MINISTRY OF EDUCATION AND TRAINING

FPT UNIVERSITY

Capstone Project Document

FoodFund - A crowdfunding system for food preparation and distribution to support communities in need

GFA25SE52	
Group Members	Trương Minh Phước - SE184067 Huỳnh Đình Luyện - SE172341 Huỳnh Lê Nhật Hoàng - SE172474 Phan Cảnh Bảo Duy - SE184173
Supervisor	Nguyễn Minh Sang
Ext Supervisor	
Capstone Project code	FA25SE052

- Ho Chi Minh, September 2025 -

Table of Contents

Acknowledgement **10**
Definition and Acronyms **11**
I. Project Introduction **13**
1. Overview 13
1.1 Project Information 13
1.2 Project Team 13
2. Product Background 13
3. Existing Systems 14
3.1 GoFundMe 14
3.2 MoMo Donation Platform 14
3.3 ZaloPay Charity 14
3.4 Thien Nguyen App 15
4. Business Opportunity 15
5. Software Product Vision 16
6. Project Scope & Limitations 16
6.1 Major Features 16
6.2 Limitations & Exclusions 17
II. Project Management Plan **18**
1. Overview 18
1.1 Scope & Estimation 18
1.2 Project Objectives 24
1.3 Project Risks 24
2. Management Approach 25
2.1 Project Process 25
2.2 Quality Management 25
2.3 Training Plan 26
3. Project Deliverables 26
4. Responsibility Assignments 27
5. Project Communications 27
6. Configuration Management 28
6.1 Document Management 28
6.2 Source Code Management 28
6.3 Tools & Infrastructures 28
III. Software Requirement Specification **29**
1. Product Overview 29
2. User Requirements 29
2.1 Actors 29
2.2 Use Case 30
3. Functional Requirements 34
3.1 System Functional Overview 34
3.2 Authentication 45

3.3 Campaign Categories..... 48
3.4 Campaigns.....53
3.5 Posts Of A Campaign.....70
3.7 Organisation.....80
3.8 Profile.....92
3.9 Manage Users..... 95
3.10 Donation..... 97
3.11 Ingredient Disbursement Request..... 102
3.12 Operation Disbursement Request.....106
3.13 Inflow Transaction.....110
3.14 Meal Batch..... 114
3.15 Expense Proof..... 117
3.16 Delivery Task..... 121
3.17 Notification..... 124
3.18 Badge..... 126
3.19 Wallet.....130
4. Non-Functional Requirements.....134
4.1 External Interfaces..... 134
4.2 Quality Attributes.....134
5. Requirement Appendix.....135
5.1 Business Rules.....135
5.2 Common Requirements..... 137
5.3 Application Messages List..... 138
5.4 Other Requirements..... 140
IV. Software Design Description.....141
1. System Design.....141
1.1 System Architecture.....141
1.2 Package Diagram.....144
2. Database Design..... 149
2.1 FoodFund Database Design.....149
2.2 Database Design for User Service..... 158
2.3 Database Design for Campaign Service.....159
2.4 Database Design for Operation Service..... 161
3. Detailed Design.....162
3.1 Authentication..... 162
3.2 Campaign Category Management.....166
3.3 User Management..... 169
3.4 Organization Management..... 173
3.5 Wallet Management..... 179
3.6 Badge Management.....182
3.7 Donation Management.....183
3.8 Disbursement Management..... 188

3.9 Campaign Management....194
3.10 Campaign Reassignment Management.... 201
3.11 Post Management....204
3.12 Post Like Management....208
3.13 Post Comment Management.... 211
3.14 Expense Proof Management....214
3.15 Meal Batch Management....218
3.16 Delivery Task Management....222
V. Software Testing Documentation.... 227
1. Scope of Testing....227
2. Test Strategy.... 227
2.1 Testing Types....227
2.2 Test Levels.... 228
2.3 Supporting Tools.... 228
3. Test Plan....228
3.1 Human Resources.... 228
3.2 Test Environment.... 228
3.3 Test Milestones.... 229
4. Test Cases.... 229
5. Test Reports....229
VI. Release Package & User Guides.... 232
1. Deliverable Package....232
2. Installation Guides....232
2.1 System Requirements.... 232
2.2 Installation Instruction....232
3. User Manual.... 259
3.1 Overview....259
3.2 Donation.... 260
3.3 User registration.... 262
3.4 Create organization....263
3.5 Create campaign.... 265
3.6 Operation execution.... 267

List of tables

Table 1. Definition and Acronyms..... 12
Table 2. Project Team..... 13
Table 3. Scope and Estimation..... 23
Table 4. Project Objectives..... 24
Table 5. Project Risks..... 24
Table 6. Training Plan..... 26
Table 7. Project Deliverables..... 26
Table 8. Responsibility Assignments..... 27
Table 9. Project Communications..... 27
Table 10. Tools & Infrastructure..... 28
Table 11. Actors..... 30
Table 12. Use Case Diagram Descriptions..... 33
Table 13. Screen Descriptions..... 40
Table 14. Screen Authorization..... 41
Table 15. Non-Screen Functions..... 42
Table 16. Entities Descriptions..... 44
Table 17. Business Rules..... 137
Table 18. Application Messages List..... 140
Table 19. Backend Package Description..... 145
Table 20. Frontend Package Description..... 147
Table 21. Mobile Package Diagram Description..... 148
Table 22 Attribute Data Dictionary..... 158
Table 23. Database Design for User Service Table Descriptions..... 159
Table 24. Database Design for Campaign Service Table Descriptions..... 160
Table 25. Database Design for Operation Service Table Descriptions..... 162
Table 26. AuthenticationResolver Class Specification..... 163
Table 27. Registration Class Specification..... 163
Table 28. UserResolver Class Specification..... 163
Table 29. CampaignCategoryQueryResolver Class Specification..... 166
Table 30. CampaignCategoryMutationResolver Class Specification..... 166
Table 31. UserQueryResolver Class Specification..... 169
Table 32. UserMutationResolver Class Specification..... 169
Table 33. UserAdminResolver Class Specification..... 169
Table 34. DonorProfileResolver Class Specification..... 173
Table 35. OrganizationAdminResolver Class Specification..... 173
Table 36. FundraiserProfileResolver Class Specification..... 174
Table 37. OrganizationMutationResolver Class Specification..... 174
Table 38. OrganizationQueryResolver Class Specification..... 174
Table 39. WalletQueryResolver Class Specification..... 179
Table 40. FundraiserWalletResolver Class Specification..... 179
Table 41. AdminWalletResolver Class Specification..... 180

Table 42. BadgeQueryResolver Class Specification.....182
Table 43. BadgeMutationResolver Class Specification.....182
Table 44. DonorQueryResolver Class Specification.....184
Table 45. DonorMutationResolver Class Specification.....184
Table 46. DonationWebhookController.....184
Table 47. SepayWebhookController.....184
Table 48. InflowTransactionAdminResolver Class Specification.....188
Table 49. IngredientRequestMutationResolver Class Specification.....188
Table 50. OperationRequestMutationResolver Class Specification.....188
Table 51. InflowTransactionFundraiserResolver Class Specification.....189
Table 52. IngredientRequestQueryResolver Class Specification.....189
Table 53. OperationRequestQueryResolver Class Specification.....189
Table 54. CampaignQueryResolver Class Specification.....195
Table 55. CampaignMutationResolver Class Specification.....195
Table 56. CampaignReassignmentQueryResolver Class Specification.....201
Table 57. CampaignReassignmentMutationResolver Class Specification.....201
Table 58. PostQueryResolver Class Specification.....204
Table 59. PostMutationResolver Class Specification.....205
Table 60. PostLikeQueryResolver Class Specification.....208
Table 61. PostLikeMutationResolver Class Specification.....208
Table 62. PostCommentQueryResolver Class Specification.....211
Table 63. PostCommentMutationResolver Class Specification.....212
Table 64. ExpenseProofQueryResolver Class Specification.....214
Table 65. ExpenseProofMutationResolver Class Specification.....215
Table 66. MealBatchQueryResolver Class Specification.....218
Table 67. MealBatchMutationResolver Class Specification.....219
Table 68. DeliveryTaskQueryResolver Class Specification.....223
Table 69. DeliveryTaskMutationResolver Class Specification.....223
Table 70. Test Levels.....228
Table 71. Supporting Tools.....228
Table 72. Human Resources.....228
Table 73. Test Environment.....228
Table 74. Test Milestones.....229
Table 75. Test Statistics Report.....230
Table 76. Unit Test Report.....230
Table 77. Deliverable Package.....232

List of figures

Figure 1. FoodFund Context Diagram..... 29
Figure 2. Use Case Diagram..... 30
Figure 3. Admin Web Screen Flow..... 34
Figure 4. Guest Web Screen Flow..... 34
Figure 5. Donor Web Screen Flow.....35
Figure 6. Fundraiser Web Screen Flow.....35
Figure 7. Donor Mobile Screen Flow.....36
Figure 8. Kitchen Staff Mobile Screen Flow.....36
Figure 9. Delivery Staff Mobile Screen Flow..... 37
Figure 10. FoodFund Entity Relationship Diagram.....42
Figure 11. System Architecture Diagram.....143
Figure 12. Backend Package Diagram.....146
Figure 13. Frontend Package Diagram..... 148
Figure 14. Mobile Package Diagram.....149
Figure 15. FoodFund Database Design.....151
Figure 16. Database Design for User Service..... 160
Figure 17. Database Design for Campaign Service.....161
Figure 18. Database Design for Operation Service.....163
Figure 19. Authentication Class Diagram..... 164
Figure 20. Sign In Sequence Diagram.....166
Figure 21. Sign Up Sequence Diagram..... 166
Figure 22. Sign Up Sequence Diagram..... 167
Figure 23. Campaign Category Class Diagram.....168
Figure 24. View Campaign Categories Sequence Diagram.....169
Figure 25. Create Campaign Category Sequence Diagram.....169
Figure 26. Update Campaign Category Sequence Diagram..... 170
Figure 27. Delete Campaign Category Sequence Diagram.....170
Figure 28. User Management Class Diagram..... 171
Figure 29. Get My Profile Sequence Diagram..... 172
Figure 30. Update Profile Sequence Diagram..... 172
Figure 31. Get All Users Sequence Diagram.....173
Figure 32. Update User Account Sequence Diagram..... 174
Figure 33. Organization Management Class Diagram..... 175
Figure 34. Request Create Organization Sequence Diagram.....177
Figure 35. Approve Request Create Organization Sequence Diagram..... 178
Figure 36. Approve Join Organization Request..... 179
Figure 37. Leave Organization Sequence Diagram.....180
Figure 38. Organization State Diagram..... 180
Figure 39. Wallet Management Class Diagram..... 181
Figure 40. Search Wallet Transactions Sequence Diagram..... 182
Figure 41. Get System Wallet Sequence Diagram.....183

Figure 42. Get Fundraiser Wallet Sequence Diagram..... 183
Figure 43. Badge Management Class Diagram.....184
Figure 44. View Profile Badge Sequence Diagram..... 185
Figure 45. Donation Management Class Diagram.....185
Figure 46. Create Donation Sequence Diagram..... 187
Figure 47. PayOS Webhook Donation Payment Process Sequence Diagram..... 188
Figure 48. Sepay Webhook Bank Transfer Process Sequence Diagram.....189
Figure 49. Donation State Diagram.....189
Figure 50. Disbursement Management Class Diagram..... 190
Figure 51. Create Ingredient Request Sequence Diagram..... 192
Figure 52. Approve Ingredient Request Sequence Diagram..... 193
Figure 53. Create Disbursement Sequence Diagram.....194
Figure 54. Confirm Disbursement Sequence Diagram..... 195
Figure 55. Campaign Management Class Diagram.....196
Figure 56. Create Campaign Sequence Diagram.....198
Figure 57. Search Campaigns Sequence Diagram..... 198
Figure 58. View Campaign Detail Sequence Diagram..... 199
Figure 59. View My Campaigns Sequence Diagram.....199
Figure 60. View Campaign Statistics Sequence Diagram.....200
Figure 61. Change Campaign Status Sequence Diagram.....200
Figure 62. Extend Campaign Sequence Diagram..... 201
Figure 63. Update Phase Status Sequence Diagram..... 202
Figure 64: Campaign State Diagram.....202
Figure 65: Campaign Phase State Diagram..... 202
Figure 66. Campaign Reassignment Management Class Diagram..... 203
Figure 67. Assign Campaign To Organizations Sequence Diagram.....204
Figure 68. Get Pending Reassignment Requests Sequence Diagram.....205
Figure 69. Respond Campaign Assignment Sequence Diagram.....205
Figure 70: Campaign Reassignment State Diagram..... 206
Figure 71. Post Management Class Diagram.....207
Figure 72. Create Post Sequence Diagram..... 208
Figure 73. Get Posts by Campaign Sequence Diagram.....209
Figure 74. Update Post Sequence Diagram.....209
Figure 75. Delete Post Sequence Diagram..... 210
Figure 76. Post Like Management Class Diagram.....211
Figure 77. Like Post Sequence Diagram..... 212
Figure 78. Unlike Post Sequence Diagram.....213
Figure 79. View Post Likers Sequence Diagram.....213
Figure 80. Post Comment Management Class Diagram.....214
Figure 81. Create Comment Sequence Diagram..... 216
Figure 82. Get Post Comments Sequence Diagram.....216
Figure 83. Update Comment Sequence Diagram.....217

Figure 84. Delete Comment Sequence Diagram..... 217
Figure 85. Expense Proof Management Class Diagram.....218
Figure 86. Create Expense Proof Sequence Diagram..... 219
Figure 87. Update Expense Proof Status Sequence Diagram..... 220
Figure 88. View Expense Proof List Sequence Diagram..... 220
Figure 89. View Detail Expense Proof Sequence Diagram..... 221
Figure 90. View Expense Proof Statistics Sequence Diagram.....221
Figure 91. Meal Batch Class Diagram..... 222
Figure 92. Create Meal Batch Sequence Diagram..... 223
Figure 93. Update Meal Batch Status Sequence Diagram.....224
Figure 94. View Meal Batches Sequence Diagram..... 224
Figure 95. View Meal Batch Detail Sequence Diagram..... 225
Figure 96: Meal Batch State Diagram.....226
Figure 97. Delivery Task Management..... 226
Figure 98. Get Delivery Tasks Sequence Diagram..... 228
Figure 99. Get Delivery Task Detail Sequence Diagram.....228
Figure 100. Assign Delivery Task to Staff Sequence Diagram..... 229
Figure 101. Update Delivery Task Status Sequence Diagram..... 229
Figure 102. Get My Delivery Tasks Sequence Diagram..... 230
Figure 103: Delivery Task State Diagram..... 230
Figure 104. Test Case Report Pie Chart..... 235

Acknowledgement

Completing this Capstone Project marks the significant conclusion of our four-year journey at FPT University. It has been a challenging yet rewarding semester, and we could not have reached the finish line without the guidance and support of our mentors and peers.

We are deeply grateful to the Board of Directors and the Software Engineering Department at FPT University for providing the resources and academic foundation necessary for this thesis. The knowledge we have gained over the years was instrumental in bringing this project to life.

A special thanks goes to our supervisor, Mr. Nguyễn Minh Sang. His mentorship, from the initial ideation phase to the final product, was critical in shaping the quality of our work. We truly appreciate the time he spent reviewing and directing our team.

We also want to thank Mr. Đỗ Tấn Nhàn, Mr. Tôn Thất Hoàng Minh, Mr. Nguyễn Thế Hoàng, Mr. Đặng Ngọc Minh Đức, and all the lecturers who took the time to provide constructive feedback. Your insights on our shortcomings helped us refine the project and deliver a more complete solution.

To my team members: thank you for the hard work, patience, and collaboration. We stuck together from the beginning, and that persistence was the key to our success.

We also extend our thanks to our families and friends. Your encouragement and outside perspectives gave us the motivation and fresh ideas we needed during this final stage.

Finally, while we have put our best effort into this project, there may still be room for improvement regarding technology and business logic. We welcome any feedback and appreciate your continued support as we start our careers.

Sincerely.

Definition and Acronyms

Acronym	Definition
ACID	Atomicity, Consistency, Isolation and Durability
API	Application Program Interface
AWS	Amazon Web Services
BR	Business Rule
CD	Change Directory
CDC	Change Data Capture
CDN	Content Delivery Network
CI	Continuous Integration
CPU	Central Processing Unit
CR	Common Requirement
DTO	Data Transfer Object
FK	Foreign Key
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
ID	Identity Document
IDE	Integrated Development Environment
JSON	JavaScript Object Notation
JWT	JSON Web Token
MFA	Multi-factor authentication
NPM	Node Package Manager
OCR	Optical Character Recognition
ORM	Object-Relational Mapping
OTP	One-time Password
PK	Primary Key

Acronym	Definition
RBAC	Role-Based Access Control
RPC	Remote Procedure Call
SQL	Structured Query Language
TDD	Test-Driven Development
TLS	Transport Layer Security
QR	Quick-response
UAT	User Acceptance Test
UI	User Interface
URL	Uniform Resource Locator
UX	User Experience
WAL	Write-Ahead Log

Table 1. Definition and Acronyms

I. Project Introduction

1. Overview

1.1 Project Information

- Project name: A crowdfunding system for food preparation and distribution to support communities in need
- Project code: FA25SE052
- Group name: GFA25SE52
- Software type: Web app, Mobile app

1.2 Project Team

Full Name	Role	Email	Mobile
Nguyễn Minh Sang	Supervisor	sangnm18@fpt.edu.vn	0934041469
Trương Minh Phước	Leader	phuoctmse184067@fpt.edu.vn	0901308975
Huỳnh Đình Luyện	Member	luyenhdse172341@fpt.edu.vn	0328858148
Huỳnh Lê Nhật Hoàng	Member	hoanghlnse172474@fpt.edu.vn	0939023883
Phan Cảnh Bảo Duy	Member	duypcbse184173@fpt.edu.vn	0938848615

Table 2. Project Team

2. Product Background

FoodFund was created in response to the persistent challenges of community-led charity campaigns in Vietnam, especially in urban centers like Ho Chi Minh City. Current food fundraising and donation activities are often managed through manual processes or scattered social media groups, leading to three major problems: lack of transparency in fund usage, difficulty in tracking allocations, and inefficiencies in coordination among fundraisers, kitchen staff, and delivery teams. Donors frequently express concerns about whether their contributions are used effectively.

The idea for FoodFund originated from students and mentors who experienced these issues firsthand during volunteering. They recognized the need for a centralized digital platform that not only collects donations but also ensures transparent fund allocation and supports the entire workflow—from ingredient purchase to meal preparation and delivery.

Designed for donors, fundraiser organizations (including their kitchen and delivery staff), and administrators. FoodFund aims to modernize charity operations, reduce manual errors, and build long-term trust. By standardizing processes and providing visibility across the chain, the platform ensures accountability and delivers more meals efficiently to people in need.

3. Existing Systems

3.1 GoFundMe

- **Description**: GoFundMe is the world's largest online fundraising platform that allows individuals, non-profits, and organizations to raise money for personal needs, emergencies, and charitable causes.
- **Link**: https://www.gofundme.com
- **System Actors**: Donors, Campaign Organizers, Platform Administrators.
- **Features**:
    - Easy campaign creation with images and descriptions.
    - Online donation through multiple payment methods.
    - Progress tracking (goal vs. amount raised).
    - Sharing campaigns via social media.
- **Pros**: Simple to use, large donor community, strong reputation.
- **Cons**: High service fees, limited transparency on how funds are spent after the campaign ends, no built-in meal/ingredient management features.

3.2 MoMo Donation Platform

- **Description**: MoMo e-wallet integrates a donation platform where users can contribute to charity campaigns directly from the mobile app. Many NGOs and government-backed initiatives partner with MoMo for fundraising.
- **Link**: https://momo.vn/cong-dong
- **System Actors**: Donors, NGOs/Charity Organizations, MoMo Admins.
- **Features**:
    - Quick in-app donation via e-wallet balance or linked bank cards.
    - Wide range of campaigns, from education to healthcare to disaster relief.
    - Transaction receipts and donation history stored in-app.
- **Pros**: Convenient for Vietnamese users, strong trust due to MoMo's brand, low transaction friction.
- **Cons**: Limited campaign management features for organizers, donors have little visibility into post-donation fund usage.

3.3 ZaloPay Charity

- **Description**: ZaloPay also integrates donation functions, enabling users to support charities directly in the Zalo app, which has a large user base in Vietnam.
- **Link**: https://donation.zalopay.vn/
- **System Actors**: Donors, NGOs/Charities, ZaloPay Admins.
- **Features**:
    - Direct donation through e-wallet.
    - Campaign promotions through Zalo's messaging ecosystem.
    - Secure payment and donation confirmation.
- **Pros**: Easy to access, strong reach due to Zalo's popularity, quick donation process.
- **Cons**: Basic transparency, no detailed reporting on how funds are used, not designed for end-to-end campaign management (ingredient purchase, meal delivery, etc.).

3.4 Thien Nguyen App

- **Description**: Thiện Nguyện (thiennguyen.app) is a national digital charity platform initiative operated by MBBank. It aims to create a transparent, centralized space for fundraising and charity activities in Vietnam, allowing organizations and individuals to launch, manage, and participate in campaigns.
- **Link**: https://thiennguyen.app/
- **System Actors**:Donors, Campaign Organizers (individuals, NGOs, groups), Platform Administrators.
- **Features**:
    - Create and manage charity campaigns (individual or organizational).
    - Real-time transaction statements for public transparency.
    - Donation via integrated e-banking and mobile apps (iOS, Android).
    - "Charity Map" (Bản đồ thiện nguyện) to locate and visualize campaigns. Community engagement with news, events, and sharing functions.
- **Pros**: Nationally backed, strong credibility, transparent transaction tracking, convenient mobile access, centralized campaign management.
- **Cons**: Limited operational workflow tracking (no ingredient or delivery management), unclear verification and auditing mechanisms, donor updates mainly financial (not impact-focused), no logistics or in-kind donation support.

4. Business Opportunity

Food insecurity remains a persistent challenge in Vietnam, where many low-income families lack regular access to nutritious meals. Existing charitable activities—whether through ad-hoc community groups or digital platforms like MoMo and ZaloPay—often stop at the point of fund transfer, leaving donors with little visibility into how their contributions are spent. Global platforms such as GoFundMe, while powerful for raising funds, are not designed for food distribution workflows. This lack of transparency and accountability reduces donor trust and long-term engagement.

The opportunity lies in FoodFund: a specialized crowdfunding and operations management system tailored for food aid. Unlike existing solutions, FoodFund integrates fundraising with ingredient purchasing, kitchen operations, and delivery tracking—ensuring transparency at every step. Donors gain confidence through real-time updates and proof of impact, while committees, kitchen staff, and delivery teams benefit from streamlined workflows.

The timing is favorable. Vietnam's digital payment adoption is accelerating, with millions using mobile wallets daily, and the culture of community-based giving has grown significantly after COVID-19. By leveraging these trends, FoodFund positions itself as a scalable, transparent, and socially impactful platform that not only delivers meals more efficiently but also strengthens long-term donor trust and participation.

5. Software Product Vision

FoodFund empowers fundraiser organizations to manage charity campaigns with full transparency and accountability. Each fundraiser represents a registered organization approved by the admin. The organization's kitchen and delivery staff collaborate digitally through structured workflows for ingredient procurement, meal preparation, and food distribution.

Donors contribute securely via a unique campaign QR code, while admins oversee all fund movements and approve staged disbursements. This ensures that every donated amount is properly allocated, verifiable through receipts, and publicly auditable.

The vision is to establish a nationwide transparent ecosystem where every step of charitable food distribution is traceable — from donation to delivery.

