

**TRƯỜNG ĐẠI HỌC CẦN THƠ
KHOA CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG**



**LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC
NGÀNH CÔNG NGHỆ THÔNG TIN**

**ỨNG DỤNG BÁN ĐỒ NỘI THẤT
SỬ DỤNG THỰC TẾ TĂNG CƯỜNG**

Sinh viên: Nguyễn Thanh Khoa

Mã số: B1606992

Khóa: K42

Cần Thơ, 06/2020

**TRƯỜNG ĐẠI HỌC CẦN THƠ
KHOA CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG
BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC
NGÀNH CÔNG NGHỆ THÔNG TIN**

**ỨNG DỤNG BẢN ĐỒ NỘI THẤT
SỬ DỤNG THỰC TẾ TĂNG CƯỜNG**

**Người hướng dẫn
TS. Trần Công Ấn**

**Sinh viên thực hiện
Nguyễn Thanh Khoa
Mã số: B1606992
Khóa: K42**

Cần Thơ, 06/2020

Lời cảm ơn

Em xin gửi lời cảm ơn chân thành đến TS. Trần Công Ân đã nhiệt tình hướng dẫn và giúp đỡ em hoàn thành tốt đề tài luận văn tốt nghiệp.

Xin gửi lời cảm ơn đến Ban Giám hiệu, toàn thể thầy cô của trường Đại học Cần Thơ nói chung và khoa CNTT và TT nói riêng, đã giảng dạy, giúp đỡ em hoàn thành tốt các môn học của chương trình đào tạo.

Xin cảm ơn gia đình đã tạo điều kiện tốt nhất cho em được học tập tại trường Đại học Cần Thơ, đã động viên em vượt qua những lúc khó khăn trong quá trình học tập.

Cảm ơn bạn bè đã đồng hành và giúp đỡ em, cũng như đưa ra những lời khuyên, cách giải quyết vấn đề khi em gặp khó khăn trong học tập.

Cuối cùng, em xin gửi lời cảm ơn đến cộng đồng lập trình viên trong nước và quốc tế đã giúp đỡ em giải quyết những vấn đề khó khăn gặp phải khi thực hiện đề tài.

Em xin trân trọng cảm ơn.

Cần Thơ, ngày 17 tháng 6 năm 2020

Tác giả luận văn

Nguyễn Thanh Khoa

Mục lục

Phần 1. Giới thiệu.....	1
1. Đặt vấn đề	2
2. Những nghiên cứu liên quan.....	2
2.1. Ứng dụng IKEA Place (iOS).....	2
2.2. Ứng dụng Curate by Sotheby's Realty - AR for Real Estate	3
3. Mục tiêu đề tài	4
4. Đối tượng và phạm vi nghiên cứu	6
4.3. Đối tượng nghiên cứu.....	6
4.4. Phạm vi nghiên cứu	6
5. Phương pháp nghiên cứu	6
6. Nội dung nghiên cứu.....	7
7. Bố cục quyền luận văn.....	10
Phần 2. Nội dung.....	11
Chương 1. Đặc tả yêu cầu.....	12
1.1. Yêu cầu chức năng	12
1.1.1. Chức năng của RESTful API server	12
1.1.2. Chức năng của ứng dụng web client	12
1.1.3. Chức năng của ứng dụng AR mobile.....	13
1.2. Yêu cầu phi chức năng	14
1.2.1. Giao diện	14
1.2.2. Phần cứng.....	14
Chương 2. Thiết kế giải pháp	15
2.1. Cơ sở lý thuyết	15
2.1.1. Giới thiệu về API và RESTful API.....	15
2.1.2. Django framework	16
2.1.3. Django rest framework.....	18
2.1.4. Giới thiệu về JWT authentication	19

2.1.5. Hệ quản trị cơ sở dữ liệu MySQL.....	21
2.1.6. Thư viện ReactJS	21
2.1.7. Thư viện Redux.....	23
2.1.8. Thư viện Axios.....	24
2.1.9. Thư viện React UI – Ant design	24
2.1.10. Game engine Unity 3D	25
2.1.11. Thư viện ARCore.....	30
2.1.12. ARCore SDK for Unity.....	38
2.1.13. Phần mềm hỗ trợ	40
2.1.14. Nội thất dựa trên phong thủy	42
2.2. Thiết kế hệ thống	44
2.2.1. RESTful API	44
2.2.2. Ứng dụng web quản lý	50
2.2.3. Mobile app	57
Chương 3. Cài đặt giải pháp	67
3.1. REST API server	67
3.1.1. Cài đặt model trong django	67
3.1.2. Cài đặt serializer trong DRF	68
3.1.3. Cài đặt View trong DRF	69
3.1.4. Routing trong DRF.....	71
3.1.5. Xác thực request bằng JWT	71
3.1.6. Xử lý yêu cầu	73
3.2. Web client app.....	74
3.2.1. Đăng nhập – Xác thực người dùng	74
3.2.2. Sử dụng React-Router để routing cho ứng dụng.....	76
3.2.3. Thao tác sử dụng CRUD.....	77
3.3. AR mobile app.....	80
3.3.1. Lưu trữ model và hình ảnh.....	80

3.3.2. Đóng gói dữ liệu bằng Scriptable Object.....	81
3.3.3. Tính toán kích thước model	83
3.3.4. Cập nhật dữ liệu	85
3.3.5. Liệt kê danh sách loại phòng	88
3.3.6. Liệt kê danh sách loại nội thất dựa theo loại phòng	89
3.3.7. Tìm kiếm nội thất.....	90
3.3.8. Liệt kê danh sách nội thất	92
3.3.9. Liệt kê nội thất yêu thích.....	94
3.3.10. Gợi ý nội thất theo phong thủy	95
3.3.11. Kéo thả nội thất	100
3.3.12. Hệ thống điều chỉnh mô hình nội thất.....	102
3.3.13. Chụp ảnh màn hình	104
3.3.14. Xóa nội thất.....	105
3.3.15. Làm mới phân cảnh.....	106
3.3.16. Một số tính năng phụ được hỗ trợ.....	106
3.4. Mô hình nội thất 3D	106
Chương 4. Đánh giá kiểm thử.....	108
4.1. Mục tiêu kiểm thử	108
4.2. Kịch bản kiểm thử	108
4.3. Đánh giá kiểm thử	111
Phần 3. Kết luận	127
1. Kết quả đạt được	128
1.1. Kết quả	128
1.2. Hạn chế.....	128
2. Hướng phát triển	129

Danh mục hình ảnh

Hình 1.1.1. Ứng dụng IKEA Place trên App Store.....	3
Hình 1.1.2. Ứng dụng Curate by Sotheby's Realty trên Google Play	4
Hình 1.1.3. Sơ đồ hoạt động của các thành phần hệ thống	5
Hình 2.2.1. Mô hình hoạt động cơ bản của REST	16
Hình 2.2.2. Logo của thư viện django.....	16
Hình 2.2.3. Mô hình MVT và MVC	17
Hình 2.2.4. Logo của django rest framework	18
Hình 2.2.5. Sơ đồ hoạt động của DRF	19
Hình 2.2.6. Sơ đồ cách thức hoạt động của JWT	20
Hình 2.2.7. Logo của MySQL.....	21
Hình 2.2.8. Logo của thư viện React	21
Hình 2.2.9. Logo của thư viện Redux	23
Hình 2.2.10. Logo Ant design và React.....	24
Hình 2.2.11. Logo của Unity 3D game engine	26
Hình 2.2.12. Addressables trong Unity packages.....	30
Hình 2.2.13. Logo của thư viện ARCore	31
Hình 2.2.14. Các hoạt động của motion tracking (Nguồn: ARCore docs)	33
Hình 2.2.15. Mặt phẳng được phát hiện bởi ARCore (Nguồn: ARCore docs).....	34
Hình 2.2.16. Ước lượng độ sáng của ARCore (Nguồn: ARCore docs).....	35
Hình 2.2.17. Mặt phẳng ARCore trong Unity.....	39
Hình 2.2.18. Logo của ứng dụng Autodesk Maya	40
Hình 2.2.19. Logo ứng dụng Substance Painter	41
Hình 2.2.20. Biểu tượng âm dương.....	42
Hình 2.2.21. Quan hệ tương sinh – tương khắc theo ngũ hành	43
Hình 2.2.22. Lược đồ cơ sở dữ liệu	44
Hình 2.2.23. Kiến trúc các thành phần của REST API.....	49
Hình 2.2.24. Giao diện màn hình đăng nhập	50

Hình 2.2.25. Giao diện trang chủ ứng dụng web quản lý	51
Hình 2.2.26. Giao diện quản lý phiên bản	52
Hình 2.2.27. Giao diện tải lên asset bundles của Unity	52
Hình 2.2.28. Giao diện quản lý nội thất	53
Hình 2.2.29. Giao diện tạo mới nội thất.....	54
Hình 2.2.30. Giao diện cập nhật thông tin nội thất	54
Hình 2.2.31. Một phần giao diện xem chi tiết nội thất	55
Hình 2.2.32. Kiến trúc các thành phần của ứng dụng web quản lý	56
Hình 2.2.33. Giao diện cập nhật ứng dụng AR mobile.....	57
Hình 2.2.34. Giao diện hướng dẫn dò tìm mặt phẳng.....	58
Hình 2.2.35. Giao diện menu của ứng dụng AR mobile.....	59
Hình 2.2.36. Giao diện liệt kê loại phòng của ứng dụng AR mobile.....	60
Hình 2.2.37. Giao diện xem thông chi tiết nội thất của ứng dụng AR mobile	61
Hình 2.2.38. Giao diện nhập thông tin người dùng	62
Hình 2.2.39. Kiến trúc các thành phần của ứng dụng AR mobile	66
Hình 2.3.1. Cài đặt bảng RoomType bằng model trong django	67
Hình 2.3.2. Cài đặt RoomTypeView trong DRF	69
Hình 2.3.3. Cài đặt router cho room-types trong DRF	71
Hình 2.3.4. Kết quả về khi tạo JWT token thành công	72
Hình 2.3.5. Lưu đồ xử lý yêu cầu của REST API server.....	73
Hình 2.3.6. Form đăng nhập của ứng dụng web quản lý	74
Hình 2.3.7. Lưu đồ xử lý của chuyển hướng đăng nhập tới trang chủ	74
Hình 2.3.8. Lưu đồ xử lý thực hiện yêu cầu đăng nhập của ứng dụng web	75
Hình 2.3.9. Lưu đồ xử lý routing của ứng dụng web.....	76
Hình 2.3.10. Component Table liệt kê danh sách nội thất.....	77
Hình 2.3.11. Lưu đồ xử lý liệt kê danh sách nội thất.....	77
Hình 2.3.12. Lưu đồ xử lý tạo mới nội thất	78
Hình 2.3.13. FurnitureScriptableObject được đánh dấu là “addressable”	80

Hình 2.3.14. Cấu hình addressable để đóng gói dữ liệu	81
Hình 2.3.15. Danh sách các ScriptableObject được đánh dấu là “addressable”	82
Hình 2.3.16. Phân cảnh dùng để tính toán kích thước mô hình	83
Hình 2.3.17. Tính toán kích thước mô hình bằng phân cảnh FurnitureCreation	84
Hình 2.3.18. Lưu đồ của hàm quản lý cập nhật dữ liệu của ứng dụng AR mobile...	85
Hình 2.3.19. Lưu đồ xử lý của hàm kiểm tra cập nhật.....	86
Hình 2.3.20. Lưu đồ xử lý của hàm tải xuống dữ liệu cập nhật.....	87
Hình 2.3.21. Menu liệt kê danh sách loại phòng.....	88
Hình 2.3.22. Lưu đồ xử lý liệt kê danh sách loại phòng	88
Hình 2.3.23. Menu liệt kê danh sách loại nội thất theo loại phòng	89
Hình 2.3.24. Lưu đồ xử lý liệt kê loại nội thất theo loại phòng	90
Hình 2.3.25. Thanh tìm kiếm trên menu	90
Hình 2.3.26. Lưu đồ xử lý tìm kiếm gần đúng theo tên nội thất.....	91
Hình 2.3.27. Danh sách nội thất trong phân cảnh AR	92
Hình 2.3.28. Lưu đồ xử lý hiển thị danh sách nội thất.....	93
Hình 2.3.29. Biểu tượng nội thất yêu thích trong popup thông tin chi tiết.....	94
Hình 2.3.30. Lưu đồ xử lý liệt kê danh sách nội thất yêu thích	94
Hình 2.3.31. Thông tin phong thủy của nội thất	95
Hình 2.3.32. Lưu đồ xử lý gợi ý nội thất dựa trên phong thủy	96
Hình 2.3.33. Popup cho phép người dùng nhập thông tin cá nhân	97
Hình 2.3.34. Lưu đồ xử lý của hàm tính can – chi – mệnh.....	98
Hình 2.3.35. Lưu đồ xử lý của hàm tính độ lợi phong thủy đồ nội thất	99
Hình 2.3.36. Bảng tiêu chí độ lợi dựa trên ngũ hành tương sinh – tương khắc	100
Hình 2.3.37. Thao tác kéo thả nội thất từ danh sách nội thất.....	100
Hình 2.3.38. Lưu đồ xử lý thao tác kéo thả nội thất	101
Hình 2.3.39. Lưu đồ xử lý của hàm cập nhật vị trí mô hình khi kéo thả nội thất ...	102
Hình 2.3.40. Ảnh chụp màn hình của 3 mô hình nội thất 2 ghế – 1 mặt bàn	104
Hình 2.3.41. Lưu đồ xử lý chụp ảnh màn hình	105

Hình 2.3.42. Trang chủ của TURBOSQID	106
Hình 2.4.1. Giao diện đăng nhập ứng dụng web quản trị	112
Hình 2.4.2. Giao diện trang chủ của ứng dụng web quản trị	112
Hình 2.4.3. Giao diện upload addressables của ứng dụng web quản trị	113
Hình 2.4.4. Giao diện quản lý đồ nội thất của ứng dụng web quản trị	113
Hình 2.4.5. Giao diện quản lý loại nội thất của ứng dụng web quản trị	114
Hình 2.4.6. Giao diện quản lý màu sắc nội thất của ứng dụng web quản trị	114
Hình 2.4.7. Giao diện quản lý vật liệu nội thất của ứng dụng web quản trị	115
Hình 2.4.8. Giao diện quản lý loại phòng của ứng dụng web quản trị	115
Hình 2.4.9. Giao diện quản lý phiên bản dữ liệu json của ứng dụng web quản trị	116
Hình 2.4.10. Giao diện kiểm tra cập nhật của ứng dụng AR	117
Hình 2.4.11. Giao diện dò tìm mặt phẳng của ứng dụng AR	118
Hình 2.4.12. Mặt phẳng phát hiện được của ứng dụng AR	119
Hình 2.4.13. Giao diện menu chính của ứng dụng AR	120
Hình 2.4.14. Giao diện menu cấp 2 và 3 của ứng dụng AR	121
Hình 2.4.15. Giao diện liệt kê đồ nội thất của ứng dụng AR	122
Hình 2.4.16. Giao diện kéo thả đặt nội thất của ứng dụng AR	123
Hình 2.4.17. Phân cảnh AR khi ẩn lưới và điểm đặc trưng	124
Hình 2.4.18. Giao diện thông tin số mệnh người dùng theo phong thủy	125
Hình 2.4.19. Giao diện thông tin sản phẩm nội thất của ứng dụng AR	126

Danh mục bảng

Bảng 1.1.1. Đối tượng nghiên cứu của đề tài.....	9
Bảng 2.2.1. Các tác vụ cơ bản của REST dựa trên phương thức HTTP.....	15
Bảng 2.2.2. Các giấy phép của Unity 3D.....	27
Bảng 2.2.3. Danh sách URI được cung cấp bởi REST API server.....	48
Bảng 2.2.4. Một số status code thông dụng của giao thức HTTP.....	49
Bảng 2.2.5. Danh sách chức năng các thao tác tay trên ứng dụng AR mobile	63
Bảng 2.2.6. Danh sách các nút chức năng chính trên ứng dụng AR mobile	64
Bảng 2.2.7. Danh sách một số nút chức năng khác trên ứng dụng AR mobile	65
Bảng 2.3.1. Danh sách các model được cài đặt.....	67
Bảng 2.3.2. Cài đặt RoomTypeSerializer trong DRF	68
Bảng 2.3.3. Danh sách các Serializer được cài đặt trong DRF	69
Bảng 2.3.4. Danh sách các View được cài đặt trong DRF.....	70
Bảng 2.3.5. Danh sách URL của room-types được Default Router hỗ trợ	71
Bảng 2.4.1. Kết quả kiểm thử kịch bản.....	111

Danh mục từ viết tắt

AR	Augmented Reality
SIR	Sotheby's International Realty
CSR	Curate by Sotheby's Realty
API	Application Programming Interface
REST	REpresentational State Transfer
SDK	Software Development Kit
HTTP	HyperText Transfer Protocol
CRUD	Create – Read – Update – Delete
JWT	Json Web Token
URL	Uniform Resource Locator
URI	Uniform Resource Identifier
JSON	JavaScript Object Notation
XML	eXtensible Markup Language
SQL	Structured Query Language
MVT	Model – View – Template
MVC	Model – View – Controller
DRF	Django Rest Framework
MIT	Massachusetts Institute of Technology
UI	User Interface
IDE	Integrated Development Environment
2D/3D	2 demensional/3 demensional

Abstract

In recent years, augmented reality – a technology allows integrating virtual contents into the real world, increasingly focused and developed strongly. Augmented reality is used in many different fields such as education, medicine, military and e-commerce. Furniture retail – a piece of e-commerce field is approaching this technology to catch up with the booming trend of information technology. The graduation thesis is a research and application of augmented reality to build a furniture retail application on Android operating system. This application is designed and built by the Unity 3D game engine and the ARCore library. The result of this graduation thesis is a furniture retail application that uses augmented reality and feng shui conception to support suggesting and designing furnishing. The application name is AR Furniture.

Keywords: Unity, ARCore, furniture, augmented reality, fengshui, AR Furniture.

Tóm tắt

Những năm gần đây, thực tế tăng cường – một công nghệ cho phép tích hợp nội dung ảo vào thế giới thực, càng ngày càng được chú trọng và phát triển mạnh mẽ. Thực tế tăng cường được ứng dụng vào nhiều lĩnh vực khác nhau như giáo dục, y học, quân đội và thương mại điện tử. Bán lẻ nội thất – một mảng của lĩnh vực thương mại điện tử, đang dần dần tiếp cận với công nghệ này để bắt kịp xu hướng phát triển vượt bậc của công nghệ thông tin. Luận văn tốt nghiệp là công trình nghiên cứu và ứng dụng công nghệ thực tế tăng cường để xây dựng một ứng dụng bán đồ nội thất trên hệ điều hành Android. Ứng dụng được thiết kế và xây dựng bằng game engine Unity 3D và thư viện ARCore. Kết quả của luận văn tốt nghiệp là một ứng dụng bán đồ nội thất sử dụng thực tế tăng cường và quan niệm phong thủy để hỗ trợ gợi ý và thiết kế trang trí nội thất. Ứng dụng được đặt tên là AR Furniture.

Từ khóa: Unity, ARCore, nội thất, thực tế tăng cường, phong thủy, AR Furniture.

Phần 1. Giới thiệu

1. Đặt vấn đề

Những năm gần đây, những từ khóa liên quan tới công nghệ thực tế tăng cường (AR – Augmented Reality) ngày càng được quan tâm hơn, những ứng dụng liên quan cũng được phát hành nhiều hơn. Trong số đó, phải kể đến sự bùng nổ của một tựa game AR trên mobile là Pokemon Go vào năm 2016. Tựa game này đã làm mưa làm gió khắp thế giới một khoảng thời gian sau khi được ra mắt. Đó là một trong những sự thành công nổi bật mà công nghệ AR từng đạt được. Công nghệ này nếu được ứng dụng ngày càng phổ biến hơn nữa vào cuộc sống thì hứa hẹn sẽ là một mảnh đất màu mỡ đầy tiềm năng và triển vọng dành cho các nhà phát triển AR trong tương lai.

Có thể thấy được những ứng dụng của AR được sử dụng ngày càng phổ biến trong cuộc sống con người như dùng trong giảng dạy, y khoa và quân đội. Nếu có thể ứng dụng AR vào đời sống hằng ngày của con người thì càng tuyệt vời hơn.

Nhận thấy được tiềm năng của việc bán đồ nội thất sử dụng công nghệ AR. Một ứng dụng AR có thể giúp người dùng lựa chọn những món đồ phù hợp với căn phòng của mình một cách trực quan (xem trước, đặt, di chuyển mô hình đồ nội thất trong căn phòng) sẽ thúc đẩy việc mua sắm của người dùng.

Ngoài ra, văn hóa người phương Đông nói chung và người Việt nói riêng rất coi trọng quan niệm phong thủy trong đời sống. Trang trí đồ nội thất cũng là một mảng quan trọng trong lĩnh vực này. Nếu có thể kết hợp giữa văn hóa cổ truyền và khoa học hiện đại với nhau thì việc gợi ý một món đồ nội thất với đầy đủ thông tin về phong thủy cũng như cái nhìn trực quan về món đồ đó ở một không gian cụ thể sẽ giúp cho người dùng lựa chọn dễ dàng hơn. Do vậy, cần phải xây dựng một ứng dụng AR bán đồ nội thất để hiện thực hóa ý tưởng này.

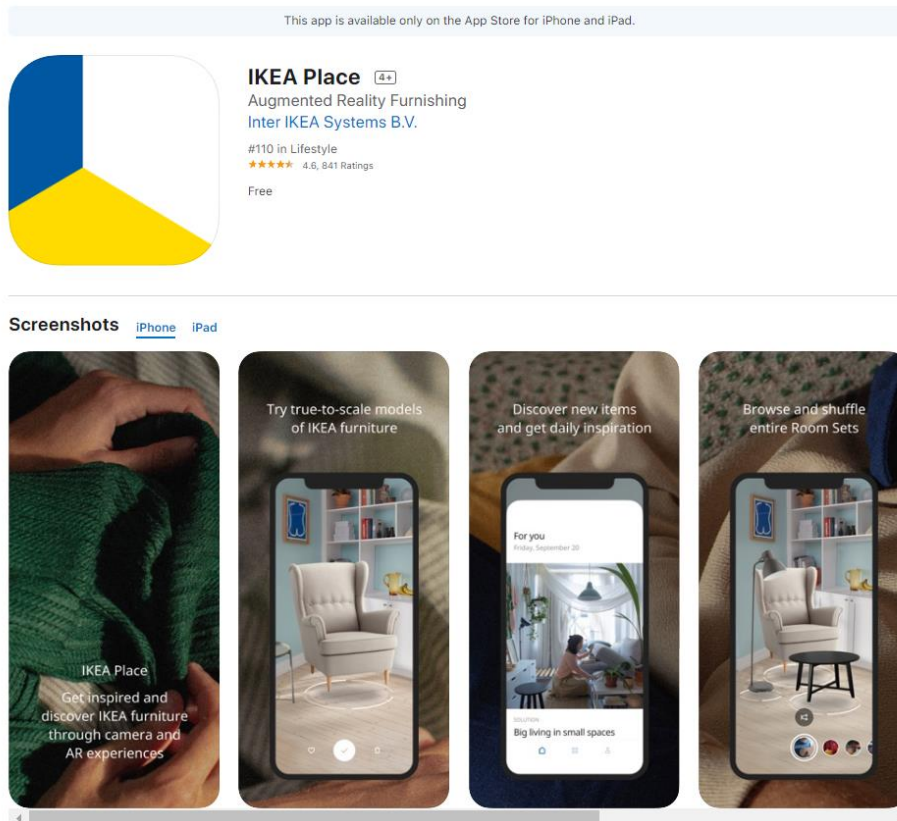
2. Những nghiên cứu liên quan

2.1. Ứng dụng IKEA Place (iOS)

IKEA là một tập đoàn sản xuất đồ nội thất lớn nhất thế giới, có tiền thân là một doanh nghiệp tư nhân của Thụy Điển. Ingvar Kamprad là người sáng lập ra IKEA vào năm 1943 lúc ông mới 17 tuổi. IKEA cũng là từ được viết tắt từ tên của ông.

IKEA Place là một ứng dụng bán đồ nội thất nền tảng mobile của IKEA được phát hành trên App store vào năm 2017. Ứng dụng này có sử dụng công nghệ AR (ARKit của iOS) cho phép người dùng đặt thử mô hình 3D của món đồ nội thất từ nhà cung cấp vào trong không gian thực của người dùng.

App Store Preview



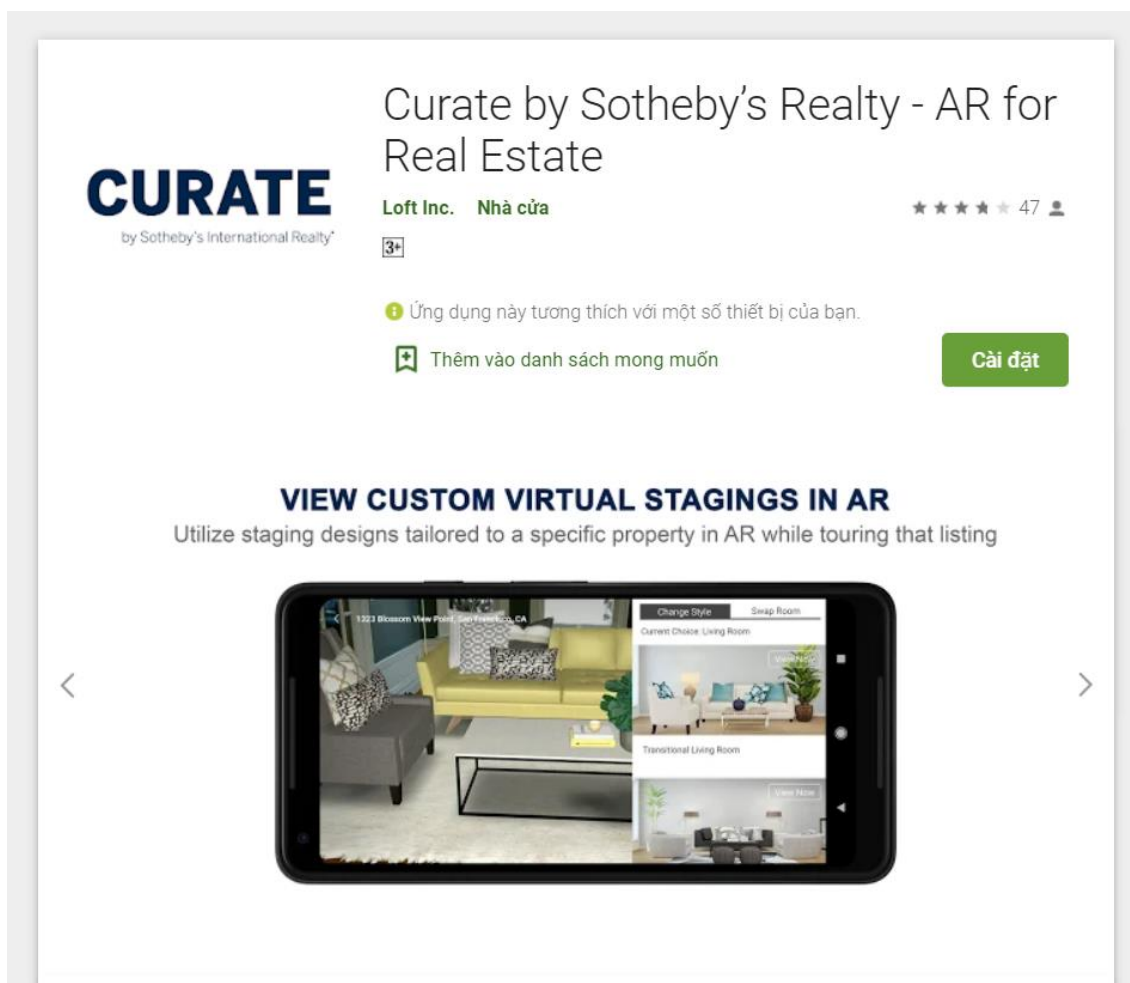
Hình 1.1.1. Ứng dụng IKEA Place trên App Store

Tuy nhiên ứng dụng này chỉ hỗ trợ nền tảng iOS 11 trở lên. Trên hệ điều hành android vẫn chưa được phát triển.

2.2. Ứng dụng Curate by Sotheby's Realty - AR for Real Estate

Sotheby's Realty (hay Sotheby's International Realty – SIR) là một thương hiệu bất động sản cao cấp được sáng lập năm 1976 bởi Sotheby's Auction House. SIR hoạt động như là một thị trường môi giới và nhượng quyền thương mại bất động sản của các khu dân cư cao cấp.

Ứng dụng Curate by Sotheby's Realty – AR for Real Estate (CSR) là một ứng dụng trải nghiệm thực tế tăng cường được tạo bởi SIR. CSR cho phép khách hàng hình dung ngôi nhà của họ trước khi mua, từ đó tăng kinh nghiệm mua nhà. SIR là công ty bất động sản đầu tiên phát hành một ứng dụng thực tế tăng cường, hỗ trợ trải nghiệm mua bán cho các cộng tác viên bán hàng và người tiêu dùng. Khi một ngôi nhà mới được mua, CSR sẽ thông qua công nghệ AR giúp người dùng thiết kế và mua sắm những món đồ nội thất cho ngôi nhà mới của họ.



Hình 1.1.2. Ứng dụng Curate by Sotheby's Realty trên Google Play

Ứng dụng này hỗ trợ cả 2 nền tảng Android (phiên bản 7 trở lên) và iOS (phiên bản 11 trở lên). Tuy nhiên, ứng dụng chỉ hỗ trợ đặt cả bộ sưu tập theo chủ đề vào không gian chứ không hỗ trợ đặt đơn lẻ từng món đồ.

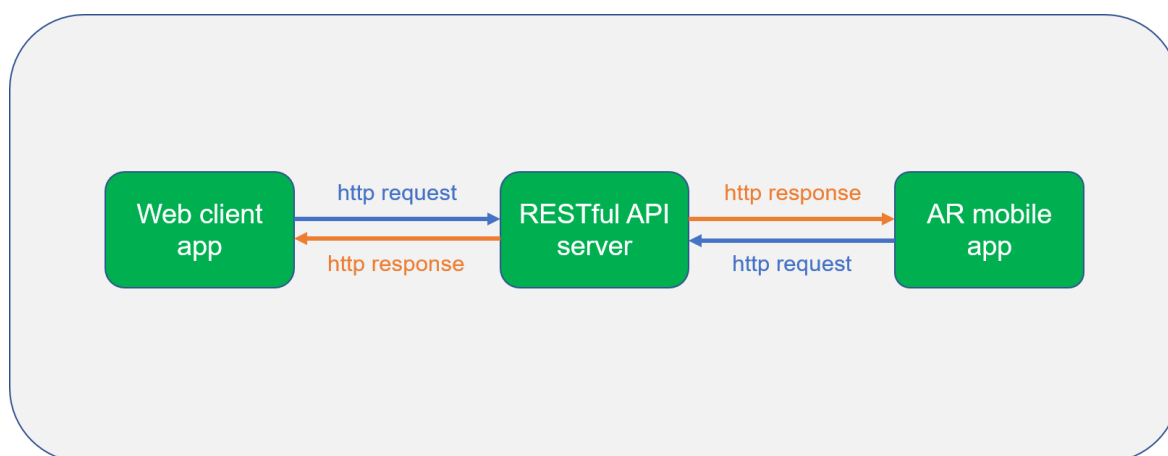
3. Mục tiêu đề tài

Mục tiêu chính của đề tài là xây dựng một ứng dụng AR bán đồ nội thất trên hệ điều hành android. Ứng dụng này sẽ bao gồm một số tính năng cơ bản:

- Chọn mô hình nội thất và kéo thả đặt vào phân cảnh trong ứng dụng (căn phòng của người dùng trong không gian thực)
- Điều khiển mô hình nội thất trong phân cảnh: dịch chuyển, xoay, xóa
- Gợi ý nội thất dựa trên phong thủy

- Liệt kê nội thất: dựa theo loại phòng và loại nội thất
- Tìm kiếm nội thất theo tên gần đúng
- Chụp ảnh màn hình
- Làm mới phân cảnh
- Quản lý nội thất yêu thích: thêm vào danh sách yêu thích hoặc bỏ thêm
- Tự động cập nhật dữ liệu

Ứng dụng AR sẽ hoạt động theo mô hình client server, dữ liệu được lưu trữ trên server, khi đó ứng dụng AR sẽ truy cập và tải xuống dữ liệu mới nhất để sử dụng về sau (offline). Để quản lý dữ liệu ở trên server thì sẽ có một ứng dụng web làm nhiệm vụ quản trị dữ liệu. Nhìn chung, mục tiêu cuối cùng của đề tài là xây dựng một hệ thống gồm 3 thành phần: API server, web client và AR mobile app.



Hình 1.1.3. Sơ đồ hoạt động của các thành phần hệ thống

Hệ thống sẽ gồm một RESTful API server đóng vai trò làm trung tâm lưu trữ dữ liệu dùng chung. Nhận các request truy xuất, tạo, cập nhật, xóa dữ liệu và tiến hành xử lý, sau đó trả về response hoặc thông báo lỗi tương ứng.

Web client đóng vai trò là một trang quản trị của admin, cho phép admin quản lý cập nhật, thêm, sửa, xóa dữ liệu của hệ thống. Người quản trị sẽ thao tác trực tiếp với ứng dụng web, ứng dụng sẽ xử lý và gửi request về API server và đợi kết quả trả về, xử lý hiển thị kết quả hoặc thông báo lỗi.

AR mobile app là ứng dụng người dùng đầu cuối, truy xuất thông tin từ server thông qua API, xử lý hiển thị, thao tác với người dùng. Ứng dụng có thể sử dụng dữ liệu đã tải về trước đó (offline). Ứng dụng này sẽ có 2 tính năng cốt lõi là kéo thả,

tính chỉnh (di chuyển, xoay, xóa) mô hình 3D của đồ nội thất trong không gian thực và hỗ trợ cung cấp thông tin phong thủy của đồ nội thất.

4. Đối tượng và phạm vi nghiên cứu

4.3. Đối tượng nghiên cứu

Đối tượng nghiên cứu chính của đề tài là sử dụng thư viện “ARCore SDK for Unity” để xây dựng nên một ứng dụng AR bán đồ nội thất trên hệ điều hành android. Ngoài ra, cũng sẽ nghiên cứu thêm về việc xây dựng một kiến trúc hệ thống ứng dụng sử dụng máy chủ dữ liệu (API server) và công cụ quản lý (Web client). Đối tượng nghiên cứu đầy đủ sẽ là một hệ thống gồm 3 thành phần:

- REST API server: xây dựng một server cung cấp các thao tác tạo, thêm, sửa, xóa (CRUD) dữ liệu của hệ thống thông qua HTTP request.
- Web client: xây dựng một ứng dụng web trên desktop giao tiếp với server thông qua REST API, nhằm mục đích quản lý dữ liệu cốt lõi của hệ thống.
- AR mobile app: xây dựng một ứng dụng AR trên hệ điều hành android cho phép truy xuất dữ liệu từ API server, lưu trữ và sử dụng dữ liệu. Ứng dụng này sẽ là ứng dụng cốt lõi của hệ thống, mang trải nghiệm AR vào hỗ trợ gợi ý lựa chọn đồ nội thất.

4.4. Phạm vi nghiên cứu

Phạm vi nghiên cứu của đề tài chủ yếu là làm việc với phần mềm game engine Unity 3D, phần mềm thiết kế đồ họa Autodesk Maya và thư viện ARCore để tạo ra ứng dụng AR bán đồ nội thất. Ngoài ra, sẽ sử dụng một vài thư viện backend như Django, Django rest framework để xây dựng nhanh một hệ thống RESTful API và một số thư viện frontend như Reactjs, Axios để giao tiếp với API server, Antd để tạo giao diện người dùng web.

5. Phương pháp nghiên cứu

Nội dung mà đề tài hướng đến còn khá mới mẻ, do đó phương pháp nghiên cứu chủ yếu là tìm hiểu thông qua tài liệu trên trang chủ của các thư viện, thông qua các video trên YouTube và làm các ví dụ trực tiếp. Từ đó, sẽ vận dụng các kiến thức đã tiếp thu được vào thực hiện đề tài. Tương ứng với mỗi thành phần của hệ thống sẽ có những phương pháp nghiên cứu khác nhau:

- API server: nghiên cứu cơ bản về các thư viện dùng để xây dựng nhanh một hệ thống RESTful API gồm django, django rest framework, JWT authentication và hệ quản trị cơ sở dữ liệu MySQL.
- Web client: nghiên cứu cơ bản về ứng dụng web sử dụng các thư viện react, redux, axios và ant design.
- AR mobile app: nghiên cứu về game engine Unity 3D, thư viện ARCore, ARCore SDK for Unity, Autodesk Maya và lĩnh vực phong thủy theo ngũ hành.

6. Nội dung nghiên cứu

Nội dung nghiên cứu được liệt kê phía dưới đây.

STT	Đối tượng	Nội dung
1	API và RESTful API	– Tìm hiểu về API và RESTful API
2	Django	– Tìm hiểu về django – Xây dựng model dữ liệu (các bảng của cơ sở dữ liệu) của đề tài trong django
3	Django rest framework (DRF)	– Tìm hiểu về django rest framework – Xây dựng các serializers, views, permissions và routing trong DRF – Xây dựng RESTful API server
4	JWT authentication	– Tìm hiểu về JWT – Tạo token xác thực người dùng bằng JWT trong DRF
5	MySQL	– Tìm hiểu hệ quản trị cơ sở dữ liệu MySQL – Cấu hình MySQL trong django
6	Reactjs	– Tìm hiểu về reactjs: components, state và props

		– Sử dụng webhook trong react – Routing bằng reactjs và chuyển hướng đăng nhập
7	Redux	– Tìm hiểu về redux: state, action, container, dispatcher – Sử dụng redux vào ứng dụng react
8	Axios	– Tìm hiểu về axios – Trao đổi với API server thông qua axios – Tạo mới access token tự động bằng axios
9	Thư viện UI Ant design	– Tìm hiểu về Antd và các components mà thư viện cung cấp – Xây dựng các components của ứng dụng, kết hợp với redux để chia sẻ các state dùng chung – Hoàn thiện các tính năng CRUD service của một ứng dụng quản lý
10	Game engine Unity 3D	– Tìm hiểu về game engine Unity 3D – Tìm hiểu các thành phần cơ bản của game engine: gameObject, component, script, transform, prefab... – Tạo dựng UI cho ứng dụng – Tạo hiệu ứng cho UI – Sử dụng thư viện Unity addressables để đóng gói các mô hình 3D – Sử dụng Unity Web Request để giao tiếp với API server
11	ARCore	– Tìm hiểu về thư viện ARCore – Tìm hiểu về khả năng của thư viện: theo dõi

		chuyển động, nhận thức về không gian, ước lượng ánh sáng – Xác định các thiết bị hỗ trợ ARCore
12	ARCore SDK for Unity	– Tìm hiểu về API của ARCore trên nền tảng game engine Unity 3D – Tìm hiểu các khái niệm cơ bản: session, detected plane, anchor – Xây dựng hệ thống kéo thả, tinh chỉnh mô hình trong không gian thực
13	Autodesk Maya	– Tìm hiểu về autodesk Maya – Tìm hiểu khái niệm pivot, material và texture – Chỉnh sửa pivot và texturing các mô hình nội thất
14	Substaince Painter 3D	– Tìm hiểu về Substaince Painter 3D – Texturing model – Xuất các kết cấu vật liệu của model 3D để nhập vào Maya
15	Phong thủy	– Tìm hiểu về phong thủy theo ngũ hành – Tính lịch âm dương, can, chi, mệnh theo ngũ hành – Tính ngũ hành theo màu sắc và vật liệu của đồ nội thất – Xây dựng hệ thống gợi ý đồ nội thất theo phong thủy

Bảng 1.1.1. Đối tượng nghiên cứu của đề tài

7. Bố cục quyển luận văn

Bố cục luận văn gồm các phần và các chương sau:

Phần 1 – Giới thiệu, gồm:

- ✓ Đặt vấn đề
- ✓ Những nghiên cứu liên quan
- ✓ Mục tiêu đề tài
- ✓ Đối tượng và phạm vi nghiên cứu
- ✓ Nội dung nghiên cứu
- ✓ Bố cục quyển luận văn

Phần 2 – Nội dung, gồm:

Chương 1. Đặc tả yêu cầu

- ✓ Yêu cầu chức năng
- ✓ Yêu cầu phi chức năng

Chương 2. Thiết kế giải pháp

- ✓ Cơ sở lý thuyết
- ✓ Thiết kế hệ thống

Chương 3. Cài đặt giải pháp

- ✓ REST API server
- ✓ Ứng dụng web quản lý
- ✓ Ứng dụng AR mobile

Chương 4. Đánh giá kiểm thử

- ✓ Mục tiêu kiểm thử
- ✓ Kịch bản kiểm thử
- ✓ Đánh giá kiểm thử

Phần 3 – Kết luận, gồm:

- ✓ Kết luận
- ✓ Hướng phát triển

Phần 2. Nội dung

Chương 1. Đặc tả yêu cầu

1.1. Yêu cầu chức năng

1.1.1. Chức năng của RESTful API server

Xác thực người dùng bằng token: tương ứng với những request cần xác thực người dùng hoặc chỉ admin mới có quyền thực hiện request đó, hệ thống sẽ kiểm tra token được gửi kèm header để xác thực người dùng. Ngoài ra, cũng có thể tạo mới token bằng email và mật khẩu.

