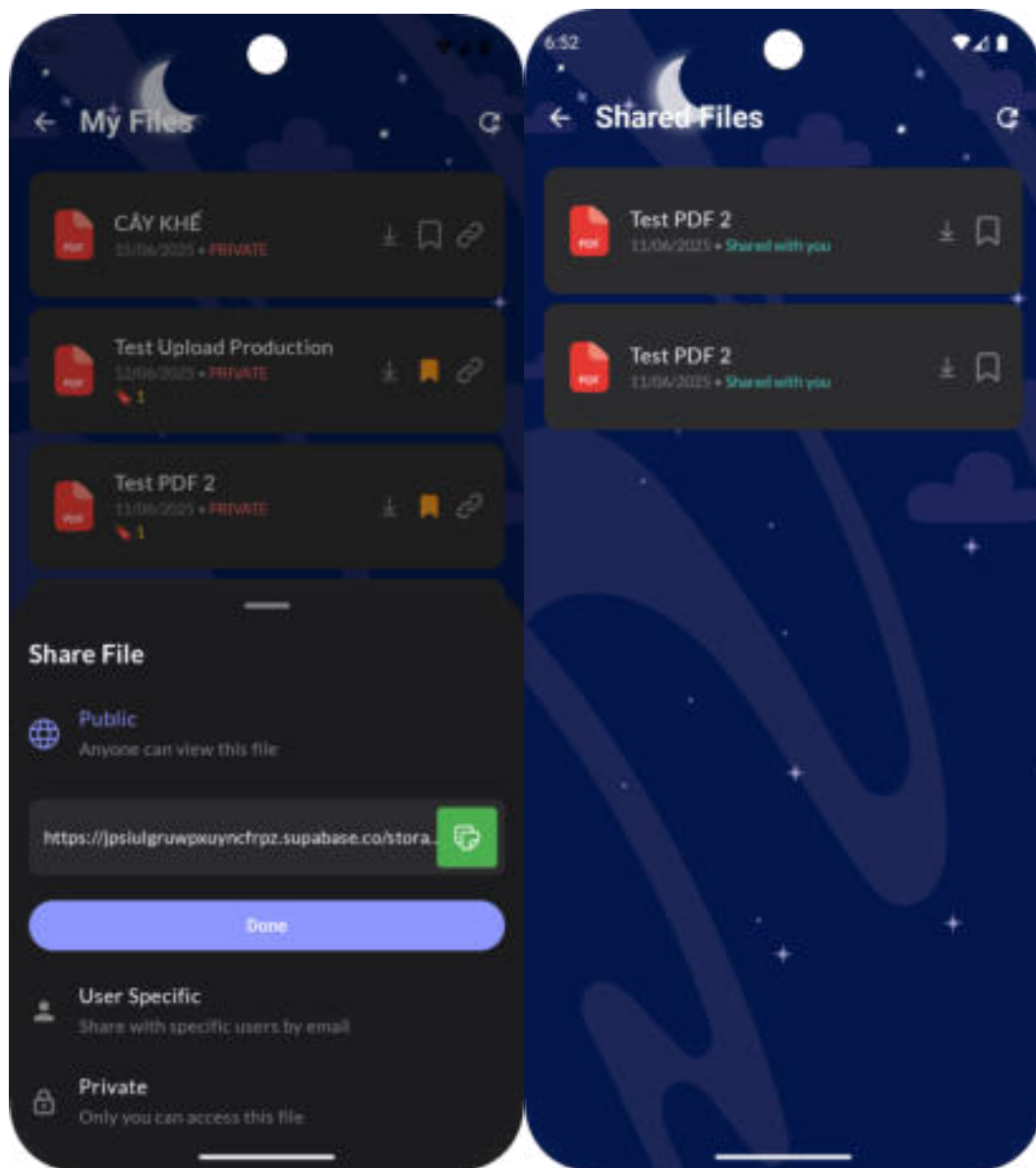
Chức năng Cloud Upload file cho phép người dùng tải các tệp từ thiết bị lên hệ thống lưu trữ đám mây một cách an toàn và tiện lợi. Ứng dụng sử dụng Supabase Storage để lưu trữ tệp, đồng thời ghi metadata (tên, loại, kích thước, thời gian) vào cơ sở dữ liệu PostgreSQL để phục vụ truy vấn và quản lý. Quá trình tải lên được tối ưu bằng tiến trình nền (background coroutine), hỗ trợ hiển thị tiến trình upload theo thời gian thực. Sau khi tải lên thành công, tệp có thể được truy cập từ nhiều thiết bị, hỗ trợ chia sẻ nhanh và đồng bộ. Hệ thống đảm bảo tính bảo mật bằng cách xác thực người dùng và sử dụng HTTPS trong truyền tải.