6. Project Scope & Limitations

6.1 Major Features

- **Donors**
  - Register and verify via email.
  - Login into the system.
  - View and contribute to active campaigns using static QR codes.
  - Track donation history and transparency reports.
  - Track operation phase status of campaigns.
  - View proof posts (images/videos) from campaigns.
- **Fundraisers**
  - Login into the system.
  - Register a new organization (name, domain of activity, contact info, legal representative details, etc.) for admin approval.
  - Manage internal Human Resources: Approve, remove, and oversee **Kitchen Staff** and **Delivery Staff** members.
  - After approval, create and manage fundraising campaigns.
  - Create fundraising campaigns with defined phases (Fundraising, Purchasing, Cooking, Distribution), budget plans and meal batch plan.
  - Receive disbursed funds from the Admin (based on staff requests) and allocate them to the appropriate operational phase.
  - Post updates (images, videos) showing campaign results.
  - Assign tasks to staff.
- **Kitchen staff**
  - Login into the system.
  - Join their organization via approval flow.
  - Execute ingredient purchases strictly according to the approved budget.

    - Upload video evidence (priority) and digital images of receipts/bills immediately after purchase to verify ingredient quality and cost.
    - Initiate a request for the "Cooking Phase" disbursement.
    - Execute the cooking process utilizing 100% of the purchased ingredients.
    - Input the exact quantity of meals produced (e.g., 500 portions) to reconcile against the ingredient input.
    - Upload video/image proof of the final prepared meals before they are handed over to the delivery team.
- **Delivery staff**
    - Login into the system.
    - Join their organization via approval flow.
    - View assigned delivery routes and quantity of meals to be transported.
    - Confirm/reject assigned delivery batches.
    - Mark delivery batches as "In Transit" and subsequently "Completed" to update the campaign status.
- **Administrators**
    - Approve or reject organization registration requests.
    - Review and approve disbursement requests initiated by fundraisers.
    - Execute fund transfers directly to the Fundraiser's bank account upon approval of staff requests.
    - Oversee fund inflows/outflows and campaign statuses.
    - Manage user accounts.
    - Manage campaign categories.

6.2 Limitations & Exclusions

6.2.1 Limitations

- Relies on third-party payment gateways (e.g., PayOS); service interruptions may affect donation records.
- Does not support offline (cash) donations.
- Fund disbursement follows manual approval — no automatic transfer.

6.2.2 Exclusions

- No integration with external logistics (Grab, Ahamove).
- No advanced financial reconciliation.
- No advanced gamification beyond basic badges.

II. Project Management Plan

1. Overview

1.1 Scope & Estimation

#	WBS Item	Complexity	Est. Effort (man-days)
1	**Project Initialization**		**33**
1.1	Define project scope	Medium	8
1.2	Define project requirement and objectives	Complex	12
1.3	Research existing solution	Medium	4
1.4	Create Project Introduction document	Simple	4
1.5	Review project with mentor	Simple	5
2	**Project Planning**		**45**
2.1	Create Project Management Plan document	Complex	8
2.2	Schedule project meeting	Simple	16
2.3	Set up project repository	Complex	21
3	**Project Execution**		**400**
3.1	**Requirement Analysis Phase**		**25**
3.1.1	Analysis requirements	Complex	8
3.1.2	Create Software Requirement Specification	Complex	12
3.1.3	Review project requirement with mentor	Simple	5
3.2	**Design Phase**		**59**
3.2.1	Create Software Design Document	Complex	28
3.2.2	Create UI/UX Design Prototype	Complex	14
3.2.3	Review software design with mentor	Medium	3
3.2.4	Database design	Complex	14
3.3	**Implement Phase**		**319**
3.3.1	**Authentication**		**5**
3.3.1.1	Sign In	Simple	1
3.3.1.2	Sign out	Simple	1
3.3.1.3	Forgot Password	Simple	1
3.3.1.4	Refresh Token	Simple	1

#	WBS Item	Complexity	Est. Effort (man-days)
3.3.1.5	Sign Up	Simple	1
3.3.2	**User management**		**5**
3.3.2.1	Admin view list of users	Simple	1
3.3.2.2	Admin update user's information	Simple	1
3.3.2.3	User update profile	Simple	1
3.3.2.4	Donor/Fundraiser/Kitchen Staff/Delivery Staff change password	Simple	1
3.3.2.5	User view profile	Simple	1
3.3.3	**Organization members actions**		**6**
3.3.3.1	Donors request join organization	Simple	1
3.3.3.2	Donors view join request	Simple	1
3.3.3.3	Donors cancel join request	Simple	1
3.3.3.4	Guests/Donors/Staffs view list of active organizations	Simple	1
3.3.3.5	Staffs leave organization	Simple	1
3.3.3.6	Guests/Donors/Staffs view organization information	Simple	1
3.3.4	**Organization management**		**7**
3.3.4.1	Fundraisers get organization joining requests	Simple	1
3.3.4.2	Fundraisers approve/reject donor's joining request	Simple	1
3.3.4.3	Fundraisers remove staff out of organization	Simple	1
3.3.4.4	Admin handle organization creation requests	Medium	3
3.3.4.5	Admin update organization information	Simple	1
3.3.5	**Campaign category management**		**12**
3.3.5.1	User view list campaign categories	Simple	2
3.3.5.2	Admin create a new campaign category	Simple	2
3.3.5.3	Admin update a campaign category	Simple	2
3.3.5.4	Admin delete a campaign category	Simple	2
3.3.5.5	Admin get category statistics	Medium	4
3.3.6	**Campaign management**		**54**

#	WBS Item	Complexity	Est. Effort (man-days)
3.3.6.1	User view list campaigns	Medium	6
3.3.6.2	User view campaign details	Complex	9
3.3.6.3	Fundraisers view their campaigns	Simple	2
3.3.6.4	Fundraisers send a campaign creation request	Complex	6
3.3.6.5	Fundraisers update their campaigns	Complex	6
3.3.6.6	Fundraisers delete their campaigns	Simple	2
3.3.6.7	Admin approves or rejects the campaign creation request.	Simple	2
3.3.6.8	Fundraisers extend days for campaign fundraising end date.	Simple	2
3.3.6.9	Admin get campaign statistics	Complex	6
3.3.6.10	Fundraiser view their campaign statistics	Complex	6
3.3.6.11	Fundraiser update campaign phases	Medium	4
3.3.6.12	System automatically handle activate approved campaigns that have reached their start date	Simple	1
3.3.6.13	System automatically handle active campaigns that have reached end date or target amount	Simple	1
3.3.6.14	Fundraiser mark campaign is completed	Simple	1
3.3.7	**Campaign post management**		**39**
3.3.7.1	User view all posts of campaign	Medium	6
3.3.7.2	Fundraisers create post	Simple	2
3.3.7.3	Fundraisers update their posts	Simple	2
3.3.7.4	Fundraisers delete their posts	Simple	2
3.3.7.5	User view post likers	Simple	3
3.3.7.6	User view post comments	Medium	6
3.3.7.7	User create comment of posts	Simple	3
3.3.7.8	User update comment of posts	Simple	3
3.3.7.9	User delete comment of posts	Simple	3
3.3.7.10	User reply comment of posts	Simple	3
3.3.7.11	User like/unlike posts	Simple	6
3.3.8	**Donation actions**		**29**

#	WBS Item	Complexity	Est. Effort (man-days)
3.3.6.1	Donors view personal donation history	Medium	3
3.3.6.2	Guests/Donors donate to campaigns	Complex	14
3.3.6.3	Guests/Donors view campaign's donation statement	Medium	4
3.3.6.4	Admin view list donation transactions	Simple	2
3.3.6.5	Fundraiser view dashboard and statistics	Medium	6
3.3.9	**Wallet Management**		**18**
3.3.9.1	Admin view system's wallet and transaction	Medium	6
3.3.9.2	Fundraisers view their wallets and transactions	Medium	6
3.3.9.3	Donors/Guests view system's wallet and transaction	Medium	2
3.3.9.4	Donors/Guest view Fundraiser's wallet and transaction	Medium	2
3.3.9.5	Admin view all Fundraiser's wallets and transaction	Medium	2
3.3.10	**Disbursement Management**		**71**
3.3.10.1	Fundraiser request ingredient disbursement	Medium	6
3.3.10.2	Fundraiser request operation disbursement	Medium	4
3.3.10.3	Admin create disbursement	Medium	3
3.3.10.4	Fundraiser confirm disbursement	Medium	3
3.3.10.5	User view ingredient disbursement request of campaign phase	Medium	4
3.3.10.6	User view detail ingredient disbursements request	Complex	6
3.3.10.7	User view their own ingredient disbursement requests	Medium	3
3.3.10.8	Admin approve/reject ingredient disbursement request	Simple	2
3.3.10.9	Kitchen staff upload expense proofs	Simple	2
3.3.10.10	User view detail expense proof	Medium	6
3.3.10.11	User view expense proof of campaigns	Simple	3
3.3.10.12	Kitchen staff view their own expense proofs	Medium	4
3.3.10.13	Admin view expense proof statistic in system	Simple	2
3.3.10.14	Admin approve/reject expense proof	Simple	3

#	WBS Item	Complexity	Est. Effort (man-days)
3.3.10.15	Kitchen/Delivery staff create cooking/delivery disbursement request	Simple	2
3.3.10.16	User view cooking/delivery disbursement request	Medium	6
3.3.10.17	User view detail cooking/delivery disbursement request	Medium	6
3.3.10.17	Kitchen/Delivery staff view their own cooking/delivery request	Simple	2
3.3.10.18	Admin approve/reject cooking/delivery disbursement request	Simple	2
3.3.10.19	Admin view cooking/delivery request statistic	Simple	2
3.3.11	**Badge Management**		**15**
3.3.11.1	Donor receive badge	Medium	4
3.3.11.2	Admin create badge	Medium	3
3.3.11.3	Admin update badge	Medium	3
3.3.11.4	Admin view all badges and details	Medium	3
3.3.11.5	User view all active badges and details	Medium	2
3.3.12	**Meal Batch Management**		**18**
3.3.12.1	Kitchen staff upload meal batch media	Medium	4
3.3.12.2	Kitchen staff update meal batch status	Simple	2
3.3.12.3	User view detail meal batch	Medium	6
3.3.12.4	User view meal batch list of campaign	Medium	6
3.3.13	**Delivery Task Management**		**15**
3.3.13.1	Fundraisers assign task to staff	Simple	2
3.3.13.2	Delivery staff approve/reject delivery task	Simple	2
3.3.13.3	User view detail delivery task	Medium	6
3.3.13.4	User view delivery task list of campaign	Simple	3
3.3.13.5	Delivery staff view their own tasks	Simple	2
3.3.14	**Notification**		**15**
3.3.14.1	User view their notification	Medium	6
3.3.14.2	User view unread notification count	Simple	3
3.3.14.3	User mark all notifications as read	Simple	3

#	WBS Item	Complexity	Est. Effort (man-days)
3.3.14.4	User delete their notification	Simple	3
3.3.15	**Campaign Reassignment**		**11**
3.3.15.1	Admin view eligible organization to take canceled campaign	Simple	2
3.3.15.2	Admin assign campaign to organizations	Simple	2
3.3.15.3	Fundraiser view assignment requests	Medium	4
3.3.15.4	Fundraiser approve/reject assignment request	Simple	2
3.3.15.5	System automatically handle expired campaign reassignments	Simple	1
3.4	**Testing Phrase**		**42**
3.4.1	Write Software Test document	Complex	14
3.4.2	Write, execute and write unit test's documents	Complex	14
3.4.3	Perform and write system test's documents	Complex	14
3.5	**Deployment Phrase**		**9**
3.5.1	Deploy to server	Complex	9
4	**Project Monitoring & Controlling**		**12**
4.1	Monitor and control risk	Complex	12
5	**Project Closing**		**16**
5.1	Write Software User Guide's document	Complex	8
5.2	Write Final Project Report	Complex	8
		Total Estimated Effort (man-days)	***504***

Table 3. Scope and Estimation

1.2 Project Objectives

#	Testing Stage	Test Coverage	No. of Defects	% of Defect	Notes
1	Reviewing	100% of documents	0	0	Review of all requirements, design, and code documents.
3	Integration Test	>= 90% of module integration	0	0	
4	System Test	100% of critical flow	0	0	

Table 4. Project Objectives

- **Milestone Timeliness (%): The project must be finished before (24 December 2025)**
- **Allocated Effort (man-days): 4 (members) * 16(weeks) * 7(days/weeks) = 448 man-days**

1.3 Project Risks

#	Risk Description	Impact	Possibility	Response Plans
1	Conflicts between team members	Medium	Medium	Organize team meetings to address and resolve conflicts
2	Solution does not meet real-world requirements	High	Medium	Extensive business requirements analysis, researching existing solution
3	Limited technical knowledge of relevant frameworks.	Medium	Low	Allocate time for technical training
4	Data security vulnerabilities	Medium	Medium	Data security vulnerabilities
5	Task deadline is too tight	Medium	Medium	Prioritize key tasks, delegate when possible, and extend the deadline.

Table 5. Project Risks

2. Management Approach

2.1 Project Process

Our software development lifecycle methodology is Agile, which is an iterative and flexible framework for developing, delivering, and maintaining products. Agile emphasizes adaptability, collaboration, and continuous delivery of value to ensure that the product can evolve alongside changing requirements and user needs. The reasons for adopting Agile in our project include:

- **Iteration planning**: This allows the team to break down the project into smaller, manageable tasks, prioritize them effectively, and set clear goals for each sprint. It helps ensure that the most important features are always being worked on first.
- **Daily Stand-ups**: These short meetings keep everyone aligned, encourage collaboration, and help quickly identify any blockers. By maintaining constant communication, the team can adapt and resolve issues before they grow into bigger problems.
- **Sprint reviews and demo**: At the end of each sprint, the team showcases the completed work to stakeholders. This not only provides transparency but also invites early feedback, making it easier to adjust direction if needed and ensuring the product evolves in line with expectations.

This iterative approach enables rapid response to changes and continuous improvement while keeping the project on track.

2.2 Quality Management

The following quality management measures will be applied to ensure that FoodFund fulfills its quality objectives:

- Defect Prevention:
    - Focus on applying coding best practices and conducting peer reviews to identify potential issues at an early stage. Maintaining clear documentation along with regular knowledge-sharing sessions will further help prevent defects.
- Reviewing:
    - Perform both formal and informal reviews of code, design, and requirements throughout the development process. In addition, holding regular sprint retrospectives enables the team to improve practices and proactively address areas needing enhancement.
- Unit Testing:
    - Unit testing is carried out continuously during development, especially under practices like Test-Driven Development (TDD) or Continuous Integration (CI). Developers create tests alongside their code to verify that each feature functions properly from the beginning.
    - Integration testing is checked whether various parts of the system interact and function correctly together. It usually takes place early in the development process, right after unit testing, and continues as new features are added and integrated into the system.
- System Testing:
    - System testing is performed at the end of each sprint to verify the complete functionality of the system in a real-world setting. Its goal is to ensure that both the new features and the overall system meet user requirements through end-to-end testing that mirrors real usage scenarios.

2.3 Training Plan

Training Area	Participants	When, Duration	Waiver Criteria
NextJS	HoangHLN	23/08/2025	Mandatory
React Native	DuyPCB	23/08/2025	Mandatory
NestJS	PhuocTM, LuyenHD	23/08/2025	Mandatory
AWS, Digital Ocean	PhuocTM	08/09/2025	Mandatory
PayOS	PhuocTM	08/10/2025	Mandatory

Table 6. Training Plan

3. Project Deliverables

#	Deliverable	Due Date	Notes
1	Project Introduction (Report 1)	02/10/2025	Project Introduction Document
2	Project Management Plan (Report 2)	06/10/2025	Project Management Plan Document
3	Software Requirement Specification (Report 3)	21/10/2025	Context Diagram, Use case diagram, ...
4	Software Design Documentation (Report 4)	10/11/2025	System architecture, class diagram, sequence diagram, ….
5	Software Testing Documentation (Report 5)	19/11/2025	Test strategy, test plan, test case, ...
6	Software User Guides (Report 6)	20/11/2025	Installation guides, user manual guides, ...
7	Final Report (Report 7)	7/12/2025	Include all project documentations
8	Code Package Back-end	10/12/2025	Code & Unit Test, Integration Test
9	Code Package Front-end	13/12/2025	Code
10	Code Package Mobile	10/12/2025	Code

Table 7. Project Deliverables

4. Responsibility Assignments

D~Do; R~Review; S~Support; I~Informed; <blank>- Omitted

Responsibility	PhuocTM	LuyenHD	DuyPCB	HoangHLN
Project Introduction (Report 1)	R	D	S	S
Project Management Plan (Report 2)	D	S	R	R
Software Requirement Specification (Report 3)	R	D	S	S
Software Design Documentation (Report 4)	D	D	S	S
Software Testing Documentation (Report 5)	R	D	S	S
Software User Guides (Report 6)	R	S	S	D
Final Report (Report 7)	D	D	D	D
Code Package Back-end	D	D	I	I
Code Package Front-end	I	I	I	D
Code Package Mobile	I	I	D	I

Table 8. Responsibility Assignments

5. Project Communications

Communication Item	Who/ Target	Purpose	When, Frequency	Type, Tool, Method(s)
Supervisor (Lecturer)	Mr. Nguyen Minh Sang	- Provide document template - Instruct the project team - Review deliverables - Supervise project status - Receive report project	Weekly	Zalo
Meeting	Team members	- Collaborate on mutual task - Planning	Weekly	Google Meets, Discord, Zalo
Messages	Team members	- Share daily progress made on project tasks - Share current and potential roadblocks	Daily	Zalo, Discord

Table 9. Project Communications

6. Configuration Management

6.1 Document Management

- Document tools: Google Docs, Google Sheets
- File Management: Google Drive

6.2 Source Code Management

- Our team uses GitHub to manage the source code version

6.3 Tools & Infrastructures

Category	Tools / Infrastructure
Technology	NextJS (Frontend), React Native (Mobile app), NestJS (Backend)
Database	PostgreSQL, OpenSearch, Valkey
IDEs/Editors	Visual Studio Code, VSCodium
Diagramming	DrawIO, Visual Paradigm, Mermaid Chart
Documentation	Google Docs/Sheets/Slides
Version Control	GitHub (Source Codes), Google Drive (Documents)
Deployment server	Digital Ocean Kubernetes, Vercel
Project management	Google Sheet

Table 10. Tools & Infrastructure

III. Software Requirement Specification

1. Product Overview

FoodFund is a web and mobile-based crowdfunding and operations management platform designed specifically for charitable food preparation and distribution campaigns. The system centralizes donation collection, transparent fund disbursement, ingredient management, kitchen workflows, and delivery tracking into a single integrated solution.

Currently, food charity initiatives in Vietnam are often managed manually or via fragmented social media groups, leading to inefficiencies, lack of transparency, and low donor trust. FoodFund addresses these challenges by digitizing and standardizing the entire workflow of a fundraising campaign—from receiving donations to preparing and delivering meals.

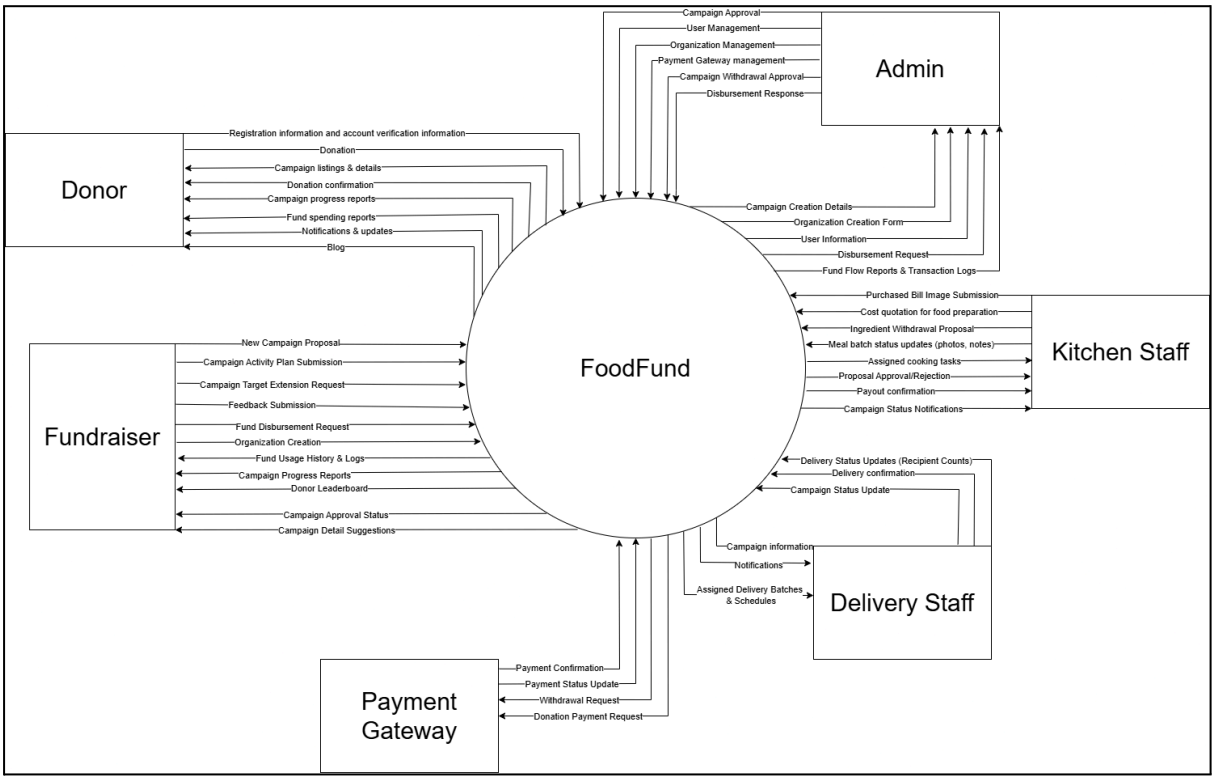


Figure 1. FoodFund Context Diagram

2. User Requirements

2.1 Actors

#	Actor	Description
1	Admin	A privileged user who oversees the entire platform and has functions to approve organizations and campaigns, authorize fund disbursements, and monitor all transactions to ensure compliance.
2	Fundraiser	A user representing a verified organization with functions to create and manage campaigns, approve team members, and submit disbursement requests to access funds.

#	Actor	Description
3	Donor	A registered user who provides financial support by donating to fundraising campaigns and has functions to track personal donation history and view campaign spending reports.
4	Kitchen Staff	A member of an organization responsible for meal preparation, with functions to view cooking tasks, upload bills for ingredient purchases, and update the status of food batches.
5	Delivery Staff	A member of an organization who handles the distribution of prepared meals, with primary functions to view assigned delivery batches and update their delivery status.
6	Guest	A person who hasn't registered an account and has basic functions like browsing and viewing the details of active fundraising campaigns.