Quản lý dữ liệu model: cho phép gửi các request tạo, cập nhật, xóa dữ liệu của các model loại phòng, loại nội thất, màu sắc, vật liệu nội thất và nội thất với mức độ xác thực là người quản trị. Những request truy xuất dữ liệu thì không cần xác thực.

Quản lý phiên bản dữ liệu json: cho phép gửi request tạo một phiên bản dữ liệu json mới với mức độ xác thực là người quản trị. Những request không được xác thực chỉ có quyền truy xuất.

Quản lý dữ liệu đóng gói bởi Unity addressables: cho phép gửi request tải lên dữ liệu được đóng gói bằng Unity addressables với mức độ xác thực là người quản trị. Những người dùng không được xác thực chỉ được quyền tải xuống.

1.1.2. Chức năng của ứng dụng web client

Đăng nhập: cho phép người quản trị dùng nhập địa chỉ email và mật khẩu để đăng nhập. Nếu đăng nhập không thành công thì thông báo lỗi.

Quản lý dữ liệu model: tính năng chỉ được cung cấp khi người quản trị đăng nhập thành công. Tương ứng với mỗi model dữ liệu loại phòng, loại nội thất, màu sắc, vật liệu nội thất và nội thất, ứng dụng sẽ cung cấp các công cụ liệt kê, tìm kiếm, tạo, cập nhật và xóa dữ liệu.

Quản lý phiên bản dữ liệu json: tính năng chỉ được cung cấp khi người quản trị đăng nhập thành công. Liệt kê toàn bộ phiên bản dữ liệu json cùng với tùy chọn tạo mới phiên bản dữ liệu json.

Quản lý dữ liệu được gói bởi Unity addressables: tính năng chỉ được cung cấp khi người quản trị đăng nhập thành công. Cho phép tải lên dữ liệu được đóng gói bởi Unity addressables.

1.1.3. Chức năng của ứng dụng AR mobile

Kiểm tra cập nhật dữ liệu: kiểm tra phiên bản dữ liệu mô hình và dữ liệu json trên thiết bị so với trên server, tiến hành tải dữ liệu mới nhất về và lưu trữ sử dụng offline.

Liệt kê danh sách đồ nội thất: liệt kê những món đồ nội thất theo loại nội thất tương ứng với loại phòng.

Tìm kiếm đồ nội thất: tìm kiếm đồ nội thất theo tên gần đúng.

Gợi ý đồ nội thất theo phong thủy: gợi ý những đồ vật phù hợp với phong thủy theo mệnh của người dùng.

Quản lý nội thất yêu thích: quản lý liệt kê, thêm, xóa nội thất yêu thích.

Kéo thả đồ nội thất đặt vào không gian người dùng: kéo thả mô hình 3D của một món đồ nội thất bất kì và đặt vào không gian thực của người dùng.

Tinh chỉnh đồ nội thất trong không gian người dùng: cho phép người dùng chọn món đồ nội thất trong không gian, di chuyển, xoay đồ nội thất.

Quản lý đồ nội thất đã đặt vào không gian người dùng: cho phép xóa một hoặc tất cả các món đồ nội thất đã đặt vào không gian của người dùng.

Quản lý hiển thị mặt phẳng, điểm: cho phép người dùng tùy chọn ẩn hoặc hiện mặt phẳng, điểm đặc trưng được phát hiện bởi ARCore.

Chụp ảnh màn hình: tính năng chụp ảnh mô hình đã đặt vào không gian của người dùng, cho phép người dùng xem lại sau.

Làm mới phân cảnh: tạo mới phiên làm việc của ARCore, tiến hành nhận dạng lại từ đầu không gian của người dùng.

1.2. Yêu cầu phi chức năng

1.2.1. Giao diện

Giao diện đơn giản, dễ sử dụng.

Đối với ứng dụng web quản lý thì giao diện sẽ thống nhất theo một mẫu chung để hướng tới tính quen thuộc, dễ sử dụng.

Đối với ứng dụng mobile: vì ứng dụng AR nên cần tối ưu hóa giao diện người dùng, chiếm ít không gian của màn hình nhất có thể.

1.2.2. Phần cứng

Ứng dụng web chỉ hỗ trợ trên trình duyệt desktop.

Ứng dụng AR chạy trên các thiết bị mobile hệ điều hành android 7.0 trở lên và phải có hỗ trợ ARCore.

Chương 2. Thiết kế giải pháp

2.1. Cơ sở lý thuyết

2.1.1. Giới thiệu về API và RESTful API

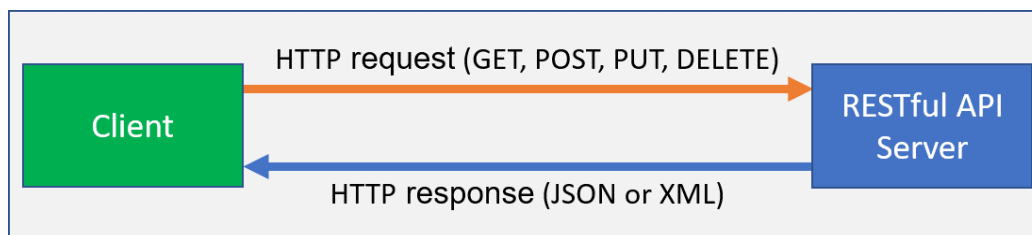
API (Application programming interface) là những quy tắc tương tác để các ứng dụng hoặc thành phần của ứng dụng có thể giao tiếp được với nhau. Mục đích của API là để đơn giản hóa việc lập trình bằng cách trừu tượng hóa đi những cài đặt phức tạp và chỉ phơi bày ra các cách thức để thực hiện hành động mà lập trình viên cần. Trong ngữ cảnh lập trình web, API là cách thức để ứng dụng trao đổi dữ liệu với một dịch vụ trực tuyến. API sẽ cung cấp các chức năng dịch vụ thông qua đường dẫn URL (Uniform Resource Locator) với định nghĩa thông số kỹ thuật như gửi request bằng HTTP (Hypertext Transfer Protocol) và trả về response message theo định dạng XML (Extensible Markup Language) hoặc JSON (JavaScript Object Notation).

REST (Representation State Transfer) là một kiến trúc phần mềm bao gồm các quy tắc để tạo ra dịch vụ web (webservice). Một webservice tuân thủ theo kiến trúc REST thì được gọi là RESTful webservice. Webservice này sử dụng giao thức HTTP để triển khai kiến trúc web. Như vậy, RESTful API chính là kiến trúc thiết kế API tuân thủ theo kiến trúc REST thông qua các phương thức của HTTP (GET, POST, PUT, DELETE,...). Tương ứng với mỗi phương thức HTTP sẽ thực hiện những tác vụ tương ứng:

Phương thức HTTP	Tác vụ
GET	Lấy dữ liệu
POST	Tạo dữ liệu mới
PUT	Cập nhật dữ liệu
DELETE	Xóa dữ liệu

Bảng 2.2.1. Các tác vụ cơ bản của REST dựa trên phương thức HTTP

Các tác vụ đọc, tạo, cập nhật, xóa được gọi là CRUD service (Create, Read, Update, Delete). Mỗi tác vụ trên phải được gọi thông qua địa chỉ URI (Uniform Resource Identifier) kèm theo phương thức và payload (có thể có hoặc không, thường là định dạng XML hoặc JSON).



Hình 2.2.1. Mô hình hoạt động cơ bản của REST

RESTful API sử dụng giao thức stateless (là một giao thức truyền thông không sử dụng session) và theo tiêu chuẩn nên hệ thống sẽ nhanh, đáng tin cậy và có thể mở rộng dễ dàng. Thông thường, RESTful API sẽ xác thực người dùng khi gửi yêu cầu đối với những tác vụ nguy hiểm như cập nhật hoặc xóa dữ liệu hoặc chỉ cho phép đối với người quản trị.

2.1.2. Django framework



Hình 2.2.2. Logo của thư viện django

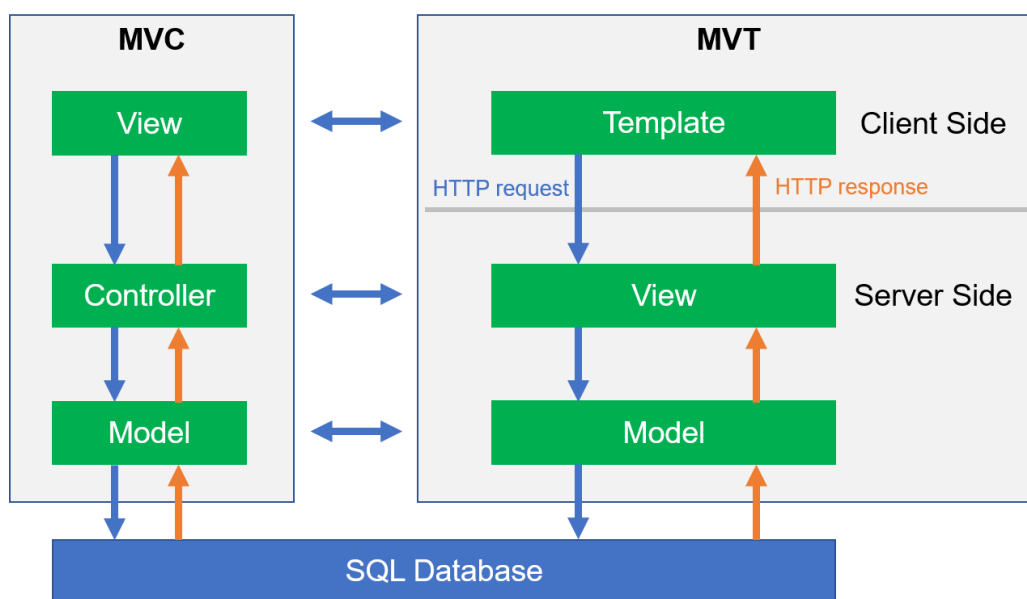
Django là một framework mã nguồn mở được viết bằng Python, dùng để đơn giản hóa việc phát triển những ứng dụng web phức tạp, hướng cơ sở dữ liệu. Được tạo ra bởi cặp đôi lập trình viên web Adrian Holovaty và Simon Willison vào năm 2003, công bố lần đầu tiên vào năm 2005 dưới giấy phép BSD. Hiện nay django được duy trì bởi tổ chức Django Software Foundation (một tổ chức được thành lập dưới dạng 501 phi lợi nhuận).

Tiêu chí của django hướng tới là tái sử dụng các thành phần, ít code, các thành phần phụ thuộc lỏng lẻo với nhau và đặc biệt là phát triển ứng dụng một cách nhanh chóng. Thay vì thao tác trực tiếp với cơ sở dữ liệu, thông qua django, chúng ta chỉ cần khai báo model dữ liệu thông qua class python. Khi đó các công việc tạo, lấy, cập nhật, xóa dữ liệu (CRUD service) sẽ được hỗ trợ tự động bởi django bằng cách gửi những HTTP request tới URL được cung cấp bởi server django.

Khi sử dụng django sẽ giúp cho việc tạo ra một ứng dụng web trở nên đơn giản, an toàn và nhanh chóng:

- Đơn giản: không cần quan tâm tới việc thao tác với cơ sở dữ liệu, xây dựng các trigger hay procedure phức tạp. Có thể thay đổi sang một hệ quản trị cơ sở dữ liệu khác một cách dễ dàng.
- An toàn: cung cấp hệ thống xác thực người dùng, quản lý tài khoản và mật khẩu an toàn, giúp lập trình viên chống các lỗi bảo mật thông thường như SQL injection.
- Nhanh chóng: django được thiết kế để giúp những nhà phát triển tạo ra ứng dụng web nhanh nhất có thể.

Django sử dụng mô hình MVT (Model – View – Template). Mô hình MVT cũng tương tự mô hình MVC, tuy nhiên dễ gây nhầm lẫn bởi View trong MVT tương ứng với Controller trong MVC còn Template thì tương ứng với View.



Hình 2.2.3. Mô hình MVT và MVC

Đối với mô hình MVT thì các thành phần được sử dụng độc lập với nhau:

- Model: là phần dùng để xử lý cơ sở dữ liệu. Đây là lớp truy xuất dữ liệu, chứa các trường bắt buộc và các hành vi xử lý dữ liệu (tạo, thêm, sửa, xóa,...).

- View: là phần dùng để thực thi các tác vụ logic và tương tác với một model để lấy dữ liệu và kết xuất ra template.
- Template: là lớp giao diện, được dùng để xử lý phần giao diện người dùng.

Tuy nhiên, khi sử dụng django để xây dựng API với django rest framework thì phần giao diện người dùng không được sử dụng nên Template sẽ được gỡ bỏ.

2.1.3. Django rest framework

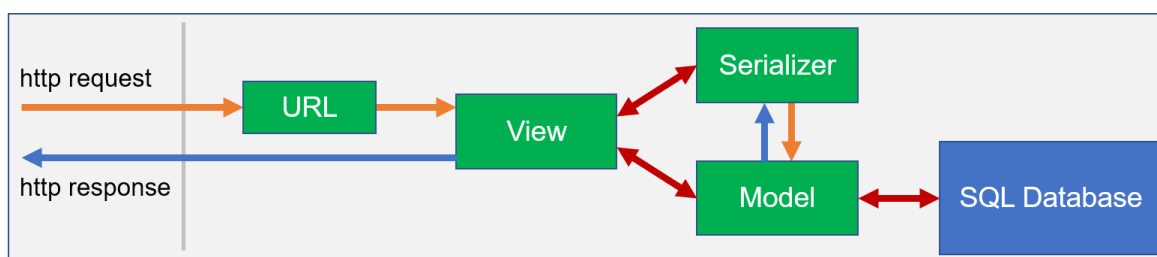


Hình 2.2.4. Logo của django rest framework

Django rest framework (DRF) là một core framework của django, cung cấp bộ công cụ tùy biến mạnh mẽ để xây dựng web APIs. DRF sẽ đơn giản hóa công việc tạo ra RESTful API. Một số khái niệm cần nắm vững để xây dựng REST API dựa trên django và django rest framework:

- **Model:** là mô hình dữ liệu được thể hiện bằng python class. Trong cơ sở dữ liệu quan hệ, tên model có thể hiểu là tên bảng, các trường của model tương ứng với các cột. Django cũng hỗ trợ các quan hệ 1-1, 1-n, n-n. Khi model dữ liệu có thay đổi, django sẽ tự động cập nhật trong cơ sở dữ liệu thông qua 2 lệnh makemigrations và migrate.
- **Serializer:** là class python hỗ trợ chuyển đổi các đối tượng được truy vấn từ model dưới dạng kiểu dữ liệu của python thành định dạng XML, JSON hoặc loại khác và ngược lại. Đây là một công cụ mạnh mẽ dùng để chuyển dữ liệu của request về kiểu dữ liệu python để xử lý và chuyển thành dữ liệu XML hoặc JSON cho response.

- **View:** cũng là một python class được dùng để nhận request, xử lý và trả về response tương ứng với loại HTTP method. Dữ liệu trả về sẽ kèm theo một status code với một ý nghĩa riêng.
- **Router:** được dùng để ánh xạ URL của request tới view tương ứng.
- **Permission:** là python class dùng để xác định tài nguyên yêu cầu có được quyền truy cập hay không. Thông thường phương GET được cho phép với tất cả người dùng (đã được xác thực hoặc không), còn các phương thức còn lại phải yêu cầu quyền tương ứng.



Hình 2.2.5. Sơ đồ hoạt động của DRF

Cách thức hoạt động cơ bản của DRF khi có một HTTP request được gửi tới có thể được mô tả như sau: request được gửi tới URL sẽ được ánh xạ tới view tương ứng, view sẽ xác định quyền truy cập và thực hiện chuyển đổi dữ liệu request về kiểu dữ liệu python bằng serializer (nếu có). Sau đó sẽ thực hiện truy vấn cần thiết tới model, nhận kết quả và dùng serializer chuyển lại thành định dạng JSON (hoặc định dạng khác) và trả về cho view. View nhận dữ liệu trả về và trả về response tương ứng kèm status code hoặc thông báo lỗi.

2.1.4. Giới thiệu về JWT authentication

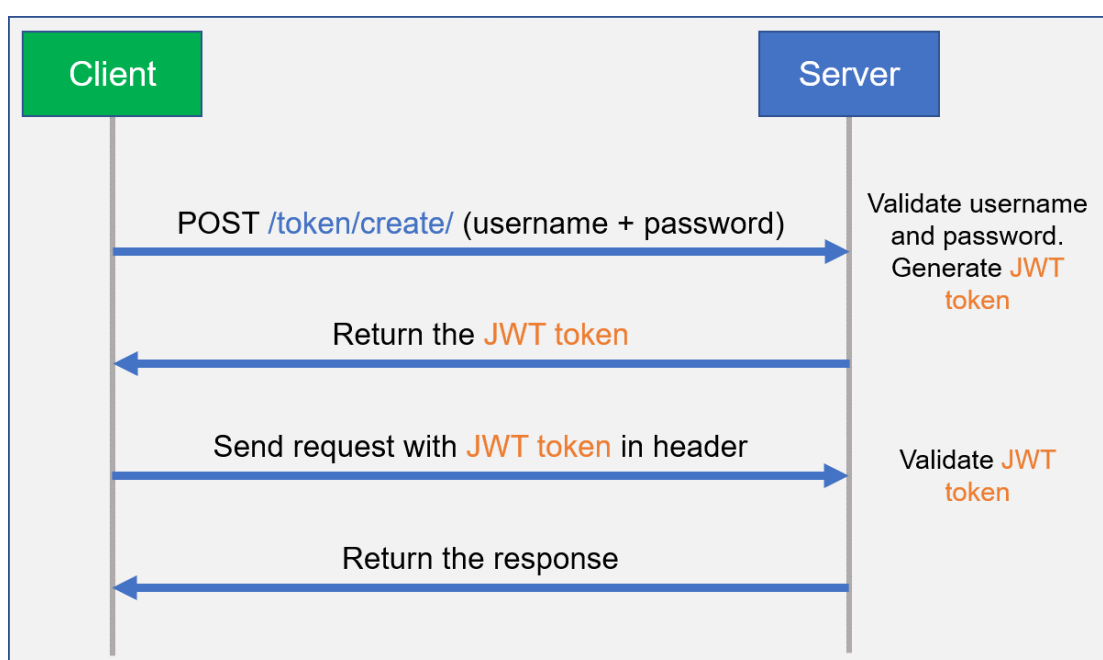
JWT (JSON Web Token) là một tiêu chuẩn mở quy định việc truyền tải những thông tin ngắn gọn dưới dạng đối tượng JSON. Những thông tin này được xác thực dưới dạng chữ ký số nên đáng tin cậy. Chữ ký số này thường được tạo bởi một khóa bí mật hoặc một cặp khóa public/private. JWT thường được dùng để cấp quyền truy cập tài nguyên (định danh người dùng) hoặc trao đổi thông tin quan trọng.

Cấu trúc của một JWT được chia thành 3 phần được phân cách bởi dấu chấm (dạng header.payload.signature), bao gồm:

- **Header:** xác định loại thuật toán hash (alg) và loại token (typ).

- **Payload:** là dữ liệu truyền tải được thêm vào bao gồm cả định danh người dùng.
- **Signature:** là dữ liệu được băm bởi 3 phần header (đã mã hóa), payload (đã mã hóa) và một khóa bí mật. Do đó, dữ liệu sẽ không thể bị thay đổi vì khi thay đổi bất kì thành phần nào của dữ liệu sẽ khiến cho chữ ký số không còn hợp lệ. Thuật toán băm thường dùng là HMAC SHA256.

JWT authentication là một thư viện sử dụng JWT để xác thực người dùng có thể tích hợp vào DRF. Vì RESTful API là stateless nên sẽ sử dụng JWT để cấp quyền cho request. Phần header của request sẽ chứa access token.



Hình 2.2.6. Sơ đồ cách thức hoạt động của JWT

Server sử dụng JWT token để xác thực cấp quyền request: đầu tiên người dùng phải đăng nhập bằng username và password. Nếu đăng nhập thành công thì server sẽ trả về một cặp JWT token (access và request token). Tương ứng với mỗi request cần yêu cầu xác thực để cấp quyền thì access token sẽ được gửi kèm phần header của request. Nếu access token còn hợp lệ (thời gian sống vẫn còn hợp lệ) thì server sẽ trả về response tương ứng, ngược lại thì trả về thông báo lỗi.

2.1.5. Hệ quản trị cơ sở dữ liệu MySQL



Hình 2.2.7. Logo của MySQL

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ mã nguồn mở miễn phí, được những doanh nghiệp lớn nhất thế giới như Facebook, Google, Adobe sử dụng để tiết kiệm thời gian và chi phí cho việc quản lý dữ liệu hiệu quả. MySQL được sáng lập bởi Michael "Monty" Widenius và David Axmark năm 1995. Hiện nay được phát triển bởi tập đoàn Oracle.

MySQL hỗ trợ trên nhiều hệ điều hành: Windows, Linux, MacOS,...

MySQL cũng được công nhận bởi DB-Engines là hệ quản trị cơ sở dữ liệu của năm 2019.

2.1.6. Thư viện ReactJS



Hình 2.2.8. Logo của thư viện React

ReactJS (hay React, React.js) là một thư viện mã nguồn mở (MIT License) dùng để xây dựng giao diện người dùng được viết bằng JavaScript. React được phát hành

lần đầu tiên vào năm 2013 bởi Facebook. React hiện nay được duy trì bởi Facebook và cộng đồng lập trình viên. React được sử dụng bởi những doanh nghiệp công nghệ hàng đầu như Facebook, Twitter, Instagram.

Sức mạnh của React nằm ở việc tự động cập nhật lại UI (user interfaces) khi dữ liệu thay đổi với chi phí ít nhất mà không cần phải tải lại toàn bộ trang web. Điều này làm cho ứng dụng phản hồi lại nhanh hơn, tiết kiệm băng thông, tăng tốc ứng dụng và tăng trải nghiệm người dùng. Trong React sẽ có một số khái niệm cơ bản về JSX, components, props, state và hook.

JSX (JavaScript XML) là một cú pháp mở rộng của JavaScript cho phép định nghĩa các thành phần HTML trong React. Nói một cách đơn giản, cú pháp này cho phép ta viết các HTML tag trực tiếp trong JavaScript.

Components là những thành phần UI được chia nhỏ ra, độc lập và có thể tái sử dụng. Component có thể là những function (stateless) hoặc class (stateful) trong JS. Component sẽ có các thuộc tính props (properties) và state (nếu được định nghĩa bằng class). Để phân biệt giữa React component và HTML tag, tất cả các React components phải được viết kiểu CamelCase (các cụm từ được viết liền nhau và bắt đầu mỗi từ bằng chữ in hoa, không có khoảng cách hoặc dấu câu xen kẽ) và phải bắt đầu bằng kí tự in hoa.

Props là những thuộc tính được truyền vào một component và chỉ có thể đọc. Ví dụ như thẻ `<button className="active">Sign up</button>` có những thuộc tính được truyền vào có thể truy xuất là className và child thông qua props. Truy xuất bằng cú pháp props.className sẽ cho giá trị là "active" và props.child có giá trị là "Sign up".

State là trạng thái thuộc về chính component đó, được quản lý bởi chính nó và không được truy xuất từ bên ngoài. Chỉ có thể sử dụng state khi dùng stateful component.

Life cycle là một vòng đời của một React component từ lúc được render lần đầu tiên và mỗi lần render lại (mounting) và khi gỡ bỏ component (unmounting). Có 2 phương thức được tự động gọi khi sự kiện mounting (component được render lần đầu tiên hoặc được render lại) và unmounting (component bị gỡ bỏ) xảy ra lần lượt là componentDidMount và componentWillUnmount. Có thể ghi đề 2 phương thức này khi sử dụng stateful component (class).

Hook là tính năng được thêm vào React ở phiên bản 16.8. Cho phép sử dụng một số tính năng chỉ có ở stateful component (class) khi dùng stateless component (function) như state (useState), life cycle (useEffect),...

2.1.7. Thư viện Redux



Hình 2.2.9. Logo của thư viện Redux

Redux là một thư viện JavaScript mã nguồn mở (MIT License) dùng để quản lý state của ứng dụng, được sử dụng phổ biến với React hoặc Angular để xây dựng giao diện người dùng. Redux được tạo bởi Dan Abramov và Andrew Clark vào năm 2015.

State trong redux sẽ có các tính chất sau:

- **Từ một nguồn duy nhất:** state của ứng dụng được là một state toàn cục được lưu trữ thành một object tree.
- **State chỉ có thể đọc:** state không thể thay đổi trực tiếp mà chỉ có thể phát động bằng một action (object) mô tả những gì đã xảy ra.
- **State được thay đổi bởi những reducer (JS function):** khi muốn thay đổi state sẽ phát động ra một action, reducer sẽ tiếp nhận và xử lý thay đổi state tương ứng. Vì reducer chỉ là những hàm js bình thường, có thể truyền vào dữ liệu, gọi theo thứ tự hoặc tái sử dụng các reducer.

Redux được thiết kế cho việc quản lý tập trung, chia sẻ state giữa các component trở nên đơn giản, nhất quán. Để sử dụng redux với react cần phải nắm một số khái niệm cơ bản sau:

- **Action:** là những thông tin payload được dùng để gửi dữ liệu đến store. Hiểu đơn giản, action chính là một event được gửi lên store kèm theo dữ liệu và loại action sẽ được thực hiện.
- **Reducer:** là những hàm thuần túy thực hiện việc nhận action được gửi lên và lấy state hiện tại của ứng dụng, xử lý và trả về một state mới. Global state chính là một object tree lưu trữ toàn bộ state của ứng dụng, action gửi lên sẽ mô tả những gì xảy ra. Khi đó, reducer sẽ tiếp nhận

action và lấy một nhánh state nhỏ tương ứng từ global state, sau đó tạo ra một nhánh state mới dựa trên action và cập nhật vào global state.

- **Store:** là một object kết nối action và reducer lại với nhau, chịu trách nhiệm lưu giữ state của ứng dụng, cho phép truy xuất và cập nhật state thông qua dispatch (là một dịch vụ của store cho phép gửi action thông qua nó).

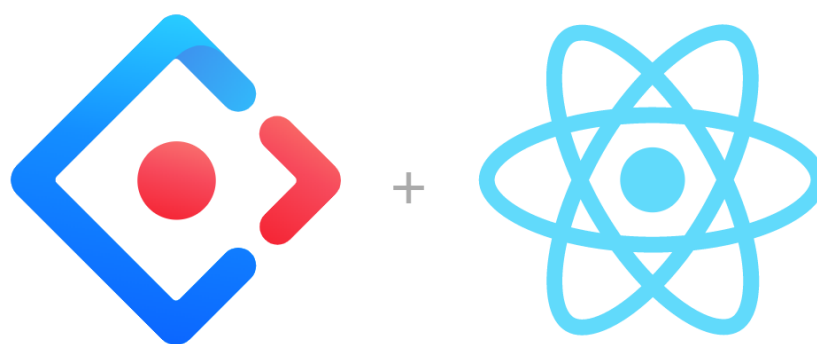
2.1.8. Thư viện Axios

Axios là một thư viện HTTP client dựa trên Promise dùng để gửi các request HTTP bất đồng bộ đến REST endpoint để sử dụng các dịch vụ CRUD.

Promise là một cải tiến để loại bỏ try catch, callback rườm rà trong xử lý bất đồng bộ, thay vào đó là then – catch. Hiểu đơn giản là khi thực hiện một tác vụ bất đồng bộ, sau khi thực hiện thành công thì “then” sẽ được gọi, ngược lại khi có lỗi phát sinh thì “catch” sẽ được gọi. Tương tự như promise, axios cũng có thể sử dụng theo kiểu gửi request lồng nhau.

Ngoài ra, axios còn hỗ trợ interceptor dùng để thực hiện một số công việc khác trước khi gửi request hoặc ngay khi nhận được response. Ví dụ như tạo mới access token khi nhận được response thông báo lỗi “401 – Unauthorized”.

2.1.9. Thư viện React UI – Ant design



Hình 2.2.10. Logo Ant design và React

Antd là một thư viện React UI mã nguồn mở (MIT license) với một tập hợp các component chất lượng cao dùng để xây dựng giao diện người. Antd được duy trì bởi Ant Design Team và cộng đồng. Các component trong Antd là những React component, hỗ trợ hầu hết các thành phần cốt yếu để tạo dựng nên một trang web

hoàn chỉnh. Khi sử dụng antd thì hầu như không cần phải cài thêm một thư viện UI bổ sung nào nữa thì cũng đủ để đáp ứng nhu cầu của dự án đưa ra.

Dưới đây là danh sách các component mà antd hỗ trợ:

- **General:** Button, Icon, Topography.
- **Layout:** Divider, Grid, Layout, Space.
- **Navigation:** Affix, Breadcrumb, Dropdown, Menu, Pagination, PageHeader, Steps.
- **Data Entry:** AutoComplete, Checkbox, Cascader, DatePicker, Form, InputNumber, Input, Mentions, Rate, Radio, Switch, Slider, Select, TreeSelect, Transfer, TimePicker, Upload.
- **Data Display:** Avatar, Badge, Comment, Collapse, Carousel, Card, Calendar, Descriptions, Empty, List, Popover, Statistic, Tree, Tooltip, Timeline, Tag, Tabs, Table.
- **Feedback:** Alert, Drawer, Modal, Message, Notification, Progress, Popconfirm, Result, Spin, Skeleton.
- **Other:** Anchor, BackTop, ConfigProvider.

2.1.10. Game engine Unity 3D

Game engine là môi trường phát triển tích hợp (IDE – Integrated Development Environment) được tạo ra để thực hiện việc thiết kế và phát triển video game. Các chức năng cốt lõi thường được cung cấp bởi một game engine bao gồm hệ thống kết xuất đồ họa 2D/3D, hệ thống vật lý hoặc phát hiện va chạm, âm thanh, ngôn ngữ kịch bản (scripting), hoạt hình, trí tuệ nhân tạo, mạng máy tính, quản lý bộ nhớ, quản lý luồng và một số chức năng khác.

Unity là một công cụ làm game đa nền tảng được phát triển bởi Unity Technologies, ra mắt lần đầu tiên vào năm 2005. Đến năm 2018, Unity đã hỗ trợ trên 25 nền tảng. Unity engine được dùng để tạo các trò chơi 2D, 3D, virtual reality và augmented reality. Game Pokemon Go cũng là một trong những game sử dụng Unity engine. Ngoài ngành công nghiệp game, Unity còn được dùng trong làm phim, kiến trúc, ô tô tự hành, kỹ thuật và xây dựng.



Hình 2.2.11. Logo của Unity 3D game engine

Tổng quan

Unity 3D là một công cụ hỗ trợ lập trình game mạnh mẽ, đầy đủ tính năng, có thể tạo game thông qua việc kéo thả và viết các chức năng game bằng các script C#. Trước đây, Unity hỗ trợ viết script bằng Boo (được loại bỏ từ Unity 5) và phiên bản JavaScript gọi là UnityScript. Tuy nhiên, từ Unity 2017 C# được lựa chọn làm ngôn ngữ chính thức để viết các script. Script có thể được sử dụng trong môi trường editing hoặc runtime.

Ngoài ra, Unity cũng hỗ trợ tích hợp những tài nguyên 2D và 3D từ những phần mềm khác vào như Maya, 3ds Max hay Photoshop. Kết hợp những tài nguyên đó lại thành những phân cảnh và môi trường trong game. Unity cũng hỗ trợ tích hợp ánh sáng, âm thanh, hiệu ứng, vật lý, chuyển động hoạt hình, khả năng tương tác và tính logic vào game.

Từ năm 2018, Unity được sử dụng để tạo ra sấp xỉ một nửa số lượng sản phẩm game mobile trên thị trường, trong số đó những nội dung về VR và AR chiếm 60%.

Nền tảng hỗ trợ

Unity engine là một game engine đa nền tảng. Unity editor được hỗ trợ trên hệ điều hành Windows và MacOS, trên Linux vẫn đang trong giai đoạn thử nghiệm.

Unity hỗ trợ xây dựng game trên hơn 25 nền tảng bao gồm desktop, mobile, console và virtual reality. Những nền tảng hỗ trợ được liệt kê bên dưới:

- **Desktop:** Windows, Universal Windows Platform, Mac, Linux.
- **Mobile:** Android, iOS, Windows Phone và Tizen.

- **VR và AR:** Oculus Rift, Gear VR, PlayStation VR, Windows Mixed Reality, Microsoft HoloLens, Steam VR, Google Daydream, Google Cardboard Android & iOS, Magic Leap, Google ARCore, Apple ARKit, Vuforia.
- **Console:** Play Station 4, Play Station Vita, Xbox one, Nintendo Switch, Nintendo 3DS.
- **Web:** WebGL.

Giấy phép

Hiện nay, Unity cung cấp 5 giấy phép bao gồm student, personal, plus, pro và enterprise. Chi phí bỏ ra vô cùng hấp dẫn so với sức mạnh của game engine này mang lại. Tương ứng với mỗi gói sẽ có giá, tính năng và giới hạn khác nhau.

Gói	Giá	Giới hạn
Student	Miễn phí	Sử dụng cho mục đích học tập
Personal	Miễn phí	Doanh thu dưới \$100K trong 12 tháng gần nhất
Plus	\$399/năm	Doanh thu dưới \$200K trong 12 tháng gần nhất
Pro	\$1800/năm	Doanh thu lớn hơn \$200K trong 12 tháng gần nhất
Enterprise	Giá tùy chỉnh	Phụ thuộc vào quy mô của doanh nghiệp

Bảng 2.2.2. Các giấy phép của Unity 3D

Các thành phần cơ bản trong Unity

Asset

Assets là tất cả những nguồn tài nguyên mà một dự án game sử dụng. Bao gồm các mô hình 3D, chất liệu, kết cấu, âm thanh, kịch bản hoặc một dự án hoàn chỉnh. Ngoài các mô hình cơ bản như hình khối, hình trụ thì Unity không thể tạo ra hầu hết

các tài nguyên này. Một số phần mềm khác như Maya, Photoshop, Substance Painter sẽ được dùng để tạo ra những tài nguyên và sau đó nhập vào Unity. Các tài nguyên này ngoài được việc được tạo ra bởi nhà phát triển thì còn có thể tải trực tiếp cửa hàng Asset của Unity. Bao gồm các tài nguyên tính phí và miễn phí.

Scene

Scene là một màn chơi của game. Một game bao gồm nhiều màn chơi, trong đó menu game cũng được xem như là một scene. Một scene được xem như một cấp độ khác nhau của game. Mỗi scene khác nhau thì độc lập với nhau, có nội dung riêng biệt với nhau giúp cho việc thiết kế và xây dựng game trở nên dễ dàng hơn.

Mỗi một scene phải có ít nhất một camera. Đó là một thiết bị giúp người chơi có thể nhìn thấy được thế giới trong scene đó.

Game Object

Game Object là một thành phần cơ bản trong Unity, đại diện cho các nhân vật, đạo cụ và phong cảnh. Hay nói cách khác, mọi thứ trong màn chơi khả dụng thì đều là Game Object. Bản thân Game Object không tự thực hiện được điều gì nhưng đóng vai trò là nơi chứa các thành phần có khả năng thực hiện chức năng thực sự. Mỗi Game Object chứa ít nhất một thành phần cơ bản là Transform. Thành phần này lưu trữ thông tin về vị trí, góc xoay và tỉ lệ của Game Object.

Đối với thành phần Transform sẽ dựa trên hệ tọa độ Oxyz. Tỉ lệ 1 đơn vị trong Unity tương ứng với 1 mét trong thế giới của nó. Ta có thể sử dụng bàn tay trái để xác định phương hướng của hệ tọa độ. Dùng bàn tay trái lật úp chỉ về phía trước, xòe ngón tay cái ra 90 độ và 4 ngón còn lại khép vào nhau. Tương ứng hướng chỉ của ngón cái là trục X, hướng lên của mu bàn tay là trục Y và hướng chỉ của 4 ngón tay còn lại là trục Z.

Component

Trong Unity, Components là những thành phần bị cô lập có thể được gắn vào một Game Object để bổ sung, thực hiện một chức năng cụ thể. Mỗi thành phần sẽ có chức năng riêng biệt, không được tạo ra trực tiếp mà thay vào đó là phải viết mã script và đính kèm script được tạo ra vào Game Object.

Game Object hoàn toàn phụ thuộc vào các Components. Khi một Game Object đòi hỏi một chức năng nào đó thì những thành phần liên quan có thể được thêm vào để có thể hoàn thành chức năng đó. Do vậy, bất cứ thứ gì mà chúng ta tưởng tượng thì đều có thể xây dựng được trong Unity.

Script

Script được xem như là một Component thiết yếu không thể thiếu trong quá trình phát triển game. Đây là thành phần được dùng để tương tác với các thao tác của người chơi, quản lý các sự kiện để thay đổi theo kịch bản tương ứng với nội dung của game.

Các ngôn ngữ được Unity hỗ trợ viết script bao gồm Boo, UnityScript (có cú pháp tương tự JavaScript) và C#. Hiện nay, ngôn ngữ được dùng phổ biến nhất là C#.

Những môi trường phát triển tích hợp tốt nhất hiện nay để viết C# script là Visual Studio (Windows và macOS), Visual Studio Code (Windows, macOS, Linux) và JetBrains Rider (Windows, macOS, Linux).