Hình 3.36: Chức năng Cloud Upload PDF

3.3.11 .Chức năng Share file Upload PDF

Chức năng **Chia sẻ tệp trên Cloud** cho phép người dùng chia sẻ các tệp đã tải lên đám mây một cách linh hoạt và bảo mật. Người dùng có thể tạo liên kết chia sẻ công khai (public link) hoặc chia sẻ riêng tư đến người dùng cụ thể thông qua địa chỉ email. Mỗi tệp chia sẻ được gắn với quyền truy cập (chỉ xem, chỉnh sửa) và thời gian hết hạn tùy chọn để đảm bảo quyền riêng tư. Hệ thống sử dụng Supabase để lưu thông tin chia sẻ và kiểm tra xác thực người nhận khi truy cập. Giao diện người dùng cung cấp tùy chọn sao chép liên kết, thu hồi chia sẻ hoặc theo dõi lượt truy cập tệp đã chia sẻ.



Hình 3.37: Chức năng Share File

3.5 Kết luận

Trong chương này, hệ thống đã được triển khai trên nền tảng Android với kiến trúc MVVM kết hợp các công nghệ hiện đại như Firebase, Supabase, Google ML Kit, MuPDF, iText và nhiều thư viện mã nguồn mở khác. Các chức năng chính như quản lý và hiển thị tệp, chỉnh sửa PDF, xem Office, nhận dạng văn bản bằng OCR, đồng bộ cloud, bảo mật người dùng và thanh toán trong ứng dụng đã được xây dựng và tích hợp đầy đủ.

Quá trình cài đặt và phát triển hệ thống tuân thủ các yêu cầu kỹ thuật đã đề ra, đảm bảo khả năng mở rộng, bảo trì và tính linh hoạt cao. Hệ thống cũng được kiểm thử trên nhiều thiết bị để đánh giá hiệu suất, khả năng sử dụng, và độ ổn định trong thực tế.

Kết quả triển khai cho thấy ứng dụng không chỉ đạt được mục tiêu của đề tài tốt nghiệp mà còn có tiềm năng phát triển thành một sản phẩm thực tế với giá trị ứng dụng và thương mại cao. Trong chương tiếp theo, quá trình kiểm thử, đánh giá hiệu năng và phân tích kết quả sẽ được trình bày chi tiết nhằm hoàn thiện toàn diện giải pháp đã xây dựng.

KẾT LUẬN

1. Kết luận

Trong suốt quá trình tìm hiểu, nghiên cứu lý thuyết và xây dựng hệ thống, đồ án đã đạt được các kết quả cụ thể sau:

Về mặt lý thuyết:

- Hiểu rõ quy trình phát triển phần mềm từ phân tích, thiết kế đến triển khai và đánh giá.
- Vận dụng hiệu quả kiến thức về kiến trúc MVVM, mô hình client-server, và thiết kế RESTful API.
- Bổ sung kiến thức về tích hợp các dịch vụ cloud (Supabase, Firebase) và triển khai hệ thống backend với Node.js.
- Nâng cao kỹ năng sử dụng các thư viện và SDK phổ biến như Google ML Kit, iText, MuPDF, và Jetpack Compose.
- Có thêm kinh nghiệm thực hành trong việc xử lý dữ liệu lớn, tối ưu hiệu suất, và bảo mật hệ thống.