Table 11. Actors

2.2 Use Case

2.2.1 Use Case Diagram(s)

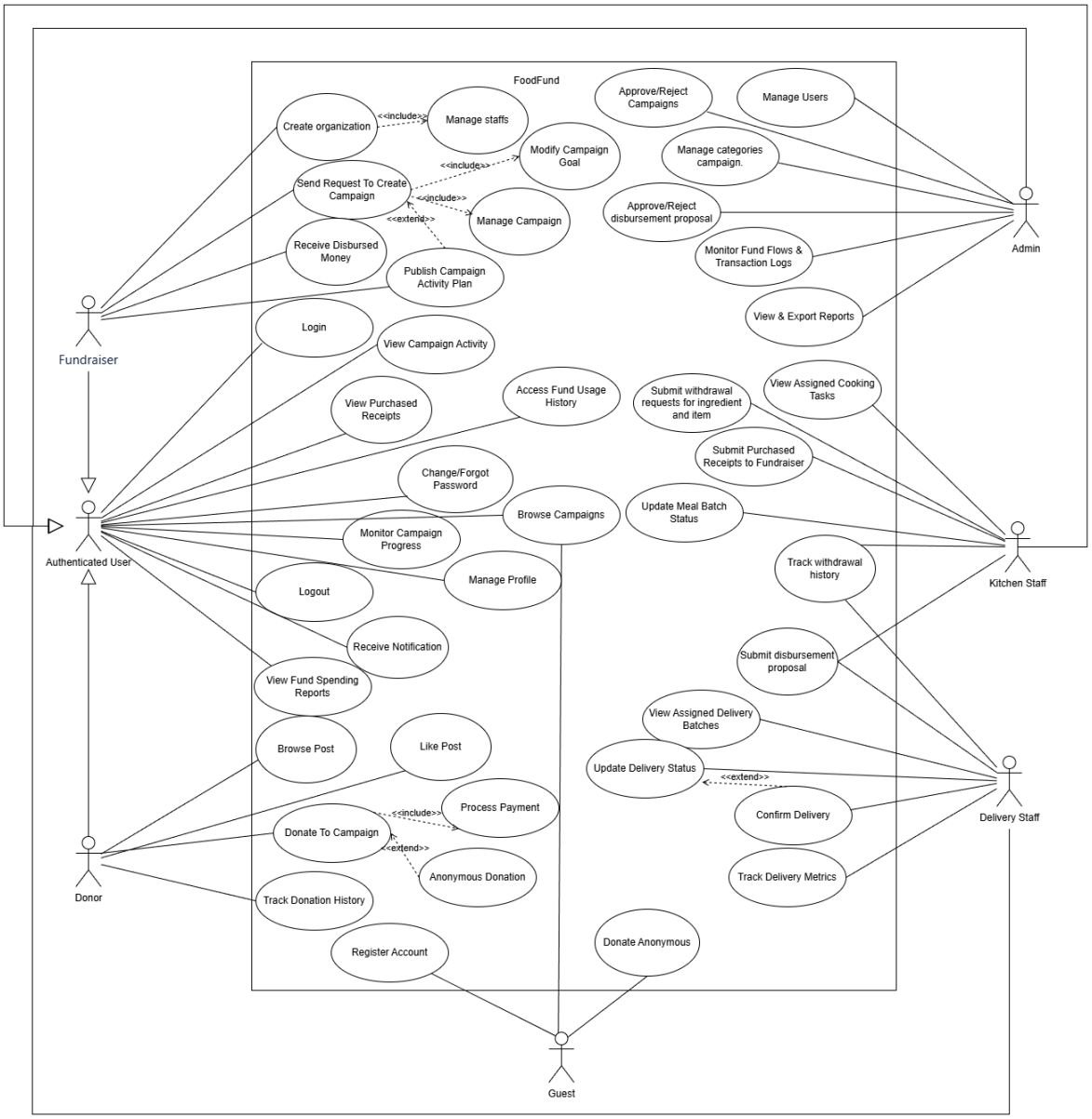


Figure 2. Use Case Diagram

2.2.2 Descriptions

ID	Use Case	Actors	Use Case Description
01	Register Account	Guest	Guest sign up for account
02	Log in	Authenticated User	Unauthenticated users login to PoemTown to verify roles, and give them access to more features in the system.

ID	Use Case	Actors	Use Case Description
03	Log out	Authenticated User	This feature allows all users to logout the system.
04	Browse Campaign	Authenticated User, Guest	Allows users to view and explore the list of active fundraising campaigns available on the platform.
05	Donate to Campaign	Donor	Allows users to view and explore the list of active fundraising campaigns available on the platform.
06	Donate as Anonymous	Guest, Donor	Allows a user, whether a Guest or a signed-in Donor, to make a donation without having their name publicly displayed on the campaign's donor list.
07	Browse Post	Guest	Allows users to read activity updates, stories, and other posts published by campaign Fundraisers.
08	Like Post	Donor	Enables a Donor to interact with a campaign post by giving it a "like".
09	View Campaign Activity	Authenticated User	Enables users to see a feed of real-time updates, such as kitchen progress or delivery status, related to a campaign.
10	Access Fund Usage History	Authenticated User	Allows Donors and Fundraisers to track a campaign's spending and view how funds are allocated to ensure transparency.
11	View Donation History	Donor	Enables a Donor to view a complete record of their past personal donations and transaction details.
12	Request to Create Organization	Fundraiser	Enables a user to submit a registration form to the Admin to become a verified Fundraiser organization on the platform.
13	Send Organization Join Request	Kitchen Staff, Delivery Staff	Allows users (Kitchen or Delivery Staff) to find a verified organization and submit a request to join its team.
14	Create Campaign	Fundraiser	Allows an approved Fundraiser to design, create, and submit a new fundraising campaign proposal for Admin approval.
15	Create Post	Fundraiser	Enables a Fundraiser to write and publish posts with photos to update donors on the campaign's progress and activities.
16	Submit Disbursement Proposal	Fundraiser	Allows a Fundraiser to formally request the release of funds from the campaign, attaching supporting documents for Admin review.

ID	Use Case	Actors	Use Case Description
17	Modify Campaign Goal	Fundraiser	Allows a Fundraiser to update the details of an active campaign, such as changing its description or extending its duration.
18	View Purchased Receipt	Authenticated User	Allows authenticated user to review the bills, receipts, and photos uploaded by Kitchen Staff as proof of purchase.
19	Ban/Unban User	Admin	Enables an Admin to restrict or block a user's account, preventing them from accessing the platform, as part of user management.
20	Change User Role	Admin	Allows an Admin to modify a user's permissions and role within the system as part of user management.
21	Approve/Reject Organization Creation	Admin	Enables an Admin to review a submitted organization registration form and either approve or reject the request.
22	Approve/Reject Campaign Creation	Admin	Enables an Admin to review a newly submitted campaign proposal and either approve it to go live or reject it.
23	Manage Campaign Categories	Admin	Allows an Admin to create, edit, and delete the categories used to classify and organize campaigns on the platform.
24	Approve/Reject Disbursement Proposal	Admin	Enables an Admin to review a Fundraiser's request for funds, check the evidence, and approve or reject the release of money.
25	Monitor Fund Flows	Admin	Allows an Admin to oversee all financial transactions and fund movements within the entire system, or a Fundraiser to monitor funds for their specific campaign.
26	View Assigned Cooking Task	Kitchen Staff	Allows a Kitchen Staff member (or their Fundraiser) to see the specific cooking tasks and schedules assigned to them for a campaign.
27	Submit Withdrawal Request For Ingredients and Items	Kitchen Staff	Enables Kitchen Staff to request funds from their Fundraiser specifically for purchasing ingredients and necessary kitchen supplies.
28	Submit Purchased Receipt	Kitchen Staff	Allows Kitchen Staff to upload photos of bills and receipts as evidence after purchasing ingredients for their assigned tasks.
29	Update Meal Batch Status	Kitchen Staff	Enables Kitchen Staff to update the progress of a batch of food (e.g.,

ID	Use Case	Actors	Use Case Description
			"Cooking," "Ready for Pickup") and attach proof.
30	Track Withdrawal History	Kitchen Staff, Delivery Staff	Allows Kitchen Staff to view the status (e.g., pending, approved) of their past fund requests and payouts.
31	View Assigned Deliveries	Delivery Staff	Enables a Delivery Staff member to see the specific list of delivery batches they are assigned to pick up and distribute.
32	Update Delivery Status	Delivery Staff	Allows a Delivery Staff member to update the real-time status of their delivery (e.g., "In Transit," "Delivered").
33	Track Delivery Metrics	Delivery Staff	Enables a Delivery Staff member to view their personal statistics, such as the total number of deliveries completed and their delivery history.
34	Receive Notification	Authenticated User	Allows users to receive notification from the system.
35	Manage Profile	Authenticated User	Allows authenticated users to change their profile.
36	Change/Forgot Password	Authenticated User	Allows authenticated users to change or get new passwords.
37	Receive Disburse Money	Fundraiser	Fundraiser note to system that they have receive the disbursement money.
38	View Campaign Activity	Authenticated User	Allows authenticated users to view campaign activity and timelines.