Prefab

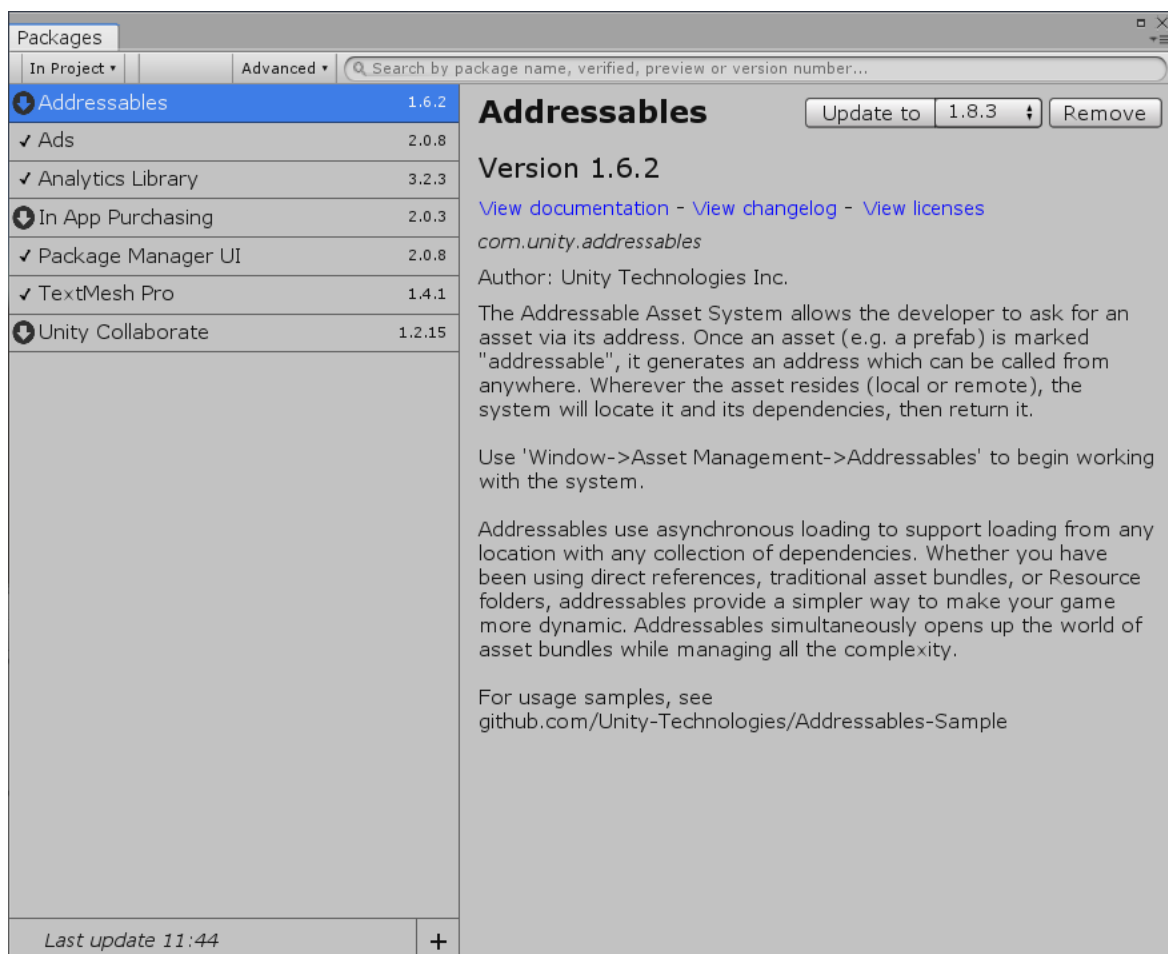
Prefabs là những Game Object được lưu lại để tái sử dụng với phần mở rộng là .prefab. Các Game Object cùng được tạo ra bởi một prefab sẽ giống nhau hoàn toàn ngoại trừ thành phần Transform để phân biệt với nhau.

Một số thành phần liên quan

Scriptable Object

ScriptableObject là một thùng chứa dữ liệu, dùng để lưu trữ những dữ liệu liên quan với nhau thành một khối. Những thể hiện khác nhau của class này sẽ độc lập dữ liệu với nhau. Mục đích sử dụng ScriptableObject để giúp quản lý asset của Project hiệu quả hơn.

Addressables



Hình 2.2.12. Addressables trong Unity packages

Addressables là một thư viện Unity dùng để đóng gói và truy xuất dữ liệu thông qua địa chỉ. Khi đánh dấu một tài nguyên (Sprite, Prefab, Scene,...) là “addressable”, thư viện sẽ tạo địa chỉ do lập trình viên chỉ định và có thể truy xuất tài nguyên đó bất kì nơi đâu thông qua địa chỉ này.

Mục đích của thư viện này là dùng để giảm tải kích thước cho file apk cài đặt ứng dụng, chỉ tải xuống những tài nguyên cần dùng khi lượng tài nguyên của toàn bộ ứng dụng có kích thước quá lớn. Ngoài ra, còn dùng để thêm các tài nguyên mới vào ứng dụng mà không cần phải cài đặt lại ứng dụng.

2.1.11. Thư viện ARCore

AR là chữ viết tắt của Augmented Reality – Thực tế tăng cường. Thực tế tăng cường là sự tích hợp thông tin kỹ thực số (âm thanh, hình ảnh, ...) với môi trường

xung quanh của người dùng trong thời gian thực. Nói cách khác, môi trường tồn tại xung quanh được che phủ thông tin kỹ thuật số lên để tạo cho người dùng trải nghiệm dường như đang được tương tác với những thông tin đó.



Hình 2.2.13. Logo của thư viện ARCore

ARCore là một bộ phát triển phần mềm do Google phát triển dùng để xây dựng những ứng dụng trải nghiệm thực tế tăng cường. Bằng cách sử dụng những API của ARCore, thiết bị di động sẽ có khả năng cảm nhận và hiểu được môi trường xung quanh, có khả năng tương tác với thông tin thu được.

ARCore được phát hành lần đầu tiên vào tháng 3 năm 2018 bởi Google. ARCore cung cấp những SDK cho những môi trường phát triển phổ biến bao gồm Android, Android NDK, iOS, Unity, Unreal.

ARCore sử dụng 3 công nghệ chính để tích hợp nội dung ảo vào thế giới thật thông qua camera của thiết bị:

- **Motion tracking:** cho phép điện thoại hiểu và tự theo dõi vị trí tương đối của thiết bị so với thế giới thật.
- **Environment understanding:** cho phép điện thoại phát hiện kích thước và vị trí của những mặt phẳng nằm ngang, thẳng đứng và có góc cạnh như mặt bàn, mặt sàn hoặc tường nhà.
- **Light estimation:** cho phép điện thoại ước lượng được tình trạng ánh sáng hiện tại của môi trường xung quanh.

Cách thức hoạt động của ARCore

Về cơ bản ARCore sẽ thực hiện 2 công việc: theo dõi vị trí của thiết bị khi nó di chuyển và tự xây dựng khả năng hiểu biết về thế giới thật xung quanh thiết bị.

Công nghệ motion tracking của ARCore sử dụng camera điện thoại để xác định những điểm đặc trưng (feature points), theo dõi những điểm này di chuyển như thế

nào theo giới gian. Bằng cách kết hợp giữa sự di chuyển của những điểm này và đọc dữ liệu từ những cảm biến quán tính (cảm biến gia tốc) của điện thoại, ARCore xác định được vị trí và góc xoay của thiết bị khi nó di chuyển trong không gian.

Ngoài việc xác định những điểm đặc trưng, ARCore còn có thể xác định được những mặt phẳng như bàn hoặc sàn nhà và cũng có thể ước tính độ sáng trung bình xung quanh nó. Kết hợp những điều này lại với nhau cho phép ARCore xây dựng được khả năng tự nhận thức về thế giới xung quanh nó.

Khả năng hiểu biết của ARCore về thế giới thật cho phép chúng ta đặt những đối tượng, những chú thích hoặc những thông tin với thế giới thực theo cách liền mạch. Motion tracking trong ngữ cảnh lúc này là khi đặt một đối tượng vào không gian thực, chúng ta có thể di chuyển thiết bị xung quanh đối tượng đó và xem từ nhiều góc độ khác nhau hoặc thậm chí khi quay camera sang hướng khác và đi xung quanh căn phòng, sau đó quay lại vị trí đã đặt đối tượng thì nó vẫn nằm ở đó.

Một số khái niệm cơ bản cần nắm về ARCore

Motion tracking

Khi điện thoại di chuyển xung quanh không gian thực, ARCore sử dụng một tiến trình gọi là nội địa hóa và lập bản đồ đồng thời (SLAM – Simultaneous Localization and Mapping) để xác định điện thoại đang ở đâu so với thế giới xung quanh. ARCore có thể phát hiện ra những sự khác biệt từ ảnh camera dựa trên điểm đặc trưng và tính toán sự thay đổi vị trí dựa trên những điểm này. Những thông tin trực quan này được kết hợp với số liệu thu được từ cảm biến đo gia tốc của thiết bị (IMU – Inertial Measurements Unit) để ước tính tư thế (pose – position và orientation) của camera thiết bị so với thế giới thực theo thời gian.

Bằng cách căn chỉnh tư thế của camera ảo (camera kết xuất nội dung 3D) với camera của thiết bị được cung cấp bởi ARCore thì những nội dung ảo có thể được kết xuất theo đúng góc nhìn. Những hình ảnh ảo (ảnh 2D) được phủ đè lên ảnh được cung cấp bởi camera thiết bị, làm cho hình ảnh ảo đó trông như là một phần của thế giới thật.



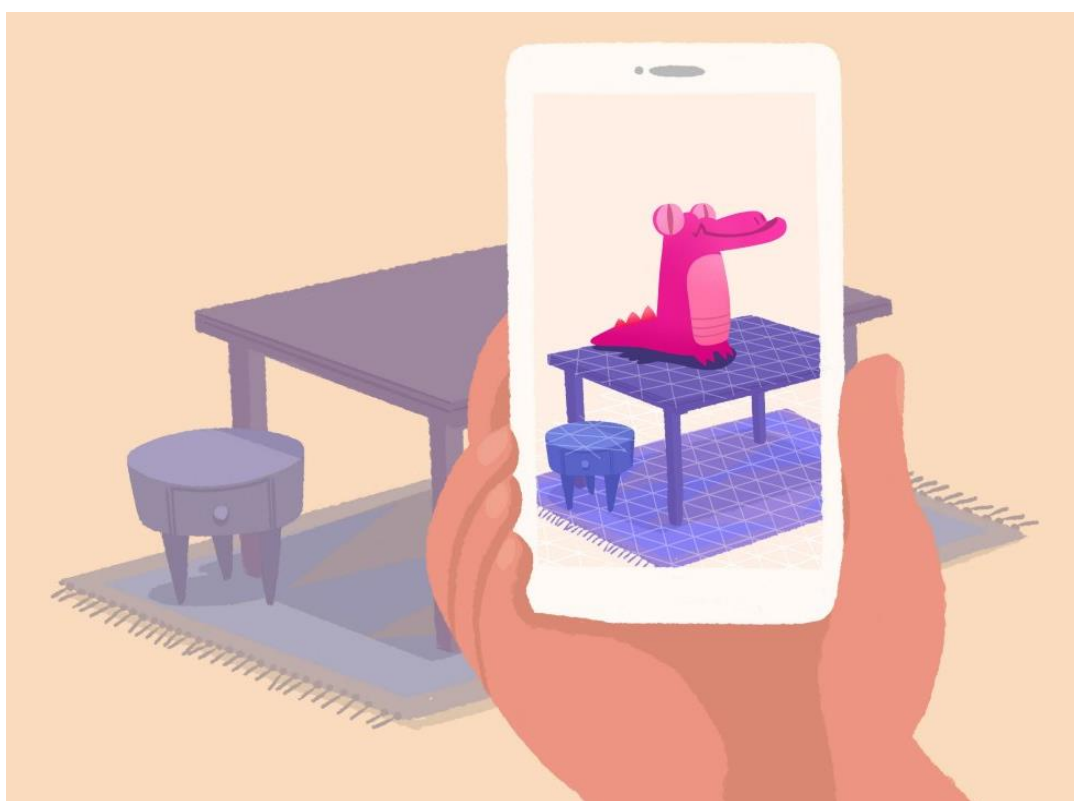
Hình 2.2.14. Các hoạt động của motion tracking
(Nguồn: ARCore docs)

Environment understanding

ARCore liên tục tăng cường khả năng hiểu biết về thế giới thật bằng cách phát hiện những điểm đặc trưng và mặt phẳng.

ARCore tìm những cụm các điểm đặc trưng có vẻ như cùng nằm trên một bề mặt (surface) nằm ngang hoặc thẳng đứng và biến những điểm này thành những mặt phẳng (plane). ARCore cũng có thể xác định đường biên giới của những mặt phẳng này và chúng ta có thể đặt những đối tượng ảo lên trên những bề mặt này.

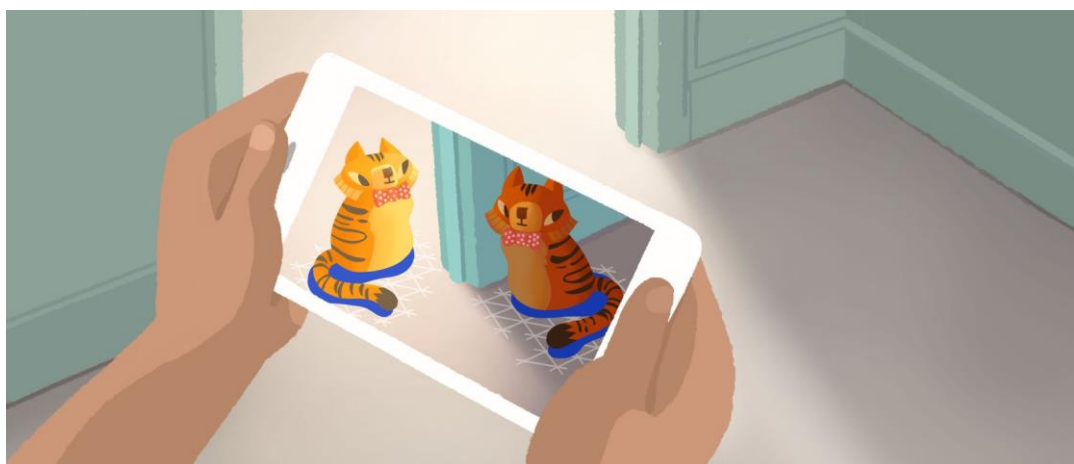
Do ARCore dựa trên những điểm đặc trưng để phát hiện ra mặt phẳng, đối với những bề mặt mà không có những kết cấu như tường trắng thì có thể không được phát hiện đúng và xảy ra hiện tượng trôi mặt phẳng (bị mất dấu mặt phẳng).



*Hình 2.2.15. Mặt phẳng được phát hiện bởi ARCore
(Nguồn: ARCore docs)*

Light estimation

ARCore có thể phát hiện thông tin về ánh sáng của môi trường và cung cấp cho chúng ta cường độ sáng trung bình và màu sắc chính xác của hình ảnh được đưa bởi camera. Thông tin này sẽ giúp chúng ta chiếu sáng những đối tượng ảo phù hợp với môi trường xung quanh chúng, tăng cảm giác chân thực của những đối tượng.



Hình 2.2.16. Ước lượng độ sáng của ARCore
(Nguồn: ARCore docs)

User interaction

ARCore sử dụng một tác vụ gọi là kiểm tra va chạm (hit testing), tác vụ này sẽ lấy một điểm có tọa độ (x, y) trên màn hình thiết bị (tọa độ này được cung cấp bởi việc gõ hoặc bất kỳ sự tương tác nào mà ứng dụng hỗ trợ như là xoay, vuốt) và chiếu một tia từ điểm này tới góc nhìn của camera về thế giới thực, trả về bất kỳ mặt phẳng hay điểm đặc trưng nào mà tia này va phải, kèm theo tư thế (pose) của điểm giao này trong thế giới thật. Điều này cho phép người dùng chọn hoặc tương tác với những đối tượng trong môi trường ảo.

Oriented points

Điểm định hướng cho phép đặt những đối tượng ảo trên bề mặt góc cạnh. Khi người dùng thực hiện một tác vụ kiểm tra va chạm nếu trả về một điểm đặc trưng thì ARCore là tìm những điểm đặc trưng gần đó và tính toán góc tại bề mặt tại điểm đặc trưng đã cho. Sau đó, ARCore trả về một tư thế (pose) giữ vị trí của góc được tạo bởi những điểm đó.

Anchors và trackables

Những tư thế có thể thay đổi do ARCore tăng cường khả năng hiểu biết của nó về vị trí thiết bị so với môi trường xung quanh. Khi chúng ta muốn đặt một đối tượng ảo, chúng ta cần định nghĩa một anchor để đảm bảo rằng ARCore sẽ theo dõi vị trí của đối tượng này theo thời gian. Thông thường để tạo một anchor sẽ dựa trên tư thế (pose) trả về của việc kiểm tra va chạm.

Sự thật thì những tư thế có thể thay đổi, điều này có nghĩa là ARCore có lẽ đã cập nhật vị trí của những đối tượng môi trường như những mặt phẳng và những điểm đặc trưng theo thời gian. Những mặt phẳng và điểm là những loại đối tượng đặc biệt gọi là trackable. Theo đúng như tên gọi, những đối tượng này sẽ được kiểm tra xuyên suốt theo thời gian. Chúng ta có thể tạo anchor dựa trên những trackable này để đảm bảo rằng mối quan hệ giữa những đối tượng ảo và những trackable được duy trì sự bền vững ngay cả khi di chuyển thiết bị vòng quanh.

Những lưu ý khi phát triển ứng dụng AR

Môi trường người dùng:

- Kích thước trải nghiệm: quyết định kích thước của trải nghiệm trước khi bắt đầu thiết kế. Trải nghiệm ở phạm vi trên một mặt bàn, trong một căn phòng hay là trong không gian thực. Điều này giúp cho quá trình thiết kế trải nghiệm phù hợp với phạm vi không gian hơn.
- Nhiều bề mặt: các đối tượng có thể thoải mái di chuyển giữa các bề mặt với nhau.
- Kích hoạt ánh sáng: khi ánh sáng có sự thay đổi thì các đối tượng trong trải nghiệm tự động thay đổi theo cho phù hợp.
- Khuyến khích di chuyển: những đối tượng to lớn có thể khuyến khích người dùng di chuyển xung quanh thay vì ngồi tại chỗ và khám phá.
- Nhiều người chơi: nhiều người dùng cùng chia sẻ dữ liệu, tương tác với nhau trong một trải nghiệm AR.
- Ánh sáng: ánh sáng xung quanh làm cho các đối tượng trông thực hơn, tăng độ chân thực cho trải nghiệm.

Sự di chuyển của người dùng:

- Khuyến khích sự khám phá: phần thưởng ẩn dành cho người chơi di chuyển và khám phá xung quanh.
- Dễ dàng truy cập: trong các ứng dụng tiện ích, người dùng sẽ được phép lật, di chuyển đối tượng xung quanh để xem xét từ bất kì góc độ nào. Các đối tượng sẽ dễ dàng truy cập cho tất cả người dùng, không có bất kì vấn đề nào về phía góc nhìn của người dùng.
- Dẫn đường: tránh để người dùng di chuyển lùi. Ngoài việc cho người dùng điều khiển cái họ nhìn thấy, dẫn đường người dùng tránh xa những chướng ngại vật mà họ không thể nhìn thấy. Giúp người dùng di chuyển khôn ngoan hơn.
- Bên trong đối tượng: khi người dùng vô tình va chạm vào đối tượng, làm cho bên trong đối tượng mờ đi để ngụ ý với người dùng là không được ở đây.

Giao diện người dùng:

- Đánh dấu ngoài phạm vi màn hình: sử dụng mũi tên định hướng để chỉ các đối tượng ngoài phạm vi màn hình. Bằng cách di chuyển thiết bị theo hướng mũi tên, người dùng có thể tìm thấy được đối tượng đó.
- Pops-up: tránh khác thông báo toàn màn hình không được mong đợi, sẽ làm cho người dùng cảm thấy khó chịu.
- Kích thước mục tiêu: duy trì kích thước của mục tiêu phù hợp để người dùng có thể dễ dàng chạm vào.
- Giao diện màn hình: cho người dùng cái nhìn rõ ràng, không bừa bộn. Dành không gian màn hình nhiều hơn cho camera, ưu tiên cho các nút điều khiển được người dùng sử dụng thường xuyên.

Sự thao tác với các đối tượng:

- Tương tác chạm: cho phép người dùng kéo, thả, di chuyển 2 ngón tay, xoay đối tượng để tăng cảm giác thực hơn.
- Bộ sưu tập đối tượng: cho phép người dùng chọn, kéo thả, thay đổi kích thước, xóa bỏ đối tượng trong cảnh.

- Tương tác chuỗi sợi quang: sử dụng chuỗi sợi quang để nhặt, di chuyển đối tượng mà không cần bao phủ toàn đối tượng. Sợi quang giúp chọn các đối tượng nhỏ dễ dàng hơn.

Thiết bị hỗ trợ ARCore: các thiết bị hỗ trợ ARCore được liệt kê tại trang <https://developers.google.com/ar/discover/supported-devices>.

2.1.12. ARCore SDK for Unity

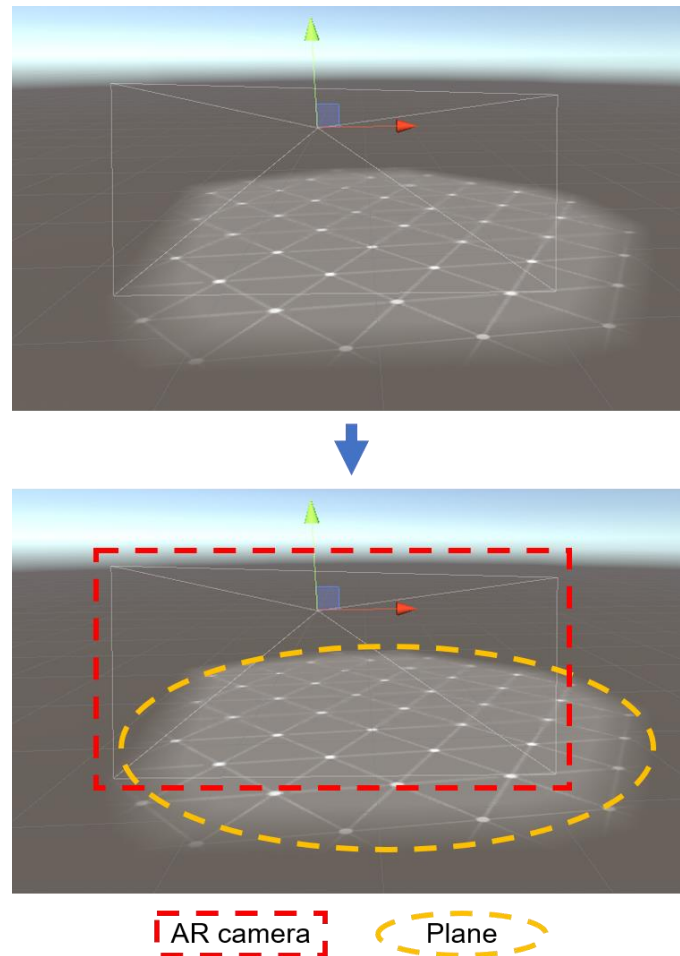
ARCore SDK cho Unity theo đúng như tên gọi chính là một bộ phát triển phần mềm ARCore dành cho Unity. Thiết bị mục tiêu của bộ SDK này là những thiết bị chạy androidOS từ phiên bản 7.0 trở về sau và có hỗ trợ dịch vụ GooglePlay Service for AR.

ARCore SDK for Unity là một SDK độc lập cho phép truy cập các tính năng sẵn có của ARCore như motion tracking, enviromental understanding, light estimation. SDK này có sẵn dưới dạng *.unitypackage, như là một Asset của Unity.

Các tính năng của ARCore SDK for Unity:

- Tìm mặt phẳng
- Truy cập hình ảnh CPU
- Sự thao tác với các đối tượng (chọn, di chuyển, thu phóng, xoay)
- Ước tính ánh sáng cơ bản
- Anchor cục bộ
- Nền camera AR
- Truy cập kết cấu GPU
- Gương mặt tăng cường
- Cloud anchors (chia sẻ anchor giữa các thiết bị android, ios)
- Ánh sáng môi trường HDR (high dynamic range – là quá trình tăng dải chênh lệch sáng tối)
- Hình ảnh tăng cường

Mặt phẳng được phát hiện bởi ARCore trong không gian thực dưới góc nhìn từ Unity được minh họa bằng hình sau:



Hình 2.2.17. Mặt phẳng ARCore trong Unity

Đường biên màu đỏ bao quanh góc nhìn của AR camera và màu vàng chính là mặt phẳng được phát hiện ở ngoài không gian thực dưới góc nhìn của Unity. Mặt phẳng và Camera chính là những GameObject trong Unity.

2.1.13. Phần mềm hỗ trợ

Autodesk Maya



Hình 2.2.18. Logo của ứng dụng Autodesk Maya

Autodesk Maya là một phần mềm đồ họa 3D chạy trên hệ điều hành Windows 64-bit, Linux và Mac OS X. Được thiết kế bởi Alias Systems Corporation và hiện tại được phát triển bởi Autodesk. Phát hành phiên bản đầu tiên vào tháng 2 năm 1998. Phiên bản mới nhất là Autodesk Maya 2019.

Maya bao gồm các công cụ mô hình hoá, kết xuất, mô phỏng, kết cấu và hoạt hình mạnh mẽ cho các nghệ sĩ, người tạo mô hình và người làm phim hoạt hình. Được dùng để thiết kế các mô hình 3D trong trò chơi điện tử, phim hoạt hình và kỹ xảo hình ảnh.

Maya hỗ trợ xuất mô hình 3D theo định dạng fbx, có thể nhập nguyên bản vào Unity. Một model từ Maya nhập vào Unity có thể bao gồm:

- Các node cấu tạo nên model tương ứng với tên ở phần mềm Maya.
- Material & texture.
- Animation, FK & IK.

Substance Painter 3D



Hình 2.2.19. Logo ứng dụng Substance Painter

Substance Painter là một phần mềm vẽ 3D cho phép vẽ chất liệu và kết xuất lưới 3D. Substance Painter được phát triển bởi Adobe. Phiên bản đầu tiên là Substance Painter 2.2 được phát hành vào ngày 21 tháng 6 năm 2016. Phiên bản mới nhất là Substance Painter 2019.

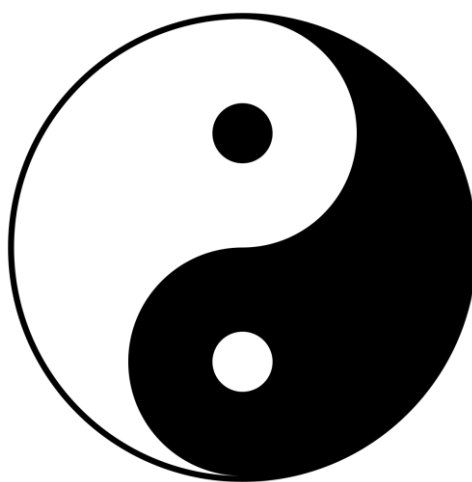
Substance Painter có nhiều vật liệu thông minh, mặt nạ thông minh và bakers tích hợp, và có khung nhìn thời gian thực “Cái bạn nhìn thấy, cái bạn thu được”. Substance Painter đủ khả năng để tạo hầu hết mọi thứ nhìn thấy trong thế giới thật thành vật liệu kỹ thuật số.

Các tính năng chính của Substance Painter:

- Công cụ vẽ mạnh mẽ: vẽ sử dụng cọ thông minh.
- Vật liệu/mặt nạ thông minh: kết cấu vật liệu hao mòn, vết trầy xước, bụi.
- Tăng tốc trên card đồ họa RTX: kết xuất bất kỳ vật liệu nào từ đối tượng 3D trong chốc lát với trình kết xuất tăng tốc của GPU.
- Trạng thái kết xuất: khung nhìn thời gian thực với trình kết xuất IRay của NVIDIA cho phép xem trước.

- Đa dạng định dạng: xuất ra định dạng hỗ trợ các công cụ game, nền tảng khác hoặc kết xuất chỉ với một cú nhấp chuột.
- Đa nền tảng: tối ưu cho Windows, Mac và Linux.
- Kỹ xảo điện ảnh: hỗ trợ UDIM, input bằng camera, ... Làm việc hiệu quả trên những tài nguyên phim chất lượng.
- Cộng đồng: học và chia sẻ với những nghệ sĩ đam mê.

2.1.14. Nội thất dựa trên phong thủy



Hình 2.2.20. Biểu tượng âm dương

Phong thủy là một học thuyết chuyên về việc nghiên cứu về sự ảnh hưởng của hướng gió, hướng khí, mạch nước đến đời sống phúc họa của con người. Không có bằng chứng khoa học chứng minh rằng phong thủy là có thật và nó được xem xét bởi cộng đồng khoa học là học giả. Bởi nó đưa ra các quan điểm trái ngược với lý luận khoa học, các khẳng định về cách thức thế giới hoạt động của nó mâu thuẫn với các kết quả nghiên cứu bằng phương pháp khoa học.

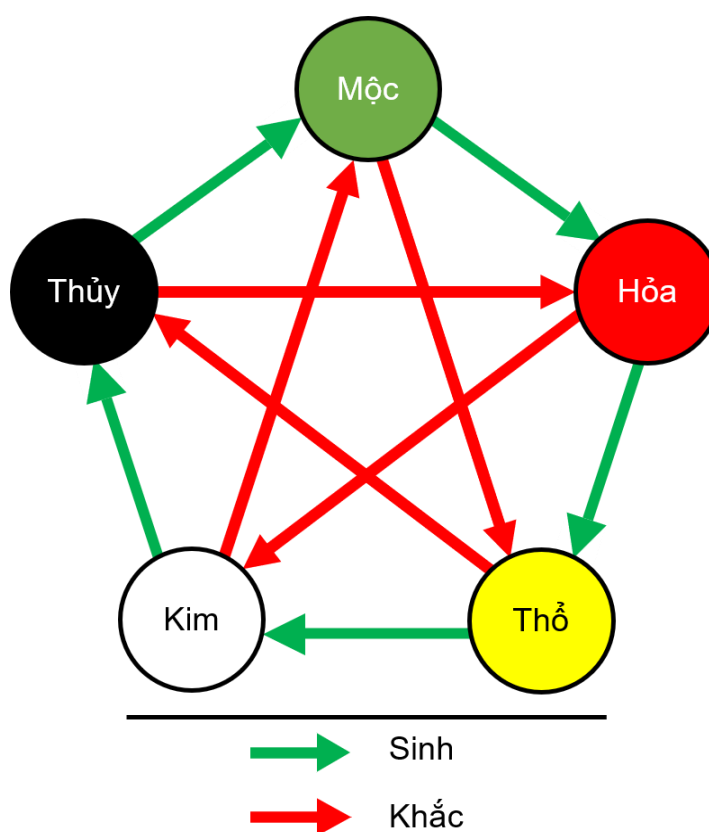
Hiện nay có rất nhiều trường phái tranh luận về nguồn gốc của phong thủy, có trường phái thì cho rằng phong thủy xuất phát từ Trung Quốc, có trường phái thì cho rằng là nền văn minh của người Việt cổ. Nhưng chính xác mà nói, không có ai dám chắc chắn về nguồn gốc thực sự của phong thủy.

Phong thủy còn là phương pháp tìm kiếm và lựa chọn nơi trú ngụ hoặc mai táng cát tường phú quý, phúc thọ bình yên, gọi là thuật phong thủy. Giống như những kỹ thuật cổ truyền khác ở Đông Á, thuật phong thủy cũng dựa vào dịch lý, thuyết âm dương, ngũ hành.

Phong thủy về nhà ở gồm rất nhiều vấn đề cần xét như huyệt vị, làm nhà, hướng nhà, cấu trúc nhà, nội thất, ... Cần phải xem xét đầy đủ các yếu tố để tạo thành phong thủy tốt cho ngôi nhà. Trang trí nội thất dựa trên phong thủy cũng là một lĩnh vực của thuật phong thủy. Vì phong thủy dựa trên ngũ hành, nên chúng ta sẽ xem xét khía cạnh ngũ hành tương sinh, tương khắc sẽ ảnh hưởng đến việc trang trí nội thất như thế nào.

Theo học thuyết ngũ hành, sinh học của vạn vật được diễn giải qua 2 nguyên lý cơ bản là tương sinh và tương khắc:

- Quan hệ tương sinh: Mộc – Hỏa, Hỏa – Thổ, Thổ – Kim, Kim – Thủy, Thủy – Mộc.
- Quan hệ tương khắc: Mộc – Thổ, Thổ – Thủy, Thủy – Hỏa, Hỏa – Kim, Kim – Mộc.



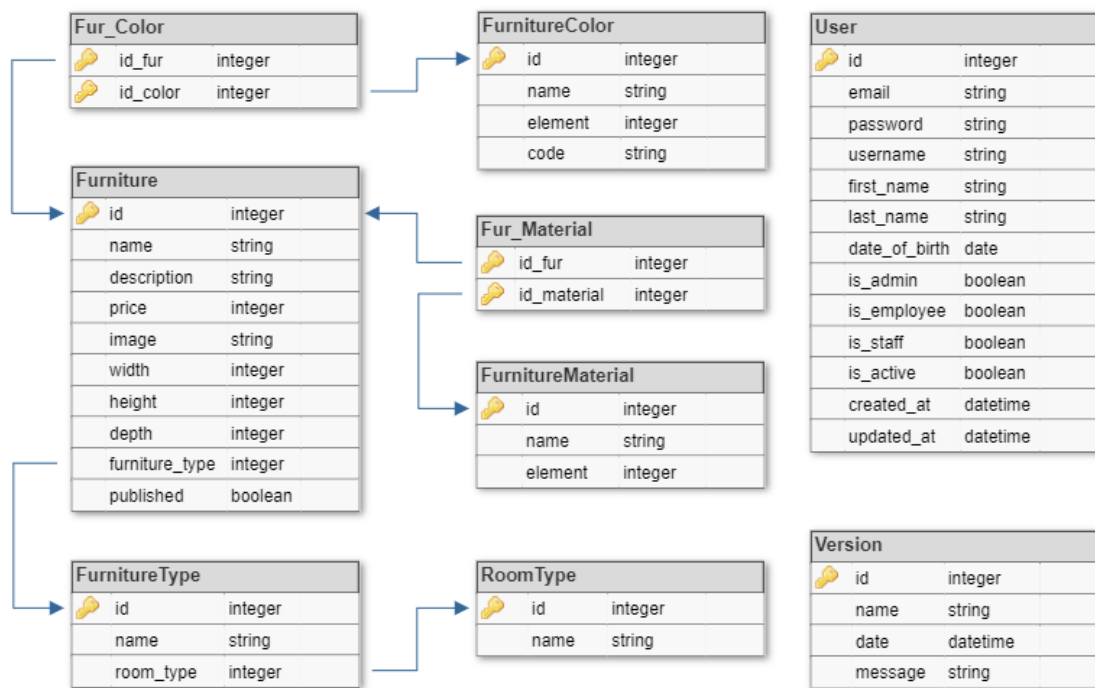
Hình 2.2.21. Quan hệ tương sinh – tương khắc theo ngũ hành

Nội thất phong thủy dựa trên ngũ hành sẽ được xem xét ở nhiều góc độ như màu sắc, chất liệu, khối lượng, hình dáng, hướng đặt của đồ nội thất dựa trên mệnh (ngũ hành) của gia chủ. Xem xét những góc độ này, sẽ có những cách phối hợp và bài trí để tăng sự thịnh vượng, giàu sang của gia chủ.

2.2. Thiết kế hệ thống

2.2.1. RESTful API

Thiết kế cơ sở dữ liệu



Hình 2.2.22. Lược đồ cơ sở dữ liệu

Lược đồ cơ sở dữ liệu được mô tả như sau:

- Bảng User: thông tin dùng để xác thực người dùng. Gồm các trường email, mật khẩu, tài khoản, họ tên, ngày sinh, ngày tạo, ngày cập nhật và 4 cờ trạng thái là người quản trị (admin), nhân viên (employee), khả dụng (active), có quyền superuser (staff).
- Bảng Version: phiên bản dữ liệu json.
- Bảng RoomType: thông tin loại phòng.
- Bảng FurnitureType: thông tin loại nội thất, mỗi loại nội thất sẽ thuộc về một loại phòng nào đó.
- Bảng FurnitureColor: thông tin màu sắc của đồ nội thất theo ngũ hành.

- Bảng FurnitureMaterial: thông tin vật liệu dùng để tạo ra đồ nội thất theo ngũ hành.
- Bảng Furniture: thông tin của đồ nội thất bao gồm tên, mô tả, giá, kích thước, màu sắc, vật liệu, loại phòng và trạng thái. Mỗi món đồ nội thất sẽ thuộc về một loại nội thất, gồm nhiều màu sắc và chất liệu tạo nên nó.

Định dạng URI

Các endpoint của REST API chính là URI. Định dạng URI chung cho các tác vụ CRUD được thiết kế theo quy tắc `/api/<version>/<model>/<id>/`. Trong đó:

- version: là phiên bản mà API cung cấp. Ví dụ v1, v2, ...
- model: là danh từ số nhiều chỉ model sẽ được thực hiện tác vụ CRUD. Ví dụ như thao tác CRUD với đồ nội thất thì model lúc này là “furnitures”.
- id: là số định danh của đối tượng, dùng để chỉ chính xác đối tượng của một model

Dưới đây là danh sách mô tả các REST endpoint v1:

STT	URI	Payload	Auth	HTTP method	Mô tả
1	/furnitures/			GET	Lấy danh sách tất cả món đồ nội thất
2	/furnitures/1/			GET	Lấy một món đồ nội thất có id là 1
3	/furnitures/	Yes	Yes	POST	Tạo mới một món đồ nội thất
4	/furnitures/1/	Yes	Yes	PUT	Cập nhật thông tin một món đồ nội thất có id là 1
5	/furnitures/1/		Yes	DELETE	Xóa một món đồ nội thất có id là 1
6	/room-types/			GET	Lấy danh sách tất cả

					loại phòng
7	/room-types/1/			GET	Lấy một loại phòng có id là 1
8	/room-types/	Yes	Yes	POST	Tạo mới một loại phòng
9	/room-types/1/	Yes	Yes	PUT	Cập nhật thông tin một loại phòng có id là 1
10	/room-type /1/		Yes	DELETE	Xóa một loại phòng có id là 1
11	/furniture-types/			GET	Lấy danh sách tất cả loại đồ nội thất
12	/furniture-types/1/			GET	Lấy loại đồ nội thất có id là 1
13	/furniture-types/	Yes	Yes	POST	Tạo mới một loại đồ nội thất
14	/furniture-types/1/	Yes	Yes	PUT	Cập nhật thông tin một loại đồ nội thất có id là 1
15	/furniture-types/1/		Yes	DELETE	Xóa một loại đồ nội thất có id là 1
16	/colors/			GET	Lấy danh sách tất cả màu sắc nội thất
17	/colors/1/			GET	Lấy một màu sắc nội thất có id là 1
18	/colors/	Yes	Yes	POST	Tạo mới một màu sắc nội thất
19	/colors/1/	Yes	Yes	PUT	Cập nhật thông tin

					một màu sắc nội thất có id là 1
20	/colors/1/		Yes	DELETE	Xóa một màu sắc nội thất có id là 1
21	/materials/			GET	Lấy danh sách tất cả vật liệu nội thất
22	/materials/1/			GET	Lấy một vật liệu nội thất có id là 1
23	/materials/	Yes	Yes	POST	Tạo mới vật liệu nội thất
24	/materials/1/	Yes	Yes	PUT	Cập nhật thông tin một vật liệu nội thất có id là 1
25	/materials/1/		Yes	DELETE	Xóa một vật liệu nội thất có id là 1
26	/versions/			GET	Lấy danh sách tất cả phiên bản json
27	/versions/1/			GET	Lấy thông tin một phiên bản json có id là 1
28	/versions/	Yes	Yes	POST	Tạo mới một phiên bản json
29	/token/create/	Yes		POST	Tạo mới một cặp access và refresh token
30	/token/refresh/	Yes		POST	Tạo mới access token bằng refresh token
31	/token/verify/	Yes		POST	Kiểm tra token hợp lệ

32	/token/blacklist/	Yes		POST	Thêm token vào blacklist
33	/upload-addressables/	Yes	Yes	POST	Tải lên dữ liệu đóng gói của Unity addressables
34	/download-addressables/			GET	Tải xuống dữ liệu đóng gói của Unity addressables

Bảng 2.2.3. Danh sách URI được cung cấp bởi REST API server

Các request gửi tới các endpoint có thể kèm theo payload là những dữ liệu định dạng json hoặc form-multipart và auth là access token.