Về mặt ứng dụng thực tiễn:

- Hoàn thiện một ứng dụng Android hỗ trợ quản lý, xem và chỉnh sửa các định dạng tài liệu phổ biến.
- Tích hợp các tính năng AI như OCR, dịch ngôn ngữ, tóm tắt văn bản, và các chức năng hữu ích khác, hỗ trợ người dùng làm việc hiệu quả hơn với tài liệu số.
- Hệ thống có khả năng mở rộng, đồng bộ dữ liệu trên cloud, hỗ trợ thanh toán và hướng đến thương mại hóa.
- Đáp ứng nhu cầu thực tế trong học tập, công việc và sử dụng cá nhân – góp phần tạo nên một sản phẩm tiềm năng để phát triển thành nền tảng thương mại.

2. Hạn chế

Mặc dù đã đạt được nhiều kết quả tích cực, hệ thống vẫn tồn tại một số điểm hạn chế sau:

- Hiệu năng của ứng dụng khi xử lý các tệp Excel có dung lượng lớn chưa thật sự tối ưu trên các thiết bị cấu hình thấp.
- Giao diện người dùng chưa thật sự hoàn chỉnh ở một số màn hình; trải nghiệm người dùng cần được cải thiện thêm.
- Một số tính năng nâng cao như chỉnh sửa định dạng Office trực tiếp, chỉnh sửa văn bản trong PDF theo đúng định dạng chữ hoặc AI nâng cao vẫn đang trong giai đoạn phát triển.
- Chưa có hệ thống kiểm thử tự động, dẫn đến khó phát hiện lỗi sớm trong quá trình phát triển lâu dài.
- Ứng dụng chủ yếu triển khai trên nền tảng Android, chưa mở rộng sang iOS hoặc nền tảng Web.

3. Hướng phát triển

Để hoàn thiện hơn trong tương lai, hệ thống sẽ được mở rộng và cải thiện theo các định hướng sau:

- Tối ưu hiệu suất, cải thiện tốc độ mở và xử lý tài liệu, đặc biệt với các tệp lớn hoặc đa phương tiện.
- Cải tiến giao diện người dùng, nâng cao trải nghiệm trực quan, nhất quán với Material Design 3.
- Mở rộng đa nền tảng, phát triển thêm phiên bản iOS và Web để tăng khả năng tiếp cận người dùng.
- Tăng cường AI, tích hợp các mô hình mạnh hơn cho OCR, nhận diện ngữ nghĩa văn bản, và gợi ý nội dung thông minh.
- Triển khai kiểm thử tự động và CI/CD, đảm bảo chất lượng ứng dụng trong môi trường phát triển chuyên nghiệp.
- Xây dựng chiến lược thương mại hóa, thông qua gói Premium, tích hợp quảng cáo hợp lý và triển khai hệ thống phân phối trên các kho ứng dụng.

TÀI LIỆU THAM KHẢO

- [1]. Android Jetpack Compose - <https://developer.android.com/courses/android-basics-compose>
- [2]. Supabase Tutorial - <https://supabase.com/docs/guides/getting-started/tutorials/with-kotlin>
- [3]. Documents | Room Database - <https://developer.android.com/training/data-storage/room?hl=vi>
- [4]. Document | Itext 7 How to use Itext 7 for android app - <https://github.com/itext/itext-android-ui>
- [5]. Document | MuPDF for Java Guide - <https://mupdf.readthedocs.io/en/1.26.2/guide/using-with-java.html>
- [6]. How to add MuPDF .so file - <https://github.com/LonelyPluto/PdfViewerDemo>
- [7]. Office Library Free Open Source Library - <https://github.com/imm-AI/office>
- [8]. ProGuard for Android Guide - <https://www.guardsquare.com/manual/configuration/usage>
- [9]. How to add Google Admob - <https://www.youtube.com/watch?v=zz7V0ggh458>
- [10]. Document | Google Play Billing - <https://developer.android.com/google/play/billing?hl=vi>
- [11]. Document | Google ML Kit - <https://developers.google.com/ml-kit/guides?hl=vi>