Table 12. Use Case Diagram Descriptions

3. Functional Requirements

3.1 System Functional Overview

3.1.1 Screens Flow

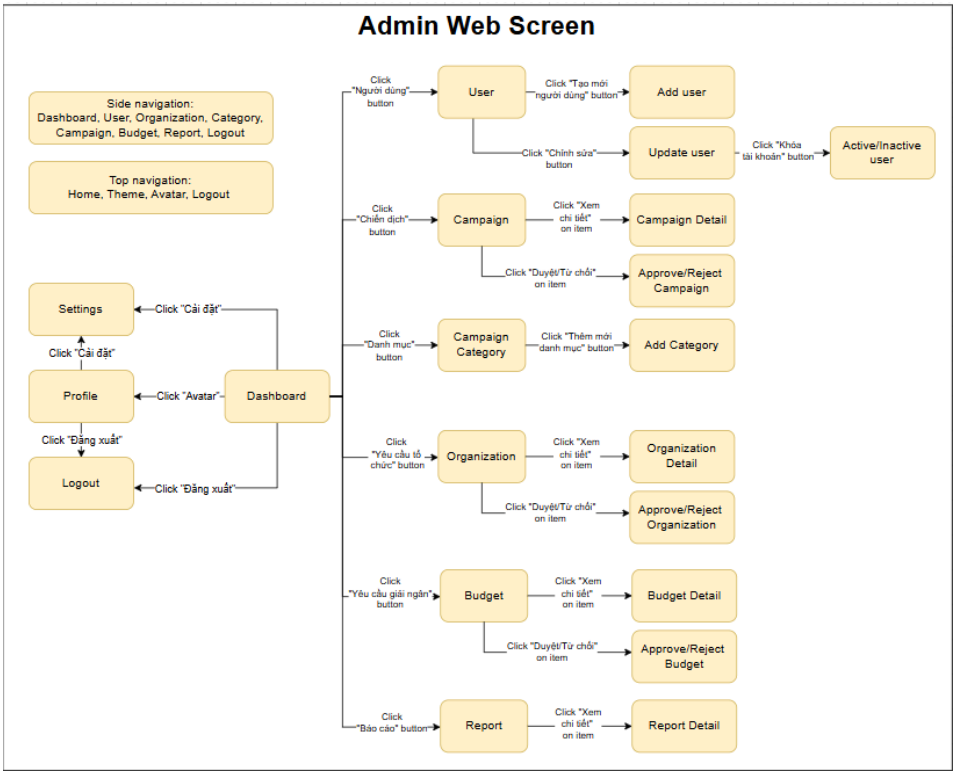


Figure 3. Admin Web Screen Flow

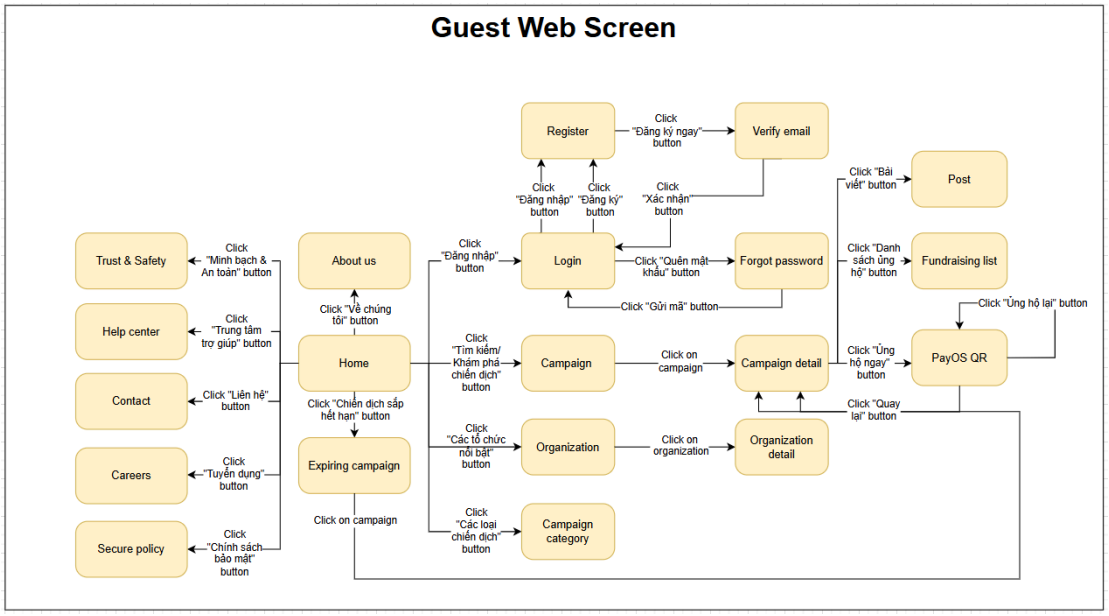


Figure 4. Guest Web Screen Flow

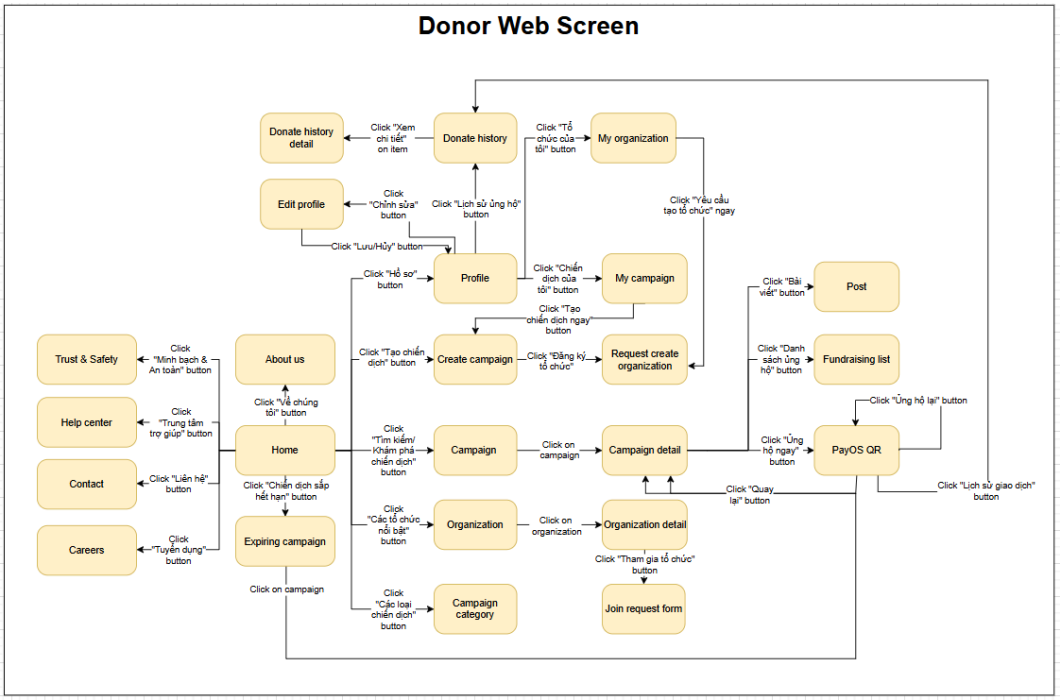


Figure 5. Donor Web Screen Flow

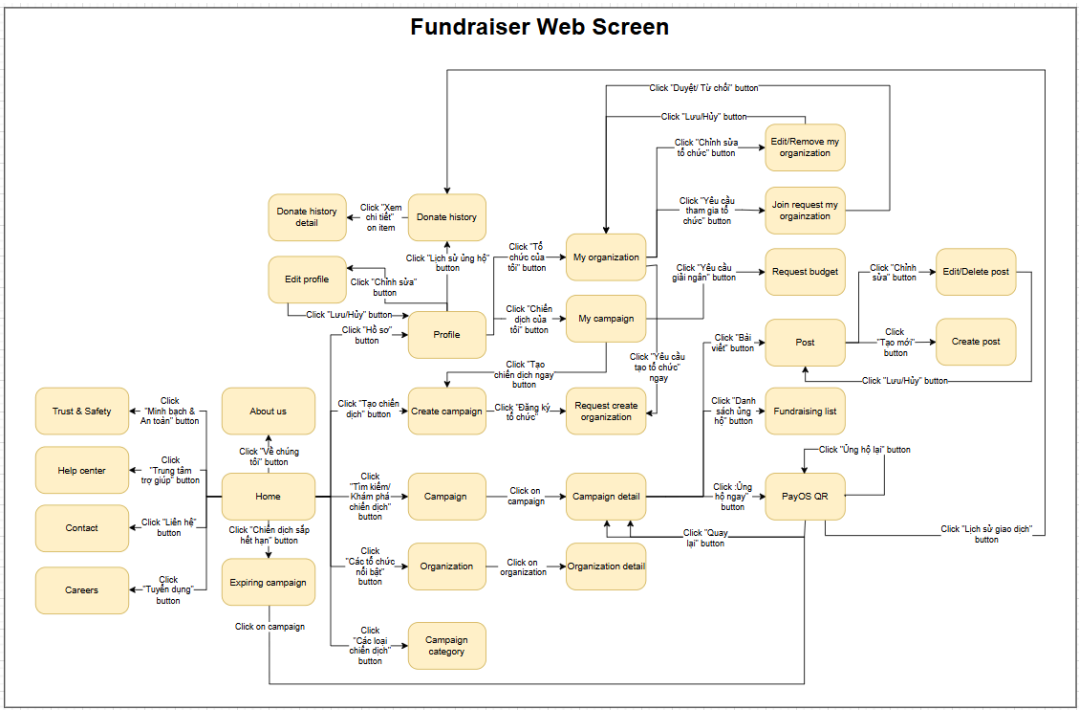


Figure 6. Fundraiser Web Screen Flow

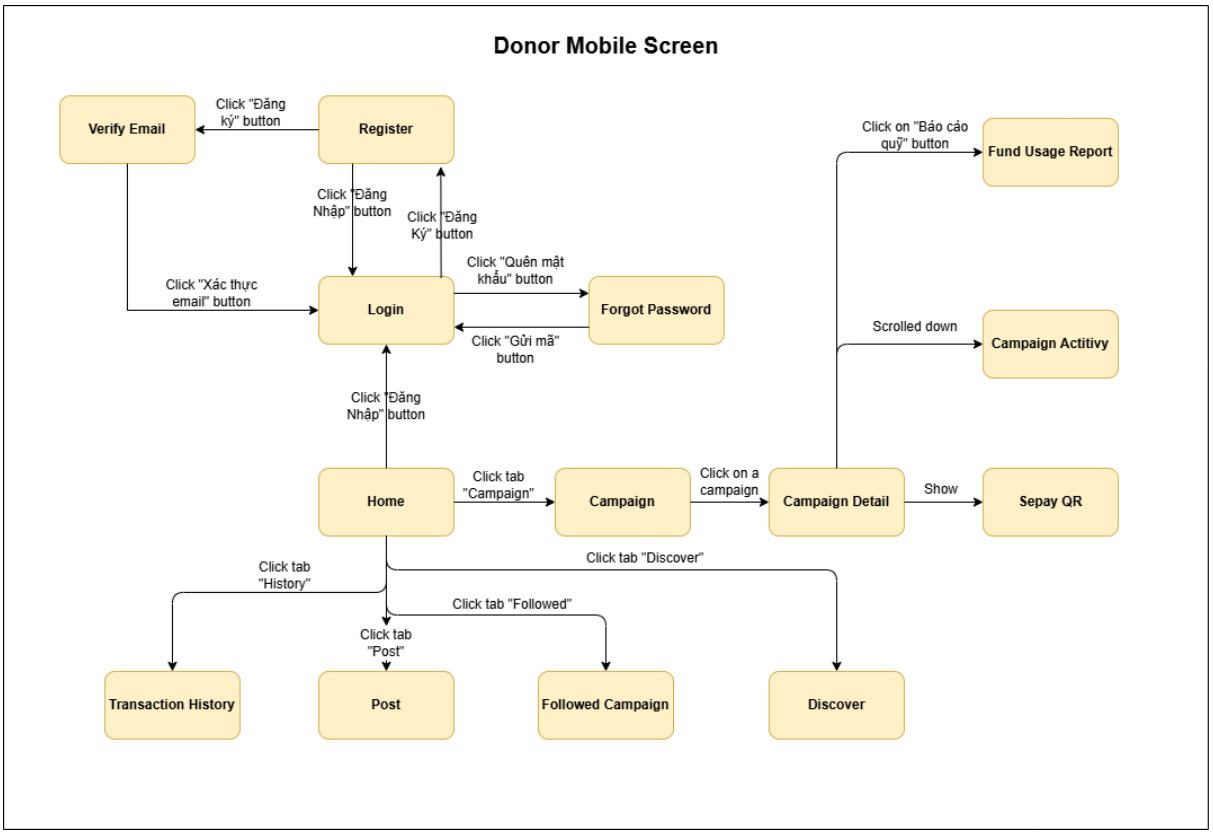


Figure 7. Donor Mobile Screen Flow

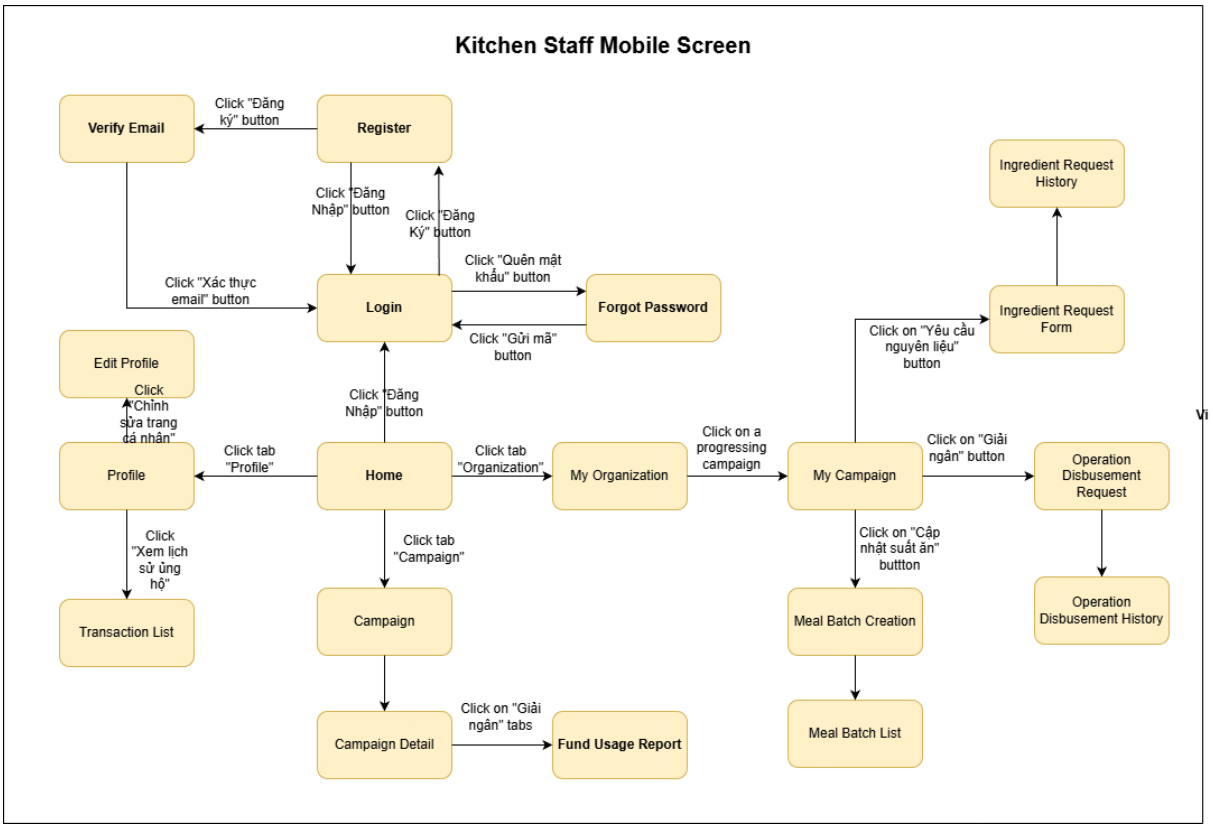


Figure 8. Kitchen Staff Mobile Screen Flow

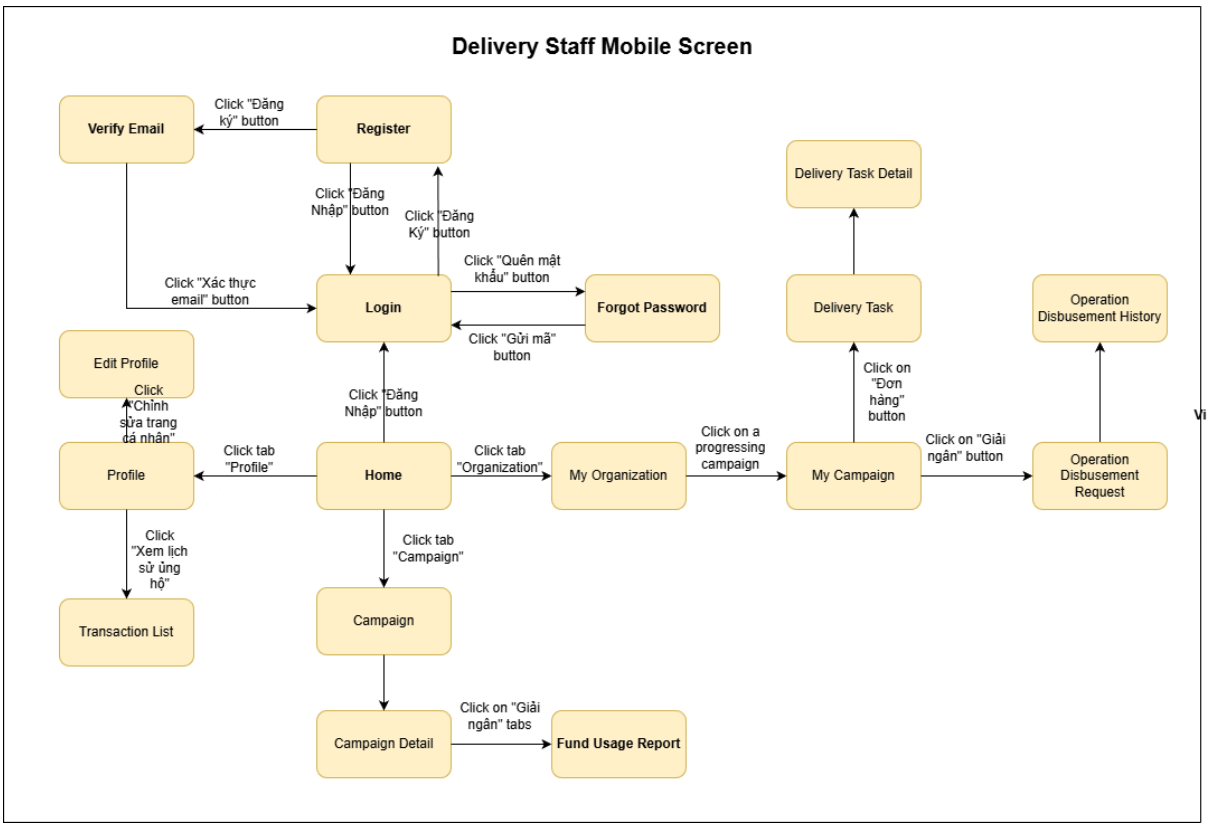


Figure 9. Delivery Staff Mobile Screen Flow

3.1.2 Screen Descriptions

#	Feature	Screen	Description
1	Login	Login	Screen allows all registered users to login to the system with their account and password.
2	Logout	Profile, Home	Screen in home and profile section allow users to logout of the system.
3	Register	Register	Screen allows guests to enter their information to create their own account as a donor role.
4	OTP Input	OTP Input	Screen allows guests to input OTP code to activate their registered account.
5	Forgot password	Forgot password	Screen allows registered users to input their account email to receive their new account password.
6	View about us	About us	Screen allows all users about our mission, vision, and the people behind FoodFund.
7	View help center	Help center	Screen allows all users guides, FAQs, and support resources.
8	View contact	Contact	Screen allows all users contact information and ways to reach the team.
9	View careers	Careers	Screen allows all users to explore job opportunities and apply to work with us.
10	View secure policy	Secure policy	Screen allows all users how we protect user data and secure the platform.

#	Feature	Screen	Description
11	View campaigns list	Campaign	Screen allows all users to view all general information of campaigns.
12	View campaign detail	Campaign detail	Screen allows all users to view all general information of campaigns.
13	View expiring campaigns	Expiring campaigns	Screen allows all users to view all campaigns that are about to end.
14	View organization featured	Organization featured	Screen allows all users to view a list of featured and verified organizations
15	View organization details	Organization details	Screen allows all users to view detailed information about the selected organization including organization info, contact details, or join option,...
16	View campaign posts	Campaign posts	Screen allows all users to view all posts and updates related to the campaign.
17	View donor list of campaign	Donor list	Screen allows all users to view the list of users who supported or donated to the campaign.
18	View campaign categories	Campaign categories	Screen allows all users to view all campaign categories for users to explore.
19	Donate	PayOs QR	Screen allows all users to make quick and secure payments via PayOS QR.
20	Request join organization	Join organization	Screen allows donors to submit a request to join the organization.
21	Request create organization	Create organization	Screen allows donors to submit organization information and verify details.
22	View profile	Profile	Screen allows registered users to view and manage profile details.
23	Edit profile	Edit profile	Screen allows registered users to edit and update their personal information.
24	View donation history	Donation history	Screen allows all users to track and review contribution history.
25	View donation history details	Donation history details	Screen allows all users to view amount, date, and campaign details for each donation.
26	Request create campaign	Create campaign request	Screen allows all fundraisers to submit a campaign creation request for approval.
27	View my campaigns	My campaigns	Screen allows all fundraisers to manage personal campaigns and viewing their status.
28	Edit my campaign	Edit my campaign	Screen allows all fundraisers to edit and update campaign information.
29	Delete my campaign	Delete my campaign	Screen allows all fundraisers to confirm and delete their campaign permanently if status of campaign is pending.
30	Edit organization	Update organization	Screen allows all fundraisers to manage organization details and contact information.
31	Handle join request my organization	Member approval	Screen allows all fundraisers to review and approve or reject membership requests.
32	Delete my organization	My organization	Screen allows all fundraisers to confirm and permanently delete the organization.

#	Feature	Screen	Description
33	Request fund disbursement	My campaign	Screen allows all fundraisers to request fund disbursement after the campaign's funding date ends.
34	Create post	Create post	Screen allows all fundraisers to write content, upload media, and publish posts.
35	Edit post	Edit post	Screen allows all fundraisers to edit and update an existing post.
35	Delete post	Delete post	Screen allows all fundraisers to confirm and delete an existing post.
36	View dashboard	Dashboard admin	Screen allows the admin to view key metrics, statistics, and management shortcuts.
37	Manage user	User management admin	Screen allows the admin to view, manage, and control user accounts.
38	Manage campaign	Campaign management admin	Screen allows the admin to view, manage, and control all campaigns.
39	Approve/Reject campaign	Campaign approval admin	Screen allows the admin to review and approve or reject campaign requests.
40	View disbursement request	Disbursement request admin	Screen allows the admin to view and manage all payout requests submitted by campaign owners.
41	View disbursement request details	Disbursement request detail admin	Screen allows the admin to review full details of a payout request
42	Approve/Reject budget request	Budget approval admin	Screen allows the admin to approve or reject payout requests.
43	Manage category	Category management admin	Screen allows the admin to manage all categories displayed.
44	Create category	Create category admin	Screen allows the admin to create a new campaign category.
45	Edit category	Edit category admin	Screen allows the admin to edit and update category information.
46	Delete category	Delete category admin	Screen allows the admin to confirm and delete a category.
47	View organization approval request	Organization approval request admin	Screen allows the admin to view and manage all organization approval requests.
48	Approve/Reject organization request	Organization approval request admin	Screen allows the admin to review organization details and approve or reject the request.
49	View organization detail	Organization detail admin	Screen allows the admin to view detailed information of an organization.
50	Ban/Unban organization	Organization detail admin	Screen allows the admin to ban or unban the organization directly.
51	Ban/Unban user	User management admin	Screen allows the admin to ban or unban the user directly.
52	View report	Report admin	Screen allows the admin to view and manage all user and campaign reports

#	Feature	Screen	Description
53	View report detail	Report admin	Screen allows the admin to view full details of a report and take actions

Table 13. Screen Descriptions

3.1.3 Screen Authorization

Screen	Donor	Fundraiser	Kitchen Staff	Delivery Staff	Admin
Login	X	X	X	X	X
Logout	X	X	X	X	X
Register	X	X	X	X	
OTP Input	X	X	X	X	
Forgot password	X	X	X	X	X
View about us	X	X	X	X	X
View help center	X	X	X	X	X
View contact	X	X	X	X	X
View careers	X	X	X	X	X
View secure policy	X	X	X	X	X
View campaigns list	X	X	X	X	X
View campaign detail	X	X	X	X	X
View expiring campaigns	X	X	X	X	X
View organization featured	X	X	X	X	X
View organization details		X	X	X	X
View campaign posts	X	X	X	X	X
View donor list of campaign	X	X		X	X
View campaign categories	X	X	X	X	X
Donate	X	X	X	X	X
Request join organization	X		X	X	
Request create organization	X		X	X	
View profile	X	X	X	X	X
Edit profile	X	X	X	X	X
View donation history	X	X	X	X	
View donation history details	X	X	X	X	
Request create campaign	X	X			
View my campaigns		X			
Edit my campaign		X			
Delete my campaign		X			
Edit organization		X			
Handle join request my organization		X			
Delete my organization		X			

Screen	Donor	Fundraiser	Kitchen Staff	Delivery Staff	Admin
Request fund disbursement		X			
Create post		X			
Edit post		X			
Delete post		X			
View dashboard					X
Manage user					X
Manage campaign					X
Approve/Reject campaign					X
View disbursement request					X
View disbursement request details					X
Approve/Reject budget request					X
Manage category					X
Create category					X
Edit category					X
Delete category					X
View organization approval request					X
Approve/Reject organization request					X
View organization detail					X
Ban/Unban organization					X
Ban/Unban user					X
View report					X
View report detail					X

Table 14. Screen Authorization

3.1.4 Non-Screen Functions

#	Feature	System Function	Description
1	Campaign Lifecycle Automation	Automatic Campaign Activation	The system runs a scheduled job to scan all campaigns with the status APPROVED. If the current system date matches or exceeds the defined fundraising start date, the system automatically updates the campaign status from APPROVED to ACTIVE, making it visible for public donations.
2	Campaign Lifecycle Automation	Automatic Campaign Conclusion & Verification	The system runs a scheduled job to evaluate ACTIVE campaigns that have reached their fundraising end date. The system checks the received amount against the fundraising goal: • Success Case: If the received amount is greater than system config percentage of the target goal, the

#	Feature	System Function	Description
			status is automatically updated to PROCESSING to initiate the operational phase. • Failure Case: If the received amount is less than system config percentage of the target goal, the status is automatically updated to ENDED, indicating the campaign did not meet the minimum viability threshold.
3	Service Health Check	/health (REST API Endpoint)	Standard endpoint used for health checks to determine if the service is running and responsive.
4	Donation Payment Processing	PayOS Webhook payment	Receives payment status updates from the PayOS payment gateway to confirm successful or failed donation transactions.
5	Donation Payment Processing	Sepay Webhook payment	Receives payment status updates from the Sepay payment gateway to store donation transactions.
6	Notification System	Asynchronous Notification Dispatch	The system utilizes a message queue architecture to handle high-volume notification delivery. This function operates independently of user screens to ensure that system alerts, donation confirmations, and status updates are delivered reliably without impacting the performance of the user interface.

Table 15. Non-Screen Functions

3.1.5 Entity Relationship Diagram

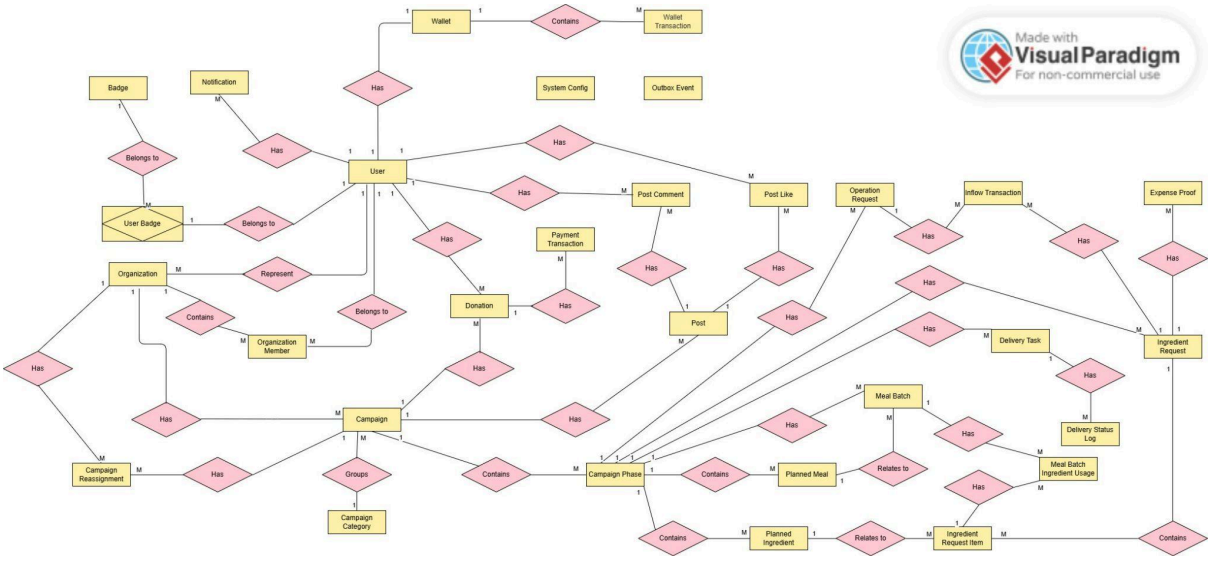


Figure 10. FoodFund Entity Relationship Diagram

Entities Description

#	Entity	Description
1	User	Represents an individual interacting with the system, including donors, fundraisers, kitchen staffs, delivery staffs and admin.
2	Organization	Represents a registered fundraising group responsible for running campaigns.
3	Organization Member	Represents the association between a user and an organization, defining their role and status within it.
4	Wallet	Represents a virtual wallet uniquely associated with a single user. This entity holds the fundraiser/admin personal balance within the platform.
5	Wallet Transaction	Represents a single transaction record against a virtual wallet.
6	Campaign	Represents a structured charitable initiative designed to raise funds and execute specific community support activities. It serves as the central record for the project's lifecycle, tracking financial goals, fundraising progress, operational phases, temporal milestones, and status transitions from creation to final conclusion.
7	Campaign Category	Represents the category a campaign belongs to.
8	Campaign Phase	Represents a distinct operational phase.
9	Campaign Reassignment	Represents a formal administrative mechanism for transferring the ownership and operational responsibility of a campaign from one organization to another.
10	Donation	Represents a single financial contribution made by a donor to a specific campaign.
11	Payment Transaction	Represents the detailed record of a financial transaction associated with a donation via the payment gateway.
12	Post	Represents an update, announcement, or proof-of-work related to a campaign, posted by the fundraiser.
13	Post Like	Represents a user liking a specific campaign post.
14	Post Comment	Represents a user commenting on a specific campaign post, potentially replying to another comment.
15	Ingredient Request	Represents a formal request from fundraiser/kitchen staff for funds needed to purchase ingredients for a campaign phase.
16	Ingredient Request Item	Represents a specific item listed within an ingredient request, detailing quantity and estimated cost.
17	Expense Proof	Represents an evidentiary document (e.g., a receipt image, images/videos ingredient purchased) uploaded to substantiate an expense.
18	Operation Request	Represents a formal request for funds for a specific purpose (e.g., delivery, cooking gas).

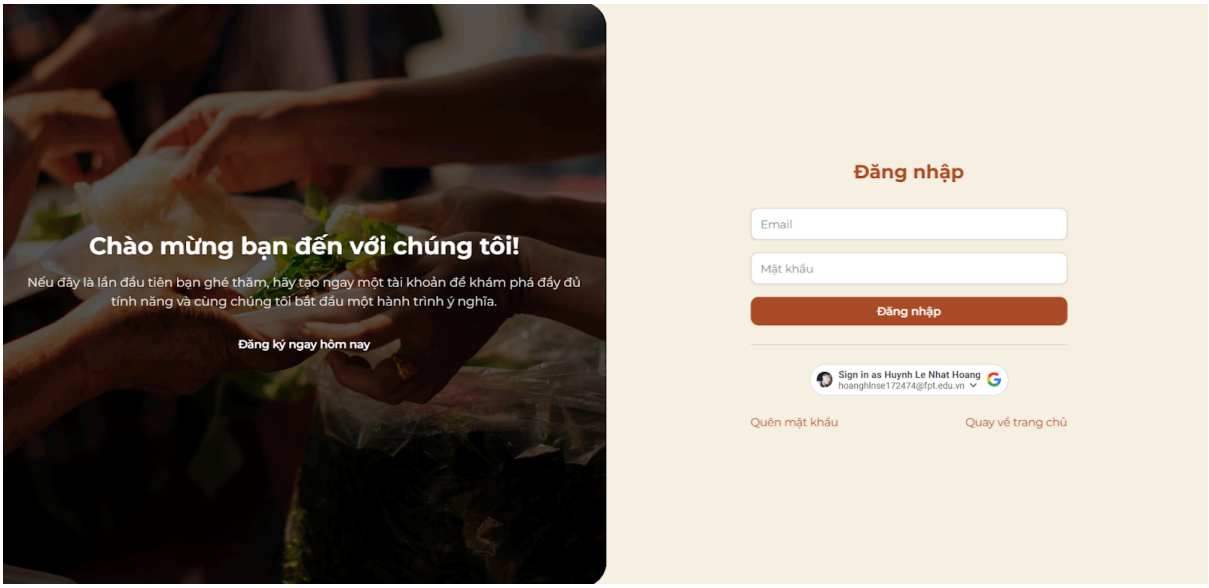
#	Entity	Description
19	Meal Batch	Represents a batch of meals prepared by the kitchen staff for a campaign phase.
20	Delivery Task	Represents an assigned task for delivery staff to deliver a specific meal batch.
21	Delivery Status Log	Represents a historical audit trail of all status changes for a single delivery.
22	Notification	Represents a message sent to a user.
23	Meal Batch Ingredient Usage	Represents the specific ingredients used in the preparation of a particular meal batch. Serves as an audit trail for ingredient consumption.
24	Badge	Represents an achievement awarded to a user (e.g., for donation milestones).
25	User Badge	Represents the association of a specific badge awarded to a specific user.
26	Inflow Transaction	Represents a record of funds moving into operational phases (e.g., disbursement for ingredients, cooking costs, delivery costs) for a campaign. Tracks fund allocation post-disbursement.
27	Outbox Event	Represents events for the Transactional Outbox Pattern, ensuring reliable, asynchronous execution of side effects.
28	System Config	Represents system configuration settings that can be dynamically modified by administrators without code deployment.
29	Planned Ingredient	Represents a specific raw material or component required for the execution of a particular campaign phase, detailing the necessary item name, quantity, and unit of measurement to facilitate accurate procurement and resource allocation.
30	Planned Meal	Represents a distinct food item or meal scheduled for preparation and distribution during a specific campaign phase, defining the target output quantity to ensure operational alignment with the campaign's humanitarian objectives.

Table 16. Entities Descriptions

3.2 Authentication

3.2.1 Sign In

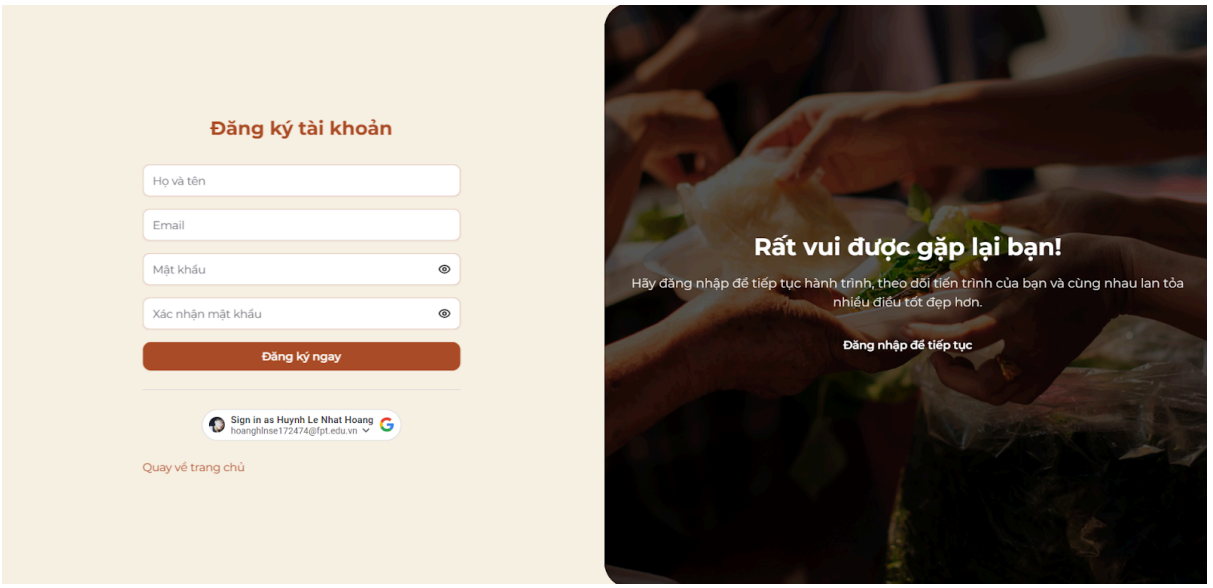
- **Function trigger:** Users click “Đăng nhập” button
- **Function description:**
  - **Actors:** Donor, Fundraiser, Kitchen Staff, Delivery Staff, Admin
  - **Purpose:** Sign in to the system



- **Function details:**
  - **Data requirement:** account’s email, password
  - **Validation:** None
  - **Business rules:** BR-02

3.2.2 Sign Up

- **Function trigger:** Users click “Đăng Ký Ngay” button
- **Function description:**
  - **Actors:** Guest
  - **Purpose:** Sign up to access the system with Donor account



- **Function details:**
    - **Data requirement:** fullname, email, password, confirm password
    - **Validation:**
        - Email:
            - Type: String
            - Must be valid email type, unique and not include blank.
        - Fullname:
            - Type: String
        - Password:
            - Type: String
            - Length: 6 - 100 characters
            - Must be valid email type, unique and not include blank.
        - Confirm Password:
            - Type: String
            - Must be the same as password
    - **Business rules:** BR-01, BR-03, BR-04