Ví dụ để tạo một cặp access và refresh token thì thông tin cần thiết của một request bao gồm:

- URI: [host-address:port]/api/v1/token/create/
- HTTP method: POST
- Content Type: JSON
- Payload: { “email”: “abc@gmail.com”, “password”: “matkhou” }

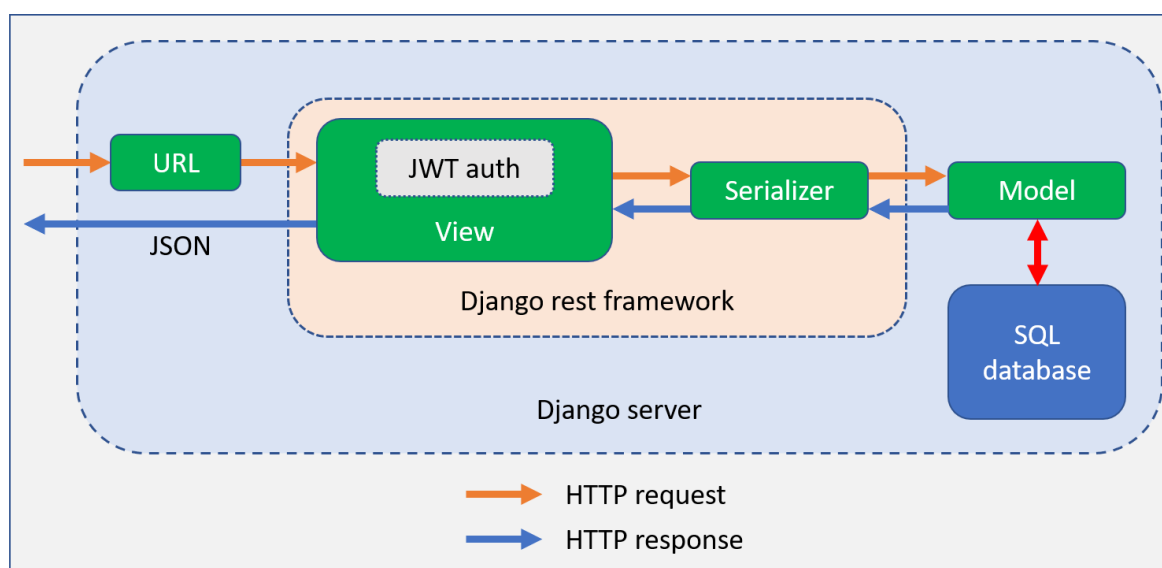
Dưới đây là danh sách một số status code thông dụng được trả về từ response:

STT	Code	TextCode	Mô tả
1	200	OK	Thành công
2	201	Created	Đã tạo ra một tài nguyên thành công
3	204	No Content	Thành công nhưng response không chứa nội dung
4	205	Reset Content	Thành công và báo hiệu cho client sắp xếp lại nội dung
5	301	Moved Permanently	Chuyển hướng tới tài nguyên được chỉ định ở địa chỉ mới

6	400	Bad Request	Lỗi server không hiểu yêu cầu
7	401	Unauthorized	Lỗi yêu cầu không được xác thực hợp lệ
8	403	Forbidden	Lỗi yêu cầu không được phép
9	404	Not Found	Lỗi yêu cầu không được tìm thấy
10	405	Method Not Allow	Lỗi server từ chối tiếp nhận loại phương thức HTTP cụ thể
11	500	Server Internal Error	Lỗi không mong muốn trên server nhằm từ chối việc thực hiện yêu cầu

Bảng 2.2.4. Một số status code thông dụng của giao thức HTTP

Kiến trúc các thành phần của REST API server



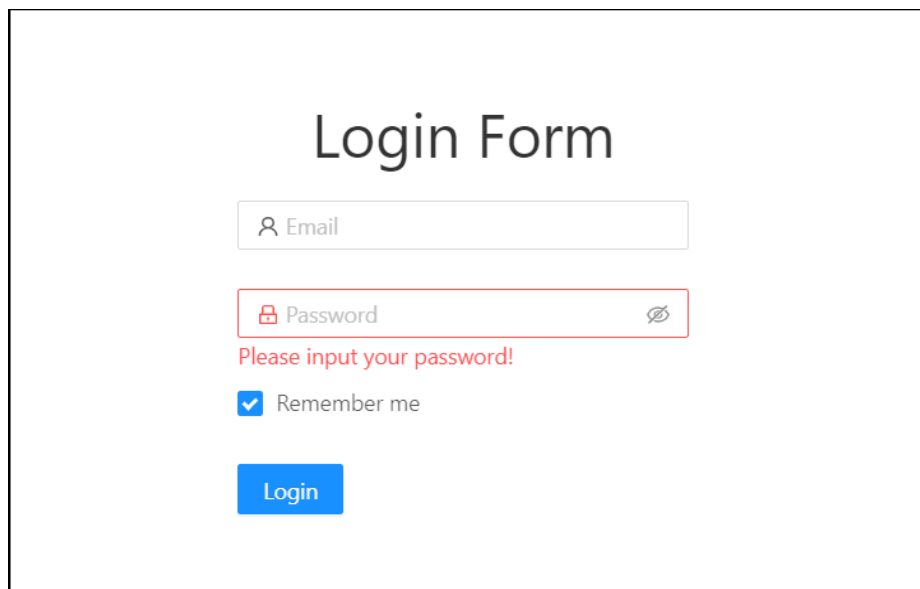
Hình 2.2.23. Kiến trúc các thành phần của REST API

Sử dụng django và django rest framework để xây dựng REST API và dùng JWT để xác thực cấp quyền cho request. Chuẩn dữ liệu để trao đổi giữa server và client là JSON. Đối với các API liên quan tới CRUD service sẽ cài đặt ModelViewSet ở lớp View và ModelSerializer ở lớp Serializer để tự động xử lý các phương thức HTTP theo các tác vụ CRUD và chuyển đổi dữ liệu giữa JSON với model tương ứng.

2.2.2. Ứng dụng web quản lý

Giao diện web

Màn hình đăng nhập

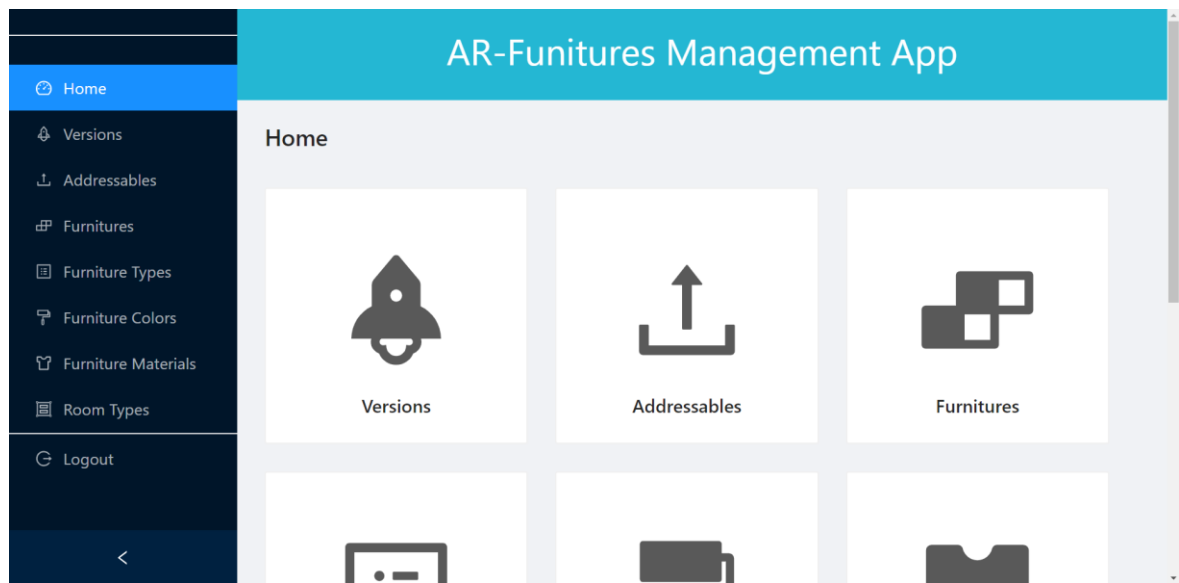


The image shows a web login form titled "Login Form". It contains an email input field with a magnifying glass icon and the placeholder text "Email". Below it is a password input field with a lock icon, the placeholder text "Password", and a toggle icon. A red error message "Please input your password!" is displayed below the password field. There is a checked checkbox labeled "Remember me" and a blue "Login" button at the bottom.

Hình 2.2.24. Giao diện màn hình đăng nhập

Màn hình đăng nhập gồm có trường nhập email, mật khẩu, ghi nhớ tự động đăng nhập, tùy chọn hiển thị mật khẩu và nút đăng nhập.

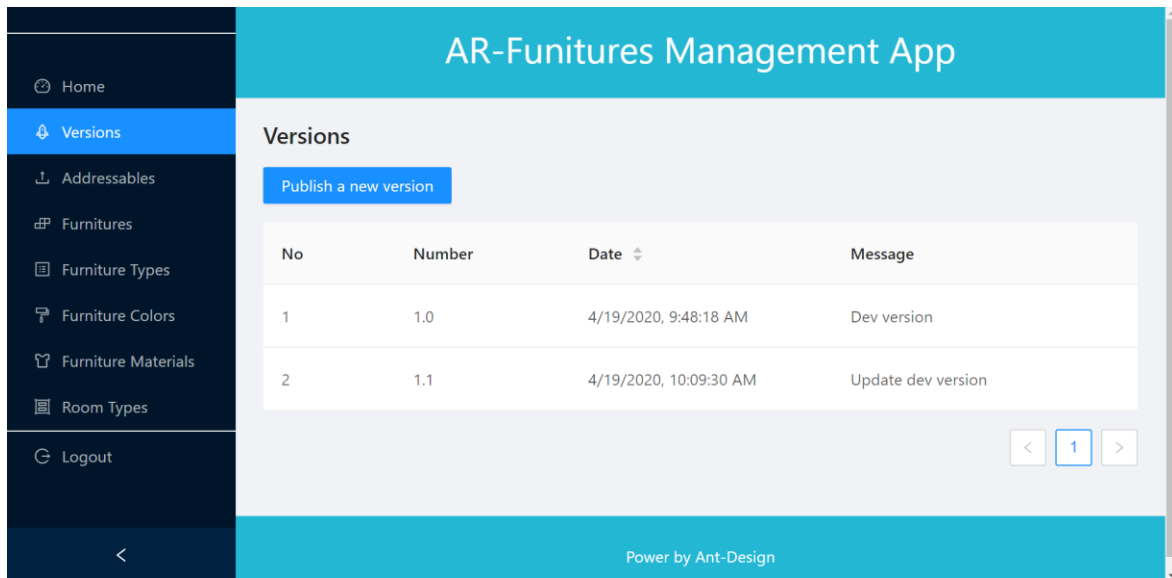
Trang chủ



Hình 2.2.25. Giao diện trang chủ ứng dụng web quản lý

Trang chủ gồm một giao diện điều hướng phía bên trái và các biểu tượng điều hướng trực tiếp ở bên phải.

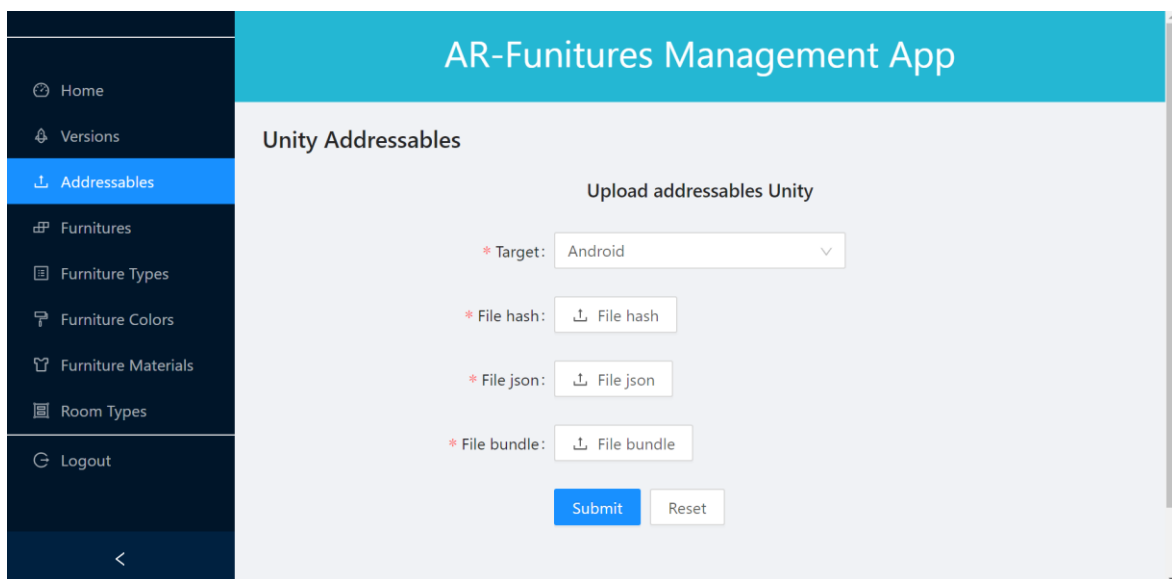
Quản lý phiên bản



Hình 2.2.26. Giao diện quản lý phiên bản

Quản lý phiên bản sử dụng một bảng dùng để hiển thị danh sách phiên bản, nút tạo mới phiên bản và cụm các nút dùng để phân trang khi danh sách quá dài.

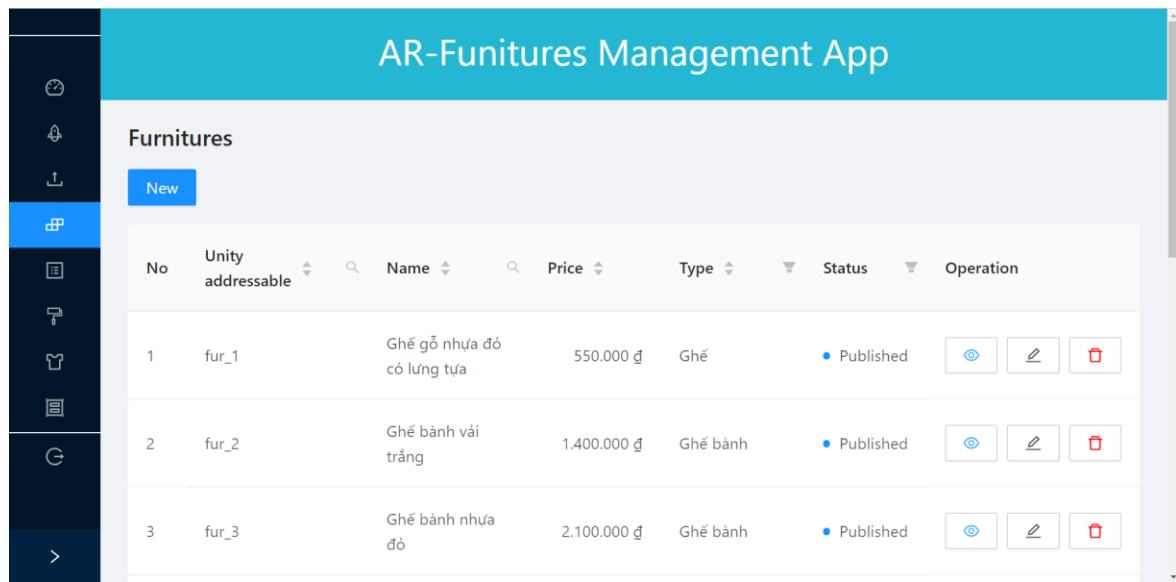
Upload Unity addressables



Hình 2.2.27. Giao diện tải lên asset bundles của Unity

Giao diện này gồm các trường dùng để chọn file asset bundles của Unity để tải lên server, các file này bao gồm một file json, một file hash và một file bundle. Nút submit để tiến hành tải lên và nút reset để bỏ chọn tất cả các file.

Quản lý đồ nội thất



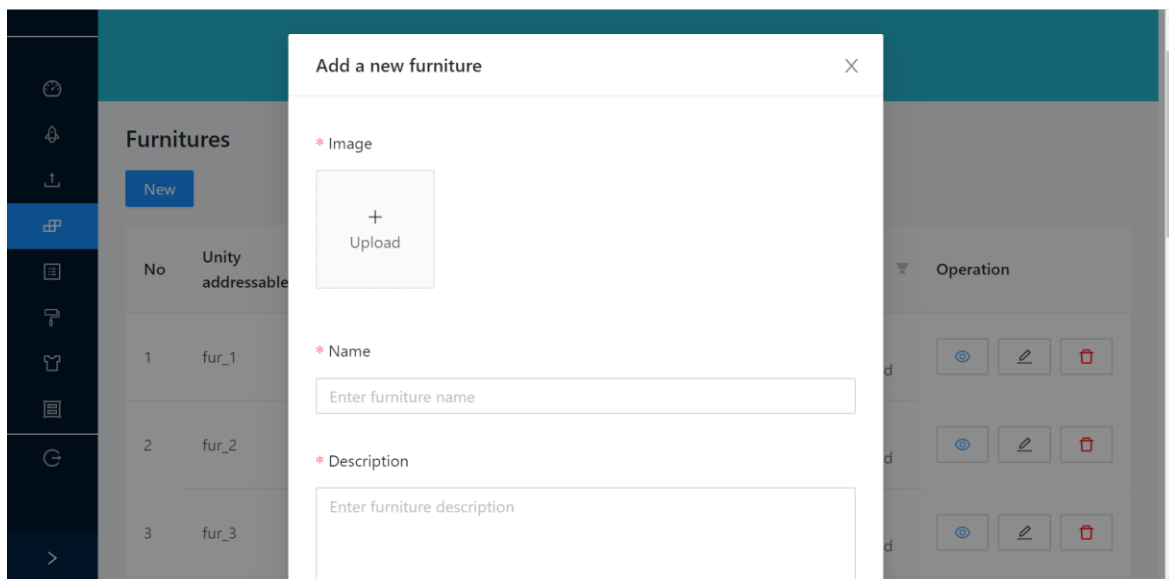
The screenshot shows the 'AR-Furniture Management App' interface. It features a sidebar with navigation icons and a main content area titled 'Furnitures'. A 'New' button is located above a table of furniture items. The table has columns for No, Unity addressable, Name, Price, Type, Status, and Operation. Three items are listed, all with a 'Published' status.

No	Unity addressable	Name	Price	Type	Status	Operation
1	fur_1	Ghế gỗ nhựa đỏ có lưng tựa	550.000 đ	Ghế	Published	[Eye] [Edit] [Delete]
2	fur_2	Ghế bành vải trắng	1.400.000 đ	Ghế bành	Published	[Eye] [Edit] [Delete]
3	fur_3	Ghế bành nhựa đỏ	2.100.000 đ	Ghế bành	Published	[Eye] [Edit] [Delete]

Hình 2.2.28. Giao diện quản lý nội thất

Tương tự như quản lý phiên bản, quản lý đồ nội thất cũng sẽ dùng một bảng để hiển thị danh sách các món đồ nội thất, một số trường sẽ được ẩn đi. Ngoài ra, còn có một số nút tùy chọn để tạo mới, xem chi tiết, chỉnh sửa và xóa đồ nội thất. Đối với bảng liệt kê danh sách đồ nội thất sẽ có một số tùy chọn như tìm kiếm, sắp xếp và lọc.

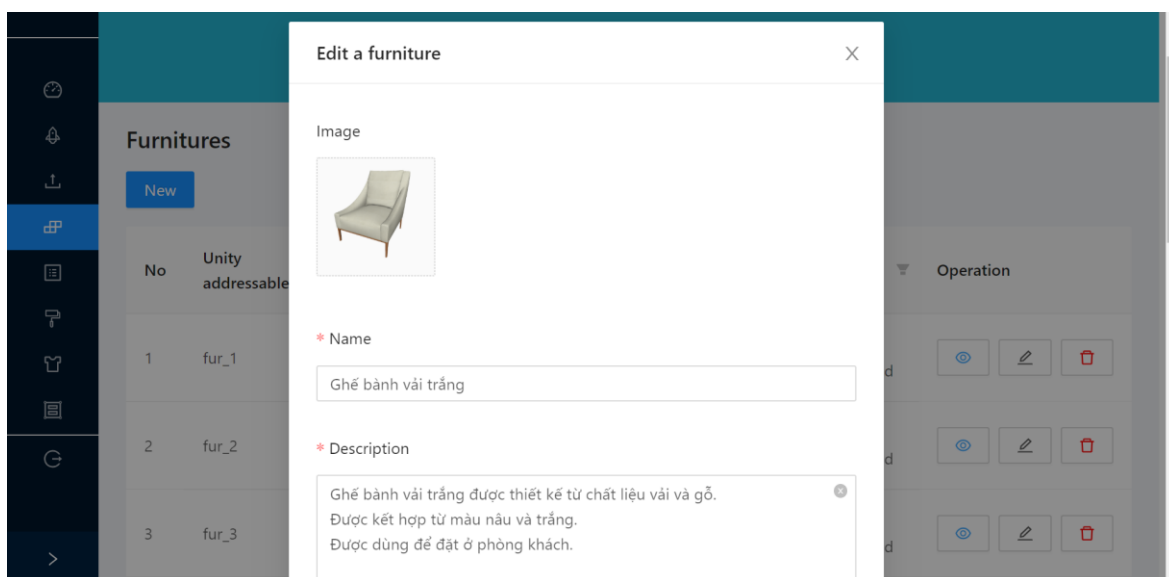
Tạo mới đồ nội thất



Hình 2.2.29. Giao diện tạo mới nội thất

Tính năng này cho phép người dùng chọn hình ảnh nội thất và nhập thông tin cần thiết để tạo mới thông tin một món đồ nội thất. Một popup được sử dụng để người dùng nhập thông tin và gửi yêu cầu.

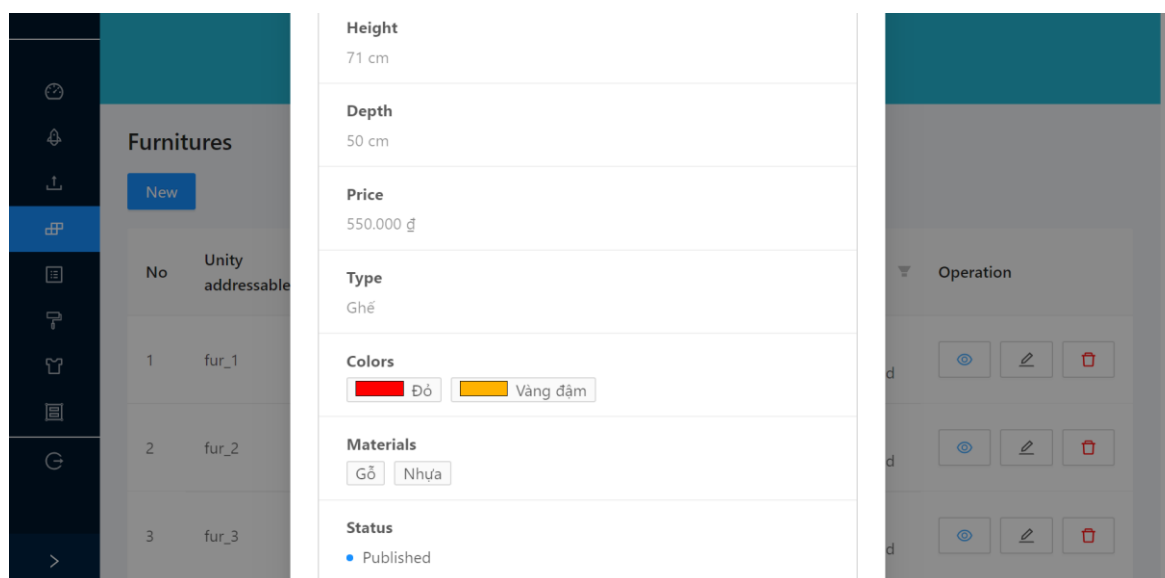
Chỉnh sửa thông tin đồ nội thất



Hình 2.2.30. Giao diện cập nhật thông tin nội thất

Tương tự như tạo mới đồ nội thất, chỉnh sửa đồ nội thất cũng sử dụng một popup hiện lên cho phép người dùng chỉnh sửa thông tin và gửi yêu cầu cập nhật. Khi người dùng nhấp vào nút chỉnh sửa trên một dòng dữ liệu trong bảng nội thất, dữ liệu của nội thất này sẽ được điền tự động vào các trường tương ứng trên popup và cho phép người dùng chỉnh sửa lại.

Xem thông tin chi tiết đồ nội thất



Hình 2.2.31. Một phần giao diện xem chi tiết nội thất

Xem chi tiết đồ nội thất cũng sử dụng một popup để hiển thị thông tin chi tiết của các món đồ nội thất. Khi người dùng nhấp vào biểu tượng xem chi tiết trên một dòng dữ liệu của bảng nội thất, dữ liệu này sẽ được điền vào các trường tương ứng trên popup và hiển thị cho người dùng xem.

Ngoài ra khi người dùng chọn vào biểu tượng xóa trên một dòng dữ liệu của bảng nội thất, một popup nhỏ sẽ hiện ra nhắc nhở người dùng có muốn tiếp tục hay không. Nếu người dùng vẫn muốn xóa thì sẽ tiến hành gửi yêu cầu tới server xóa dữ liệu của nội thất tương ứng với dòng dữ liệu đó.

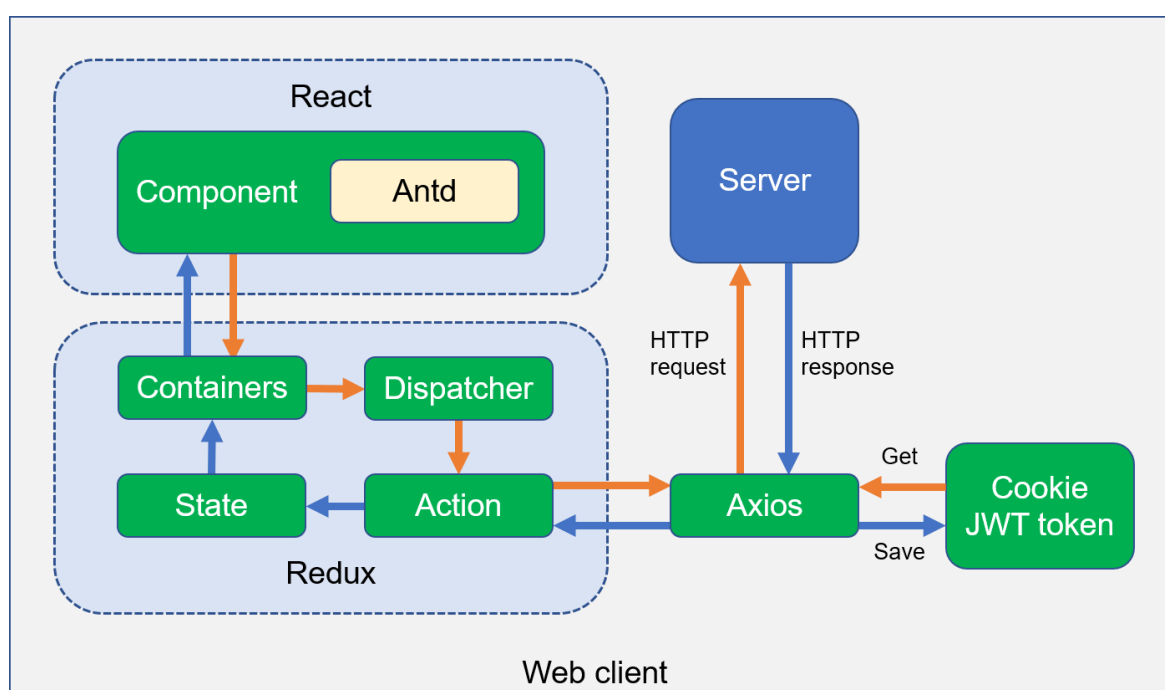
Giao diện dành cho quản lý loại phòng, loại đồ nội thất, màu sắc, vật liệu nội thất cũng tương tự.

Một số tính năng phụ:

- Đăng xuất: kết thúc phiên làm việc.

- Thông báo: hiện thông báo thành công hoặc thông báo lỗi.
- Tìm kiếm: tìm kiếm những dòng dữ liệu của một cột trong bảng.
- Sắp xếp: sắp xếp dữ liệu của một cột bảng.
- Phân trang: cụm phân trang dành cho bảng.
- Lọc: lọc ra các trường trùng của một cột trong bảng.

Kiến trúc các thành phần của web client



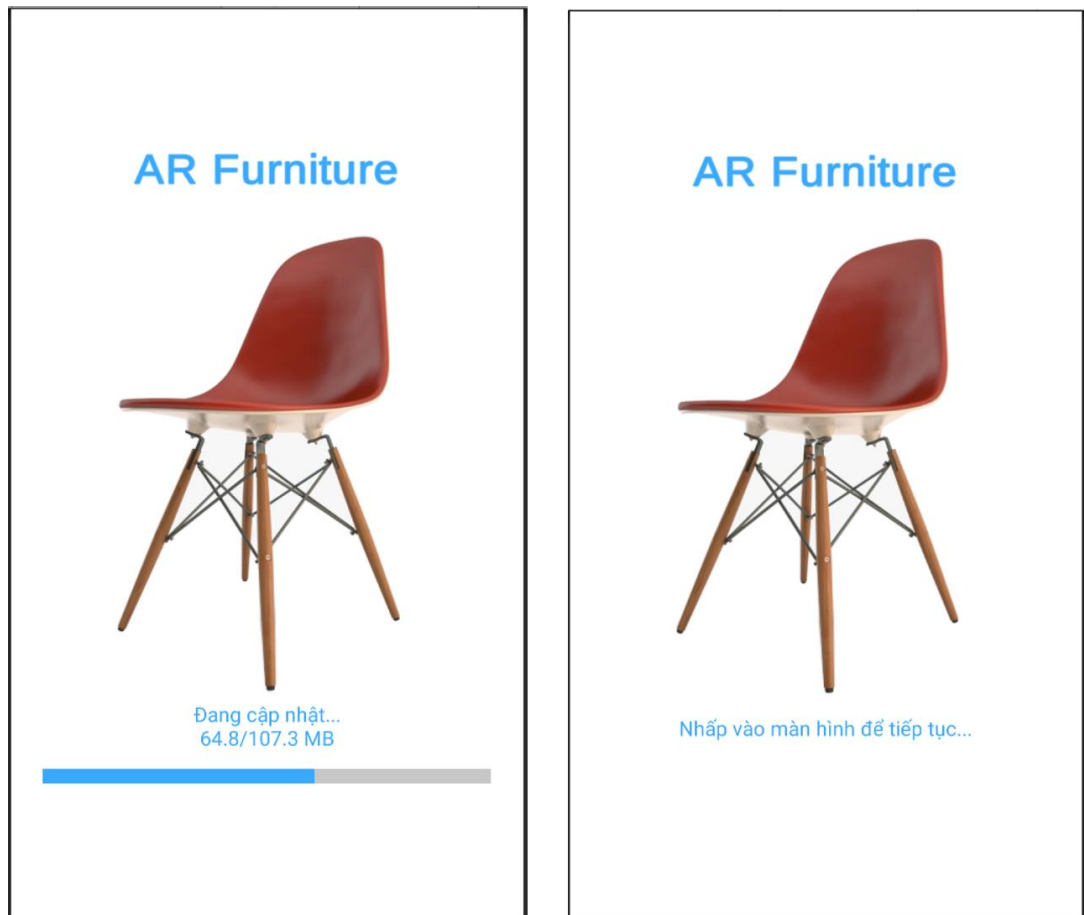
Hình 2.2.32. Kiến trúc các thành phần của ứng dụng web quản lý

Web client sẽ kết hợp react và redux để quản lý state dễ dàng hơn. Sử dụng Antd (Ant design) để tạo dựng giao diện người dùng và axios để trao đổi với REST API server. Ngoài ra, sử dụng cookie lưu trữ token để ghi nhớ phiên đăng nhập và tạo mới access token khi hết hạn.

2.2.3. Mobile app

Giao diện

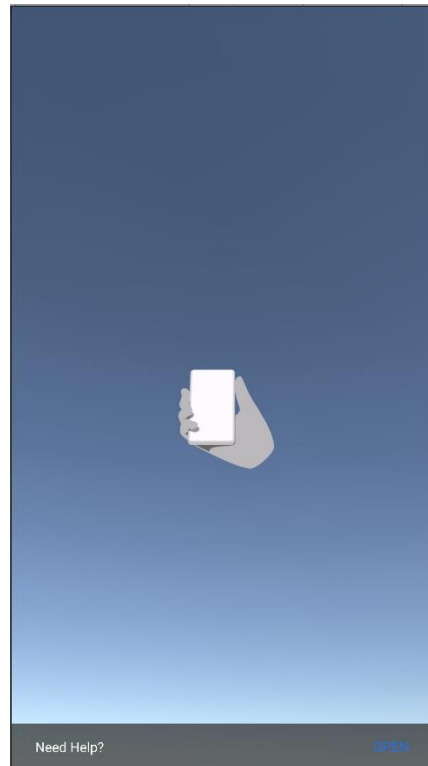
Kiểm tra cập nhật



Hình 2.2.33. Giao diện cập nhật ứng dụng AR mobile

Sử dụng một phân cảnh để thực hiện tính cập nhật dữ liệu. Màn hình cập nhật được thiết kế đơn giản, hiện đại bao gồm tên ứng dụng, hình ảnh nền, dòng chữ trạng thái và thanh phân trăm cập nhật.

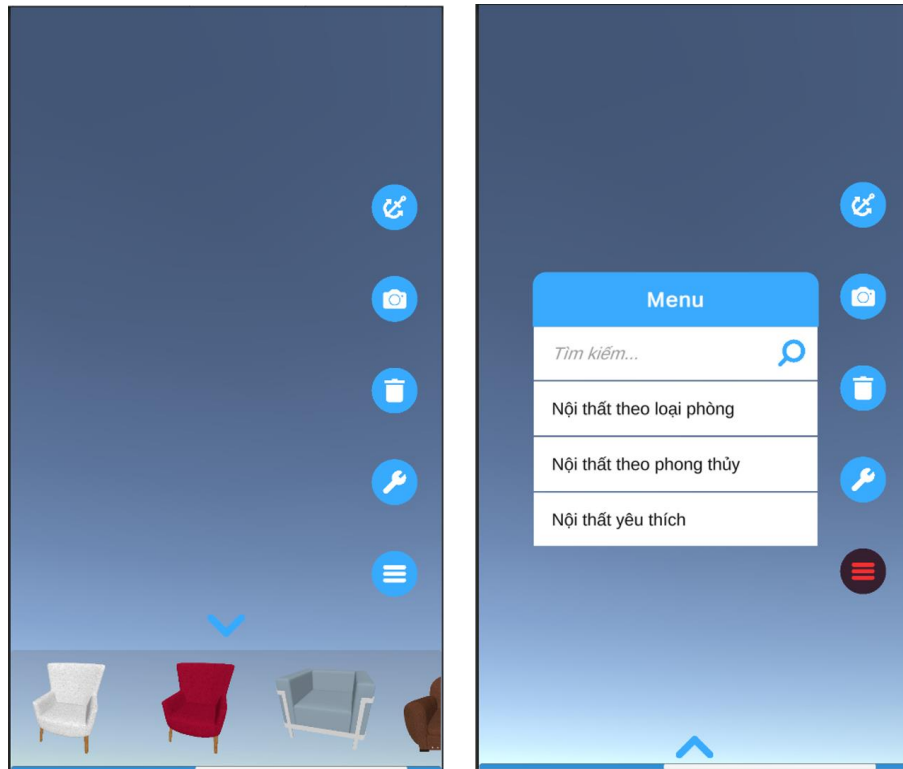
Tìm mặt phẳng



Hình 2.2.34. Giao diện hướng dẫn dò tìm mặt phẳng

Giao diện hướng dẫn dò tìm mặt phẳng được hỗ trợ bởi thư viện ARCore, gồm có một ảnh động hướng dẫn di chuyển điện thoại quét xung quanh và nút open để hiện hướng dẫn chi tiết.

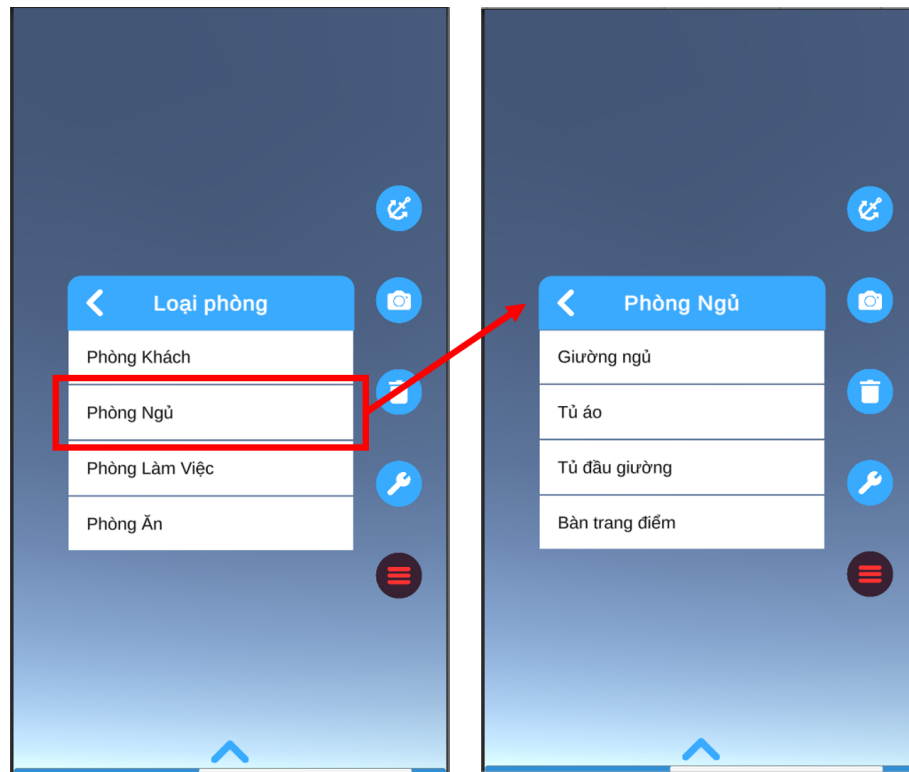
Menu chính



Hình 2.2.35. Giao diện menu của ứng dụng AR mobile

Giao diện menu chính sẽ gồm danh sách các mô hình nội thất 3D phía bên dưới và một menu danh sách các tùy chọn liệt kê đồ nội thất.

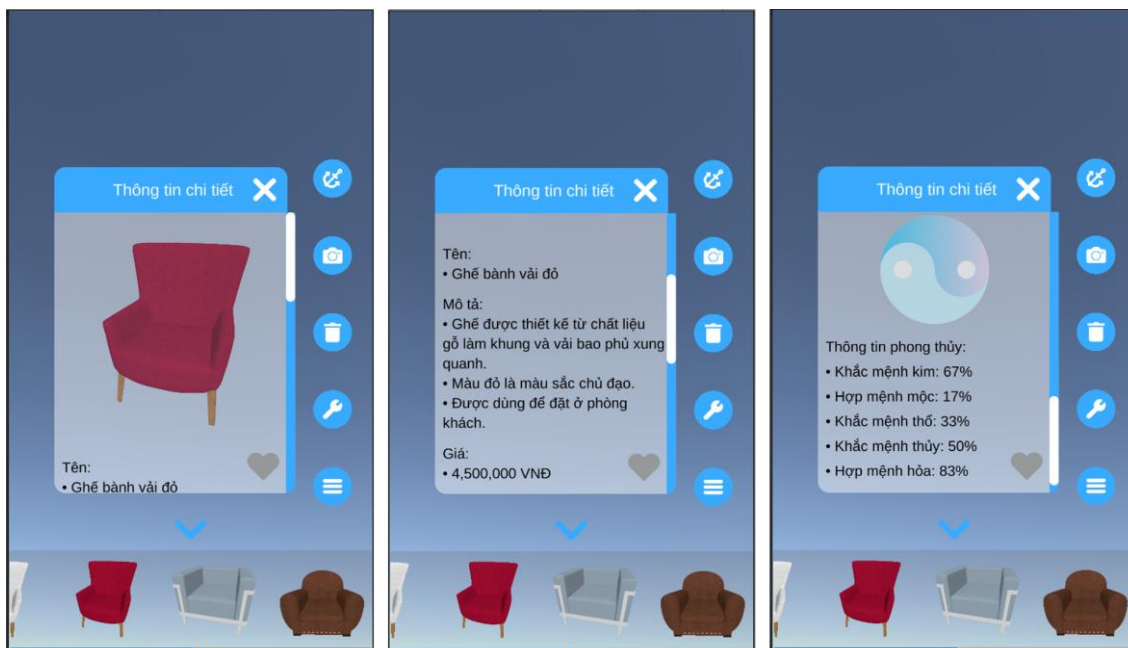
Liệt kê nội thất theo loại phòng



Hình 2.2.36. Giao diện liệt kê loại phòng của ứng dụng AR mobile

Liệt kê nội thất dựa theo loại nội thất và loại phòng là một menu điều hướng 2 cấp độ: cấp độ 1 là danh sách loại phòng, cấp độ 2 là danh sách loại nội thất dựa theo loại phòng.