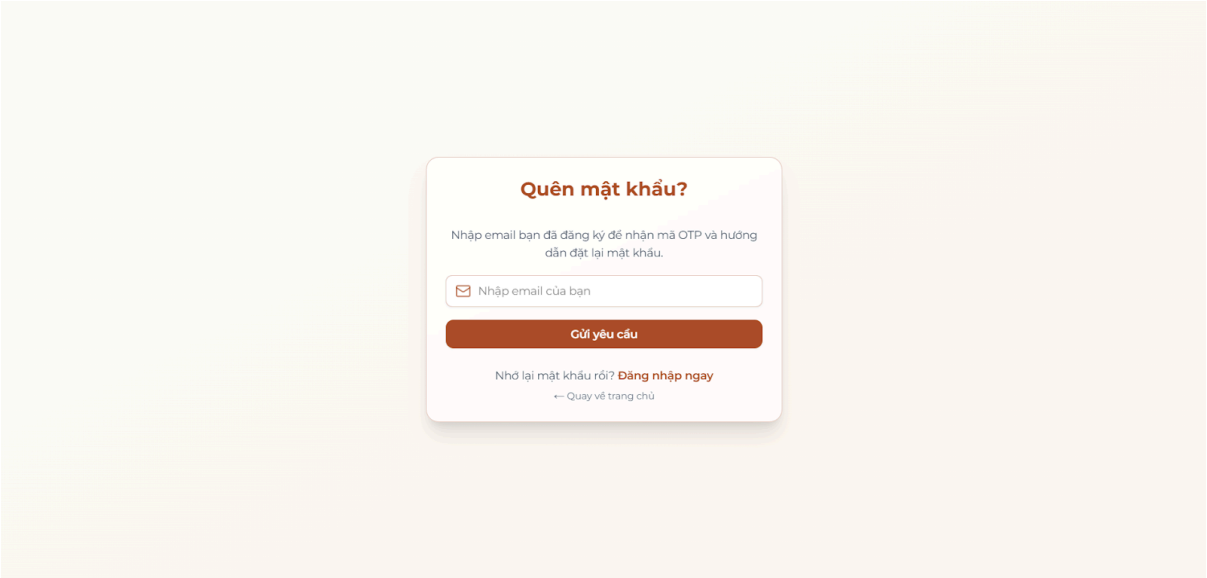
3.2.3 Logout

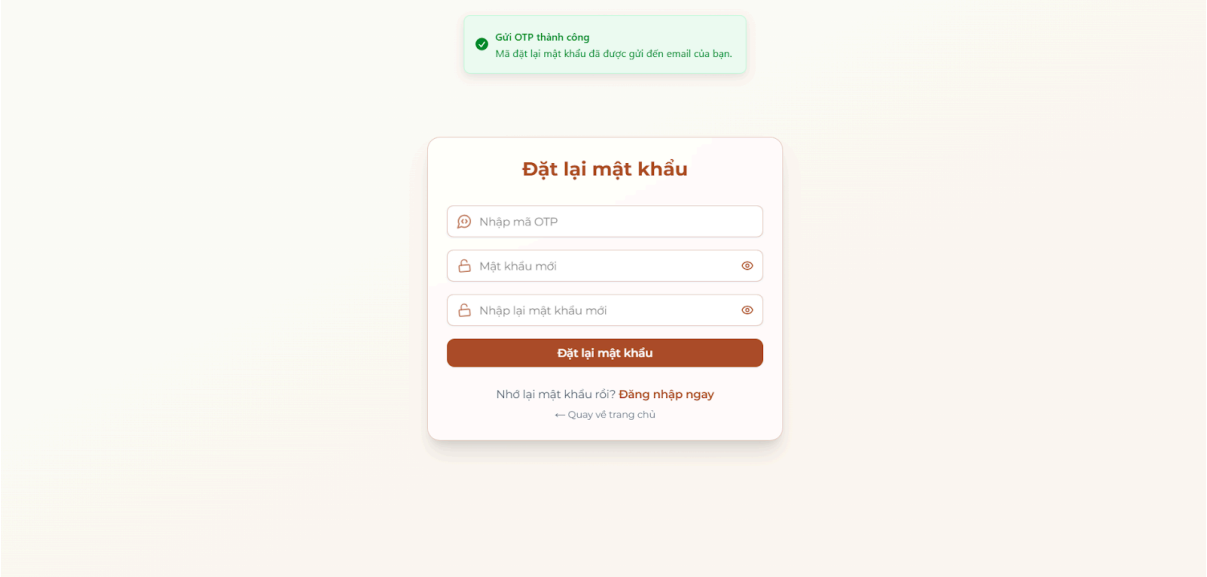
- **Function trigger:** Users click "Đăng xuất" button
- **Function description:**
    - **Actors:** Donor, Fundraiser, Admin, Kitchen Staff, Delivery Staff
    - **Purpose:** Logout of the system

- **Function details:**
    - **Data requirement:** none
    - **Validation:** none
    - **Business rules:** none

3.2.4 Forgot password

- **Function trigger:** Users click "Quên mật khẩu" button
- **Function description:**
    - **Actors:** Donor, Fundraiser, Admin, Kitchen Staff, Delivery Staff
    - **Purpose:** Registered users forgot their password and want to get a new password



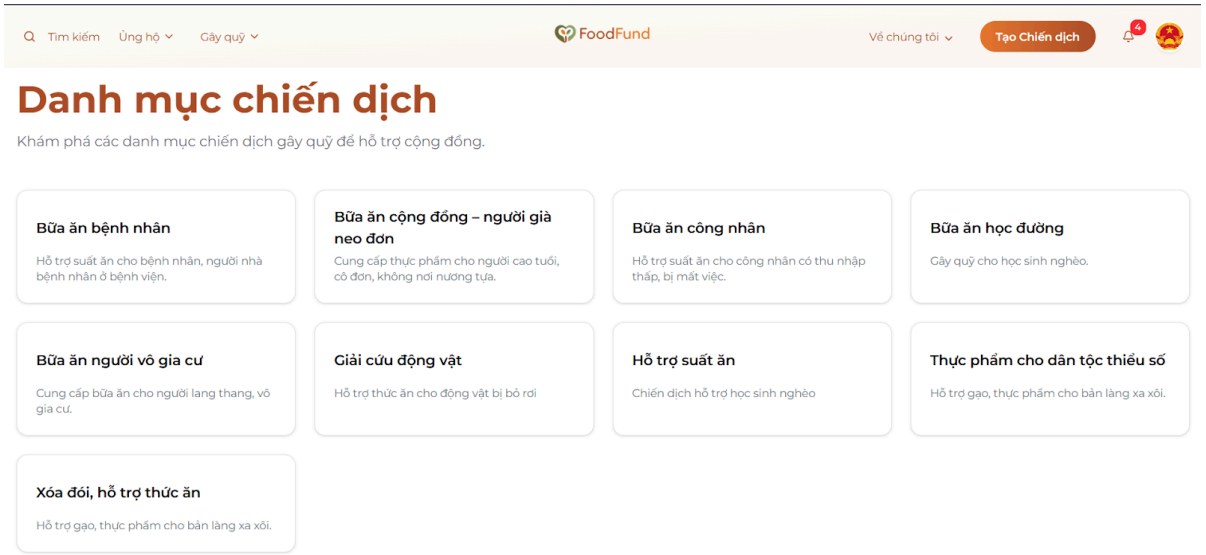


- **Function details:**
  - **Data requirement:** email
  - **Validation:**
    - Email:
      - Type: String
      - Must be valid email type, unique and not include blank.
    - OTP:
      - Type: String
    - New password:
      - Type: String
      - Length: 6 - 100 characters
      - Must be valid email type, unique and not include blank.
    - Confirm new password:
      - Type: String
  - **Business rules:** BR-05

3.3 Campaign Categories

3.3.1 View All Campaign Categories

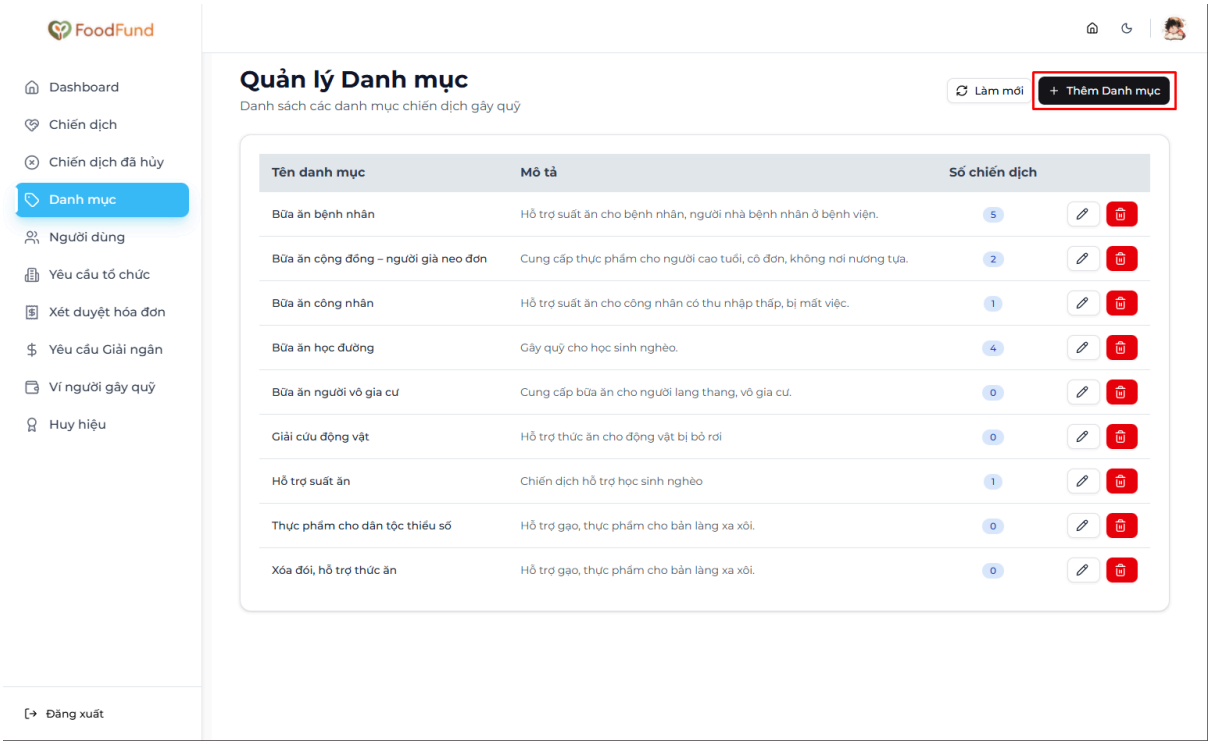
- **Function trigger:** Users hover "Ủng hộ" on header in the home page and click on "Các loại chiến dịch" button
- **Function description:**
  - **Actors:** Guest, Donor, Fundraiser, Admin, Kitchen Staff, Delivery Staff
  - **Purpose:** Users view all categories

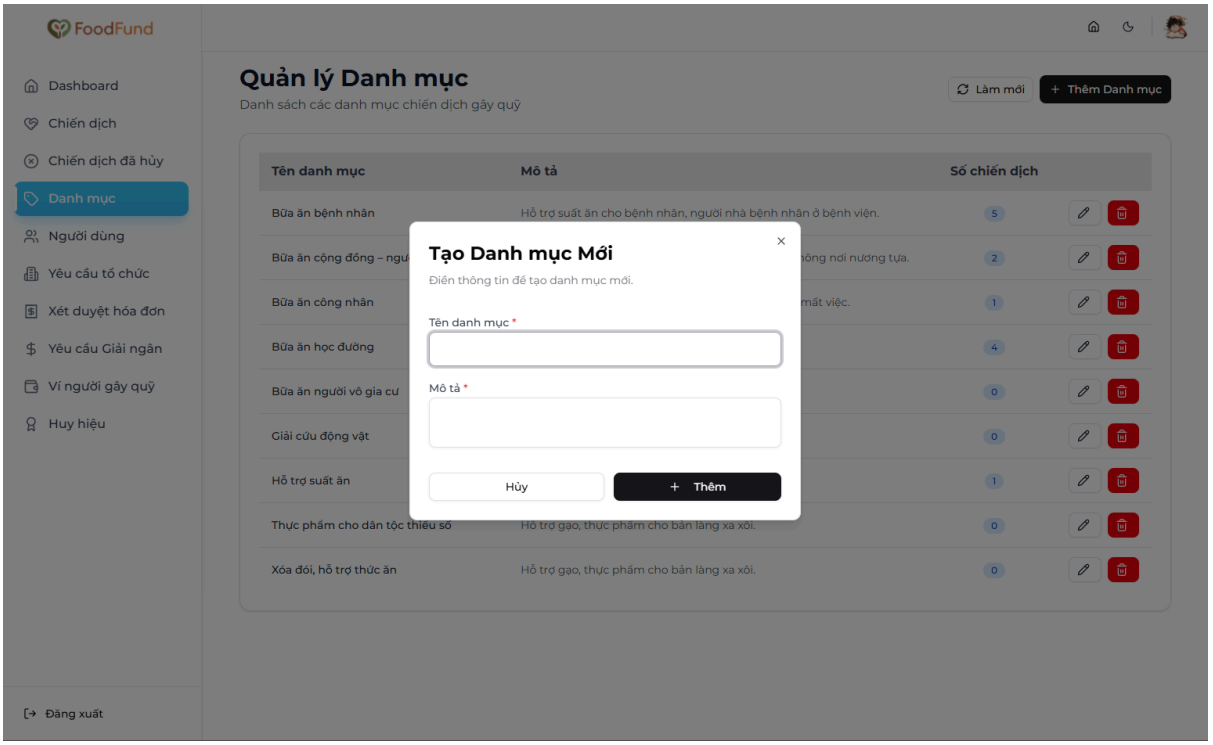


- **Function details:**
  - **Data requirement:** none
  - **Validation:** none
  - **Business rules:** none

3.3.2 Create Category

- **Function trigger:** Admin clicks "Thêm danh mục" button in the categories page
- **Function description:**
  - **Actors:** Admin
  - **Purpose:** Admin wants to add a new category



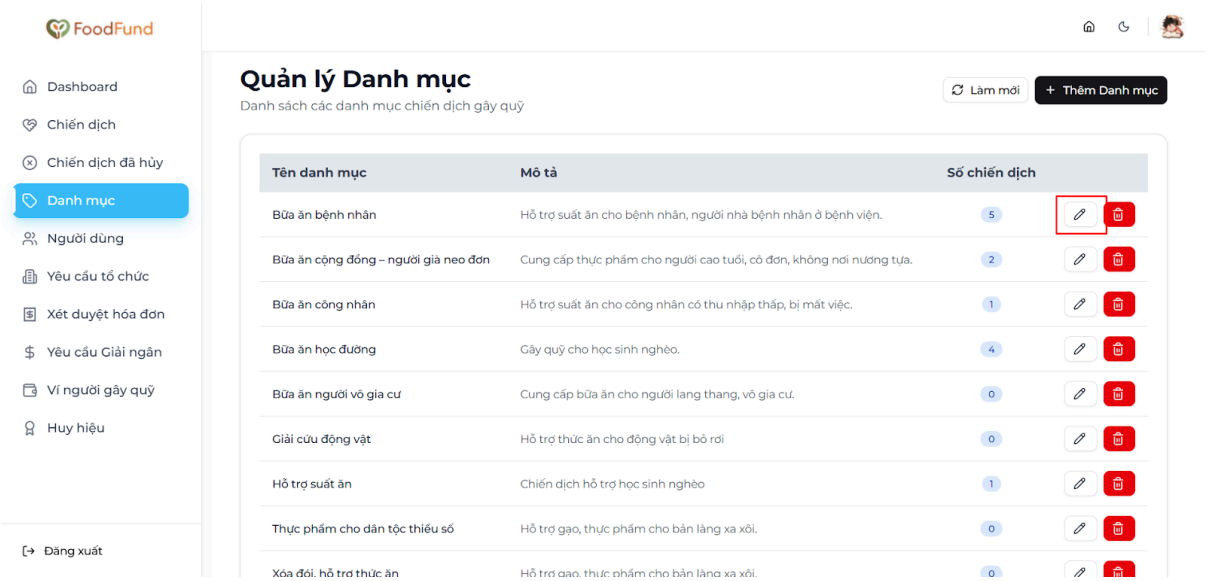


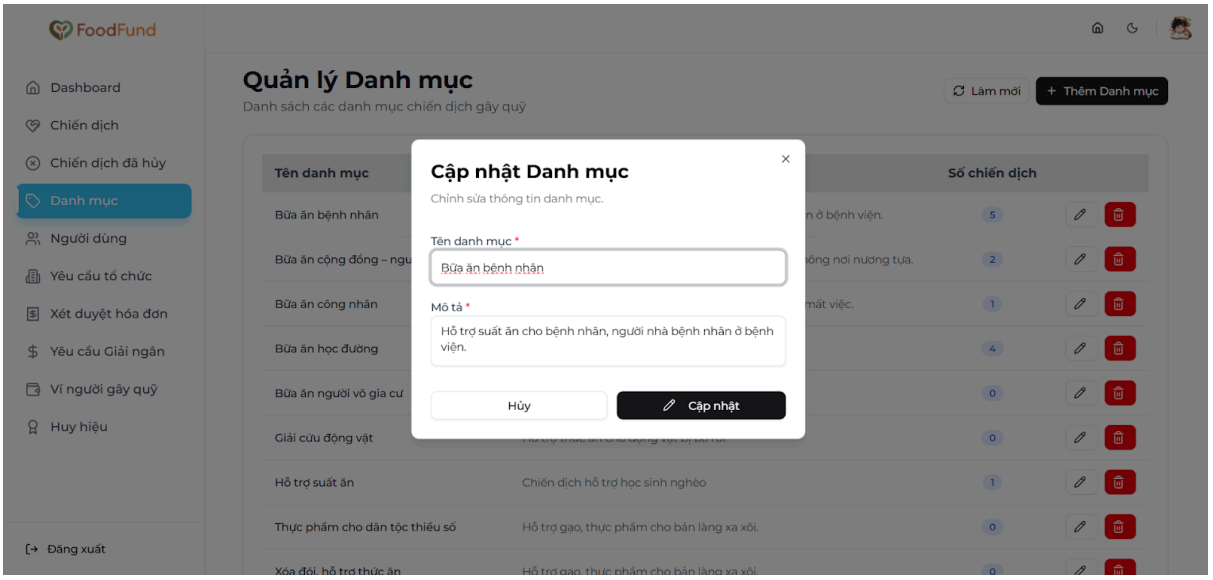
- **Function details:**
  - **Data requirement:** title, description
  - **Validation:**
    - Title:
      - Type: String
      - Length: 5-30 characters
    - Description:
      - Type: String

- Length: 6-100 characters
  - **Business rules:** BR-02

3.3.3 Update Category

- **Function trigger:** Admin clicks update icon button in the categories page
- **Function description:**
  - **Actors:** Admin
  - **Purpose:** Admin wants to update a category

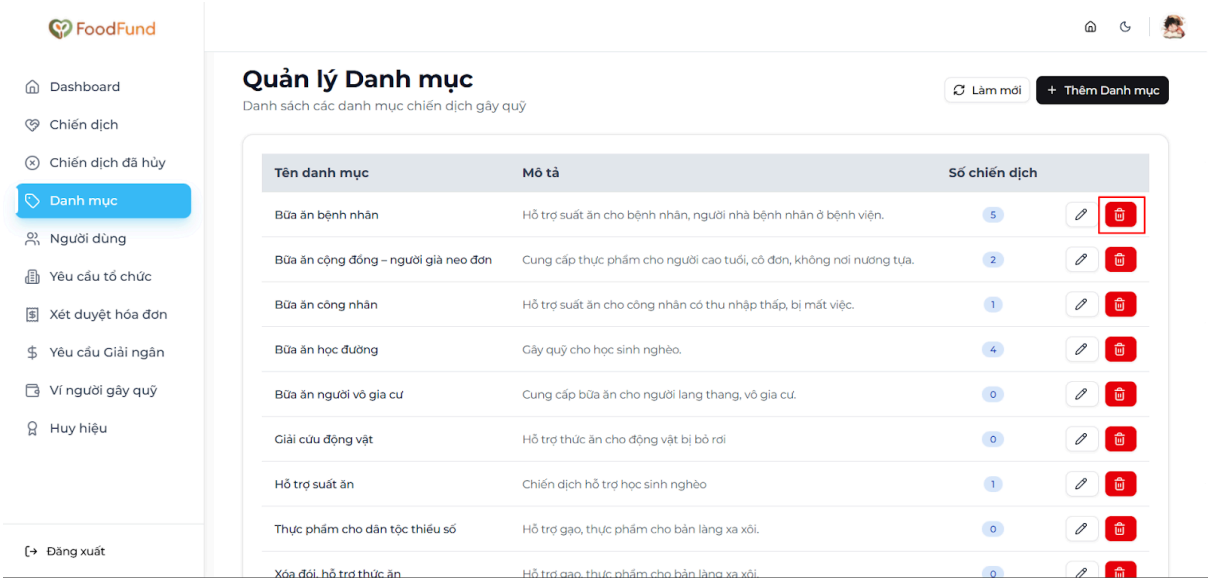


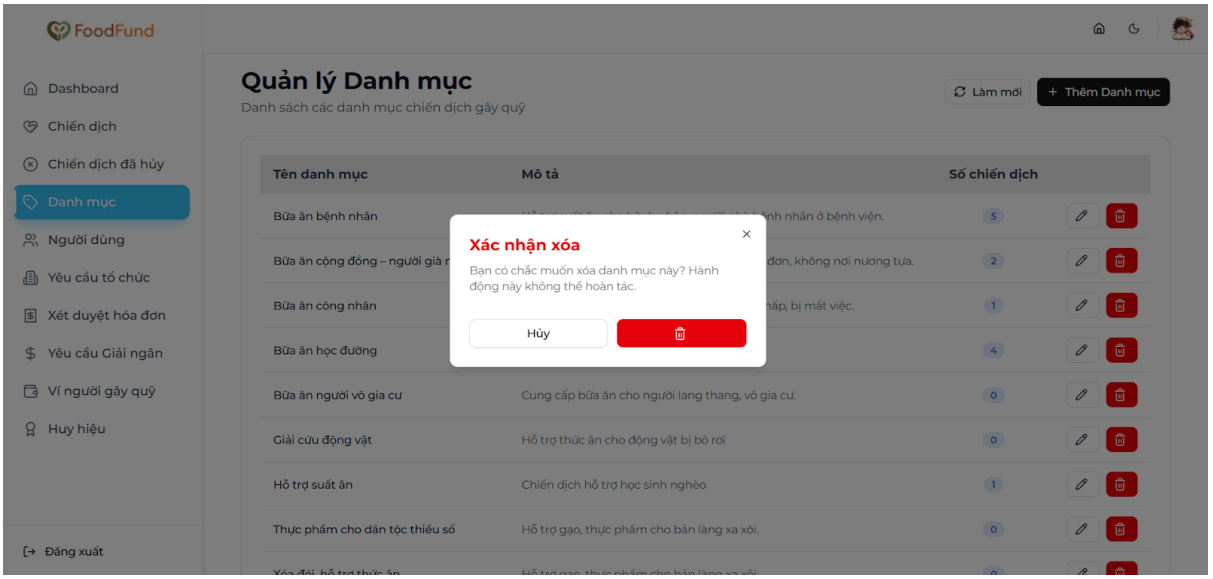


- **Function details:**
    - **Data requirement:** id, title, description
    - **Validation:**
        - Id:
            - Type: String
        - Title:
            - Type: String
            - Length: 5-30 characters
        - Description:
            - Type: String
            - Length: 6-100 characters
    - **Business rules:** BR-02

3.3.4 Delete Category

- **Function trigger:** Admin clicks delete icon button in the categories page
- **Function description:**
    - **Actors:** Admin
    - **Purpose:** Admin wants to delete a category

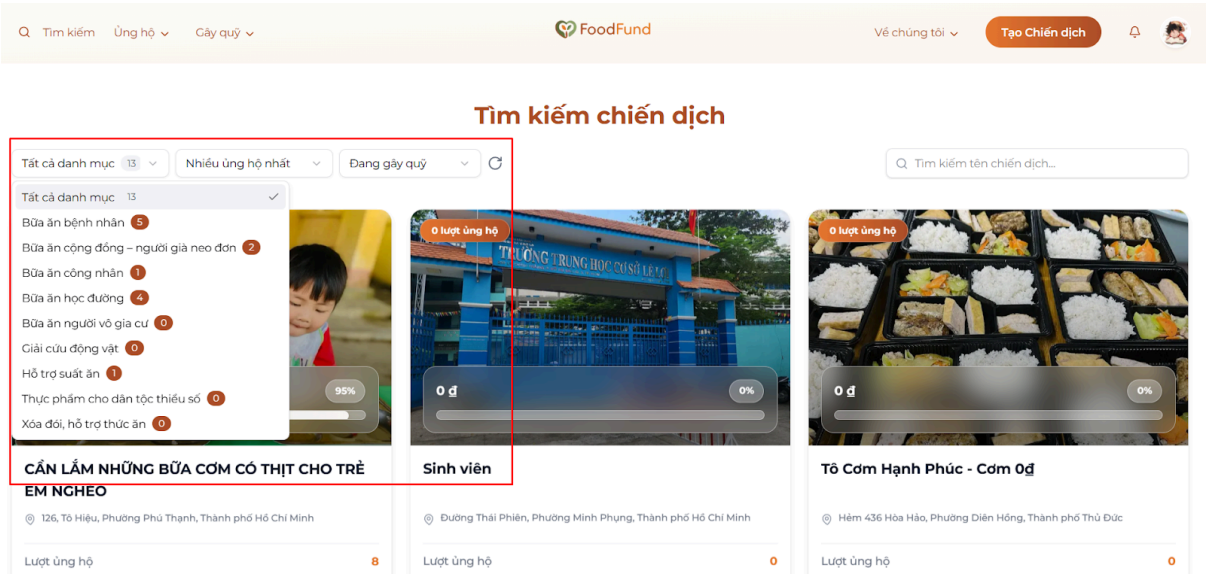




- **Function details:**
  - **Data requirement:** id
  - **Validation:**
    - Id:
      - Type: String
  - **Business rules:** BR-02, BR-50

3.3.5 View Categories With Campaign Count

- **Function trigger:** Users click “Tất cả danh mục” in search page
- **Function description:**
  - **Actors:** Guest, Donor, Fundraiser, Admin, Kitchen Staff, Delivery Staff
  - **Purpose:** Users want to view categories with campaign count

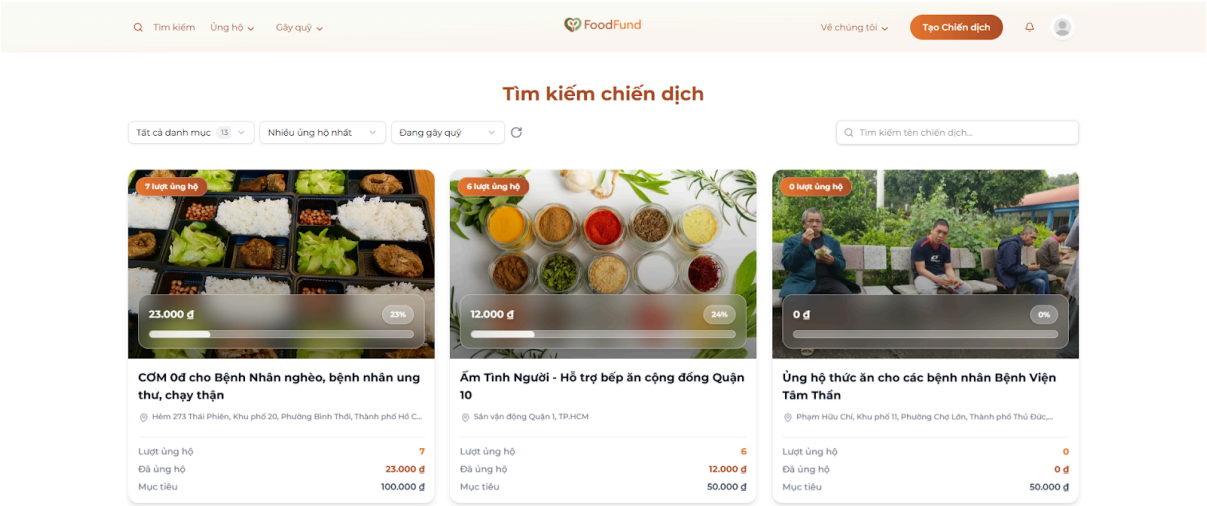


- **Function details:**
  - **Data requirement:** none
  - **Business rules:** none

3.4 Campaigns

3.4.1 View All Campaigns

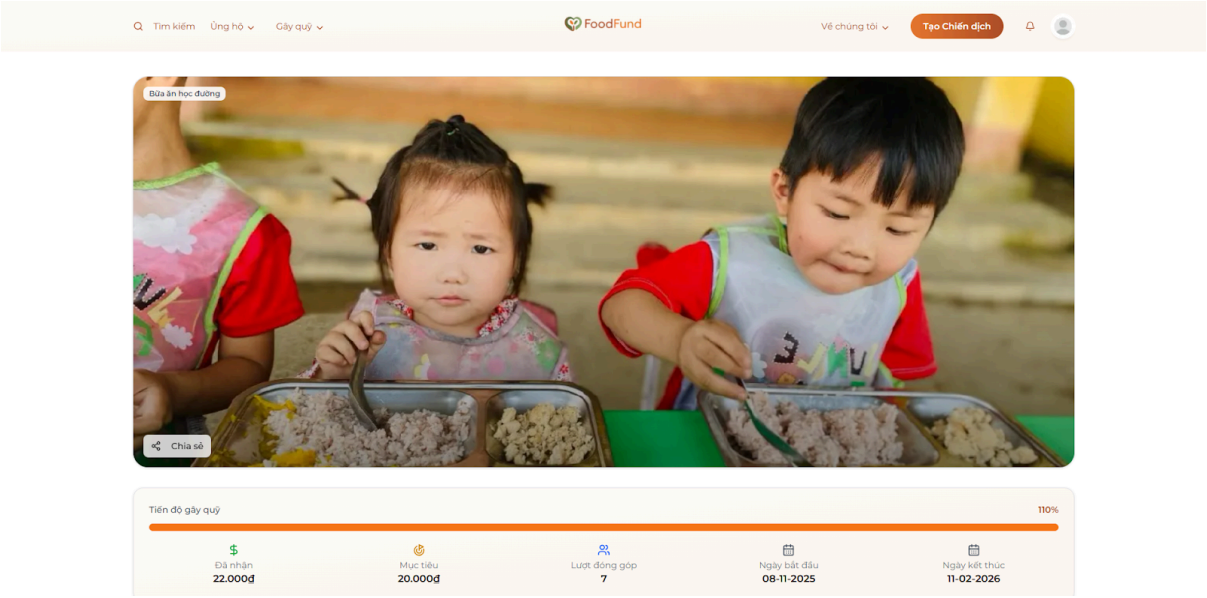
- **Function trigger:** Users click “Tìm kiếm” or “Xem tất cả” button in the home page
- **Function description:**
  - **Actors:** Guest, Donor, Fundraiser, Admin, Kitchen Staff, Delivery Staff
  - **Purpose:** Users view all active campaigns

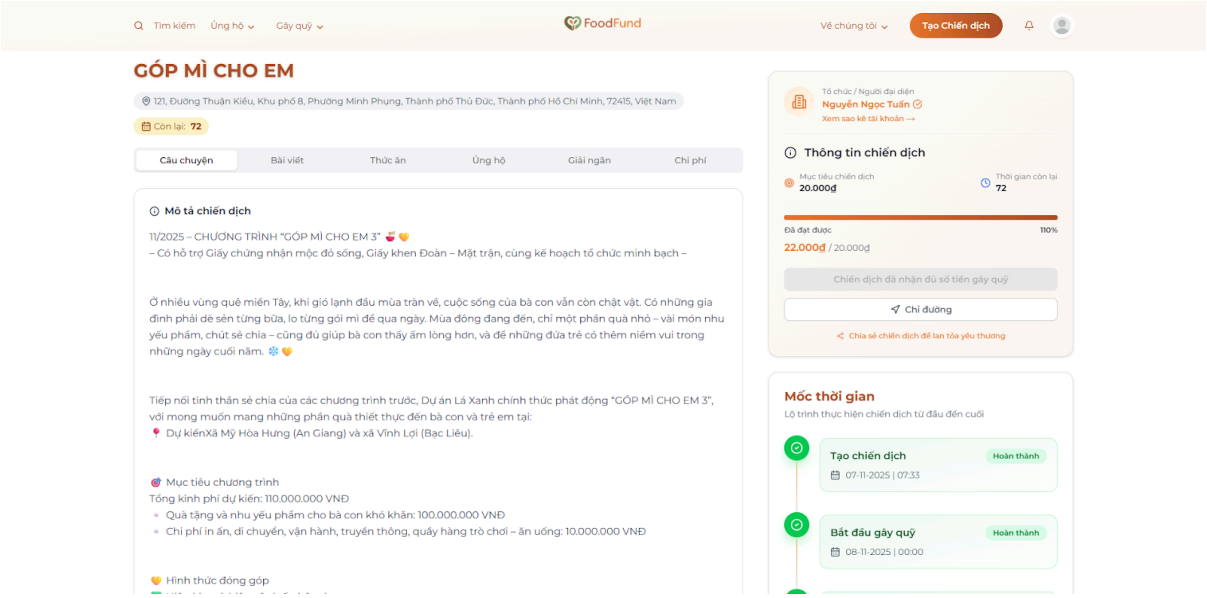


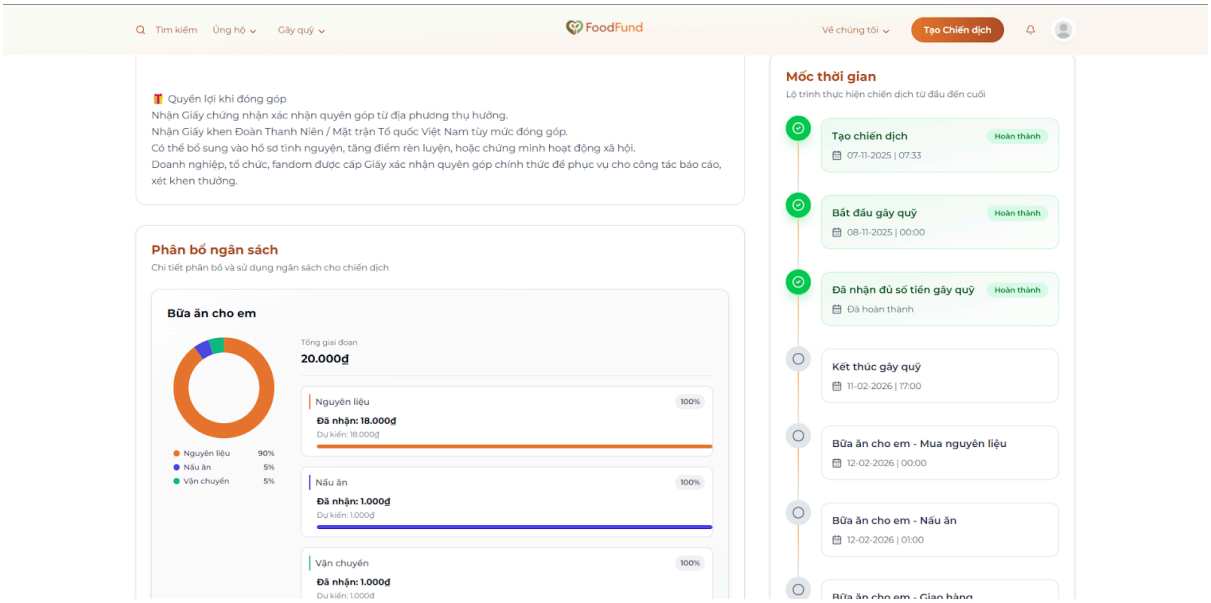
- **Function details:**
  - **Data requirement:** none
  - **Validation:** none
  - **Business rules:** BR-15, BR-25

3.4.2 View Campaign Detail

- **Function trigger:** Users access campaign detail page based on campaigns list.
- **Function description:**
  - **Actors:** Guest, Donor, Fundraiser, Admin, Kitchen Staff, Delivery Staff
  - **Purpose:** Users want to view campaign detail


Tìm kiếm
Ủng hộ
Gây quỹ
FoodFund
Về chúng tôi
Tạo Chiến dịch
Bữa ăn học đường
Chia sẻ
Tiến độ gây quỹ
110%
Đã nhận
22.000đ
Mục tiêu
20.000đ
Lượt đóng góp
7
Ngày bắt đầu
08-11-2025
Ngày kết thúc
11-02-2026

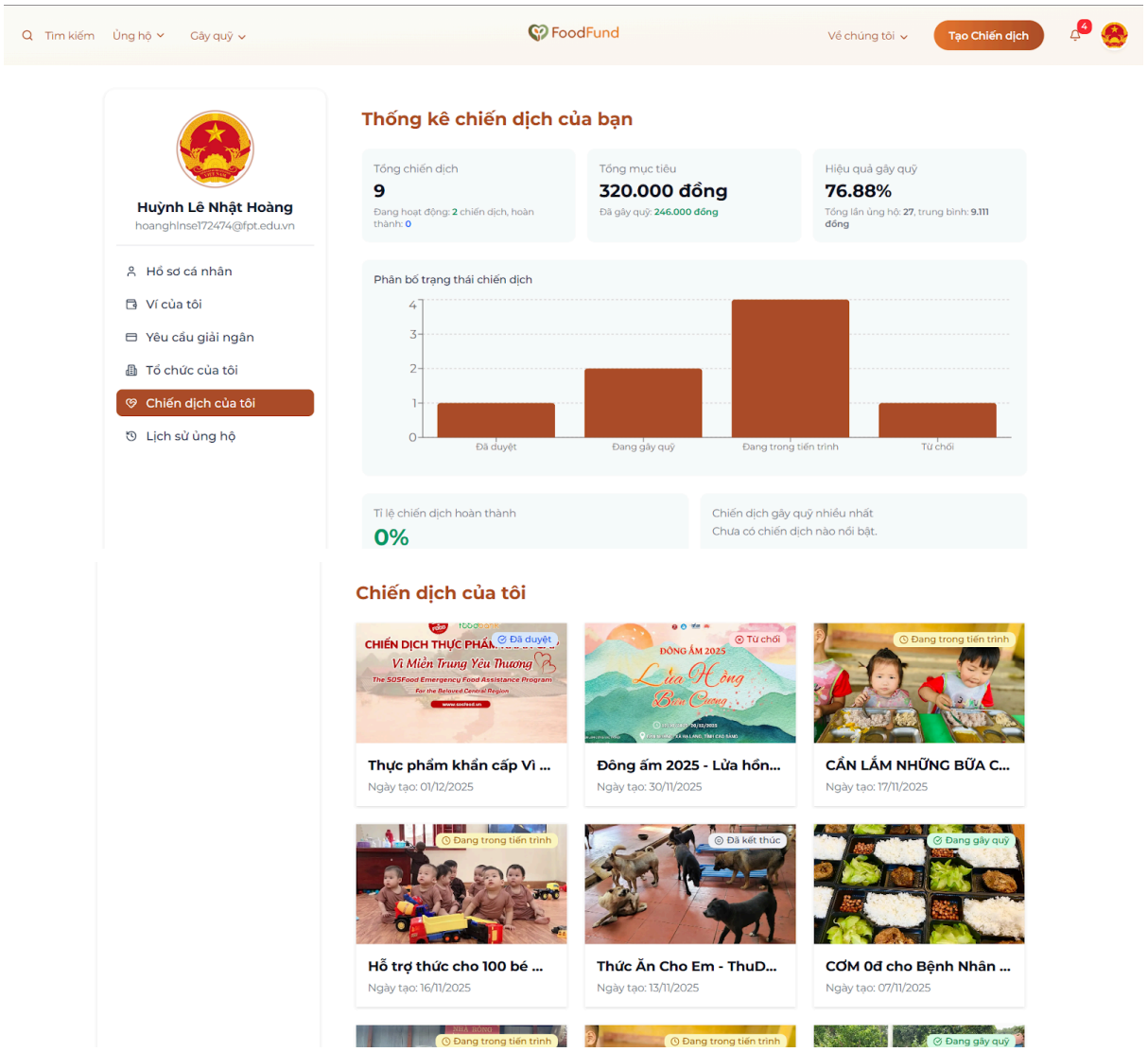

Tìm kiếm
Ủng hộ
Gây quỹ
FoodFund
Về chúng tôi
Tạo Chiến dịch
GÓP MÌ CHO EM
121, Đường Thuận Kiều, Khu phố 8, Phường Minh Phụng, Thành phố Thủ Đức, Thành phố Hồ Chí Minh, 72415, Việt Nam
Còn lại: 72
Câu chuyện
Bài viết
Thức ăn
Ủng hộ
Giải ngân
Chi phí
Mô tả chiến dịch
11/2025 – CHƯƠNG TRÌNH "GÓP MÌ CHO EM 3"
– Có hỗ trợ Giấy chứng nhận mộc đỏ sống, Giấy khen Đoàn – Mặt trận, cùng kế hoạch tổ chức minh bạch –
Ở nhiều vùng quê miền Tây, khi gió lạnh đầu mùa tràn về, cuộc sống của bà con vẫn còn chật vật. Có những gia đình phải dè sẻn từng bữa, lo từng gói mì để qua ngày. Mùa đông đang đến, chỉ một phần quà nhỏ – vài món nhu yếu phẩm, chút sẻ chia – cũng đủ giúp bà con thấy ấm lòng hơn, và để những đứa trẻ có thêm niềm vui trong những ngày cuối năm.
Tiếp nối tinh thần sẻ chia của các chương trình trước, Dự án Lá Xanh chính thức phát động "GÓP MÌ CHO EM 3", với mong muốn mang những phần quà thiết thực đến bà con và trẻ em tại:
Dự kiếnXã Mỹ Hòa Hưng (An Giang) và xã Vĩnh Lợi (Bạc Liêu).
Mục tiêu chương trình
Tổng kinh phí dự kiến: 110.000.000 VNĐ
Quà tặng và nhu yếu phẩm cho bà con khó khăn: 100.000.000 VNĐ
Chi phí in ấn, di chuyển, vận hành, truyền thông, quầy hàng trò chơi – ăn uống: 10.000.000 VNĐ
Hình thức đóng góp
Tổ chức / Người đại diện
Nguyễn Ngọc Tuấn
Xem sao kê tài khoản →
Thông tin chiến dịch
Mục tiêu chiến dịch
20.000đ
Thời gian còn lại
72
Đã đạt được
110%
22.000đ / 20.000đ
Chiến dịch đã nhận đủ số tiền gây quỹ
Chỉ đường
Chia sẻ chiến dịch để lan tỏa yêu thương
Mốc thời gian
Lộ trình thực hiện chiến dịch từ đầu đến cuối
Tạo chiến dịch
Hoàn thành
07-11-2025 | 07:33
Bắt đầu gây quỹ
Hoàn thành
08-11-2025 | 00:00


Tìm kiếm
Ủng hộ
Gây quỹ
FoodFund
Về chúng tôi
Tạo Chiến dịch
Quyền lợi khi đóng góp
Nhận Giấy chứng nhận xác nhận quyền góp từ địa phương thụ hưởng.
Nhận Giấy khen Đoàn Thanh Niên / Mặt trận Tổ quốc Việt Nam tùy mức đóng góp.
Có thể bổ sung vào hồ sơ tình nguyện, tăng điểm rèn luyện, hoặc chứng minh hoạt động xã hội.
Doanh nghiệp, tổ chức, fandom được cấp Giấy xác nhận quyền góp chính thức để phục vụ cho công tác báo cáo, xét khen thưởng.
Phân bổ ngân sách
Chi tiết phân bổ và sử dụng ngân sách cho chiến dịch
Bữa ăn cho em
Tổng giai đoạn
20.000đ
Nguyên liệu 90%
Nấu ăn 5%
Vận chuyển 5%
Nguyên liệu
100%
Đã nhận: 18.000đ
Dự kiến: 18.000đ
Nấu ăn
100%
Đã nhận: 1.000đ
Dự kiến: 1.000đ
Vận chuyển
100%
Đã nhận: 1.000đ
Dự kiến: 1.000đ
Mốc thời gian
Lộ trình thực hiện chiến dịch từ đầu đến cuối
Tạo chiến dịch
Hoàn thành
07-11-2025 | 07:33
Bắt đầu gây quỹ
Hoàn thành
08-11-2025 | 00:00
Đã nhận đủ số tiền gây quỹ
Hoàn thành
Đã hoàn thành
Kết thúc gây quỹ
11-02-2026 | 17:00
Bữa ăn cho em - Mua nguyên liệu
12-02-2026 | 00:00
Bữa ăn cho em - Nấu ăn
12-02-2026 | 01:00

- **Function details:**
    - **Data requirement:** Id
    - **Validation:**
        - Id:
            - Type: string
            - Object Id of campaign item
    - **Business rules:** none

3.4.3 View My Campaigns

- **Function trigger:** Fundraiser accesses my campaigns page.
- **Function description:**
    - **Actors:** Fundraiser
    - **Purpose:** Fundraiser views all their campaigns



- **Function details:**
    - **Data requirement:** none
    - **Validation:** none
    - **Business rules:** BR-02

3.4.5 View Expiring Campaigns

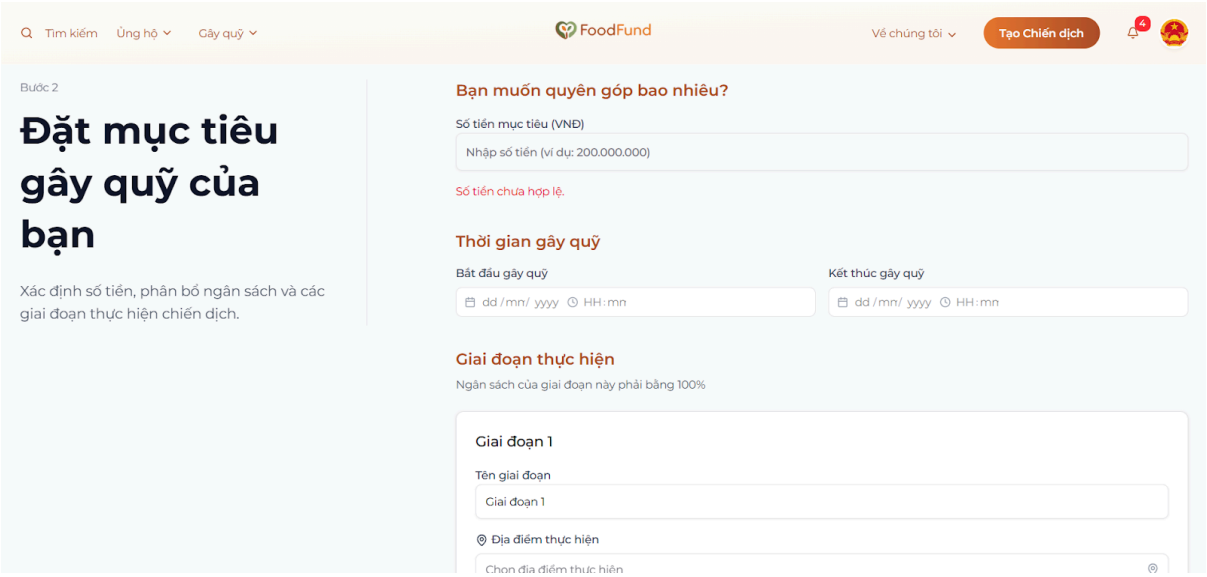
- **Function trigger:** Users access expiring campaigns page.
- **Function description:**
  - **Actors:** Guest, Donor, Fundraiser, Admin, Kitchen Staff, Delivery Staff
  - **Purpose:** Users view all expiring campaigns

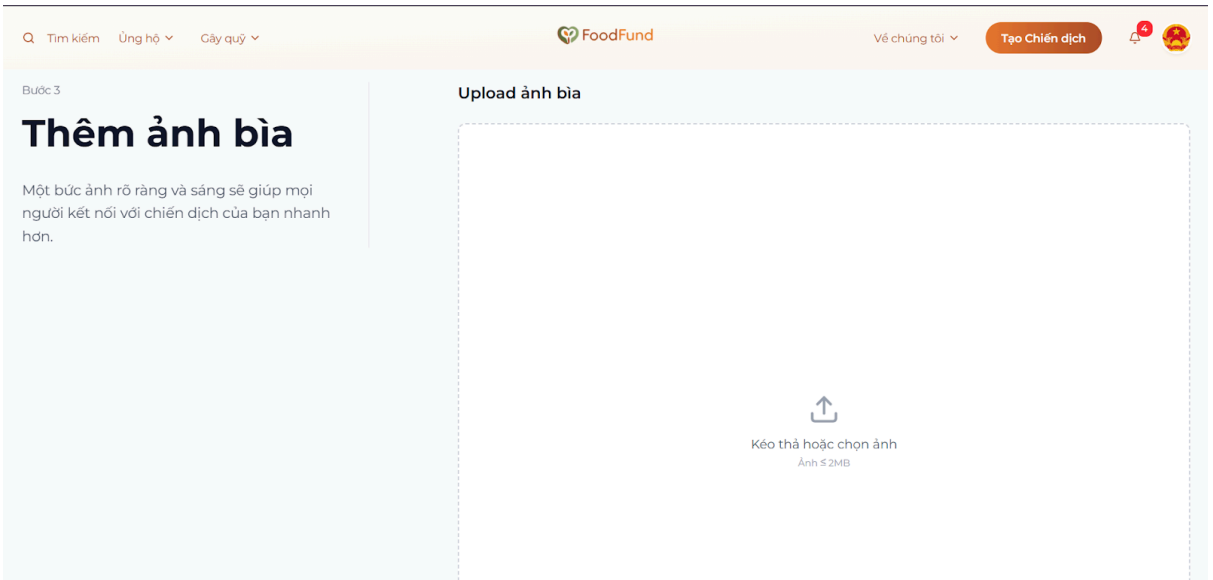
- **Function details:**
  - **Data requirement:** none
  - **Validation:** none
  - **Business rules:** none