Thông tin chi tiết của đồ nội thất



Hình 2.2.37. Giao diện xem thông chi tiết nội thất của ứng dụng AR mobile

Thông tin chi tiết của một món đồ nội thất được hiển thị dạng popup gồm ảnh, tên, mô tả, giá, kích thước, màu sắc, vật liệu và thông tin phong thủy. Khi nhấn vào hình ảnh một món đồ nội thất trong danh sách thì hiển thị thông tin chi tiết của món đồ.

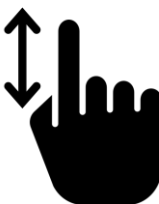
Thông tin người dùng

The screenshot shows a mobile application interface with a dark blue background. A central white form titled 'Thông tin cá nhân' (Personal Information) is displayed. The form contains the following elements: a text input field for 'Tên' (Name) with the value 'Nguyễn Thanh Khoa'; a date picker for 'Ngày sinh' (Date of Birth) showing '1997 / Tháng 8 / 29'; a checkbox labeled 'Âm lịch' (Lunar Calendar) which is checked; and two lines of text: 'Dương lịch: 30/09/1997.' and 'Can: Đinh, Chi: Sửu, Mệnh: Thủy.' At the bottom of the form are two buttons: 'OK' and 'Hủy' (Cancel). To the right of the form is a vertical sidebar with five circular icons: a share icon, a camera icon, a trash can icon, a wrench icon, and a hamburger menu icon. A blue upward-pointing arrow is visible at the bottom center of the screen.

Hình 2.2.38. Giao diện nhập thông tin người dùng

Thông tin người dùng là một popup gồm các trường để người dùng nhập vào tên, ngày sinh, checkbox tùy chọn ngày âm lịch và thông tin can – chi – mệnh của người dùng.

Thao tác màn hình đối với tính năng AR

Thao tác	Tên thao tác	Chức năng
	Kéo thả bằng một ngón tay	Đặt mô hình đồ nội thất vào bối cảnh cảnh: sử dụng một ngón tay để chọn và kéo thả mô hình đồ nội thất trong danh sách vào mặt phẳng nằm ngang trong phân cảnh.
	Nhấn	Chọn hoặc bỏ chọn đồ nội thất: dùng một ngón tay nhấp một lần vào mô hình đồ nội thất đã đặt vào phân cảnh để chọn và vòng tròn màu xanh xuất hiện. Nhấp một lần nữa vào mô hình hoặc vị trí ngoài mô hình để bỏ chọn.
	Vuốt sang trái/phải bằng một ngón tay	Di chuyển mô hình nội thất theo trục X: vuốt sang trái/phải bằng một ngón tay để di chuyển mô hình đồ nội thất đang được chọn sang trái/phải, tương ứng với trục X của mặt phẳng nằm ngang trong phân cảnh.
	Vuốt lên/xuống bằng một ngón tay	Di chuyển mô hình đồ nội thất theo trục Z: vuốt lên/xuống bằng một ngón tay để di chuyển mô hình đồ nội thất đang được chọn ra xa hoặc lại gần, tương ứng với trục Z của mặt phẳng nằm ngang trong phân cảnh.
	Xoay 2 ngón tay	Xoay đồ nội thất theo trục Y: sử dụng 2 ngón tay xoay theo hoặc ngược chiều kim đồng hồ để quay mô hình nội thất đang được chọn quanh trục Y của chính nó.

Bảng 2.2.5. Danh sách chức năng các thao tác tay trên ứng dụng AR mobile

Danh sách các nút chức năng chính nằm ở thanh bên phải màn hình

Nút chức năng	Tên	Thao tác	Chức năng
	Refresh	Nhấn giữ 2 giây	Tạo mới lại phân cảnh
	Screenshot	Nhấn	Chụp ảnh màn hình
	Delete	Nhấn	Xóa mô hình nội thất đang được chọn trong phân cảnh
		Nhấn giữ 2 giây	Xóa hết tất cả các mô hình nội thất trong phân cảnh
	Setting	Nhấn	Hiển thị danh sách cài đặt
	Menu	Nhấn	Hiển thị danh sách menu

Bảng 2.2.6. Danh sách các nút chức năng chính trên ứng dụng AR mobile

Danh sách một số nút chức năng khác

Nút chức năng	Tên	Thao tác	Chức năng
	Search	Nhấn	Tìm kiếm theo tên nội thất
	Close	Nhấn	Đóng popup đang mở
	Like	Nhấn	Thêm một món đồ nội thất vào danh sách yêu thích

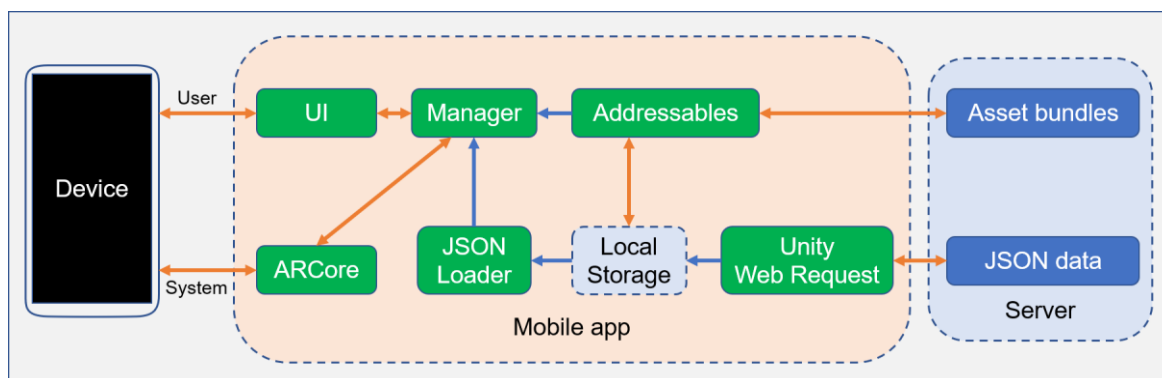
	Dislike	Nhấn	Bỏ một món đồ nội thất từ danh sách yêu thích
	Back	Nhấn	Trở về tùy chọn trước đó
	Check	Nhấn	Chọn một tùy chọn
	Uncheck	Nhấn	Bỏ chọn một tùy chọn
	Show	Nhấn	Hiện thị danh sách đồ nội thất
	Hide	Nhấn	Ẩn danh sách đồ nội thất
	Furniture	Nhấn	Hiện thông tin chi tiết đồ nội thất
		Kéo thả	Đặt mô hình nội thất vào trong phân cảnh

Bảng 2.2.7. Danh sách một số nút chức năng khác trên ứng dụng AR mobile

Phiên bản ARCore SDK for Unity

Phiên bản ARCore được sử dụng trong đề tài luận văn là phiên bản 1.16.0.

Kiến trúc các thành phần của ứng dụng mobile



Hình 2.2.39. Kiến trúc các thành phần của ứng dụng AR mobile

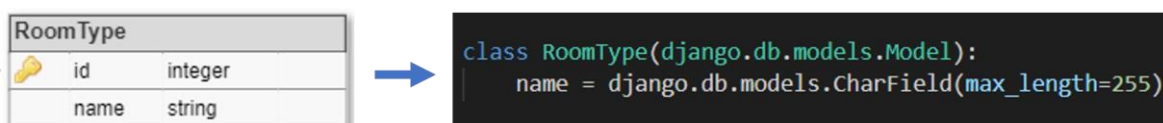
Ứng dụng mobile gồm phần UI sẽ tương tác trực tiếp với người dùng, thư viện ARCore sẽ trao đổi ở phía hệ thống thông qua ảnh camera và các cảm biến gia tốc IMU. Về phía trao đổi dữ liệu với server, sử dụng Unity Web Request giao tiếp với server REST để tải dữ liệu JSON và Addressables để tải những tài nguyên đóng gói bởi Unity Editor. Những dữ liệu được cung cấp bởi server sẽ được lưu trữ lại ở bộ nhớ trong và được truy xuất khi cần. Lúc truy xuất, bộ JSON Loader sẽ được sử dụng để đọc dữ liệu JSON và Addressables sẽ đọc các tài nguyên đã tải trước đó trong bộ nhớ. Manager là thành phần dùng để xử lý yêu cầu của người dùng thông qua UI và giao tiếp với thư viện ARCore khi cần thiết như đặt mô hình 3D vào mặt phẳng nằm ngang của ARCore trong phân cảnh.

Chương 3. Cài đặt giải pháp

3.1. REST API server

3.1.1. Cài đặt model trong django

Model là những lớp dùng để truy xuất cơ sở dữ liệu, chứa các trường bắt buộc. Tương ứng với các bảng RoomType, FurnitureType, FurnitureColor, FurnitureMaterial, Furniture và Version sẽ được cài đặt thành các model trong django. Mỗi model sẽ bao gồm các trường là các cột của bảng tương ứng.



Hình 2.3.1. Cài đặt bảng RoomType bằng model trong django

Ví dụ bảng RoomType gồm các cột id, name thì khi cài đặt bằng ngôn ngữ python trong django thì sẽ chỉ có trường name còn trường id sẽ được tạo tự động. Các bảng còn lại cũng được cài đặt tương tự.

Danh sách các model được sử dụng:

Tên model	Mô tả
User	Quản lý tài khoản người dùng
RoomType	Quản lý loại phòng
FurnitureType	Quản lý loại đồ nội thất
Color	Quản lý màu sắc nội thất
Material	Quản lý vật liệu nội thất
Furniture	Quản lý đồ nội thất
Version	Quản lý phiên bản dữ liệu json

Bảng 2.3.1. Danh sách các model được cài đặt

3.1.2. Cài đặt serializer trong DRF

Serializer là class dùng để chuyển đổi dữ liệu giữa json (hoặc xml hay multipart-form) thành các kiểu dữ liệu python và ngược lại, mục đích là làm người phiên dịch trung gian giữa dữ liệu được gửi lên từ client và dữ liệu được truy xuất từ Model. Một Serializer có thể có một hoặc không có Model.

Đối với các thao tác CRUD thì sẽ sử dụng ModelSerializer. ModelSerializer là một python class kế thừa từ Serializer, tự động kiểm tra dữ liệu được gửi lên từ client và chuyển đổi dữ liệu tương ứng sang các kiểu dữ liệu python dựa trên các trường của model được cung cấp.

```
class RoomType(django.db.models.Model):
    name = django.db.models.CharField(max_length=255)
```



```
class RoomTypeSerializer(rest_framework.serializers.ModelSerializer):
    class Meta:
        model = RoomType
        fields = ('id', 'name',)
```

Bảng 2.3.2. Cài đặt RoomTypeSerializer trong DRF

Ví dụ khi tạo Serializer cho model RoomType thì chỉ cần cung cấp class model RoomType và các trường dùng để xác thực và chuyển đổi dữ liệu gồm id và name. Các model còn lại cũng được cài đặt tương tự.

Danh sách các Serializer:

Tên Serializer	Model sử dụng	Mô tả
UserSerializer	User	Chuyển đổi dữ liệu và thao tác dữ liệu
RoomTypeSerializer	RoomType	
FurnitureTypeSerializer	FurnitureType	
ColorSerializer	Color	
MaterialSerializer	Material	

FurnitureSerializer	Furniture	
VersionSerializer	Version	
AddressablesFileSerializer	Không có	Tải lên các asset bundles của Unity

Bảng 2.3.3. Danh sách các Serializer được cài đặt trong DRF

3.1.3. Cài đặt View trong DRF

View trong DRF được xem như là một Controller dùng để phân quyền, điều khiển truy cập và thao tác dựa trên các HTTP method (GET, POST, PUT, DELETE). ModelViewSet được sử dụng để tự động chuyển các request liên quan đến thao tác CRUD tới hàm xử lý phù hợp (hàm xử lý của Serializer – list, retrieve, create, update, destroy) dựa trên các phương thức HTTP.

```
class RoomTypeView(rest_framework.viewsets.ModelViewSet):
    queryset = RoomType.objects.all()
    serializer_class = RoomTypeSerializer
    permission_classes = [IsAdminOrReadOnly]
```

Hình 2.3.2. Cài đặt RoomTypeView trong DRF

Ví dụ để cài đặt một ModelViewSet cơ bản cho model RoomType thì chỉ cần cung cấp cho nó một tập các đối tượng được nó quản lý thông qua queryset, một class serializer dùng để chuyển đổi và thao tác dữ liệu dữ liệu, một class dùng để phân quyền. Để tạo các ModelViewSet cho các model còn lại cũng tương tự.

Danh sách các View:

View	Loại View	Mô tả
RoomTypeView	ModelViewSet	Hỗ trợ thao tác CRUD dựa trên HTTP method (GET, POST, PUT, DELETE)
FurnitureTypeView		
ColorView		
MaterialView		

FurnitureView		
VersionView	ListCreateAPIView	Cho phép tạo mới (POST) và lấy (GET) danh sách phiên bản json
FileUploadView	APIView	Hỗ trợ tải lên (POST) các asset bundles của Unity
FileDownloadView		Hỗ trợ tải xuống (GET) các asset bundles của Unity
MediaView		Hỗ trợ tải xuống (GET) hình ảnh tại /media/

Bảng 2.3.4. Danh sách các View được cài đặt trong DRF

ListCreateAPIView là một View chỉ hỗ trợ 2 hàm create (POST) và list (GET). APIView là một DRF View nhưng không tự động xử lý mà phải ghi đè các hàm cần dùng.

3.1.4. Routing trong DRF

Khi sử dụng Model View Set thì việc routing bằng DRF trở nên vô cùng đơn giản bằng cách sử dụng DefaultRouter. Chỉ cần cung cấp cho router này một tên đường dẫn URL tương ứng với view thì mọi việc vạch đường dựa trên phương thức HTTP sẽ được router này hỗ trợ đầy đủ.

```
router = rest_framework.routers.DefaultRouter()
router.register('room-types', RoomTypeView, basename='room_type')
urlpatterns = [django.urls.path('', django.urls.include(router.urls))]
```

Hình 2.3.3. Cài đặt router cho room-types trong DRF

Ví dụ khi sử dụng DRF routing cho một ModelViewSet thì chỉ cần cung cấp cho DefaultRouter một đường dẫn URL là /room-types/ và việc vạch đường dựa trên HTTP request được router thực hiện như sau:

URL	HTTP method	Chức năng
/room-types/	POST	Tạo mới một loại phòng
/room-types/	GET	Lấy tất cả danh sách loại phòng
/room-types/1/	GET	Lấy một loại phòng có id là 1
/room-types/1/	PUT	Cập nhập một loại phòng có id là 1
/room-types/1/	DELETE	Xóa một loại phòng có id là 1

Bảng 2.3.5. Danh sách URL của room-types được Default Router hỗ trợ

Các model còn lại cũng sẽ được cài đặt tương tự. Các routing khác sẽ sử dụng router mặc định được cung cấp bởi django ví dụ như tạo JWT token.

3.1.5. Xác thực request bằng JWT

Sử dụng một phiên bản thu gọn của thư viện JWTAuthentication là SimpleJWTAuthentication. Khi tạo token bằng email và password thành công thì server sẽ trả về một cặp access và refresh token. Thời gian sống của access token là 5 phút và refresh token là 15 ngày. Khi access token hết hạn thì có thể sử dụng refresh token để tạo một access token mới.

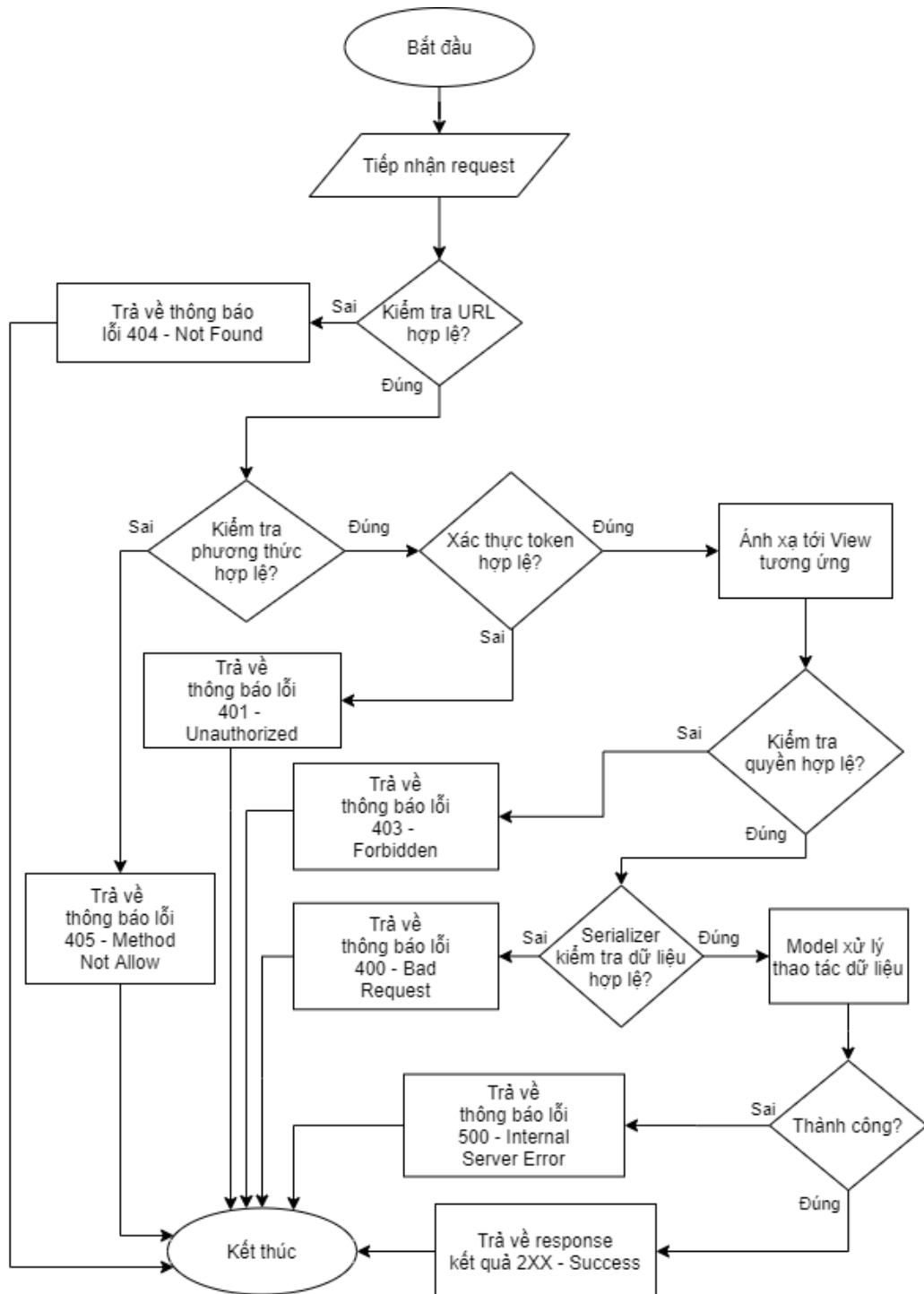
```
{  
  "refresh": "header.payload.signature",  
  "access": "header.payload.signature",  
  "expired_access_at": 1590044742000,  
  "expired_refresh_at": 1591254042000,  
  "timezone": "utc"  
}
```

Hình 2.3.4. Kết quả về khi tạo JWT token thành công

Kết quả của việc tạo mới một cặp access và refresh token bao gồm: access_tok, refresh_tok, thời gian hết hạn (kiểu DateTime trong JavaScript) access và refresh cùng với múi giờ.

3.1.6. Xử lý yêu cầu

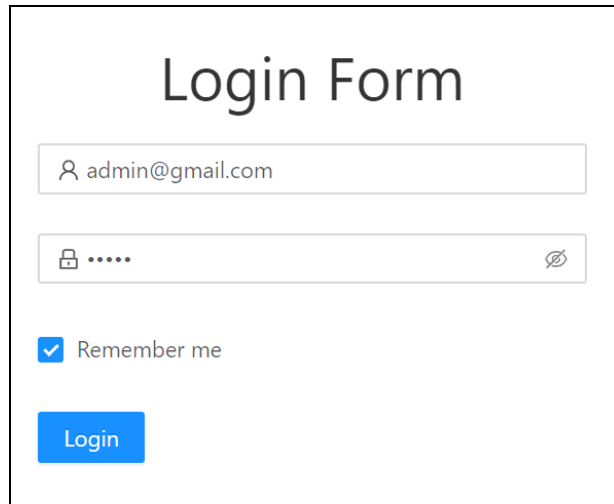
Quá trình xử lý một request của server có thể mô tả đơn giản hóa bằng lưu đồ sau:



Hình 2.3.5. Lưu đồ xử lý yêu cầu của REST API server

3.2. Web client app

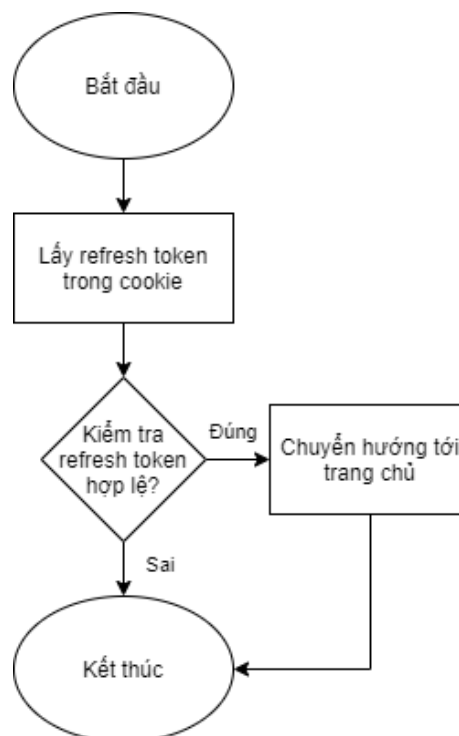
3.2.1. Đăng nhập – Xác thực người dùng



The image shows a web login form titled "Login Form". It contains two input fields: the first for an email address with the placeholder "admin@gmail.com", and the second for a password with masked dots ".....". Below the password field is a checkbox labeled "Remember me" which is checked. At the bottom is a blue button labeled "Login".

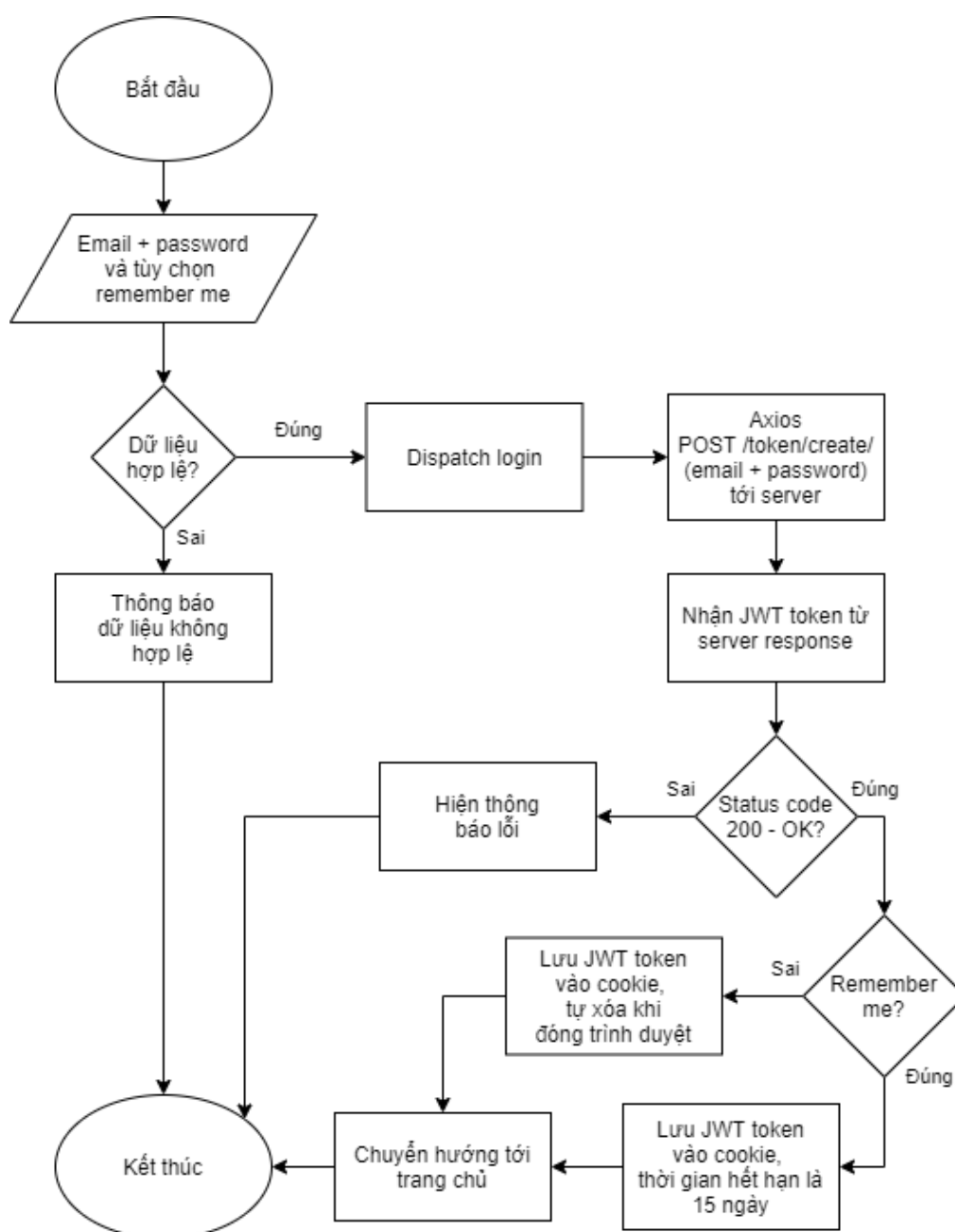
Hình 2.3.6. Form đăng nhập của ứng dụng web quản lý

Khi người dùng truy cập trang đăng nhập thì tại đây sẽ kiểm tra xem người dùng đã được đăng nhập hay chưa, nếu đã đăng nhập rồi thì sẽ chuyển hướng về trang chủ. Tính năng này sẽ sử dụng JWT token, dùng nó thay cho session để ghi nhớ phiên đăng nhập và xác thực các request.



Hình 2.3.7. Lưu đồ xử lý của chuyển hướng đăng nhập tới trang chủ

Khi người dùng nhập thông tin email, mật khẩu và tùy chọn ghi nhớ đăng nhập, sau đó người dùng nhấp chọn đăng nhập thì web client sẽ tiến hành xử lý như sau:

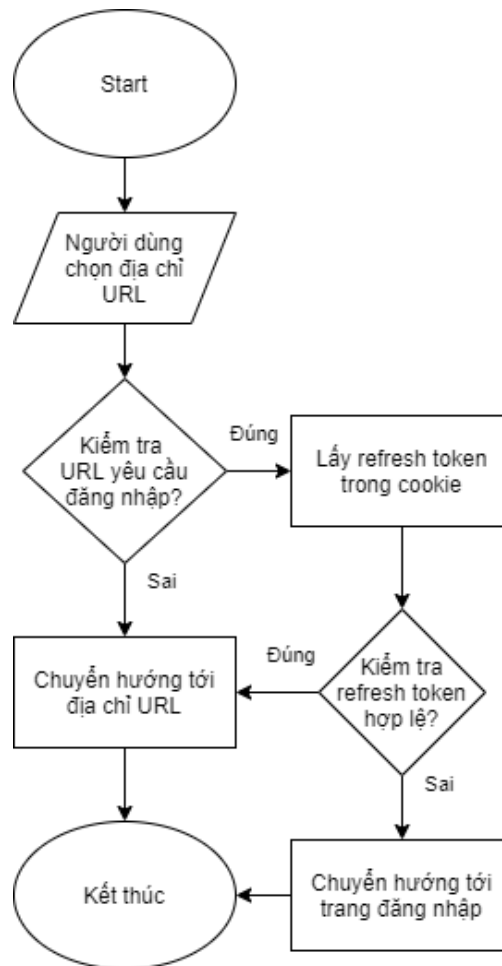


Hình 2.3.8. Lưu đồ xử lý thực hiện yêu cầu đăng nhập của ứng dụng web

Đầu tiên, ứng dụng sẽ kiểm tra tính hợp lệ của dữ liệu nhập vào bao gồm email và mật khẩu. Tiếp theo, sẽ sử dụng axios để gửi yêu cầu tạo mới JWT token tới server. Nếu tạo token thành công thì sẽ lưu trữ token vào cookie và chuyển hướng tới trang chủ, ngược lại thì thông báo lỗi.

3.2.2. Sử dụng React-Router để routing cho ứng dụng

Ứng dụng được routing thành 2 phần là URL public và URL private: URL public là những địa chỉ URL không yêu cầu đăng nhập và URL private là những địa chỉ URL yêu cầu đăng nhập.



Hình 2.3.9. Lưu đồ xử lý routing của ứng dụng web

Khi người dùng truy cập tới một địa chỉ URL bất kì, nếu địa chỉ đó yêu cầu người dùng đăng nhập thì ứng dụng sẽ chuyển hướng người dùng tới trang đăng nhập. Đối với những URL không được hỗ trợ bởi ứng dụng thì sẽ chuyển hướng người dùng đến trang thông báo lỗi 404 – Not Found.

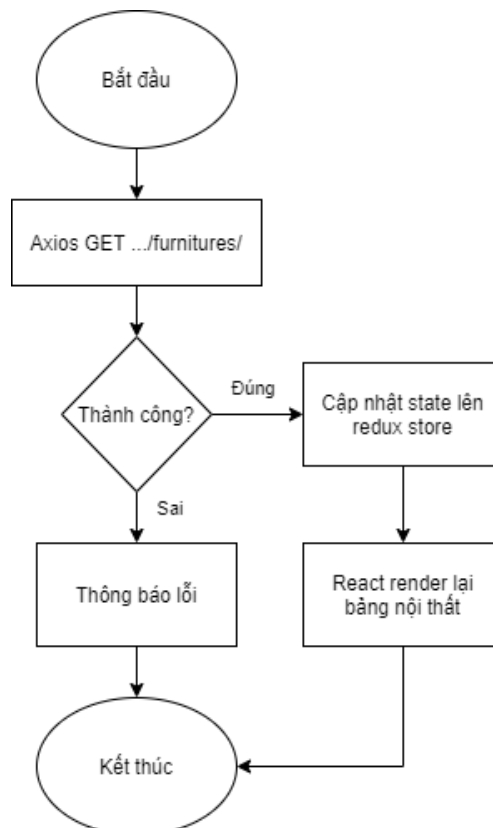
3.2.3. Thao tác sử dụng CRUD

Liệt kê danh sách nội thất

No	Unity addressable	Name	Price	Type	Status	Operation
1	fur_1	Ghế gỗ nhựa đỏ có lưng tựa	550.000 đ	Ghế	Published	  
2	fur_2	Ghế bành vải trắng	1.400.000 đ	Ghế bành	Published	  
3	fur_3	Ghế bành nhựa đỏ	2.100.000 đ	Ghế bành	Published	  
4	fur_4	Sofa phong cách hiện đại	2.800.000 đ	Sofa	Published	  

Hình 2.3.10. Component Table liệt kê danh sách nội thất

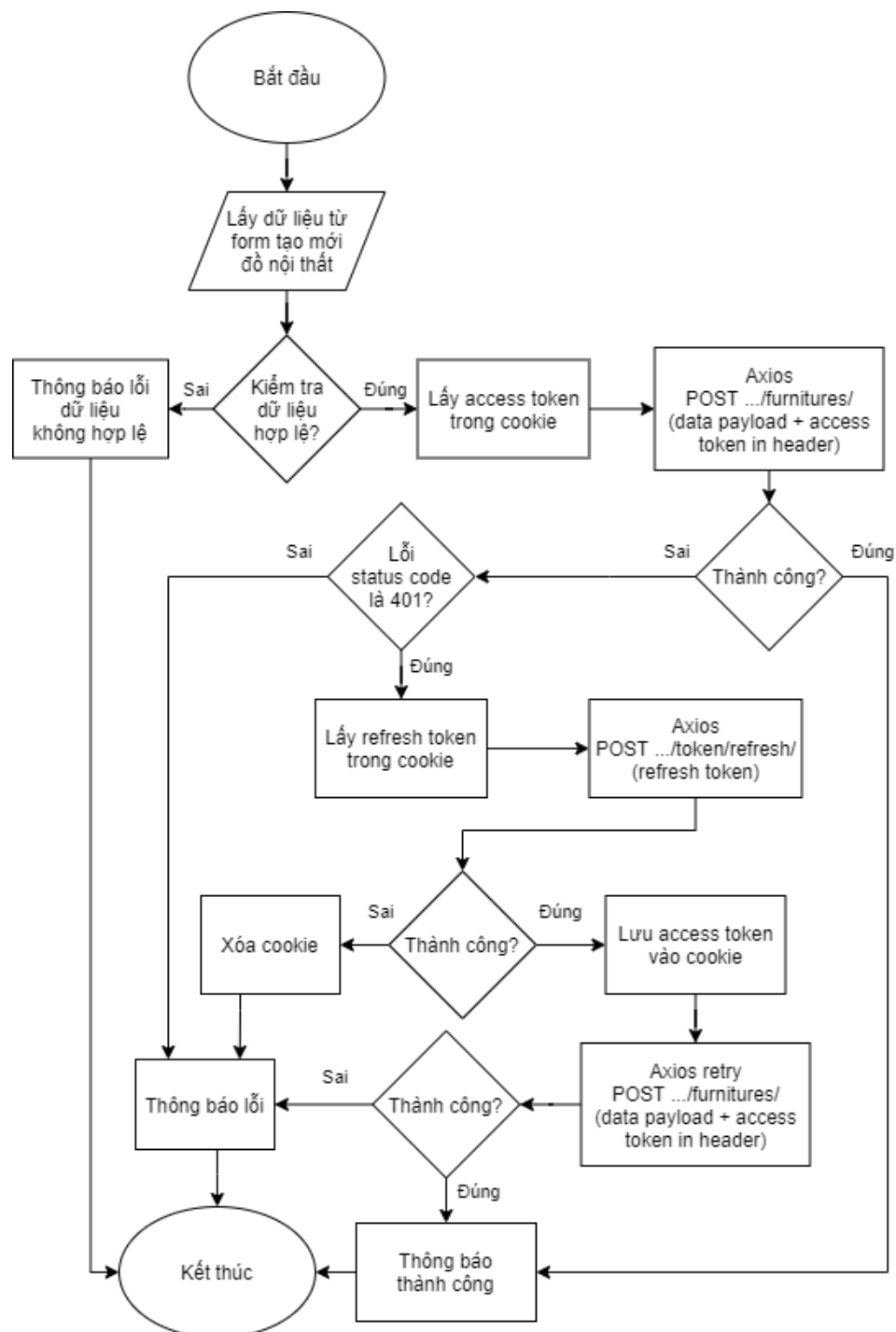
Sử dụng component Table được cung cấp bởi Ant design để hiển thị danh sách nội thất. Khi người dùng truy cập vào tính năng quản lý nội thất thì ứng dụng sẽ tự động gửi các yêu cầu cần thiết tới REST server để lấy và render dữ liệu.



Hình 2.3.11. Lưu đồ xử lý liệt kê danh sách nội thất

Tạo mới nội thất

Sử dụng một popup hiện cho phép người dùng nhập những thông tin cần thiết và submit lên server. Sau khi tạo mới thành công sẽ cập nhật lại danh sách nội thất. Quá trình kiểm tra dữ liệu và gửi request lên server để tạo mới nội thất có thể được mô tả bằng lưu đồ sau:



Hình 2.3.12. Lưu đồ xử lý tạo mới nội thất

Các tính năng tạo mới, cập nhật, xóa nội thất là những tính năng yêu cầu đăng nhập nên sẽ được cài đặt lưu đồ xử lý gần như tương tự nhau.

Các tính năng quản lý khác

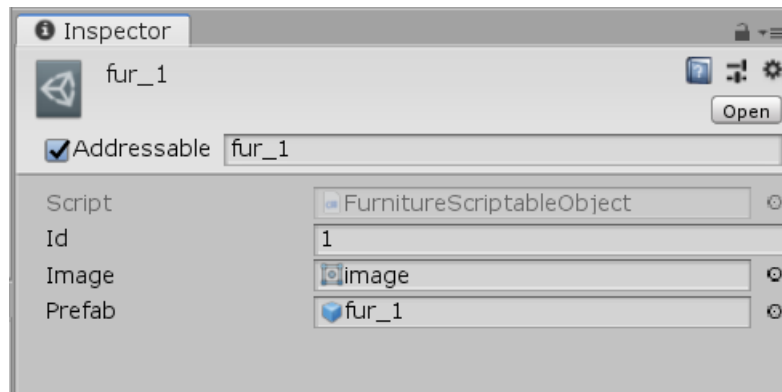
Vì điểm mạnh của React là xây dựng các component có khả năng tái sử dụng nên các tính năng quản lý khác như loại nội thất, loại phòng, màu sắc, vật liệu, phiên bản sẽ được cài đặt tương tự như quản lý nội thất. Riêng tính năng tải lên các Addressables bundles sẽ được thiết kế khác đôi chút, chỉ hỗ trợ tính năng upload. Tuy nhiên, lưu đồ xử lý gửi một request lên server cũng tương tự như tạo mới nội thất.

3.3. AR mobile app

3.3.1. Lưu trữ model và hình ảnh

ScriptableObject

Sử dụng ScriptableObject để lưu trữ dữ liệu



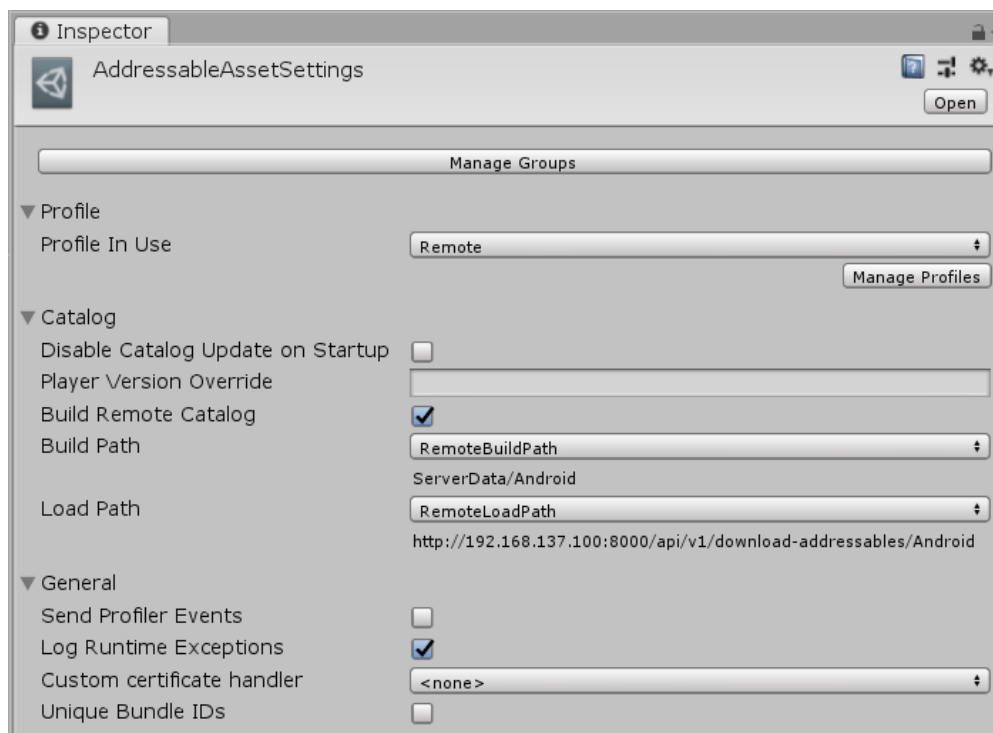
Hình 2.3.13. FurnitureScriptableObject được đánh dấu là “addressable”

Tạo một script FurnitureScriptableObject kế thừa từ ScriptableObject để lưu trữ prefab model 3D và hình ảnh model 2D. Đồng thời đánh dấu đối tượng này là một “addressable” và chỉ định một địa chỉ để cho Addressable quản lý.