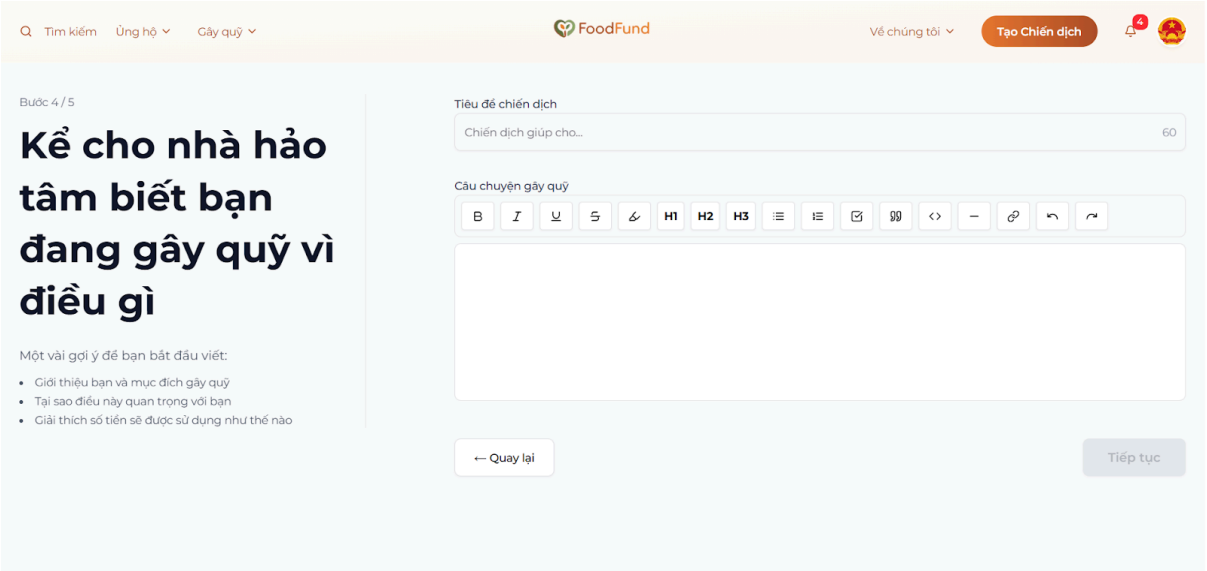
3.4.6 Create Campaign

- **Function trigger:** Fundraiser clicks “Tạo chiến dịch” button on the campaign page.
- **Function description:**
  - **Actors:** Fundraiser
  - **Purpose:** Fundraiser wants to create a campaign

Tìm kiếm
Ủng hộ
Gây quỹ
FoodFund
Về chúng tôi
Tạo Chiến dịch
Bước 1
Chọn loại chiến dịch với FoodFund
Chọn danh mục phù hợp với chiến dịch gây quỹ của bạn
Bữa ăn bệnh nhân
Hỗ trợ suất ăn cho bệnh nhân, người nhà bệnh nhân ở bệnh viện.
Bữa ăn cộng đồng – người già neo đơn
Cung cấp thực phẩm cho người cao tuổi, cô đơn, không nơi nương tựa.
Bữa ăn công nhân
Hỗ trợ suất ăn cho công nhân có thu nhập thấp, bị mất việc.
Bữa ăn học đường
Gây quỹ cho học sinh nghèo.
Bữa ăn người vô gia cư
Cung cấp bữa ăn cho người lang thang, vô gia cư.
Giải cứu động vật
Hỗ trợ thức ăn cho động vật bị bỏ rơi


Tìm kiếm
Ủng hộ
Gây quỹ
FoodFund
Về chúng tôi
Tạo Chiến dịch
Bước 2
Đặt mục tiêu gây quỹ của bạn
Xác định số tiền, phân bổ ngân sách và các giai đoạn thực hiện chiến dịch.
Bạn muốn quyên góp bao nhiêu?
Số tiền mục tiêu (VNĐ)
Nhập số tiền (ví dụ: 200.000.000)
Số tiền chưa hợp lệ.
Thời gian gây quỹ
Bắt đầu gây quỹ
Kết thúc gây quỹ
dd /mm/ yyyy HH:mm
dd /mm/ yyyy HH:mm
Giai đoạn thực hiện
Ngân sách của giai đoạn này phải bằng 100%
Giai đoạn 1
Tên giai đoạn
Giai đoạn 1
Địa điểm thực hiện


Tìm kiếm
Ủng hộ
Gây quỹ
FoodFund
Về chúng tôi
Tạo Chiến dịch
Bước 3
Thêm ảnh bìa
Một bức ảnh rõ ràng và sáng sẽ giúp mọi người kết nối với chiến dịch của bạn nhanh hơn.
Upload ảnh bìa
Kéo thả hoặc chọn ảnh
Ảnh ≤ 2MB

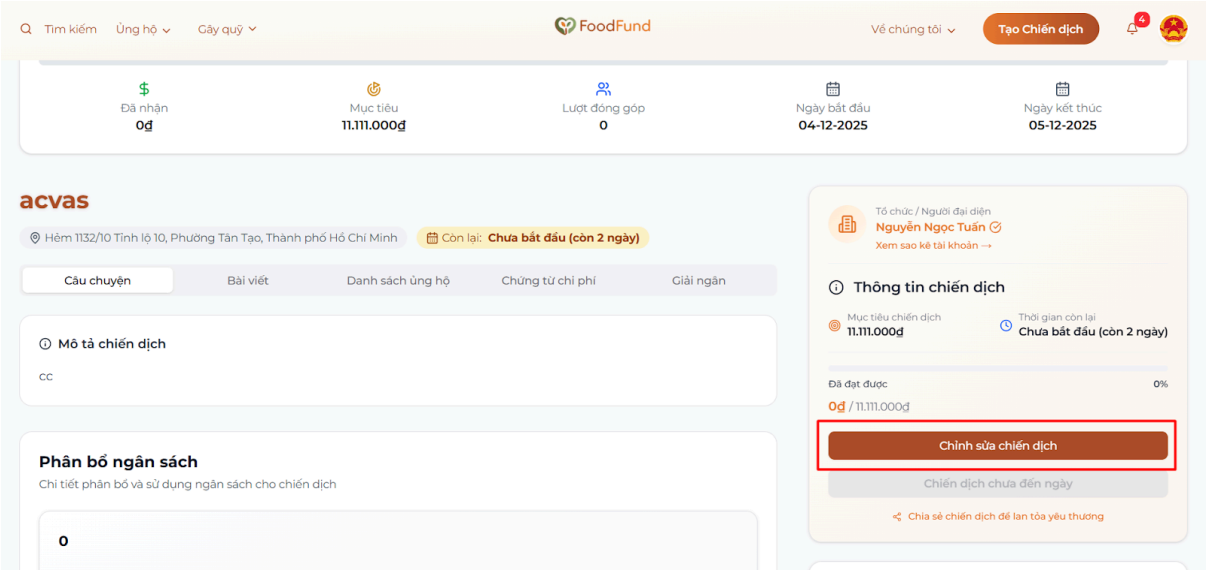


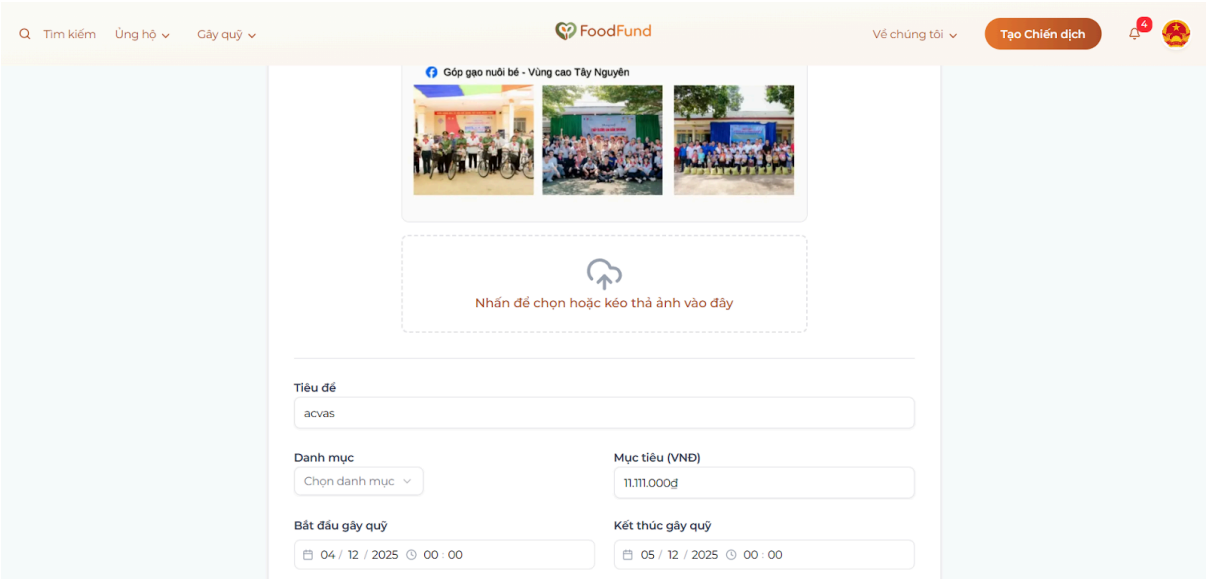
- **Function details:**
  - **Data requirement:** title, description, coverImageFileKey, targetAmount, categoryId, fundraisingStartDate, fundraisingEndDate, phases, phaseName, location, ingredientPurchaseDate, cookingDate, deliveryDate, cookingBudgetPercentage, deliveryBudgetPercentage
  - **Validation:**
    - Title:
      - Type: String
      - Length: 5-100 characters
    - Description
      - Type: String
      - Length: 5-100 characters
    - CoverImageFileKey
      - Type: String
    - TargetAmount:
      - Type: BigInt

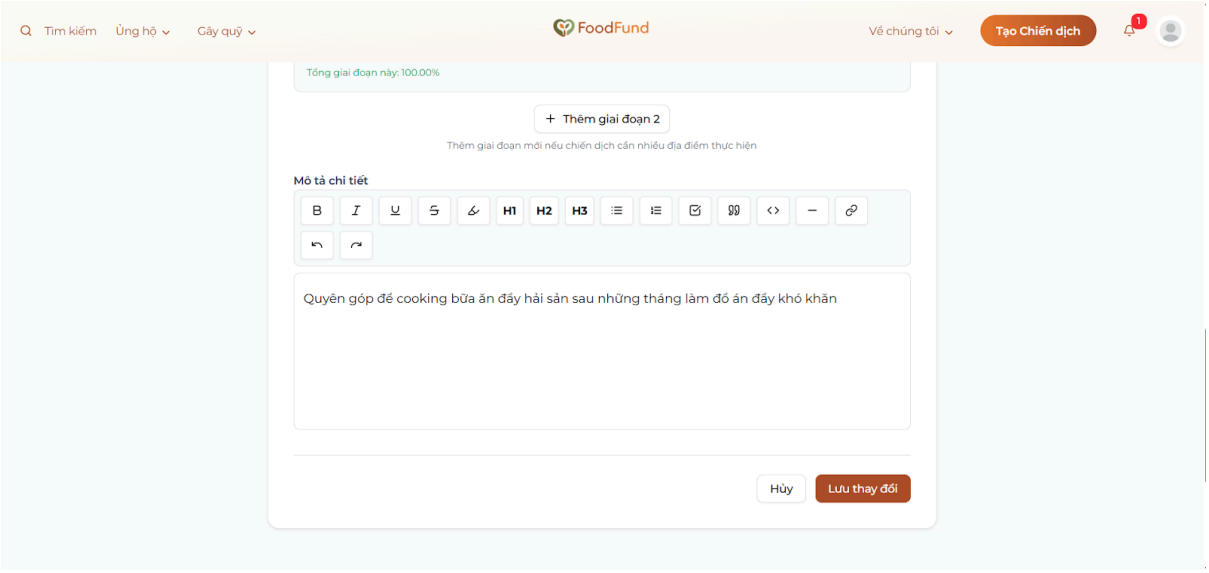
        - CategoryId:
            - Type: String
            - Must be match to category id
        - FundraisingStartDate:
            - Type: Timestamp
        - FundraisingEndDate:
            - Type: Timestamp
        - Phases:
            - Type: Array
        - PhaseName:
            - Type: String
            - Length: 5-100 characters
        - Location:
            - Type: String
            - Length: 5-100 characters
        - IngredientPurchaseDate:
            - Type: Timestamp
        - CookingDate:
            - Type: Timestamp
        - DeliveryDate:
            - Type: Timestamp
        - IngredientBudgetPercentage:
            - Type: Number
        - IngredientBudgetPercentage:
            - Type: Number
        - CookingBudgetPercentage:
            - Type: Number
        - DeliveryBudgetPercentage:
            - Type: Number
    - **Business rules:** BR-02, BR-12, BR-15

3.4.6 Edit Campaign

- **Function trigger:** Fundraiser click “Chỉnh sửa” button on their campaign of the profile page.
- **Function description:**
    - **Actors:** Fundraiser
    - **Purpose:** Fundraiser wants to create a campaign


Tìm kiếm
Ủng hộ
Gây quỹ
FoodFund
Về chúng tôi
Tạo Chiến dịch
Đã nhận
0đ
Mục tiêu
11.111.000đ
Lượt đóng góp
0
Ngày bắt đầu
04-12-2025
Ngày kết thúc
05-12-2025
acvas
Hẻm 1132/10 Tỉnh lộ 10, Phường Tân Tạo, Thành phố Hồ Chí Minh
Còn lại: Chưa bắt đầu (còn 2 ngày)
Câu chuyện
Bài viết
Danh sách ủng hộ
Chứng từ chi phí
Giải ngân
Mô tả chiến dịch
cc
Phân bổ ngân sách
Chi tiết phân bổ và sử dụng ngân sách cho chiến dịch
0
Tổ chức / Người đại diện
Nguyễn Ngọc Tuấn
Xem sao kê tài khoản →
Thông tin chiến dịch
Mục tiêu chiến dịch
11.111.000đ
Thời gian còn lại
Chưa bắt đầu (còn 2 ngày)
Đã đạt được
0%
0đ / 11.111.000đ
Chỉnh sửa chiến dịch
Chiến dịch chưa đến ngày
Chia sẻ chiến dịch để lan tỏa yêu thương


Tìm kiếm
Ủng hộ
Gây quỹ
FoodFund
Về chúng tôi
Tạo Chiến dịch
Góp gạo nuôi bé - Vùng cao Tây Nguyên
Nhấn để chọn hoặc kéo thả ảnh vào đây
Tiêu đề
acvas
Danh mục
Chọn danh mục
Mục tiêu (VNĐ)
11.111.000đ
Bắt đầu gây quỹ
04 / 12 / 2025 00 : 00
Kết thúc gây quỹ
05 / 12 / 2025 00 : 00


Tìm kiếm
Ủng hộ
Gây quỹ
FoodFund
Về chúng tôi
Tạo Chiến dịch
Tổng giai đoạn này: 100.00%
+ Thêm giai đoạn 2
Thêm giai đoạn mới nếu chiến dịch cần nhiều địa điểm thực hiện
Mô tả chi tiết
H1
H2
H3
Quyên góp để cooking bữa ăn đầy hải sản sau những tháng làm đồ án đầy khó khăn
Hủy
Lưu thay đổi

- **Function details:**
    - **Data requirement:** id, title, description, coverImageFileKey, targetAmount, categoryId, fundraisingStartDate, fundraisingEndDate, phases, phaseName, location, ingredientPurchaseDate, cookingDate, deliveryDate, cookingBudgetPercentage, deliveryBudgetPercentage
    - **Validation:**
        - Title:
            - Type: String
            - Length: 5-100 characters
        - Description
            - Type: String
            - Length: 5-100 characters
        - CoverImageFileKey
            - Type: Array<String>
        - TargetAmount:
            - Type: BigInt
        - CategoryId:
            - Type: String
            - Must be match to category id
        - FundraisingStartDate:
            - Type: Timestamp
        - FundraisingEndDate:
            - Type: Timestamp
        - Phases:
            - Type: Array
        - PhaseName:
            - Type: String
            - Length: 5-100 characters
        - Location:
            - Type: String
            - Length: 5-100 characters
        - IngredientPurchaseDate:
            - Type: Timestamp
        - CookingDate:
            - Type: Timestamp
        - DeliveryDate:
            - Type: Timestamp
        - IngredientBudgetPercentage:
            - Type: Number
        - IngredientBudgetPercentage:
            - Type: Number
        - CookingBudgetPercentage:
            - Type: Number
        - DeliveryBudgetPercentage:

        - Type: Number
    - **Business rules:** BR-02, BR-13

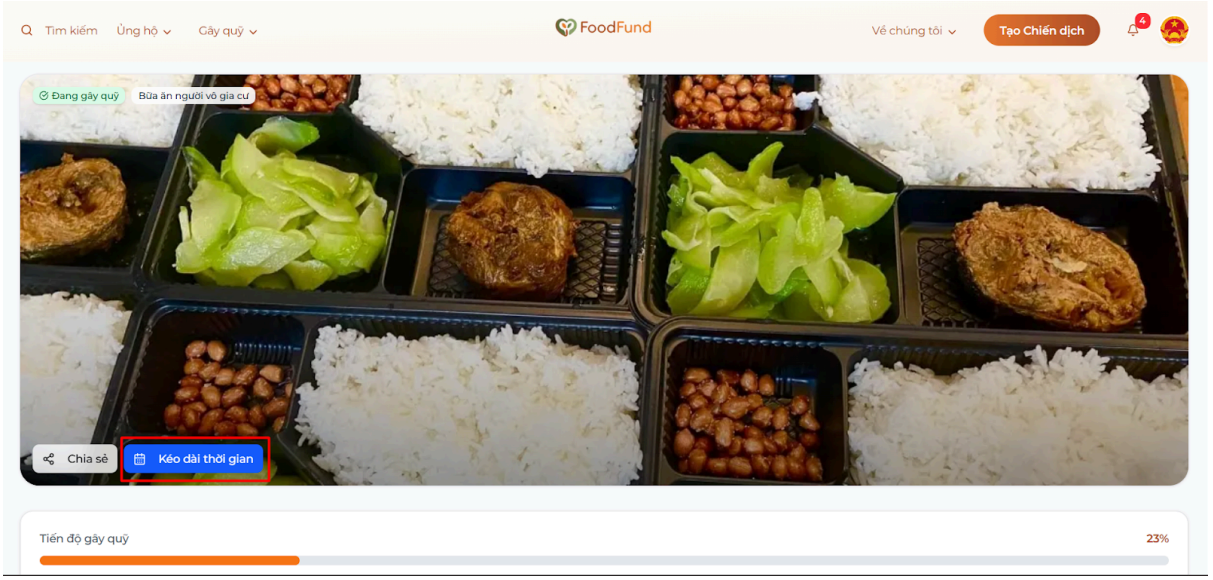
3.4.7 Delete Campaign

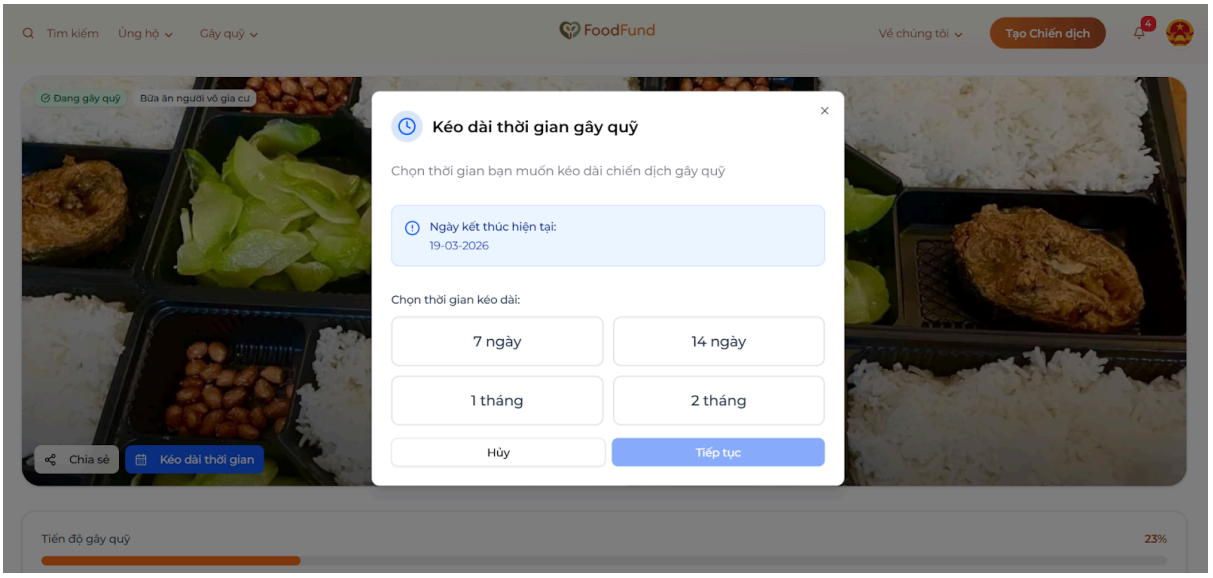
- **Function trigger:** Fundraiser clicks the “Xóa” button on their campaign of the profile page.
- **Function description:**
    - **Actors:** Fundraiser
    - **Purpose:** Fundraiser wants to delete a campaign

- **Function details:**
    - **Data requirement:** id
    - **Validation:** none
    - **Business rules:** BR-02, BR-14

3.4.8 Extend Campaign

- **Function trigger:** Fundraiser clicks the “Kéo dài thời gian” button on their campaign of the profile page.
- **Function description:**
    - **Actors:** Fundraiser
    - **Purpose:** Fundraiser want to delete a campaign

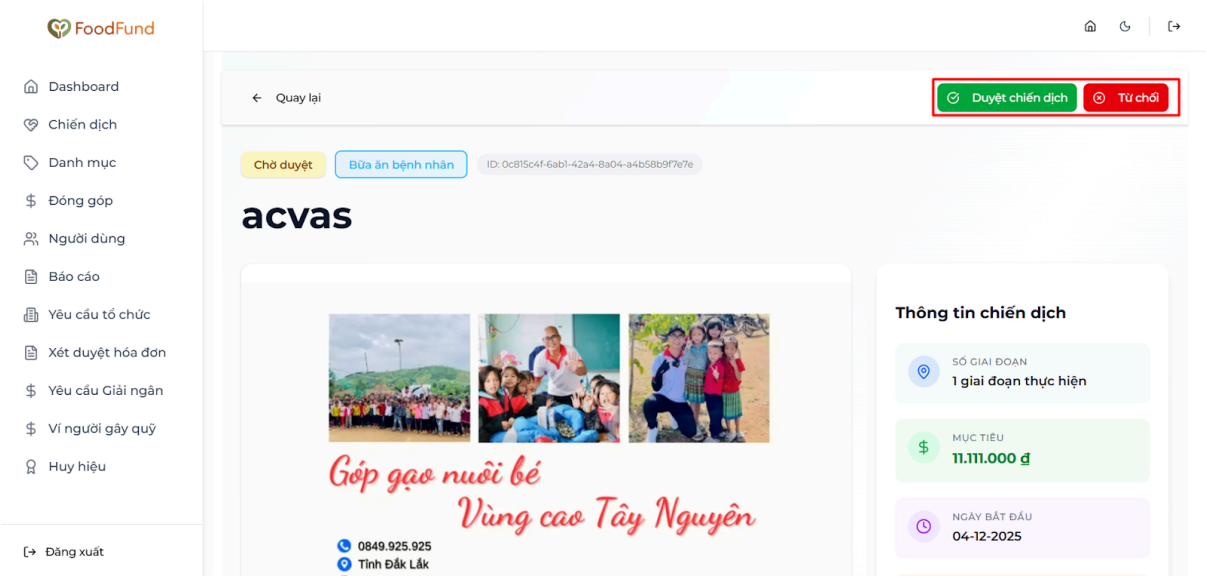




- **Function details:**
  - **Data requirement:** id, extensionDays
  - **Validation:**
    - Id:
      - Type: String
    - ExtensionDays:
      - Type: Number
  - **Business rules:** BR-02, BR-19, BR-24, BR-20