3.3.2. Đóng gói dữ liệu bằng Scriptable Object

Sử dụng Addressables để đóng gói dữ liệu

Cấu hình Addressables trong Unity 3D

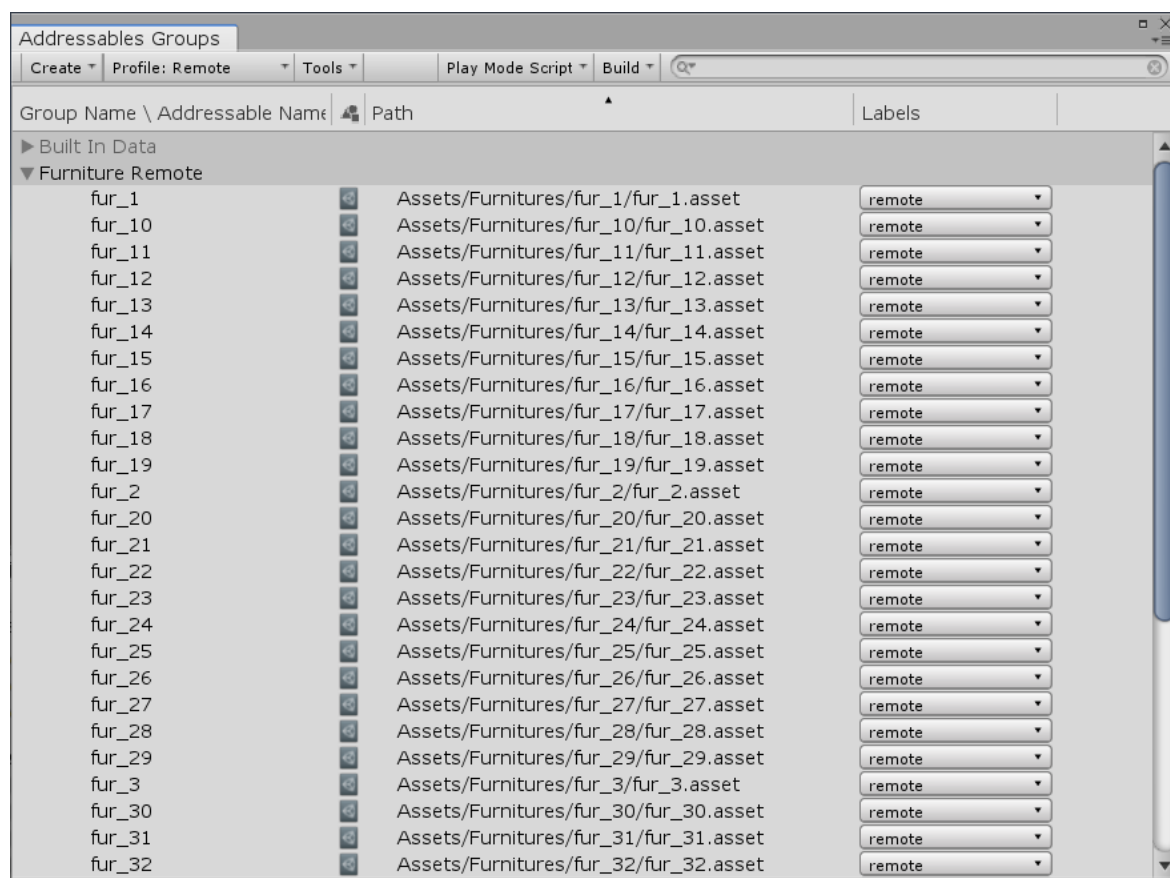


Hình 2.3.14. Cấu hình addressable để đóng gói dữ liệu

Truy cập theo đường dẫn Window > Asset Management > Addressables > Settings để cấu hình Addressables:

- Build Path: là đường dẫn dùng để build asset bundles
- Load Path: là đường dẫn dùng để truy cập và tải xuống asset bundles.

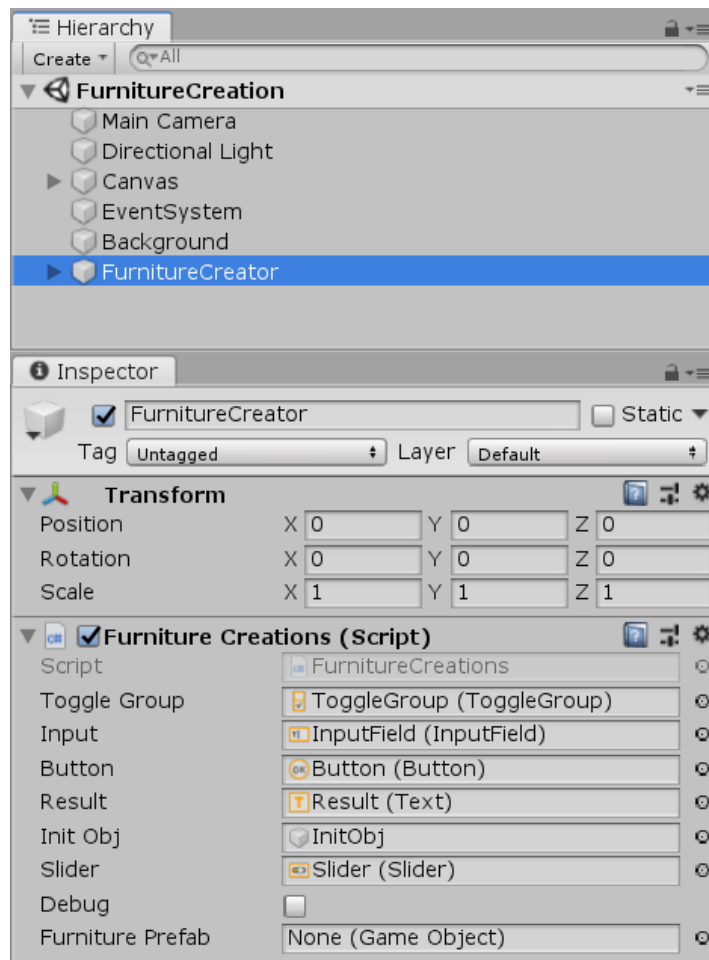
Truy cập theo đường dẫn Window > Asset Management > Addressables > Groups để quản lý các asset được đặt làm addressables:



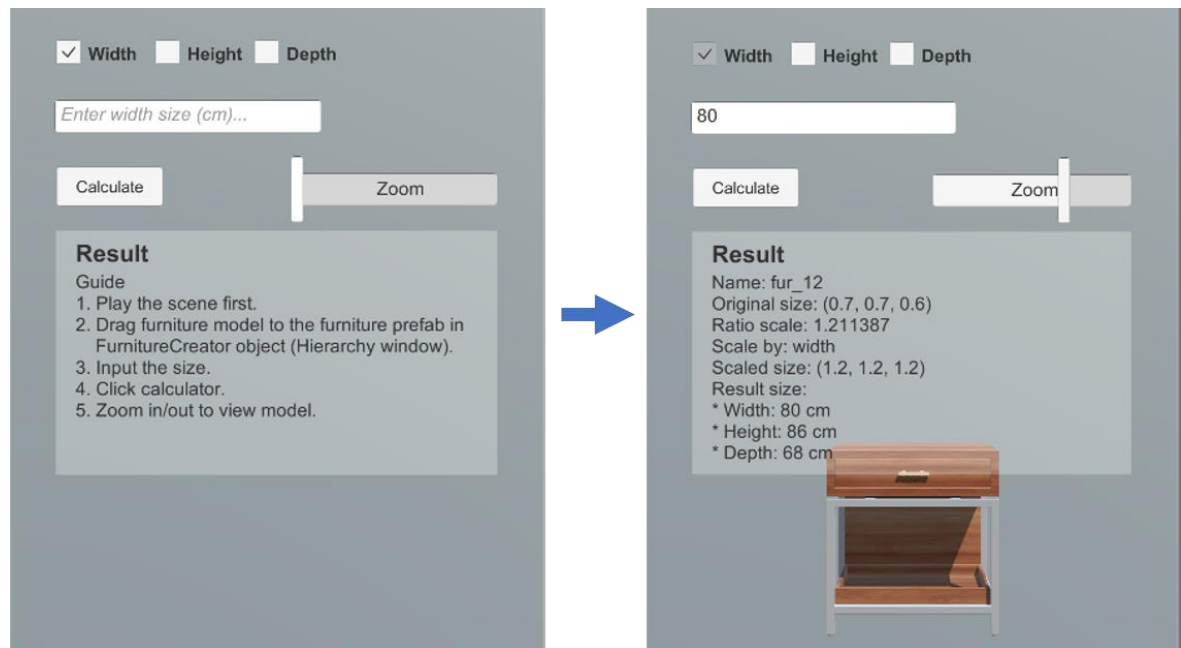
Hình 2.3.15. Danh sách các ScriptableObject được đánh dấu là “addressable”

3.3.3. Tính toán kích thước model

Sử dụng một phân cảnh chạy trực tiếp trên môi trường Unity Editor để chọn mô hình và tính toán tỉ lệ kích thước model:



Hình 2.3.16. Phân cảnh dùng để tính toán kích thước mô hình



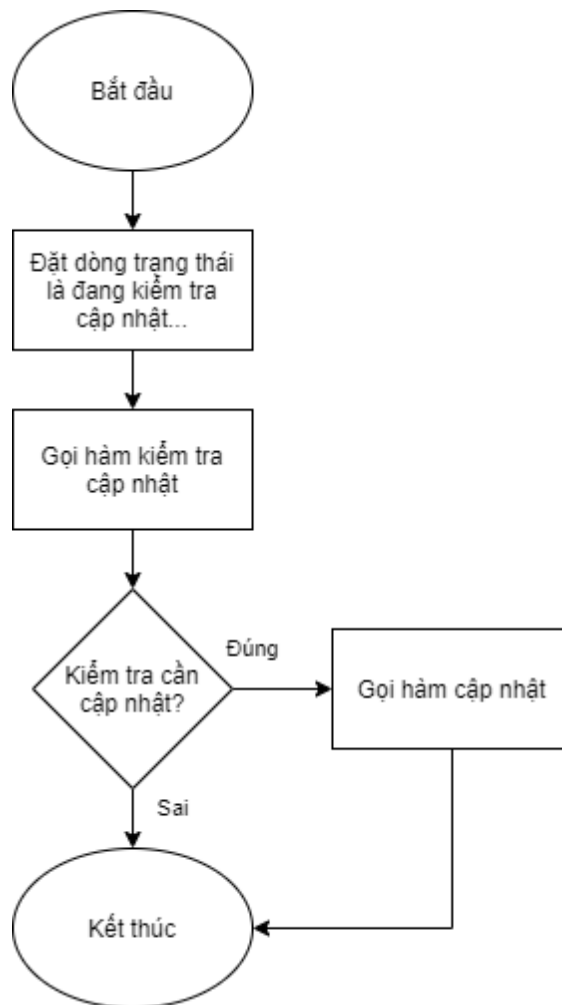
Hình 2.3.17. Tính toán kích thước mô hình bằng phân cảnh FurnitureCreation

Phân cảnh này sẽ đo kích thước lưới của mô hình nội thất được chọn, các kích thước này sẽ có chiều ngang, cao và sâu. Tương ứng với kích thước theo một chiều được chỉ định, phân cảnh sẽ tính toán và cho ra kết quả của mô hình nội thất theo chiều tỉ lệ đó.

3.3.4. Cập nhật dữ liệu

Khi khởi động ứng dụng thì phân cảnh kiểm tra cập nhật sẽ được khởi chạy đầu tiên. Phân cảnh này chịu trách nhiệm kiểm tra cập nhật, tải về và hỗ trợ chuyển sang phân cảnh chức năng AR.

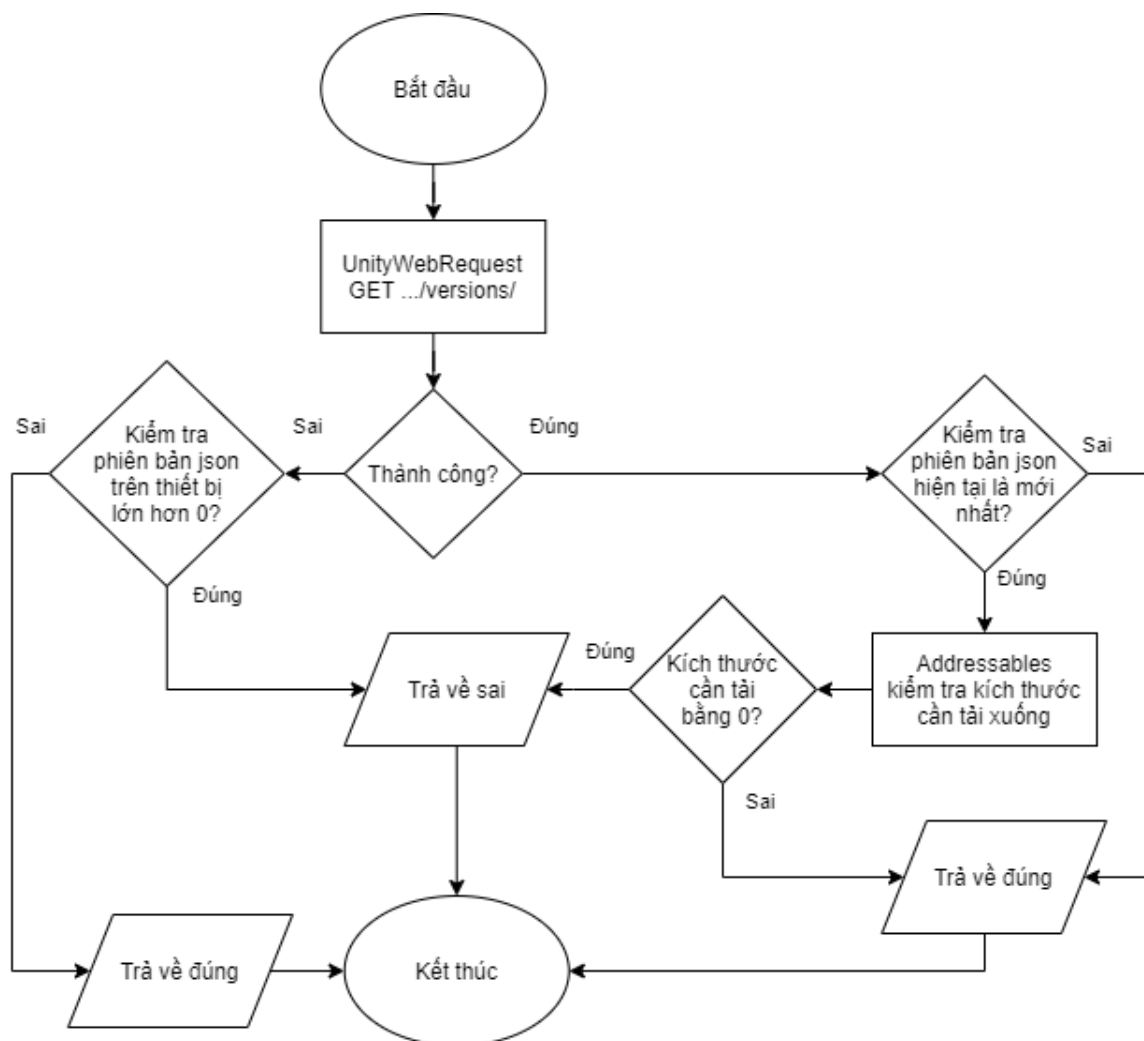
Hàm quản lý cập nhật được mô tả bằng lưu đồ sau:



Hình 2.3.18. Lưu đồ của hàm quản lý cập nhật dữ liệu của ứng dụng AR mobile

Hàm này sẽ xác định có tiến hành cập nhật dữ liệu hay không thông qua việc gọi hàm kiểm tra cập nhật. Nếu hàm kiểm tra cập nhật trả về kết quả là cần cập nhật thì sẽ gọi hàm tiến hành cập nhật và kết thúc.

Hàm kiểm tra cập nhật

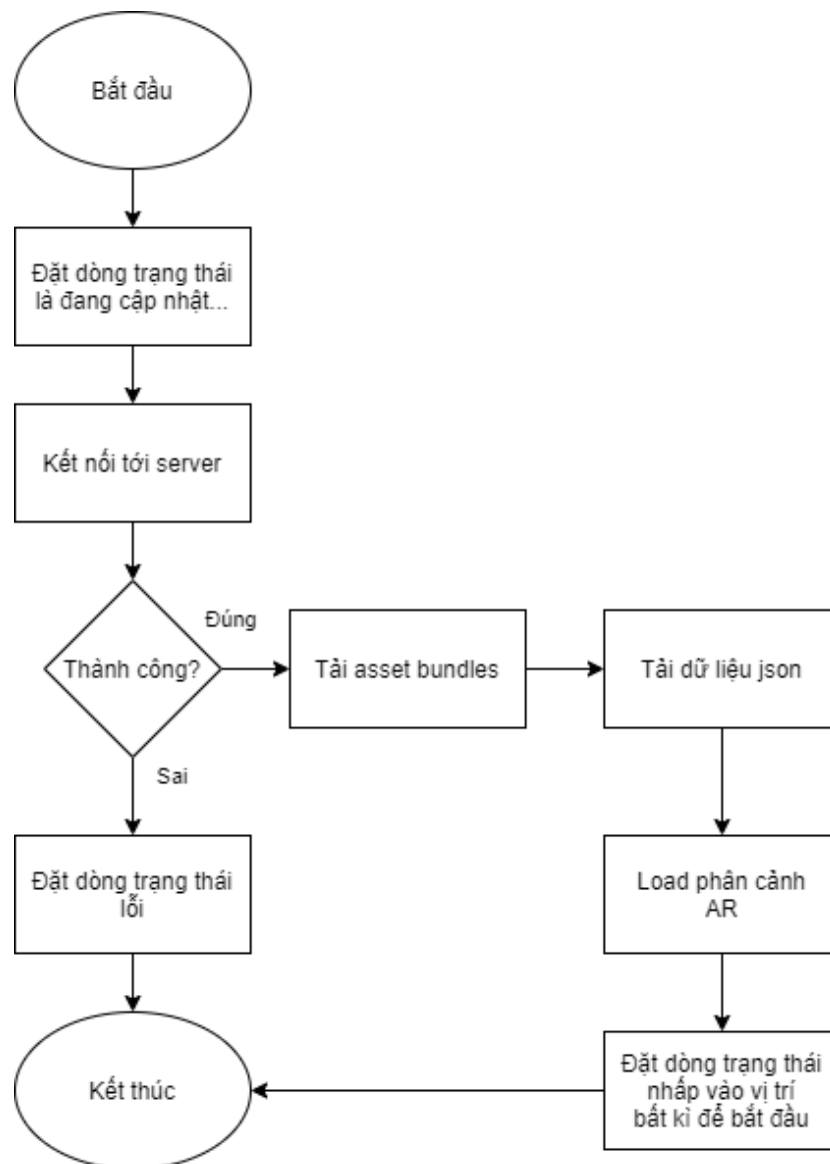


Hình 2.3.19. Lưu đồ xử lý của hàm kiểm tra cập nhật

Hàm kiểm tra cập nhật sẽ xác định có cần cập nhật hay không thông qua việc kiểm tra phiên bản dữ liệu json và kích thước cần tải xuống của addressables. Hàm này sẽ trả về đúng nếu thỏa một trong 3 điều kiện sau

- Ứng dụng lần đầu được cài đặt, hoặc đã bị xóa dữ liệu (clear data)
- Phiên bản dữ liệu json bị lỗi thời
- Các dữ liệu addressables bị thay đổi ở trên server

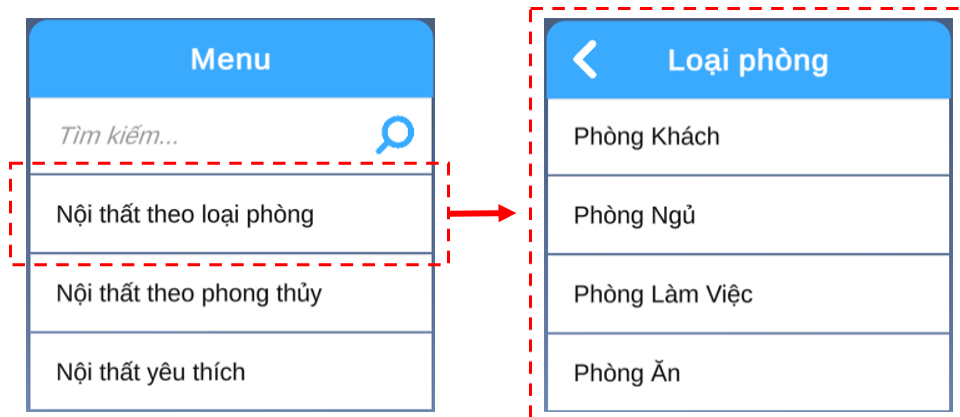
Hàm tải xuống dữ liệu cập nhật



Hình 2.3.20. Lưu đồ xử lý của hàm tải xuống dữ liệu cập nhật

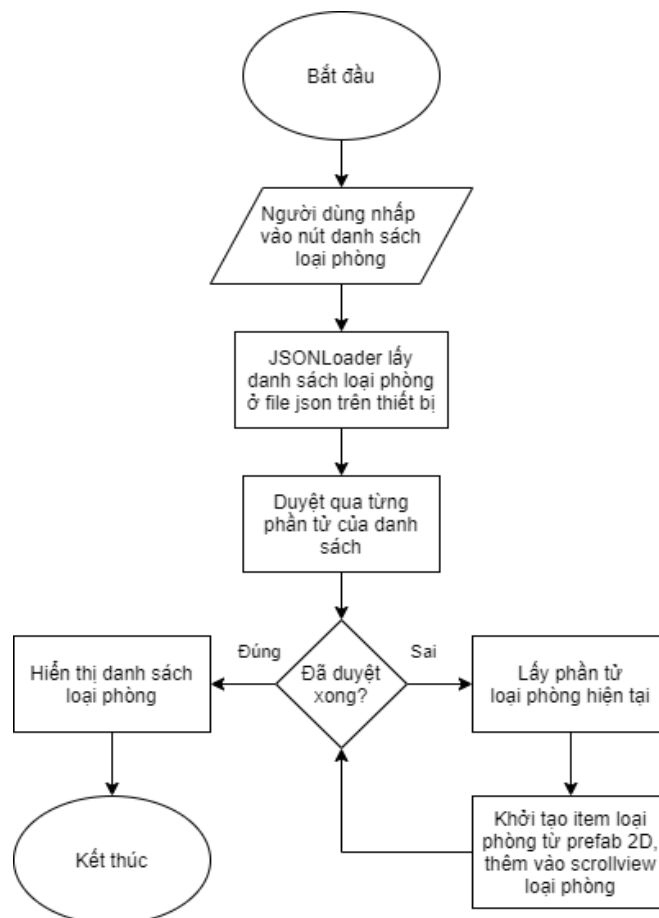
Hàm sẽ tải xuống lần lượt dữ liệu asset bundles của addressables và các dữ liệu JSON (furnitures, furniture-types, room-types, colors, materials, version) được cung cấp bởi REST API.

3.3.5. Liệt kê danh sách loại phòng



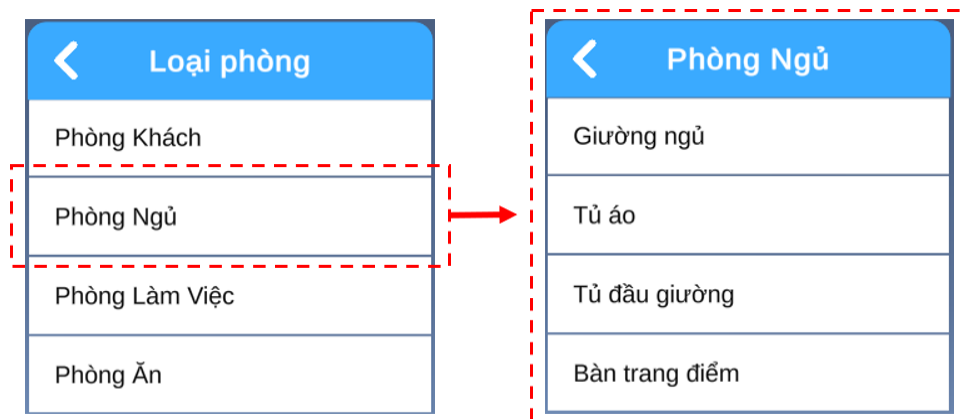
Hình 2.3.21. Menu liệt kê danh sách loại phòng

Tính năng này là một menu cấp độ 1 liệt kê tất cả các loại phòng dưới dạng một danh sách cho phép người dùng chọn và liệt kê theo loại phòng. Quá trình liệt kê được mô tả đơn giản hóa bằng lưu đồ sau:



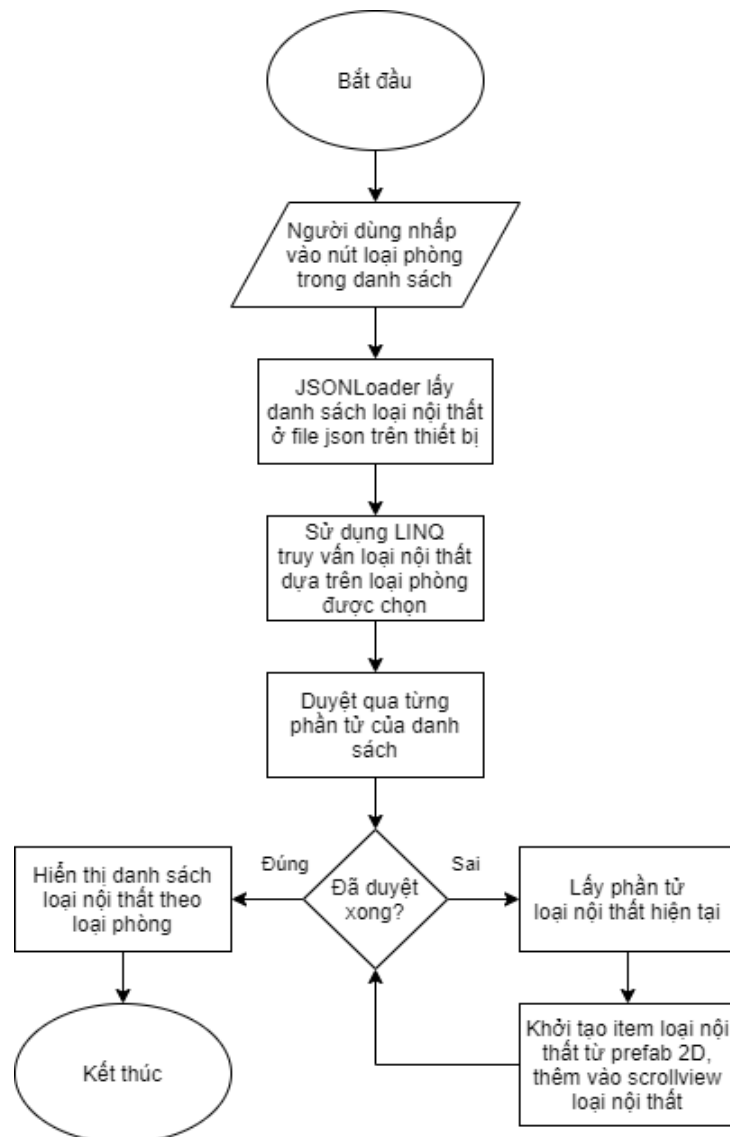
Hình 2.3.22. Lưu đồ xử lý liệt kê danh sách loại phòng

3.3.6. Liệt kê danh sách loại nội thất dựa theo loại phòng



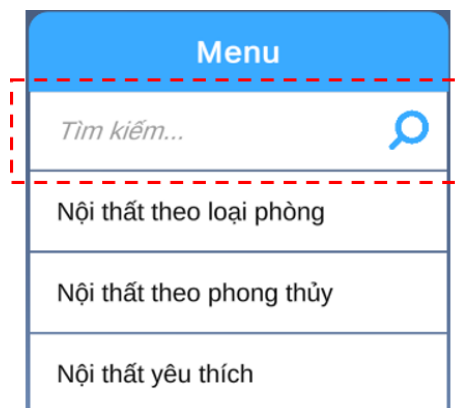
Hình 2.3.23. Menu liệt kê danh sách loại nội thất theo loại phòng

Tính năng này là một menu cấp độ 2 cho phép liệt kê danh sách loại nội thất dựa theo loại phòng. Quá trình liệt kê danh sách loại nội thất được mô tả bằng lưu đồ ở trang tiếp theo.



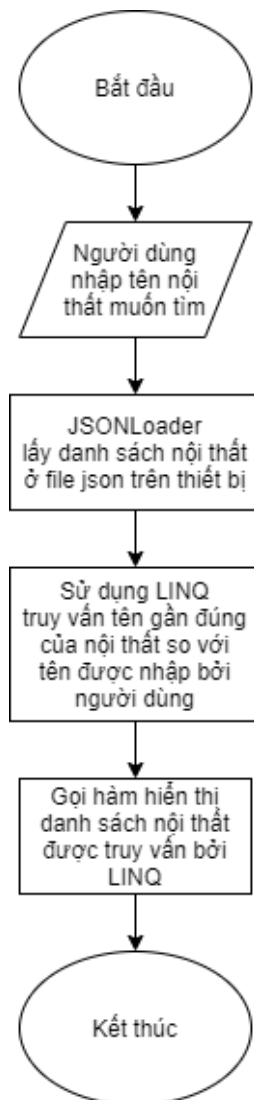
Hình 2.3.24. Lưu đồ xử lý liệt kê loại nội thất theo loại phòng

3.3.7. Tìm kiếm nội thất



Hình 2.3.25. Thanh tìm kiếm trên menu

Tính năng cho phép người dùng tìm kiếm theo tên gần đúng của nội thất, được mô tả bằng lưu đồ sau:



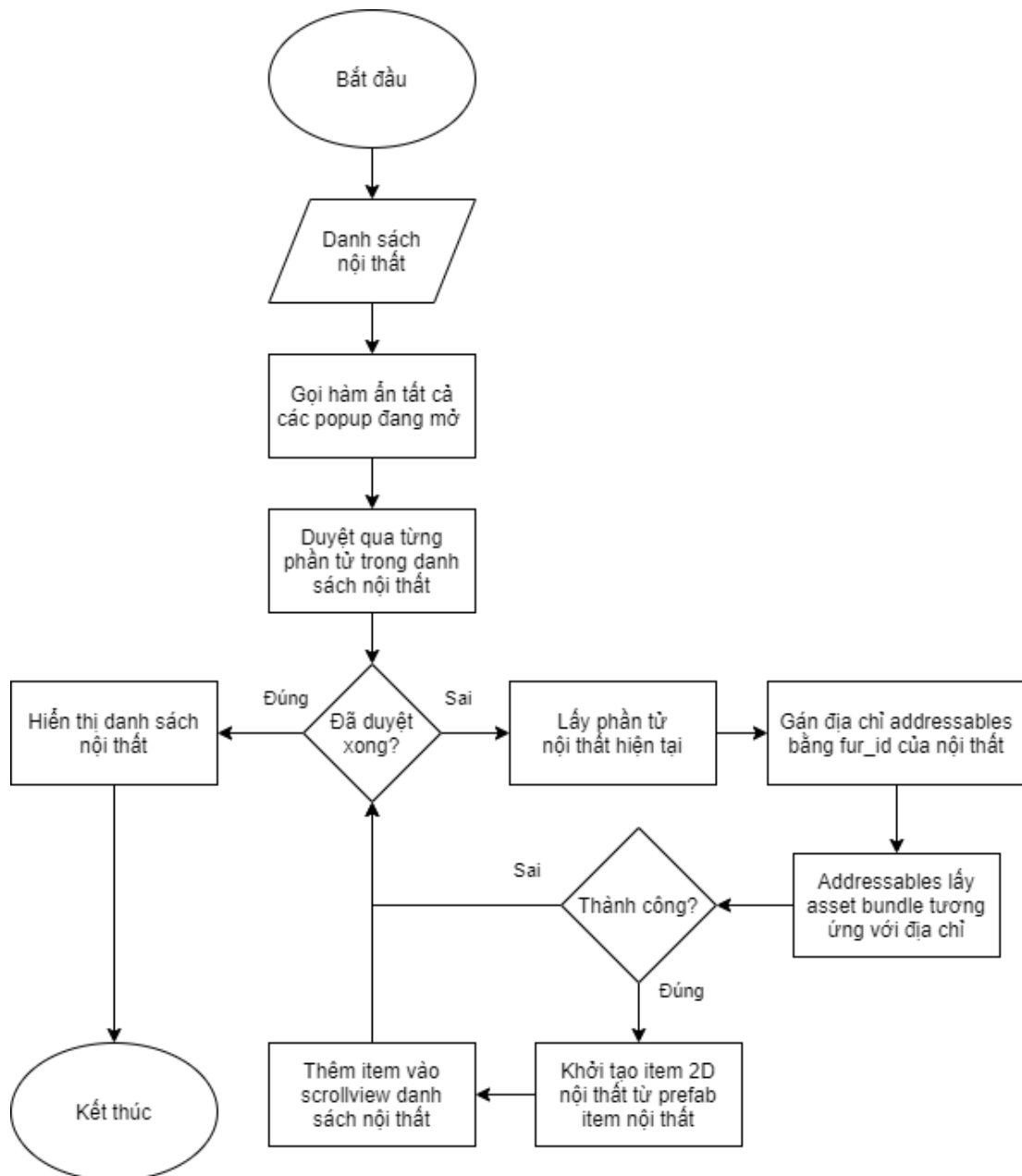
Hình 2.3.26. Lưu đồ xử lý tìm kiếm gần đúng theo tên nội thất

3.3.8. Liệt kê danh sách nội thất



Hình 2.3.27. Danh sách nội thất trong phân cảnh AR

Tính năng này thực chất là một tính năng bị động, nó chỉ được gọi bởi những tính năng khác bao gồm tìm kiếm nội thất, liệt kê nội thất theo loại nội thất, gợi ý nội thất theo phong thủy và nội thất yêu thích. Đầu vào là một danh sách đồ nội thất, sau đó sẽ sử dụng thư viện Addressables để lấy FurnitureScriptableObject dựa trên địa chỉ “addressable” tương ứng. Quá trình lấy và hiển thị danh sách nội thất được thể hiện bằng lưu đồ ở trang tiếp theo.



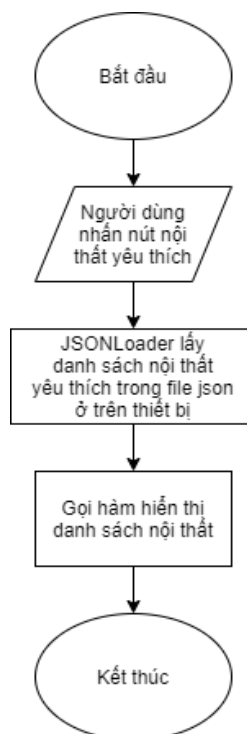
Hình 2.3.28. Lưu đồ xử lý hiển thị danh sách nội thất

3.3.9. Liệt kê nội thất yêu thích



Hình 2.3.29. Biểu tượng nội thất yêu thích trong popup thông tin chi tiết

Tính năng này dùng để liệt kê những nội thất được đánh dấu yêu thích (biểu tượng trái tim màu đỏ). Tính năng này được cài đặt khá đơn giản bằng lưu đồ sau:



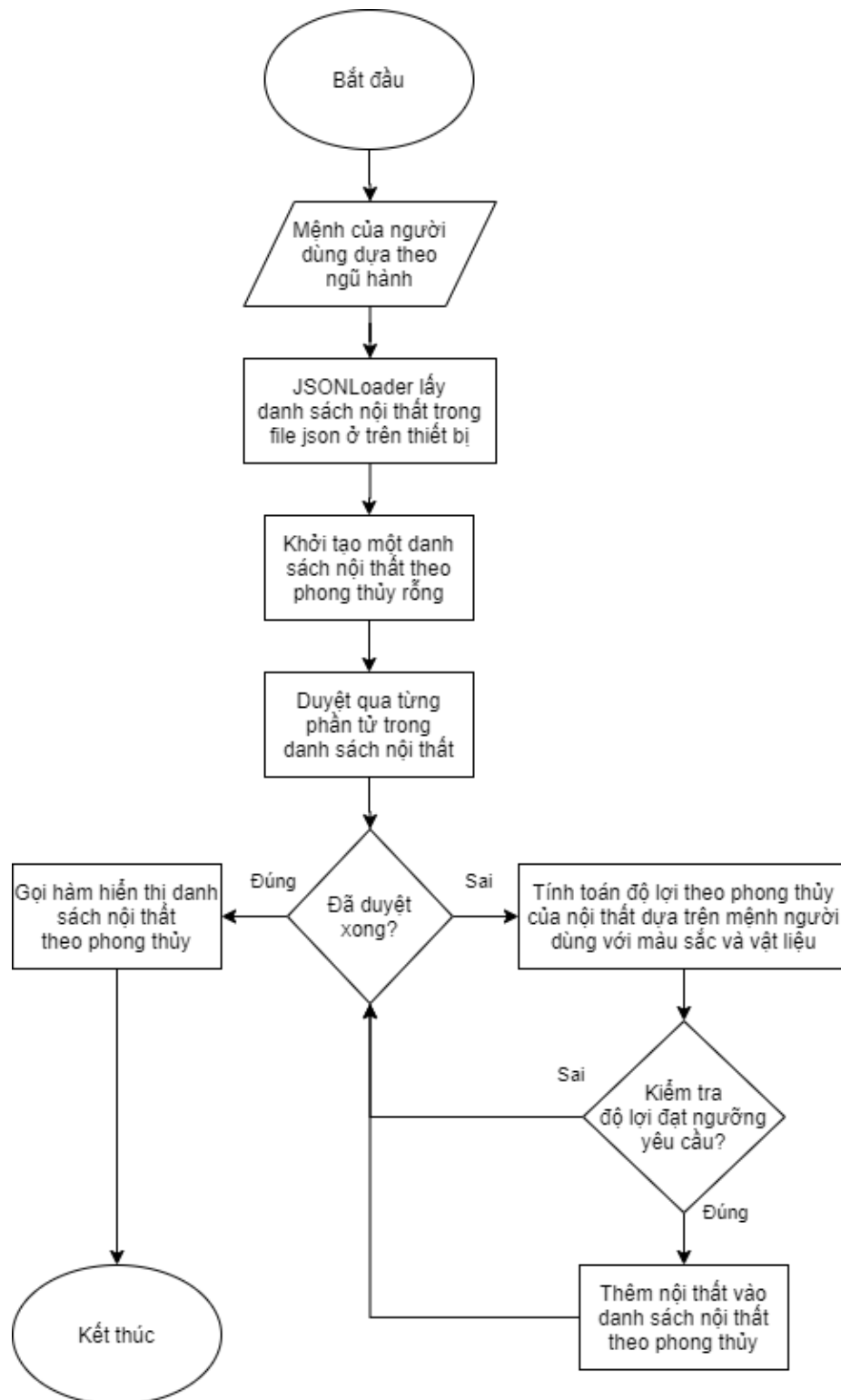
Hình 2.3.30. Lưu đồ xử lý liệt kê danh sách nội thất yêu thích

3.3.10. Gợi ý nội thất theo phong thủy



Hình 2.3.31. Thông tin phong thủy của nội thất

Mỗi món đồ nội thất được cấu tạo nên từ một vài vật liệu và màu sắc. Mỗi màu sắc và vật liệu sẽ có liên hệ tới ngũ hành, hợp với số mệnh nào theo ngũ hành. Tiêu chí đánh giá độ lợi phong thủy dựa trên ngũ hành của một món đồ nội thất sẽ dựa vào trung bình cộng độ lợi phong thủy theo mệnh tương ứng với mỗi màu sắc và vật liệu. Cách thức tính độ lợi sẽ được mô tả chi tiết ở phần sau (hàm tính độ lợi phong thủy của nội thất dựa trên màu sắc và vật liệu). Việc gợi ý nội thất dựa trên phong thủy được mô tả bằng lưu đồ tổng quát sau:



Hình 2.3.32. Lưu đồ xử lý gợi ý nội thất dựa trên phong thủy

Mệnh của người dùng dựa trên ngũ hành được tính toán từ ngày sinh mà người dùng cung cấp thông qua popup sau:

Thông tin cá nhân

Tên

Nguyễn Thanh Khoa

Ngày sinh

1997 / Tháng 8 / 29

☒ Âm lịch

Dương lịch: 30/09/1997.

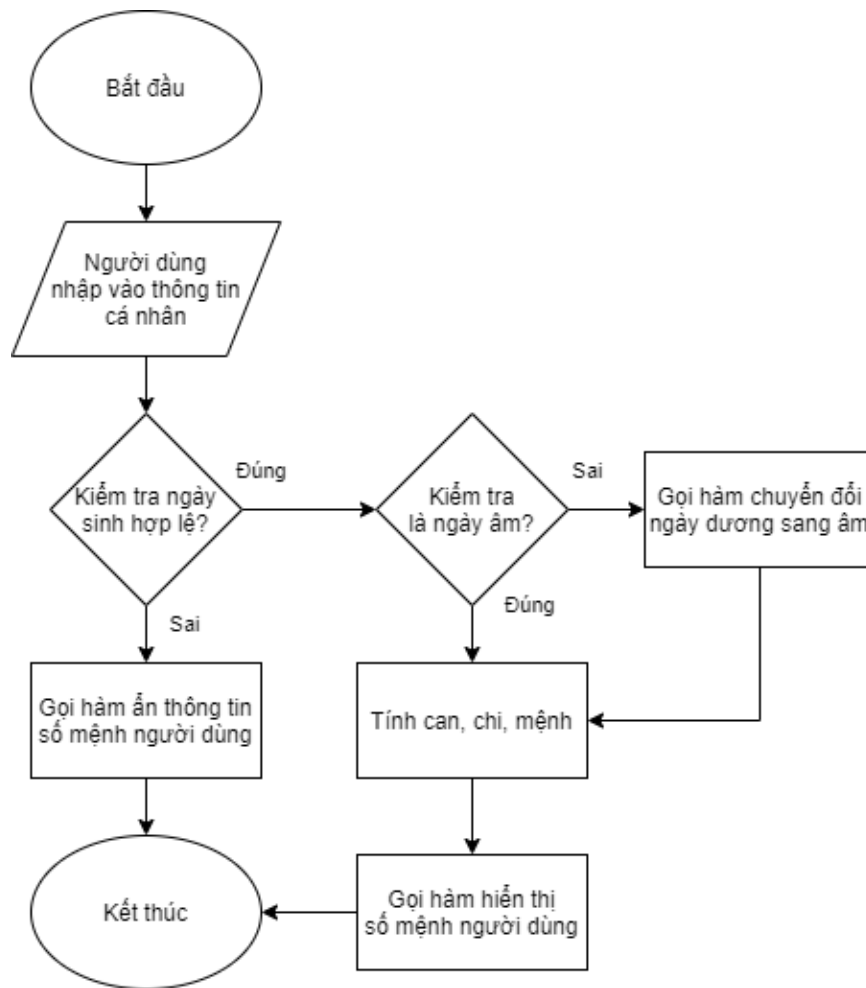
Can: Đinh, Chi: Sửu, Mệnh: Thủy.

OK Hủy

Hình 2.3.33. Popup cho phép người dùng nhập thông tin cá nhân

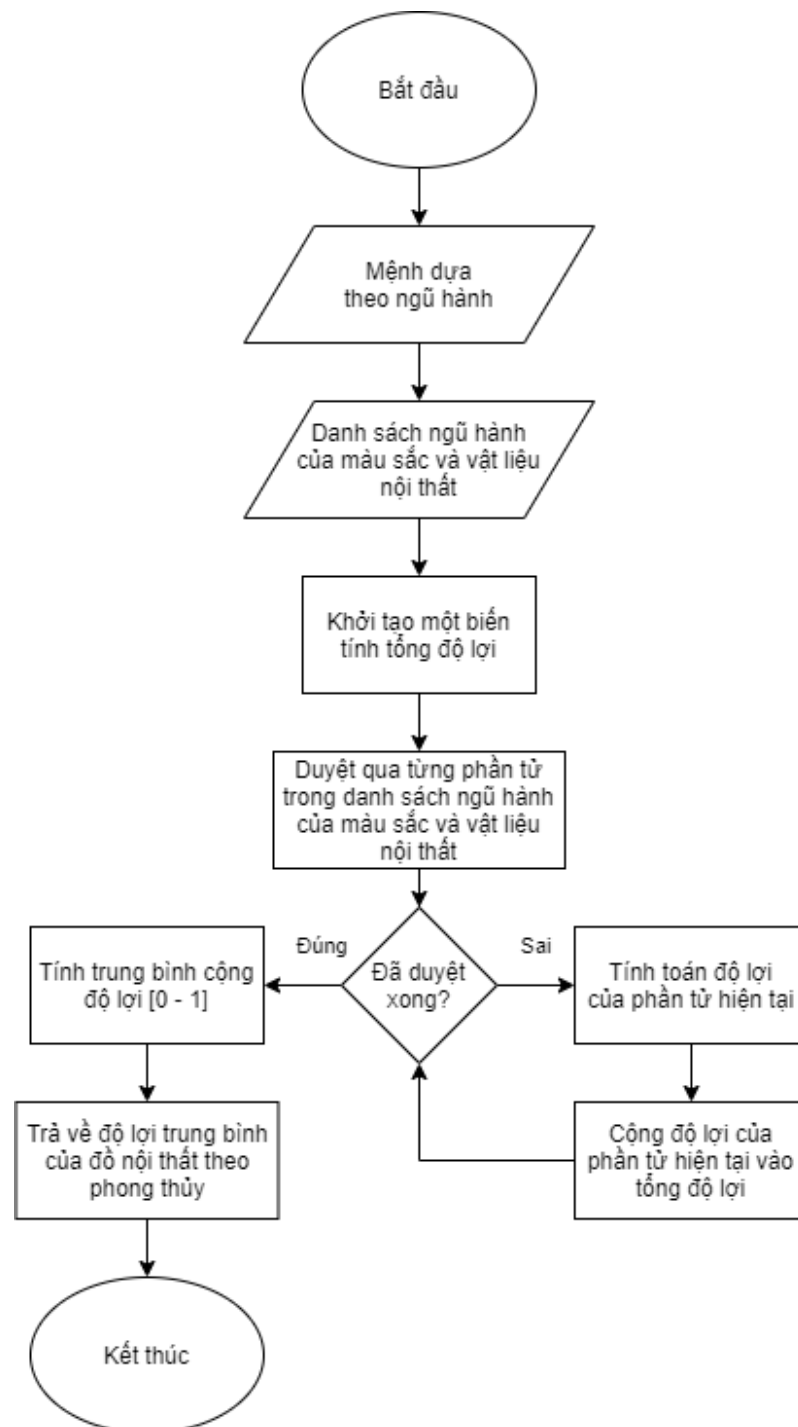
Hàm tính mệnh dựa trên ngày sinh của người dùng cung cấp

Hàm này sử dụng một thư viện chuyển đổi ngày âm sang dương được tạo bởi tác giả Hồ Ngọc Đức ^[7] và các công thức tính can, chi, mệnh dựa trên năm âm lịch ^[8]. Tuy nhiên các công thức khá phức tạp nên không trình bày trong đề tài này. Lưu đồ xử lý của hàm tính can – chi – mệnh được trình bày như sau:



Hình 2.3.34. Lưu đồ xử lý của hàm tính can – chi – mệnh

Hàm tính độ lợi phong thủy của nội thất dựa trên màu sắc và vật liệu



Hình 2.3.35. Lưu đồ xử lý của hàm tính độ lợi phong thủy đồ nội thất

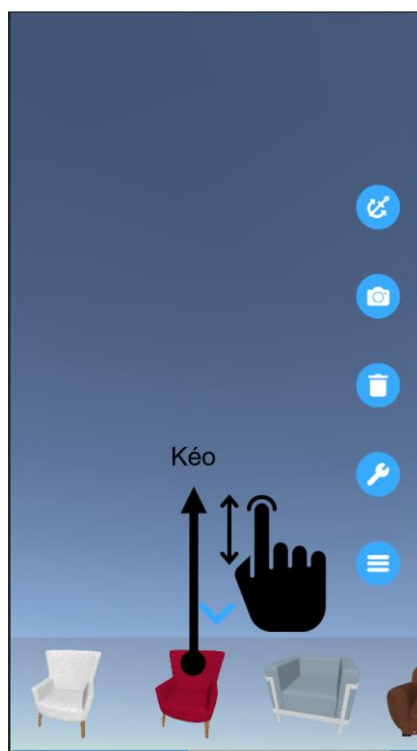
Hàm tính độ lợi phong thủy của nội thất dựa trên màu sắc và vật liệu về cơ bản là tính trung bình cộng dựa trên mối quan hệ tương sinh – tương khắc của màu sắc và vật liệu so với một mệnh trong ngũ hành. Tiêu chí giá độ lợi được trình bày trong bảng sau:

Tiêu chí	Giá trị
Trùng	1
Được sinh	1
Sinh	0.25
Khắc	0.25
Bị khắc	0

Hình 2.3.36. Bảng tiêu chí độ lợi dựa trên ngũ hành tương sinh – tương khắc

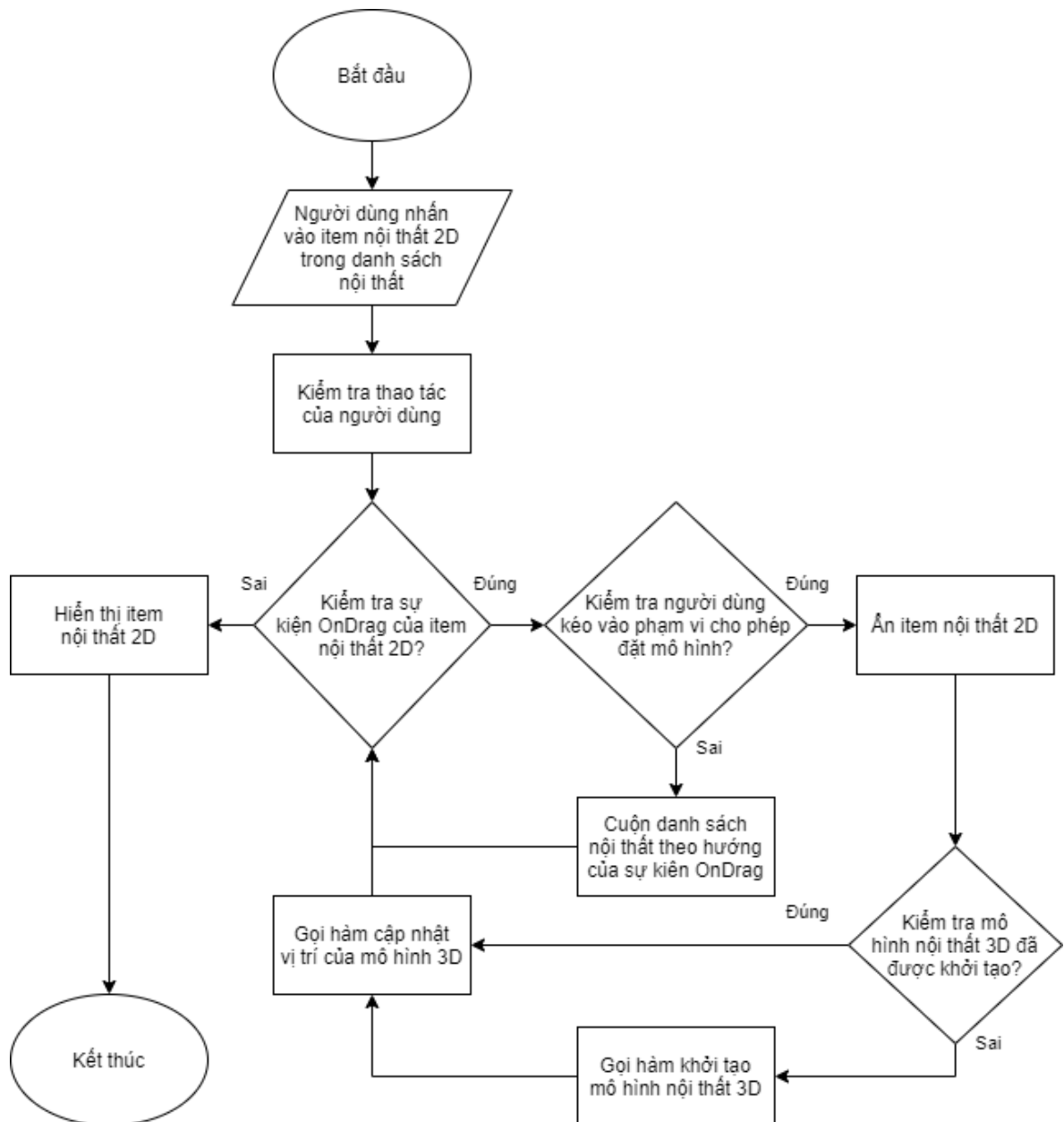
Độ lợi 0.5 là cân bằng, khi quy đổi sang thang phần trăm thì 1 tương đương với hợp 100%, 0 là khắc 100%.

3.3.11. Kéo thả nội thất



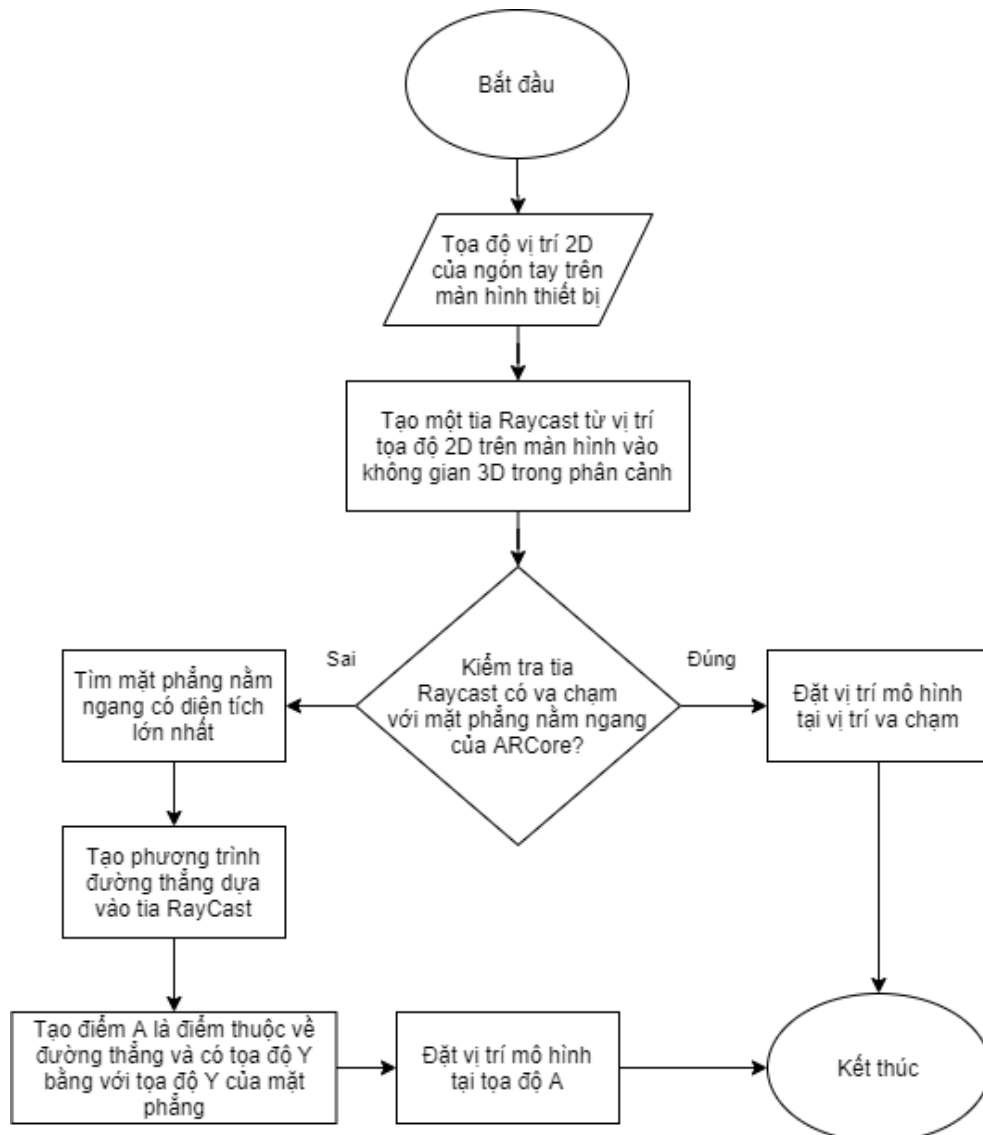
Hình 2.3.37. Thao tác kéo thả nội thất từ danh sách nội thất

Khi người dùng nhấn chọn vào ảnh 2D của mô hình nội thất, giữ và kéo lên phía trên đến một khoảng nhất định thì mô hình 3D của nội thất đó sẽ được khởi tạo. Mô tả đơn giản hóa quá trình kéo thả bằng lưu đồ sau:



Hình 2.3.38. Lưu đồ xử lý thao tác kéo thả nội thất

Hàm cập nhật vị trí khi kéo thả nội thất



Hình 2.3.39. Lưu đồ xử lý của hàm cập nhật vị trí mô hình khi kéo thả nội thất

3.3.12. Hệ thống điều chỉnh mô hình nội thất

Các tính năng chọn, di chuyển và xoay mô hình nội thất được hỗ trợ bởi thư viện ARCore. Tuy nhiên, tính năng chọn và di chuyển còn chưa phù hợp nên cần phải tinh chỉnh lại.

Tính năng chọn mô hình nội thất đang nằm trong phân cảnh: ARCore sẽ dựa vào vị trí ngón tay đặt vào màn hình để tạo một tia Raycast dùng để chọn mô hình. Tuy nhiên, nếu có thành phần UI 2D phía trước thì tính năng này vẫn hoạt động. Tinh

chỉnh lại một chút tại lúc chọn mô hình, nếu có thành phần 2D nào phía trước thì không cho phép chọn.

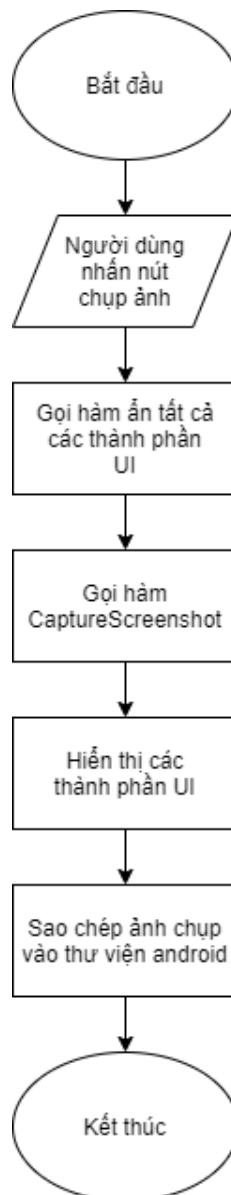
Tính năng di chuyển nội thất trong phân cảnh AR: ARCore cho phép di chuyển các mô hình nội thất xung quanh mặt phẳng được xác định bởi ARCore, không thể di chuyển ra ngoài các mặt phẳng. Để di chuyển mô hình nội thất vào nơi chưa được ARCore phát hiện ra mặt phẳng thì sẽ tinh chỉnh lại tương tự như hàm cập nhật vị trí nội thất khi kéo thả.

3.3.13. Chụp ảnh màn hình



Hình 2.3.40. Ảnh chụp màn hình của 3 mô hình nội thất 2 ghế – 1 mặt bàn

Tính năng này được cài đặt bằng một hàm đơn giản, khi người dùng nhấn vào nút chụp ảnh thì hàm này sẽ được gọi. Nguyên lý hoạt động của hàm này được mô tả ở trang tiếp theo.



Hình 2.3.41. Lưu đồ xử lý chụp ảnh màn hình

3.3.14. Xóa nội thất

Khi người dùng kích hoạt tính năng xóa một mô hình nội thất trong phân cảnh thì sử dụng Addressables để giải phóng mô hình nội thất do nó load lên và dùng hàm Destroy để xóa bỏ anchor dùng để cố định mô hình nội thất. Tương tự với tính năng xóa tất cả mô hình nội thất trong phân cảnh, tương ứng với mỗi mô hình trong phân cảnh sẽ được thực hiện giống như cách xóa một mô hình.

3.3.15. Làm mới phân cảnh

Khi người dùng kích hoạt tính năng làm mới phân cảnh, ứng dụng sẽ tiến hành tải lại phân cảnh AR.

3.3.16. Một số tính năng phụ được hỗ trợ

Hiển thị ToastMessage: sử dụng API native trên android để gọi hàm hiển thị Toast message

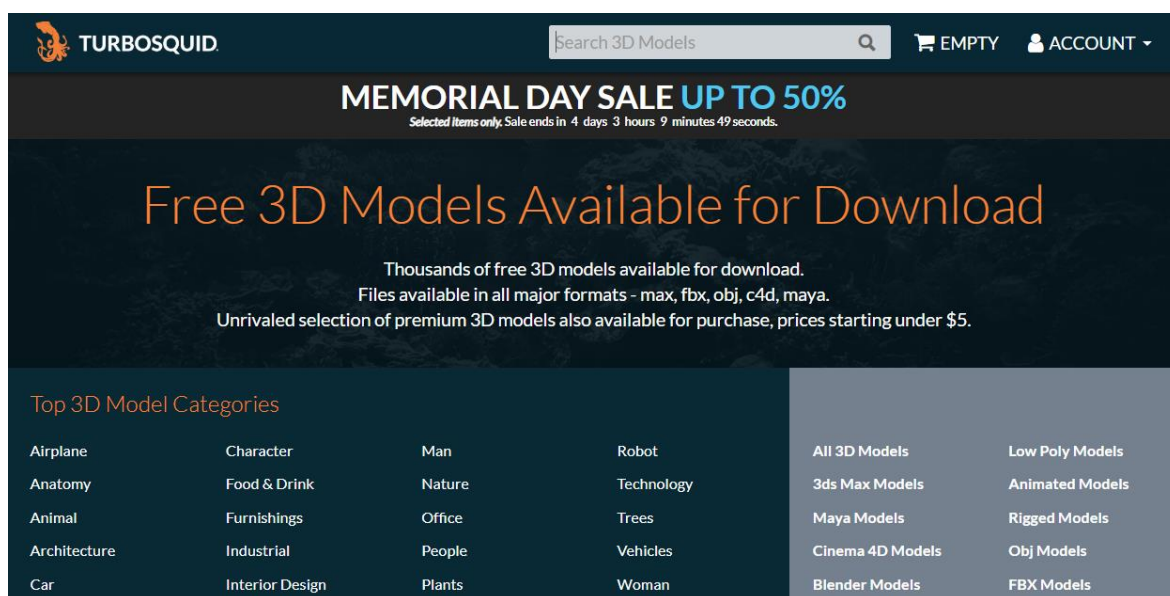
Thoát: nhấn 2 lần thoát để đóng ứng dụng

Hiển thị mặt phẳng ARCore: tùy chọn ẩn hoặc hiển thị mặt phẳng được phát hiện bởi ARCore

Hiển thị điểm đặc trưng ARCore: tùy chọn ẩn hoặc hiển thị những điểm đặc trưng được phát hiện bởi ARCore

Hướng dẫn quét mặt phẳng: hướng dẫn quét điện thoại xung quanh căn phòng để ARCore phát hiện mặt phẳng. Tính năng này được hỗ trợ bởi thư viện ARCore.

3.4. Mô hình nội thất 3D



Hình 2.3.42. Trang chủ của TURBOSQUID

TURBOSQUID là trang web cung cấp những mô hình nội thất miễn phí và có tính phí (giá bắt đầu từ dưới \$5), hỗ trợ nhiều định dạng như max, fbx, obj, c4d và maya. Những chủ đề của mô hình được cung cấp bởi TURBOSQUID rất nhiều, từ con

người, máy bay, xe hơi, kiến trúc, động vật cho tới nội thất và còn nhiều chủ đề khác nữa.

Các mô hình nội thất 3D được tải miễn phí từ trang web TURBOSQID. Sau đó dùng phần mềm Substance Painter để chỉnh sửa màu sắc, kết cấu và đưa sang phần mềm Maya để chỉnh sửa pivot và texturing các mô hình. Cuối cùng là xuất ra định dạng fbx để nhập vào phần mềm Unity 3D.

Hiện tại ứng dụng đang sử dụng **40 mô hình** nội thất được cung cấp miễn phí bởi TURBOSQID.

Chương 4. Đánh giá kiểm thử

4.1. Mục tiêu kiểm thử

Mục tiêu kiểm thử trọng tâm chính là ứng dụng AR bán đồ nội thất, kiểm tra xem ứng dụng có hoạt động tốt với các tính năng được lập trình hay không, có phát sinh lỗi ngoài ý muốn hay không nhằm mục đích đưa ra hướng khắc phục hoặc cách khắc phục trong tương lai. Bên cạnh đó, mục tiêu thứ cấp kiểm thử chính là REST API server và ứng dụng web quản lý, để xác định khả năng đáp ứng yêu cầu đưa ra hay không và xác định những lỗi phát sinh cần được khắc phục.

4.2. Kịch bản kiểm thử

Kịch bản 1: Ứng dụng web quản lý – REST API server

Tiến hành kiểm thử mô hình ứng dụng web quản lý với REST API server:

- Đăng nhập sai email hoặc mật khẩu
- Đăng nhập đúng email, mật khẩu, không ghi nhớ đăng nhập và đăng xuất
- Đăng nhập đúng email, mật khẩu và ghi nhớ đăng nhập
- Truy cập trang chủ
- Truy cập chức năng quản lý phiên bản json. Sau đó tạo mới phiên bản
- Truy cập chức năng upload Unity addressables. Sau đó tải lên các asset bundles
- Truy cập chức năng quản lý nội thất. Sau đó tạo, xem chi tiết, cập nhật, xóa một nội thất
- Truy cập chức năng quản lý loại nội thất. Sau đó tạo, xem chi tiết, cập nhật, xóa một loại nội thất
- Truy cập chức năng quản lý màu sắc nội thất. Sau đó tạo, xem chi tiết, cập nhật, xóa một màu sắc nội thất
- Truy cập chức năng quản lý vật liệu nội thất. Sau đó tạo, xem chi tiết, cập nhật, xóa một vật liệu nội thất
- Truy cập chức năng quản lý loại phòng. Sau đó tạo, cập nhật và xóa một loại phòng
- Truy cập URL không được cung cấp

- Đăng xuất

Kịch bản 2: Ứng dụng AR bán đồ nội thất – REST API server

Tiến hành kiểm thử mô hình ứng dụng AR bán đồ nội thất với REST API server:

- Cài ứng dụng lần đầu và tắt kết nối mạng tới server
- Cài ứng dụng lần đầu và có kết nối mạng tới server
- Xóa dữ liệu, mở ứng dụng và tắt kết nối mạng tới server
- Xóa dữ liệu, mở ứng dụng và có kết nối mạng tới server

Kịch bản 3: Ứng dụng AR bán đồ nội thất – Phân cảnh AR

- Di chuyển thiết bị phát hiện mặt phẳng
- Liệt kê danh sách loại nội thất theo loại phòng
- Liệt kê danh sách nội thất theo loại nội thất
- Tìm kiếm theo tên nội thất
- Liệt kê danh sách nội thất dựa trên phong thủy
- Xem chi tiết nội thất
- Chọn ẩn mặt phẳng và hiển thị mặt phẳng
- Chọn ẩn điểm đặc trưng và hiển thị điểm đặc trưng
- Chọn và kéo thả nội thất đặt vào phân cảnh
- Chọn và di chuyển nội thất trong phân cảnh
- Chọn và xoay nội thất trong phân cảnh
- Chụp ảnh màn hình
- Chọn và xóa nội thất trong phân cảnh
- Chọn làm mới phân cảnh

Môi trường kiểm thử

REST API server và ứng dụng web quản lý

- Thiết bị kiểm thử: Laptop ASUS GL553VX
- Cấu hình thiết bị:
 - ✓ Chip Intel Core i5 6300HQ
 - ✓ RAM 8GB
 - ✓ Hệ điều hành Windows 10
 - ✓ Trình duyệt Microsoft Edge 81

Ứng dụng AR bán đồ nội thất

- Thiết bị kiểm thử: Samsung Galaxy A50S
- Cấu hình thiết bị:
 - ✓ Chip Exynos 9611
 - ✓ RAM 4GB
 - ✓ Hệ điều hành Android 10

Kết nối mạng: Mạng LAN

4.3. Đánh giá kiểm thử

Kết quả kiểm thử kịch bản

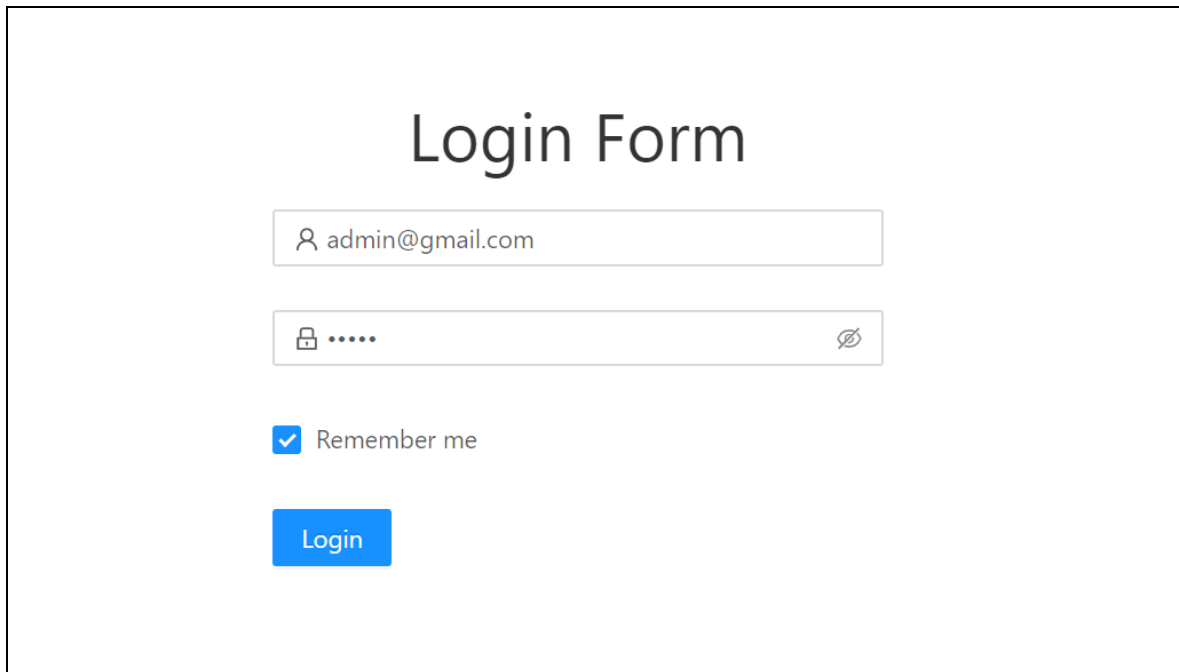
Kịch bản	Mức độ đạt yêu cầu
Kịch bản 1	Đạt
Kịch bản 2	Đạt
Kịch bản 3	Đạt

Bảng 2.4.1. Kết quả kiểm thử kịch bản

Kết quả kiểm thử

Một số hình ảnh kiểm thử web quản trị

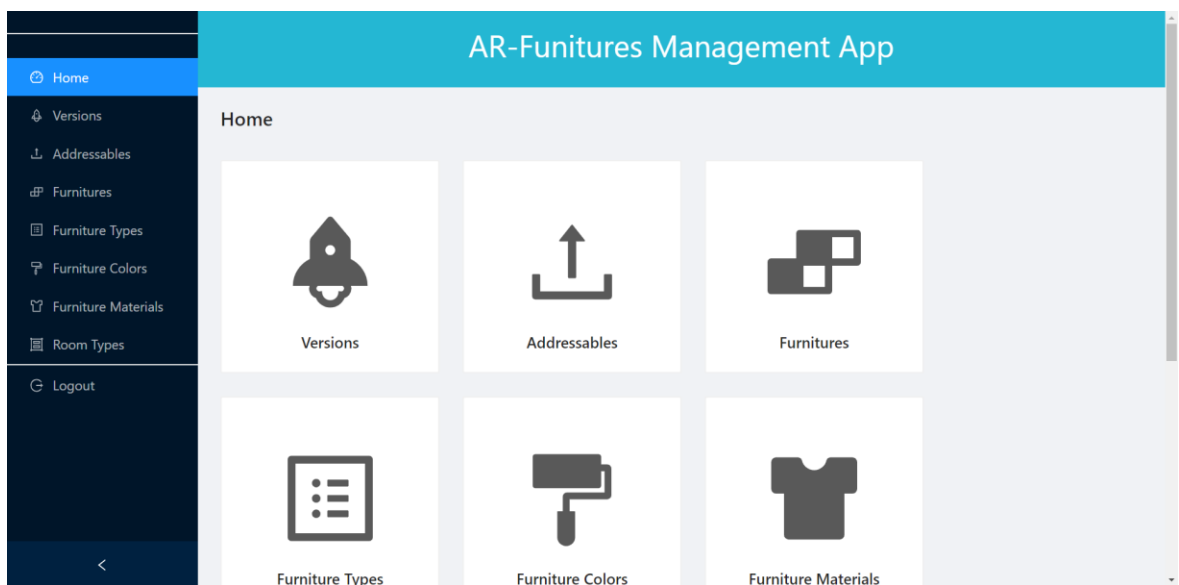
Giao diện đăng nhập:



The screenshot shows a 'Login Form' with a title 'Login Form' in a large, dark font. Below the title, there are two input fields: the first contains the email 'admin@gmail.com' and has a user icon on the left; the second contains masked characters '.....' and has a password icon on the left and a toggle icon on the right. Below these fields is a checkbox labeled 'Remember me' which is checked. At the bottom is a blue 'Login' button.

Hình 2.4.1. Giao diện đăng nhập ứng dụng web quản trị

Giao diện trang chủ:



Hình 2.4.2. Giao diện trang chủ của ứng dụng web quản trị

Giao diện upload dữ liệu đóng gói bởi Unity:

The screenshot shows the 'AR-Furniture Management App' interface. On the left is a dark sidebar with navigation links: Home, Versions, Addressables (highlighted), Furnitures, Furniture Types, Furniture Colors, Furniture Materials, Room Types, and Logout. The main content area is titled 'Unity Addressables' and contains a form titled 'Upload addressables Unity'. The form has four fields: 'Target' (a dropdown menu set to 'Android'), 'File hash' (a text input with a file icon), 'File json' (a text input with a file icon), and 'File bundle' (a text input with a file icon). Below these fields are 'Submit' and 'Reset' buttons. At the bottom right of the main area, it says 'Power by Ant-Design'.

Hình 2.4.3. Giao diện upload addressables của ứng dụng web quản trị

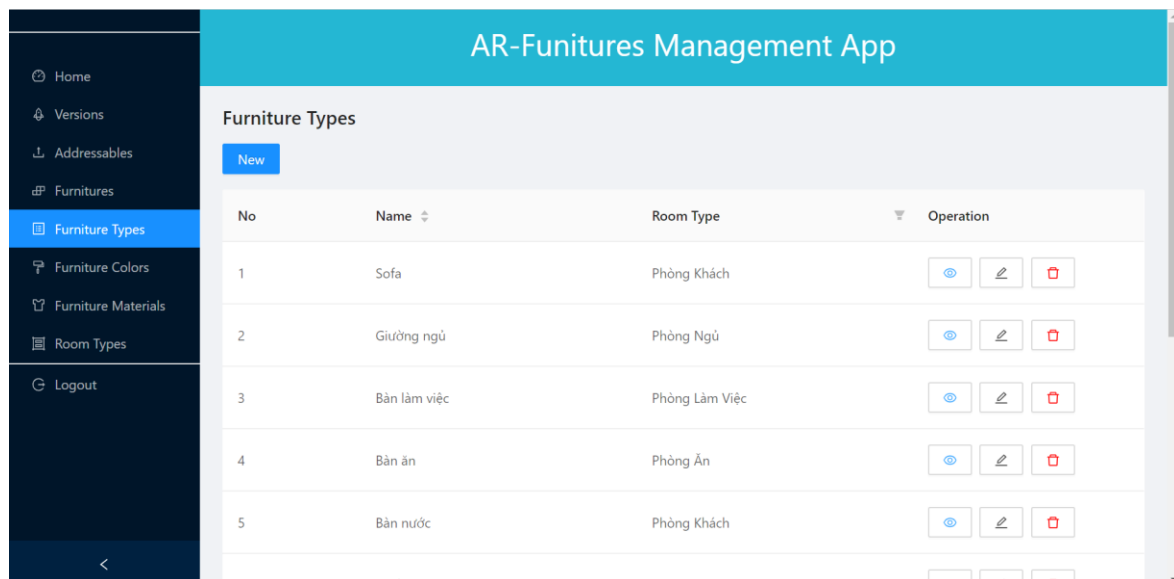
Giao diện quản lý đồ nội thất:

The screenshot shows the 'AR-Furniture Management App' interface with the 'Furnitures' section selected. A 'New' button is visible. Below it is a table with the following columns: No, Unity addressable, Name, Price, Type, Status, and Operation. The table contains five rows of furniture data.

No	Unity addressable	Name	Price	Type	Status	Operation
1	fur_1	Ghế gỗ nhựa đỏ có lưng tựa	550.000 đ	Ghế	Published	  
2	fur_2	Ghế bành vải trắng	1.400.000 đ	Ghế bành	Published	  
3	fur_3	Ghế bành nhựa đỏ	2.100.000 đ	Ghế bành	Published	  
4	fur_4	Sofa phong cách hiện đại	2.800.000 đ	Sofa	Published	  
5	fur_5	Ghế bành vải vàng	1.800.000 đ	Ghế bành	Published	  

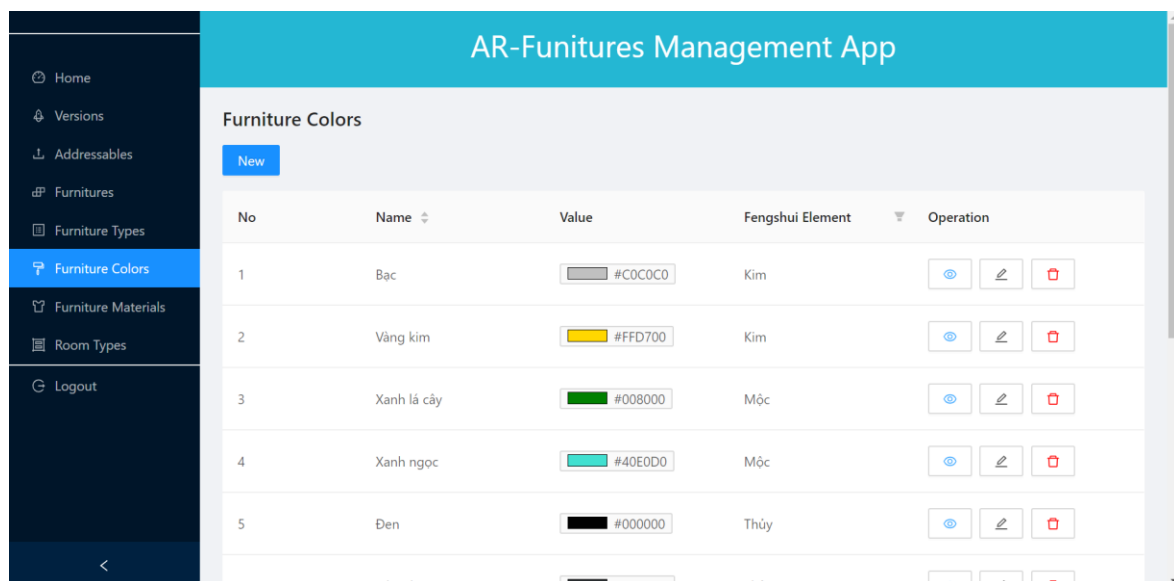
Hình 2.4.4. Giao diện quản lý đồ nội thất của ứng dụng web quản trị

Giao diện quản lý loại nội thất:



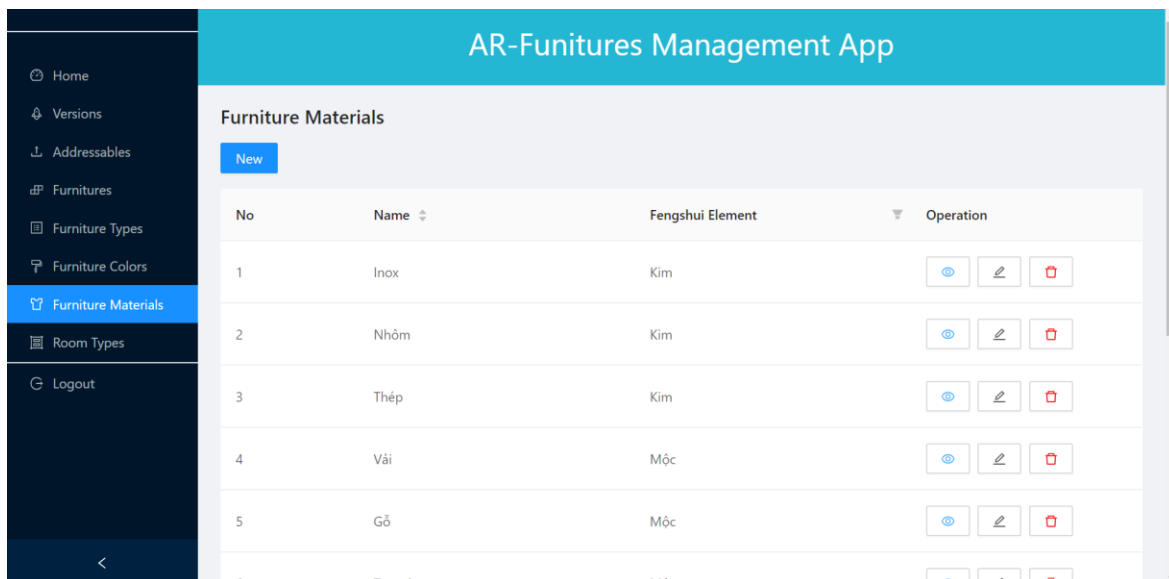
Hình 2.4.5. Giao diện quản lý loại nội thất của ứng dụng web quản trị

Giao diện quản lý màu sắc của nội thất:



Hình 2.4.6. Giao diện quản lý màu sắc nội thất của ứng dụng web quản trị

Giao diện quản lý vật liệu của nội thất:

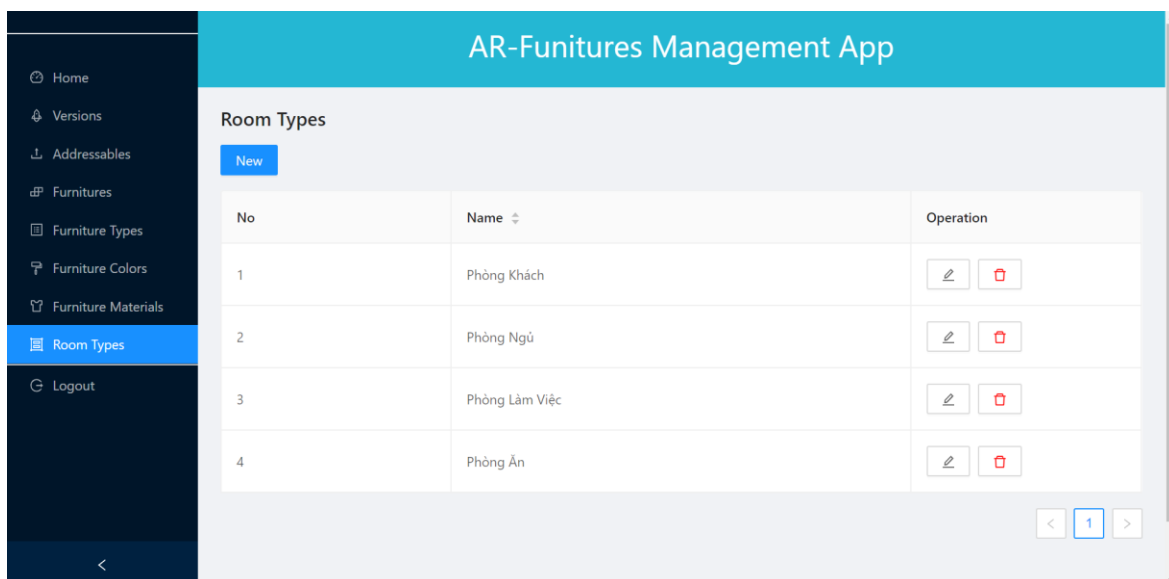


The screenshot displays the 'AR-Furniture Management App' interface. On the left is a dark sidebar with navigation options: Home, Versions, Addressables, Furnitures, Furniture Types, Furniture Colors, Furniture Materials (highlighted), Room Types, and Logout. The main content area is titled 'Furniture Materials' and includes a 'New' button. Below this is a table with columns: No, Name, Fengshui Element, and Operation. The table contains five rows of data.

No	Name	Fengshui Element	Operation
1	Inox	Kim	[Edit] [Delete]
2	Nhôm	Kim	[Edit] [Delete]
3	Thép	Kim	[Edit] [Delete]
4	Vải	Mộc	[Edit] [Delete]
5	Gỗ	Mộc	[Edit] [Delete]

Hình 2.4.7. Giao diện quản lý vật liệu nội thất của ứng dụng web quản trị

Giao diện quản lý loại phòng:

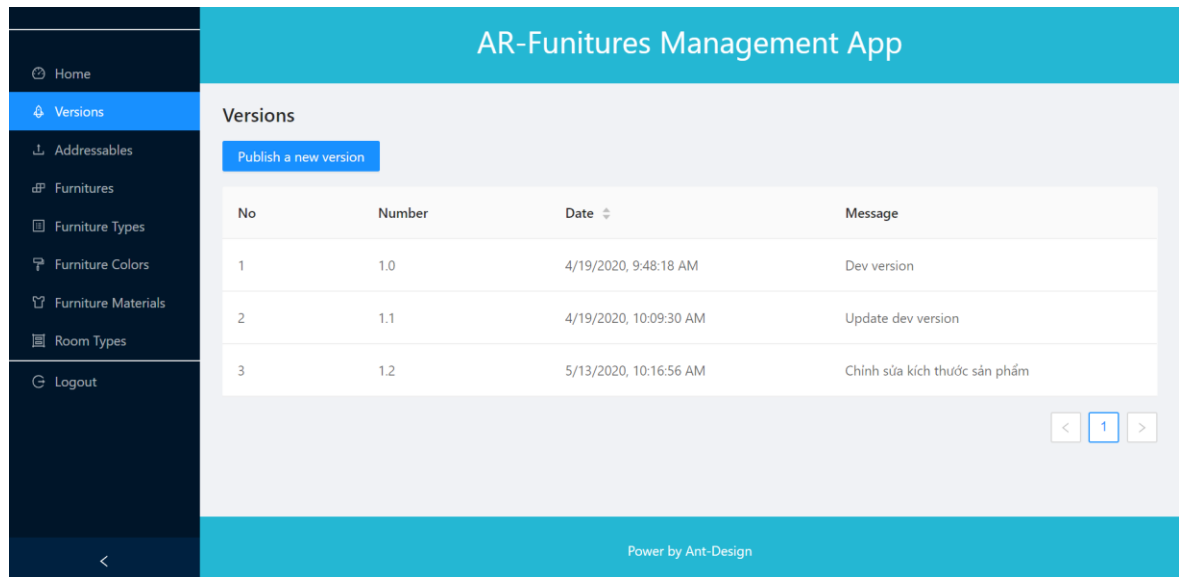


The screenshot displays the 'AR-Furniture Management App' interface. On the left is a dark sidebar with navigation options: Home, Versions, Addressables, Furnitures, Furniture Types, Furniture Colors, Furniture Materials, Room Types (highlighted), and Logout. The main content area is titled 'Room Types' and includes a 'New' button. Below this is a table with columns: No, Name, and Operation. The table contains four rows of data. At the bottom right of the table area, there are pagination controls showing '< 1 >'.

No	Name	Operation
1	Phòng Khách	[Edit] [Delete]
2	Phòng Ngủ	[Edit] [Delete]
3	Phòng Làm Việc	[Edit] [Delete]
4	Phòng Ăn	[Edit] [Delete]

Hình 2.4.8. Giao diện quản lý loại phòng của ứng dụng web quản trị

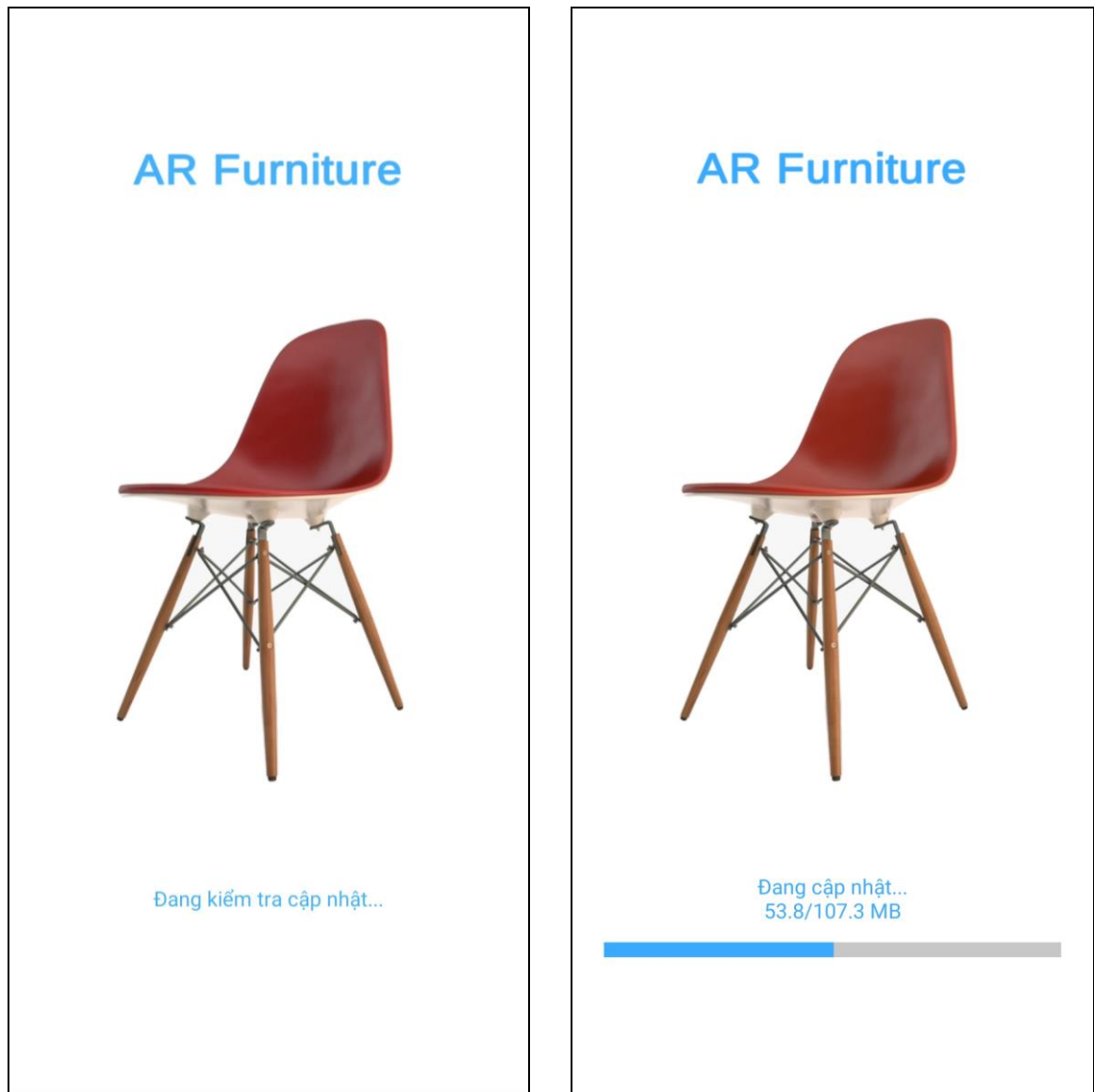
Giao diện quản lý phiên bản:



Hình 2.4.9. Giao diện quản lý phiên bản dữ liệu json của ứng dụng web quản trị

Một số hình ảnh kiểm thử ứng dụng AR mobile

Kiểm tra cập nhật:



Hình 2.4.10. Giao diện kiểm tra cập nhật của ứng dụng AR

Dò tìm mặt phẳng:

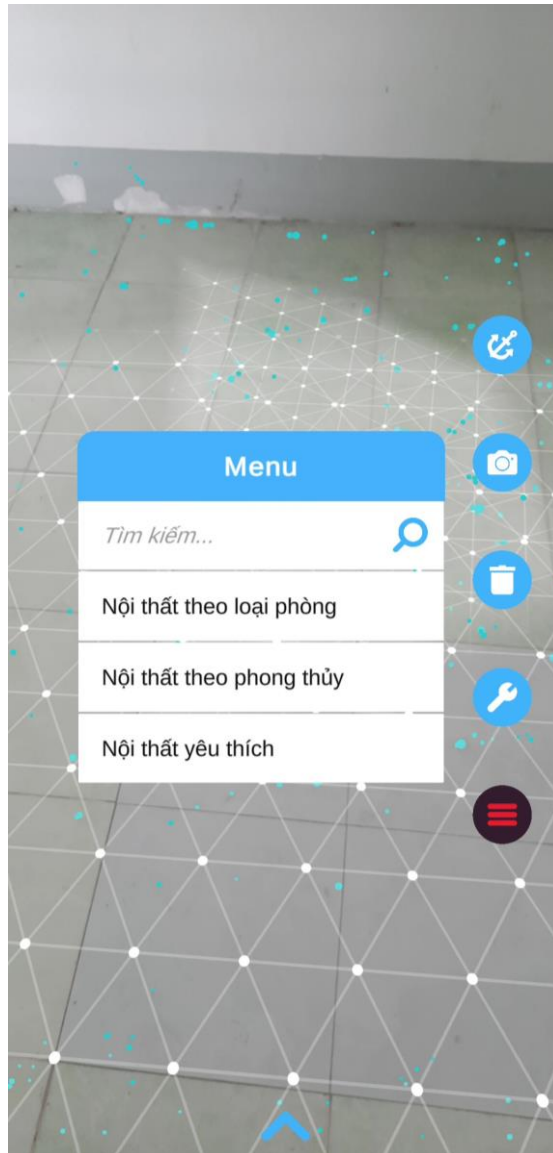


Hình 2.4.11. Giao diện dò tìm mặt phẳng của ứng dụng AR

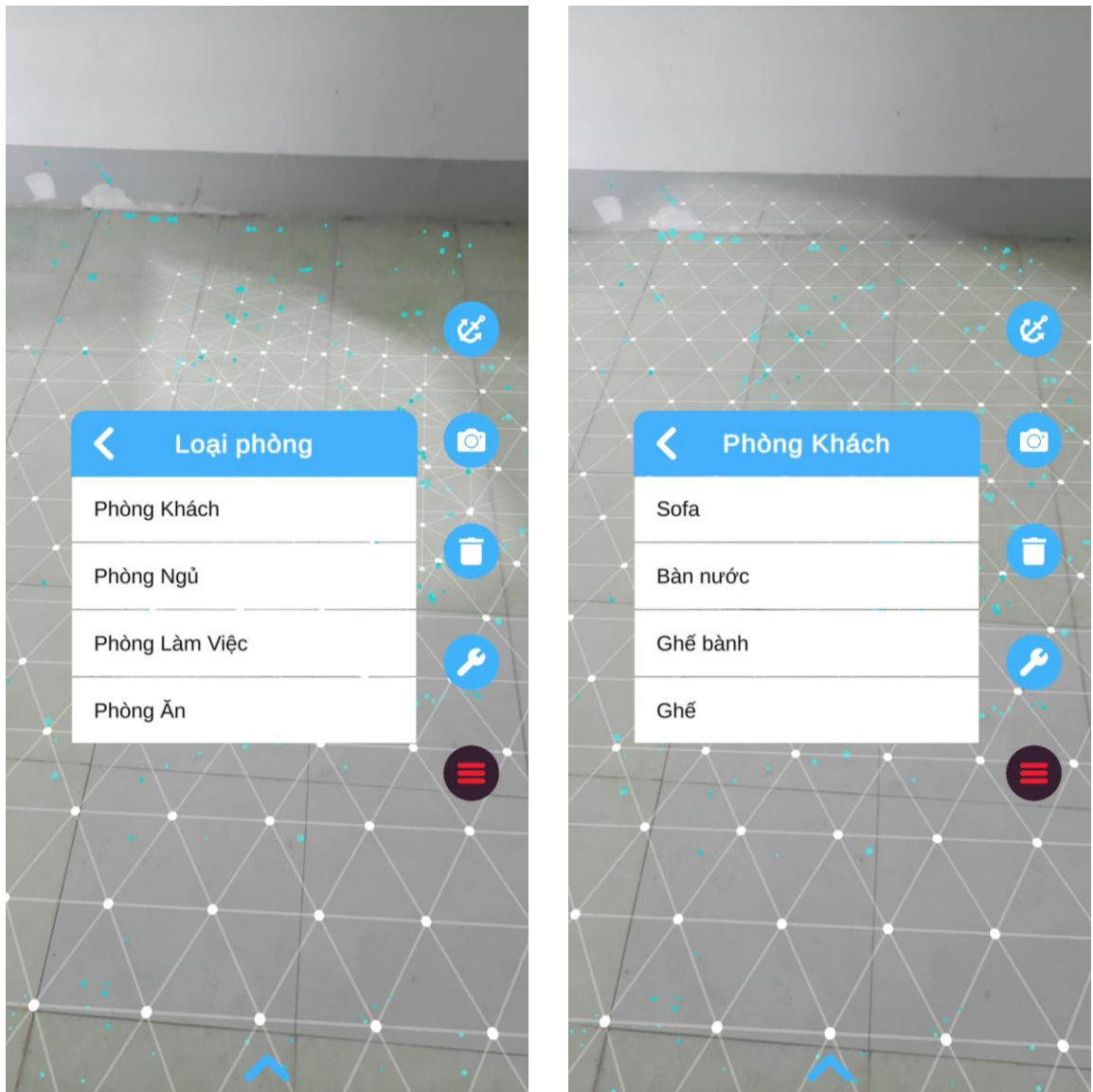


Hình 2.4.12. Mặt phẳng phát hiện được của ứng dụng AR

Giao diện menu:

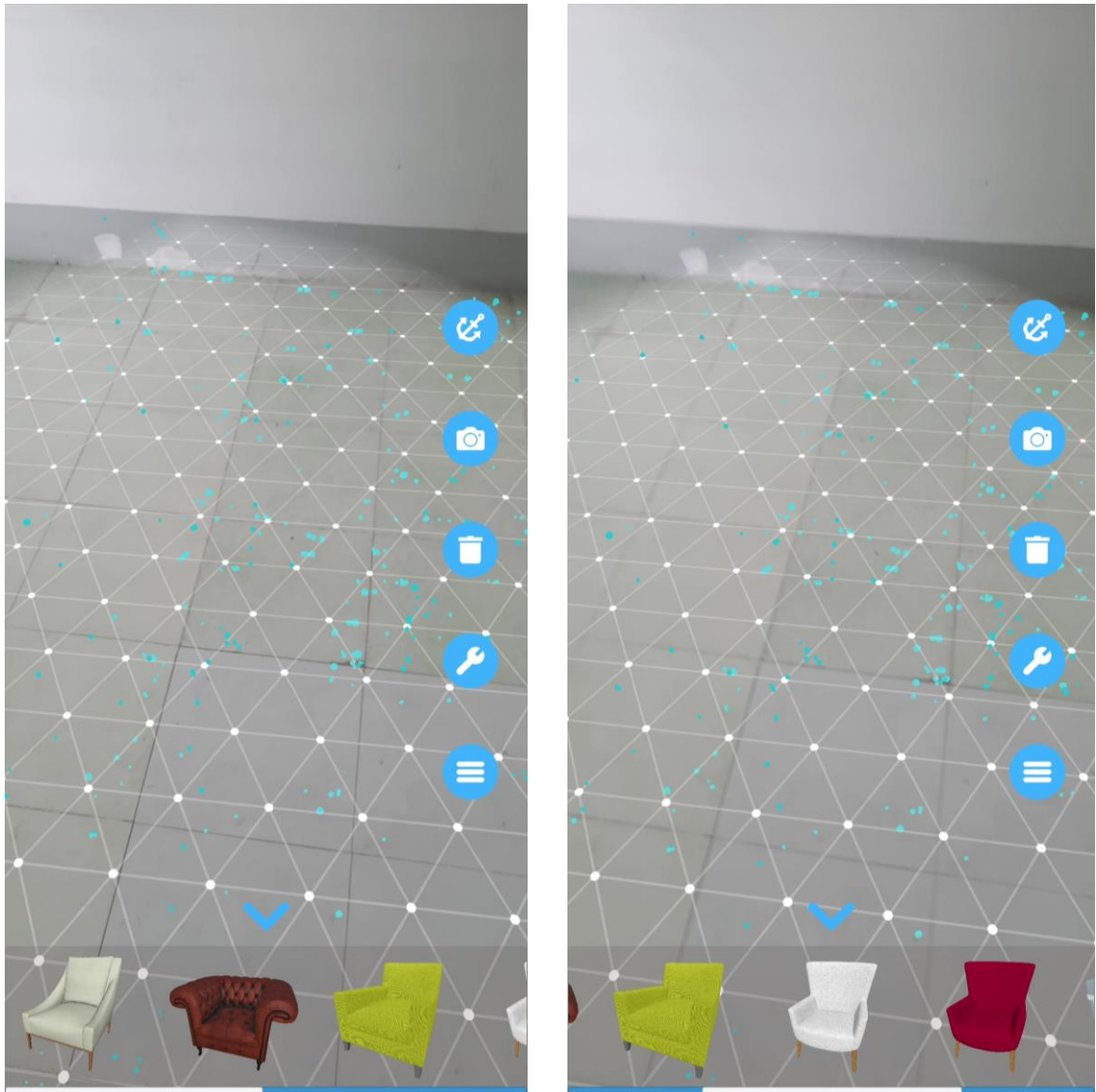


Hình 2.4.13. Giao diện menu chính của ứng dụng AR



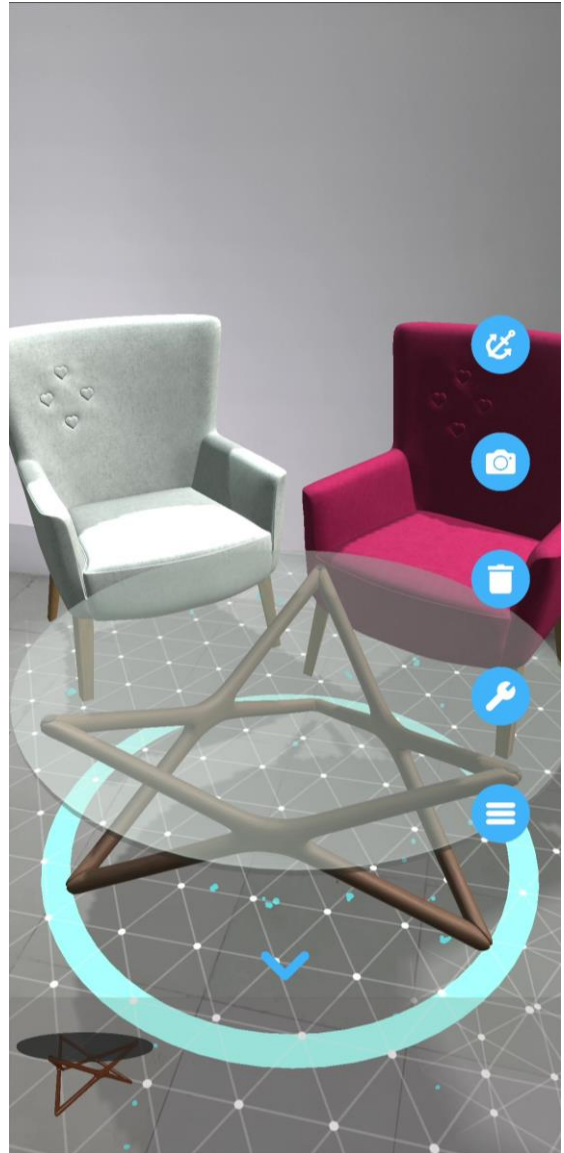
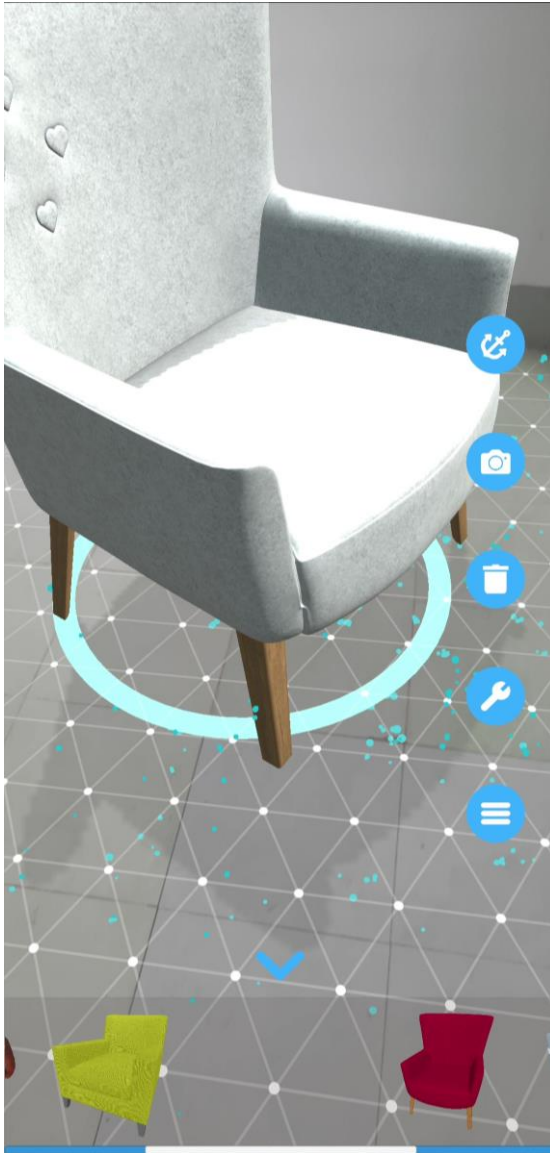
Hình 2.4.14. Giao diện menu cấp 2 và 3 của ứng dụng AR

Danh sách đồ nội thất:



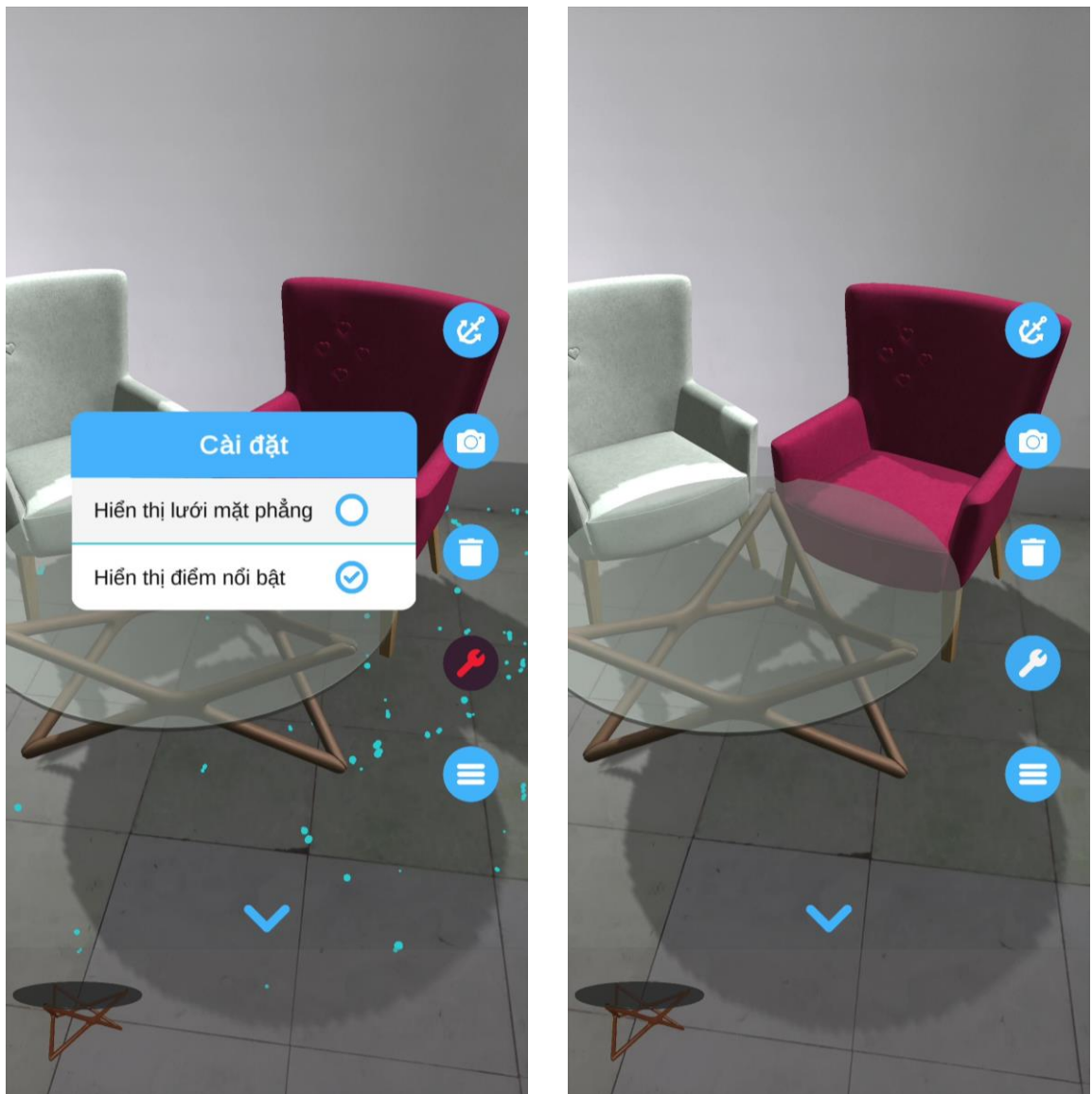
Hình 2.4.15. Giao diện liệt kê đồ nội thất của ứng dụng AR

Kéo thả nội thất:



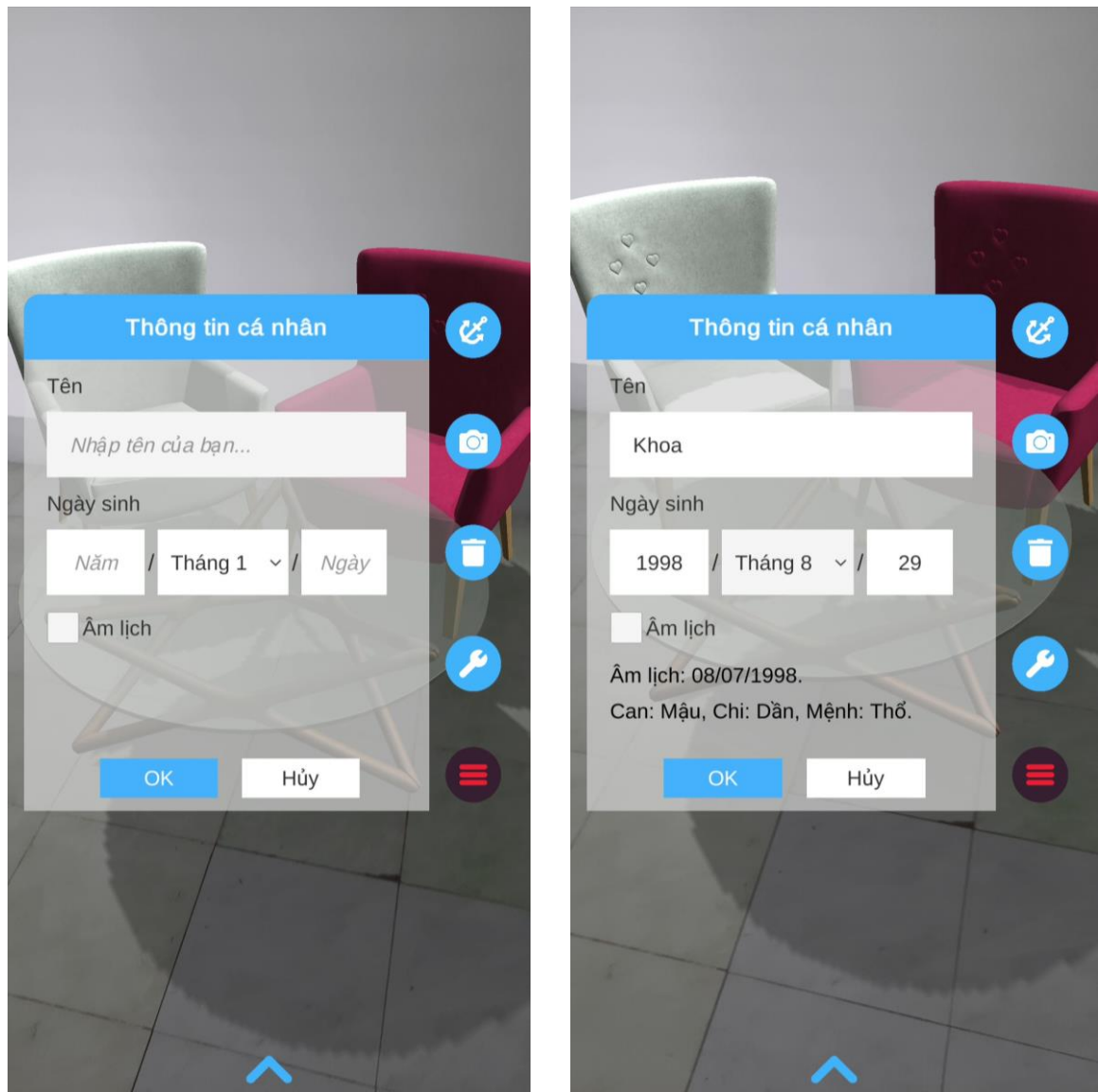
Hình 2.4.16. Giao diện kéo thả đặt nội thất của ứng dụng AR

Tùy chọn cài đặt:



Hình 2.4.17. Phân cảnh AR khi ẩn lưới và điểm đặc trưng

Nội thất theo phong thủy:



Hình 2.4.18. Giao diện thông tin số mệnh người dùng theo phong thủy

Thông tin sản phẩm nội thất:



Hình 2.4.19. Giao diện thông tin sản phẩm nội thất của ứng dụng AR

Phần 3. Kết luận

1. Kết quả đạt được

1.1. Kết quả

Xây dựng hoàn thiện hệ thống gồm 3 thành phần RESTful API server, Web client và AR mobile app đạt mục tiêu đề ra ban đầu.

Hoàn thành mục tiêu trọng tâm của đề tài nghiên cứu là xây dựng ứng dụng AR bán đồ nội thất với các tính năng cơ bản:

- Chọn mô hình nội thất và kéo thả đặt vào phân cảnh trong ứng dụng (căn phòng của người dùng trong không gian thực)
- Điều khiển mô hình nội thất trong phân cảnh: dịch chuyển, xoay, xóa
- Gợi ý nội thất dựa trên phong thủy
- Liệt kê nội thất: dựa theo loại phòng và loại nội thất
- Tìm kiếm nội thất theo tên gần đúng
- Chụp ảnh màn hình
- Làm mới phân cảnh
- Quản lý nội thất yêu thích: thêm vào danh sách yêu thích hoặc bỏ thêm
- Tự động cập nhật dữ liệu

1.2. Hạn chế

Ứng dụng AR bán đồ nội thất đang được phát triển ngày một hoàn thiện hơn nên hiện tại chưa hỗ trợ tính năng thương mại như giỏ hàng, mua hàng trên ứng dụng. Các thiết bị không hỗ trợ dịch vụ Google for AR thì sẽ không sử dụng được ứng dụng này. Ngoài ra, vẫn còn tồn tại một số lỗi nhỏ và đang tìm cách khắc phục ở phía ứng dụng AR bán đồ nội thất như trôi mặt phẳng ARCore hoặc bị crash ứng dụng khi xử lý mô hình nội thất phức tạp (kích thước lớn, kết cấu phức tạp).

Các thông tin của response cung cấp bởi REST API server còn hạn chế, khiến việc xử lý ở phía frontend phức tạp đôi chút. Bên cạnh đó, ứng dụng web quản lý chưa hỗ trợ responsive nên chỉ có thể sử dụng trên trình duyệt desktop.

2. Hướng phát triển

Trong tương lai, sẽ tích hợp thêm vào ứng dụng AR bán đồ nội thất những tính năng thương mại như giỏ hàng, mua hàng trực tiếp trên ứng dụng. Những cải tiến của hệ thống ứng dụng trong tương lai:

- Khắc phục những lỗi còn tồn tại ở phiên bản hiện tại
- Cải tiến/bổ sung dữ liệu của response REST API để giảm tải xử lý ở phía frontend
- Tạo responsive cho ứng dụng web quản lý
- Thêm tính năng mới cho ứng dụng web như quản lý bán hàng

Tài liệu tham khảo

- [1] Google, “ARCore,” Google Developers, ngày 28/08/2019. [Online]. Có sẵn tại: <https://developers.google.com/ar>. [Truy cập ngày 20/01/2020].
- [2] Google, “ARCore SDK for Unity,” Google Developer, 20/01/2020. [Online]. Có sẵn tại: <https://developers.google.com/ar/develop/unity>. [Truy cập ngày 23/05/2020].
- [3] Trung Nguyễn, “Tự học Unity 3D,” STDIO, ngày 13/09/2015. [Online]. Có sẵn tại: <https://www.stdio.vn/tutorials/topics/tu-hoc-unity-3>. [Truy cập ngày 30/09/2019].
- [4] Wikipedia, “Representational state transfer,” Wikipedia. [Online]. Có sẵn tại: https://en.wikipedia.org/wiki/Representational_state_transfer. [Lần cuối truy cập ngày 18/05/2020].
- [5] Wikipedia, “Phong thủy,” Wikipedia. [Online]. Có sẵn tại: https://vi.wikipedia.org/wiki/Phong_thủy. [Lần cuối truy cập ngày 18/05/2020].
- [6] Wikipedia, “Ngũ hành,” Wikipedia. [Online]. Có sẵn tại: https://vi.wikipedia.org/wiki/Ngũ_hành. [Lần cuối truy cập ngày 18/05/2020].
- [7] Hồ Ngọc Đức, “Thuật toán âm lịch,” Infomatik, 18/08/2008. [Online]. Có sẵn tại: <https://www.informatik.uni-leipzig.de/~duc/amlich/calrules.html>. [Truy cập ngày 11/03/2020].
- [8] Phong Linh Stone, “Hướng Dẫn Cách Tính Can Chi Theo Tuổi — Âm Lịch,” Medium, 24/02/2020. [Online]. Có sẵn tại: <https://medium.com/@phonglinggems/2e8572b65c51>. [Lần cuối truy cập ngày 11/03/2020].
- [9] Django, “Django overview,” Django Project. [Online]. Có sẵn tại: <https://www.djangoproject.com/start/overview/>. [Lần cuối truy cập ngày 15/05/2020].
- [10] JWT, “JSON Web Token Introduction,” JWT. [Online]. Có sẵn tại: <https://jwt.io/introduction/>. [Lần cuối truy cập ngày 18/05/2020].
- [11] Wikipedia, “MySQL,” Wikipedia. [Online]. Có sẵn tại: <https://en.wikipedia.org/wiki/MySQL>. [Lần cuối truy cập ngày 18/05/2020].
- [12] Wikipedia, “Redux (JavaScript library),” Wikipedia. [Online]. Có sẵn tại: [https://en.wikipedia.org/wiki/Redux_\(JavaScript_library\)](https://en.wikipedia.org/wiki/Redux_(JavaScript_library)). [Lần cuối truy cập ngày 19/05/2020].

- [13] Wikipedia, “*React (web framework)*,” Wikipedia. [Online]. Có sẵn tại: [https://en.wikipedia.org/wiki/React_\(web_framework\)](https://en.wikipedia.org/wiki/React_(web_framework)). [Lần cuối truy cập ngày 19/05/2020].
- [14] Wikipedia, “*Unity (game engine)*,” Wikipedia. [Online]. Có sẵn tại: [https://en.wikipedia.org/wiki/Unity_\(game_engine\)](https://en.wikipedia.org/wiki/Unity_(game_engine)). [Lần cuối truy cập ngày 19/05/2020].
- [15] Trần Nguyễn Anh Huy, “*Ứng dụng Unity3D xây dựng game Không chiến – Air Combat*,” **Luận văn tốt nghiệp** đại học, Đại học Cần Thơ, Việt Nam, 2019.