3.4.9 Campaign Approval

- **Function trigger:** Admin clicks “Duyệt/Từ chối” button on manage campaign page.
- **Function description:**
  - **Actors:** Admin
  - **Purpose:** Admin wants to approve/reject a campaign



- **Function details:**
  - **Data requirement:** campaignId, newStatus, reason
  - **Validation:**
    - CampaignId:
      - Type: String
    - NewStatus:
      - Type: String
    - Reason:
      - Type: String
      - Optional
  - **Business rules:** BR-02, BR-16, BR-22

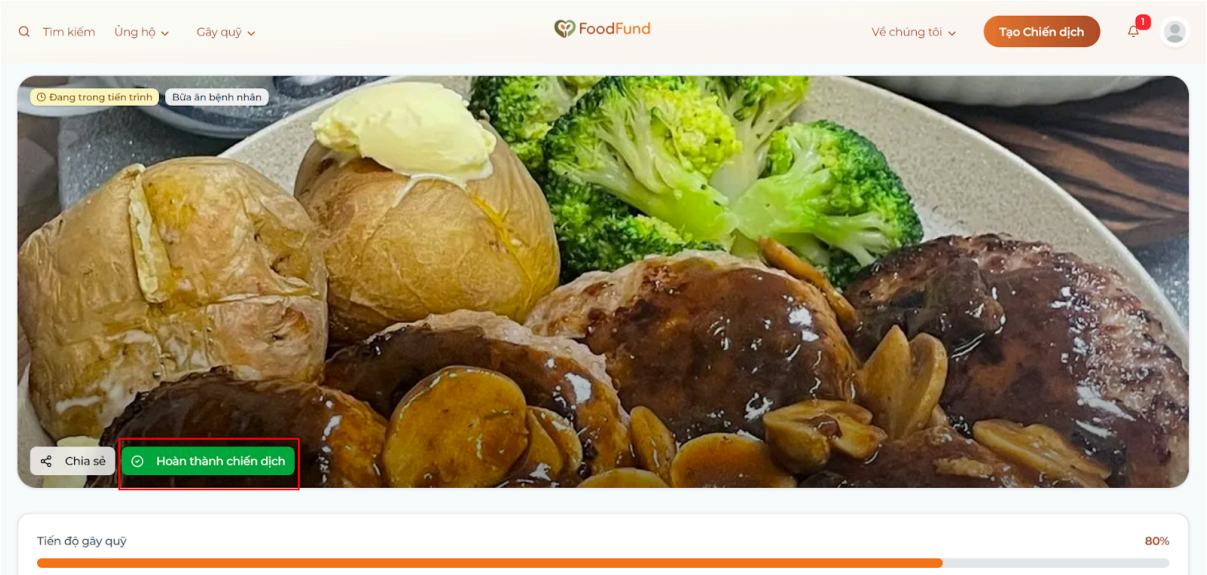
3.4.10 Cancel campaign

- **Function trigger:** Admin clicks “Hủy chiến dịch” button on manage campaign page.
- **Function description:**
  - **Actors:** Admin
  - **Purpose:** Admin wants to cancel a campaign

- **Function details:**
  - **Data requirement:** campaignId, newStatus, reason
  - **Validation:**
    - CampaignId:
      - Type: String
    - NewStatus:
      - Type: String
    - Reason:
      - Type: String
      - Optional
  - **Business rules:** BR-02, BR-30

3.4.11 Mark Complete Campaign

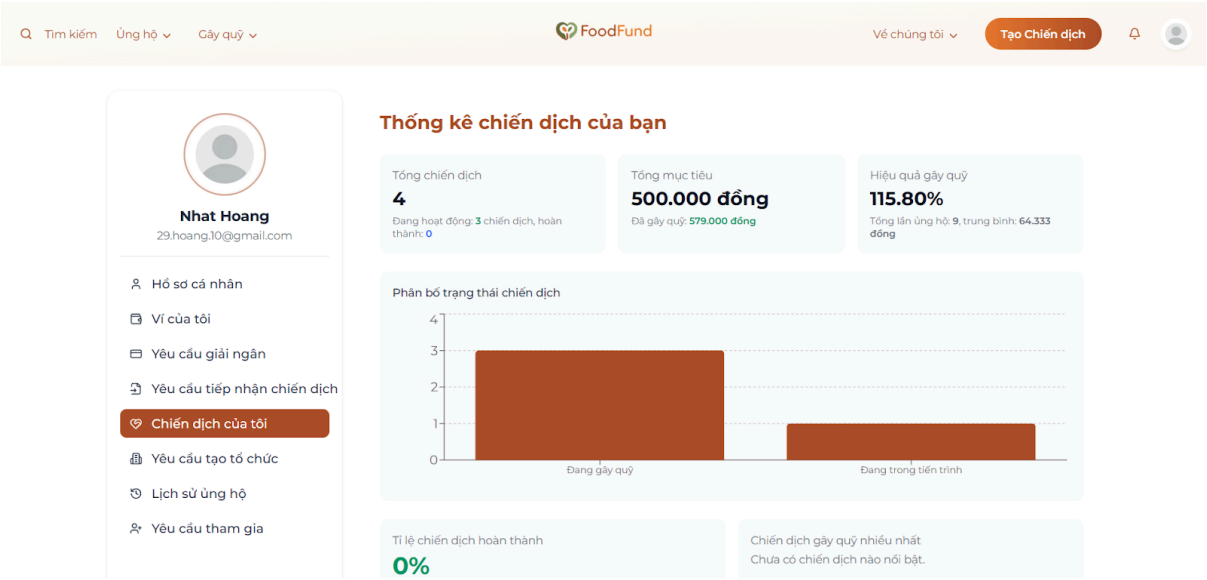
- **Function trigger:** Fundraiser clicks “Hoàn thành chiến dịch” button on their campaign page.
- **Function description:**
  - **Actors:** Fundraiser
  - **Purpose:** Fundraiser wants to mark complete their campaign



- **Function details:**
  - **Data requirement:** campaignId
  - **Validation:**
    - CampaignId:
      - Type: String
  - **Business rules:** BR-02, BR-21, BR-23, BR-26

3.4.12 View Campaign Stats

- **Function trigger:** Fundraiser clicks “Chiến dịch của tôi” tab on profile page.
- **Function description:**
  - **Actors:** Fundraiser
  - **Purpose:** Fundraiser wants to view their campaign stats



- **Function details:**
  - **Data requirement:** none
  - **Validation:** none
  - **Business rules:** BR-02, BR-12

3.4.13 View all eligible organization to take cancelled campaign

- **Function trigger:** Admin clicks “Điều phối” button on tab “Chiến dịch đã hủy”.
- **Function description:**
  - **Actors:** Admin
  - **Purpose:** Admin wants to view all eligible organization to take cancelled campaign

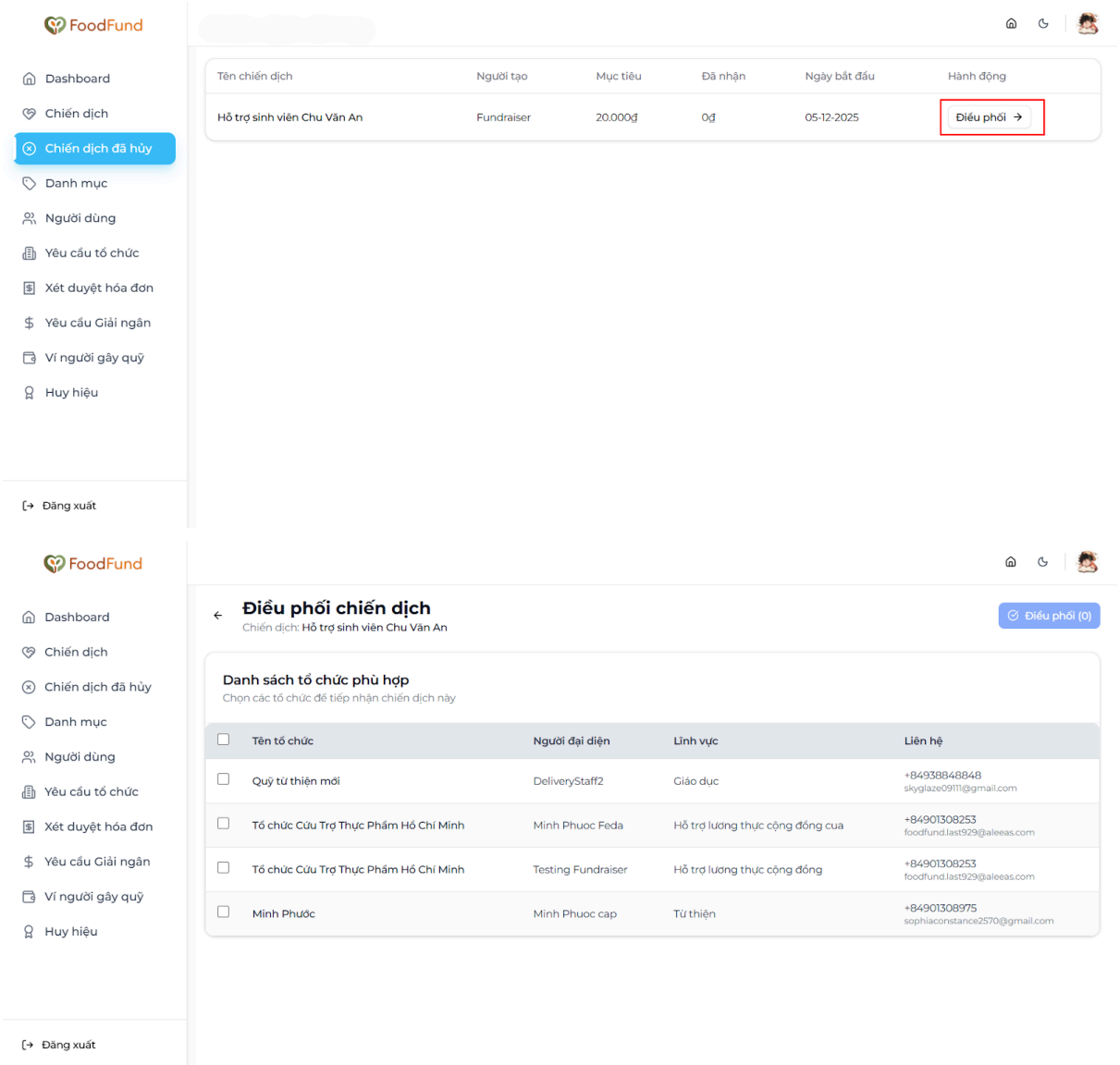


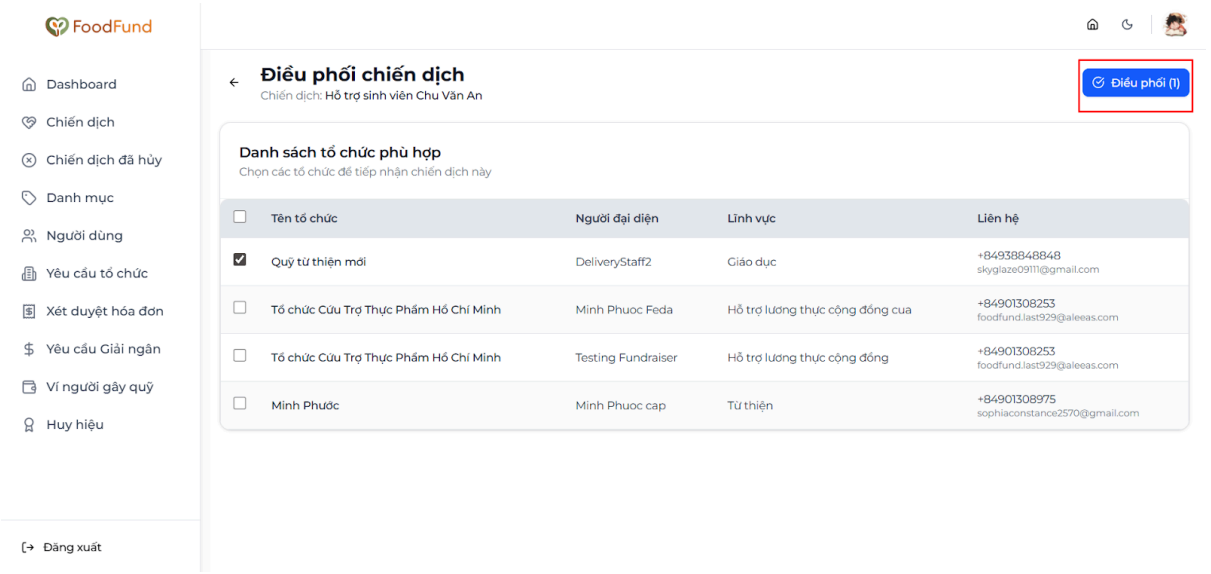
Figure 3.4.13: View all eligible organization to take cancelled campaign

- **Function details:**
  - **Data requirement:** campaignId
  - **Validation:** none
  - **Business rules:** BR-02, BR-31

3.4.14 Assign Campaign To Organizations

- **Function trigger:** Admin clicks radio box to choose organisations
- **Function description:**
  - **Actors:** Admin

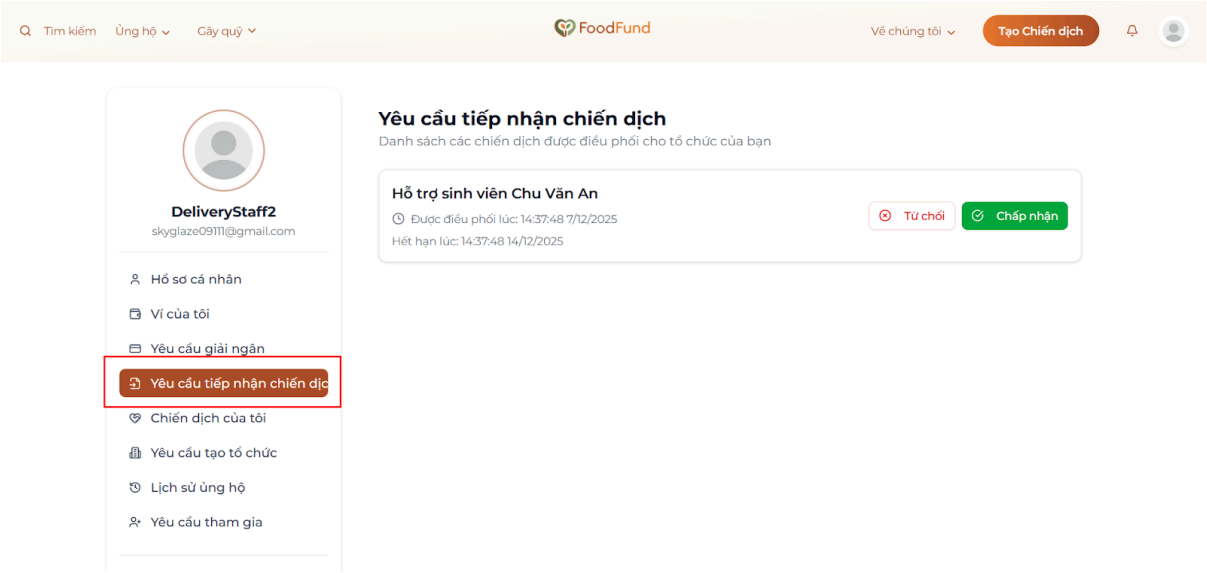
- **Purpose:** Admin wants to assign campaign to organizations



- **Function details:**
  - **Data requirement:** campaignId, organizationIds, reason
  - **Validation:**
    - CampaignId:
      - Type: String
    - OrganizationIds:
      - Type: Array
    - Reason:
      - Type: String
  - **Business rules:** BR-02, BR-30

3.4.15 View Assignment Requests

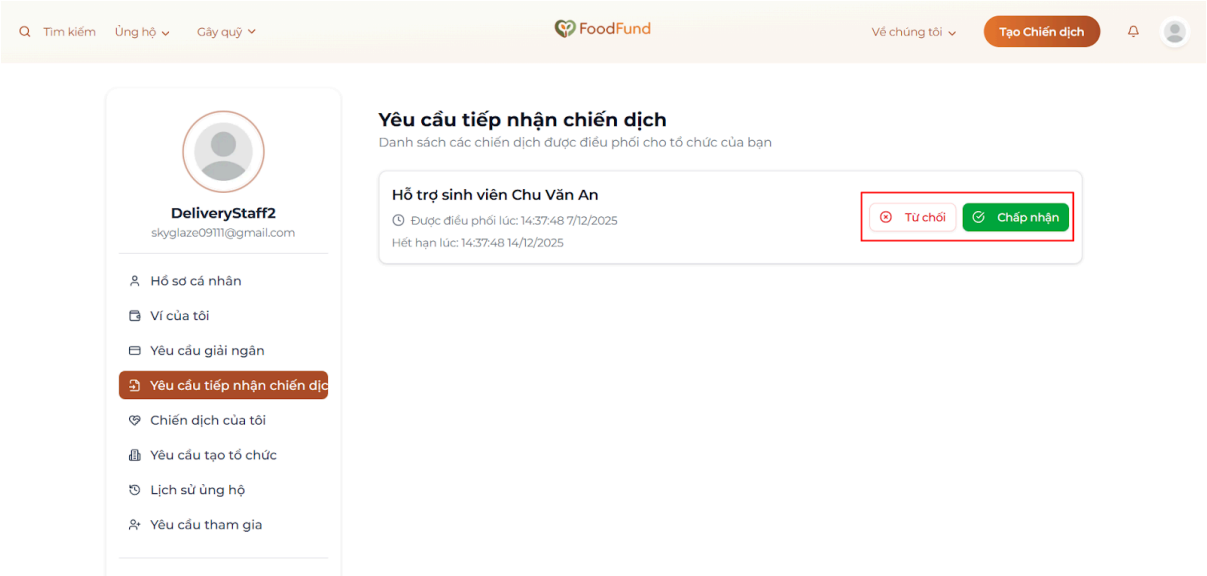
- **Function trigger:** Fundraiser clicks “Yêu cầu tiếp nhận chiến dịch” tab in profile page
- **Function description:**
  - **Actors:** Fundraiser
  - **Purpose:** Fundraiser wants to view assignment requests

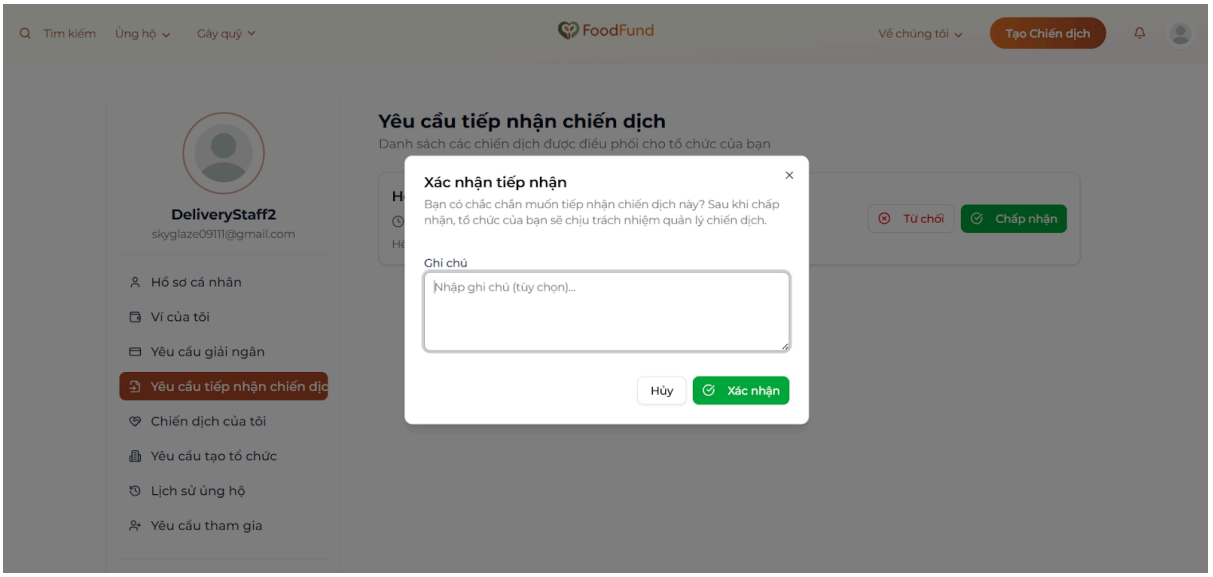


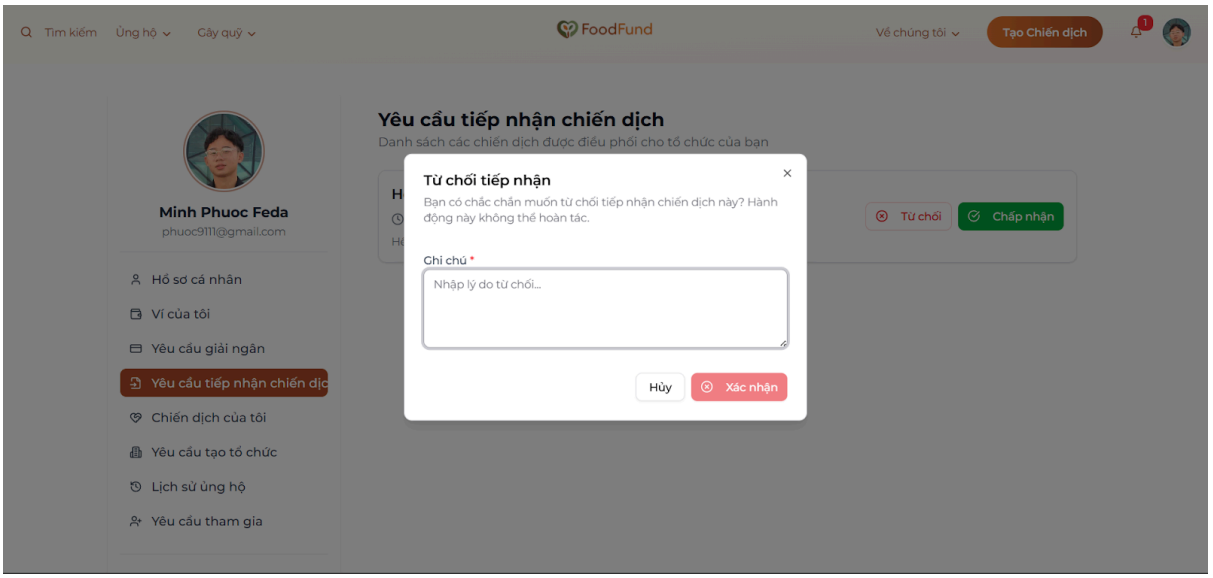
- **Function details:**
  - **Data requirement:** none
  - **Validation:** none
  - **Business rules:** BR-02, BR-32

3.4.16 Assignment Requests Approval

- **Function trigger:** Fundraiser clicks “Từ chối/ Chấp nhận” button in “Yêu cầu tiếp nhận chiến dịch” tab in profile page
- **Function description:**
  - **Actors:** Fundraiser
  - **Purpose:** Fundraiser wants to approve/reject assignment request





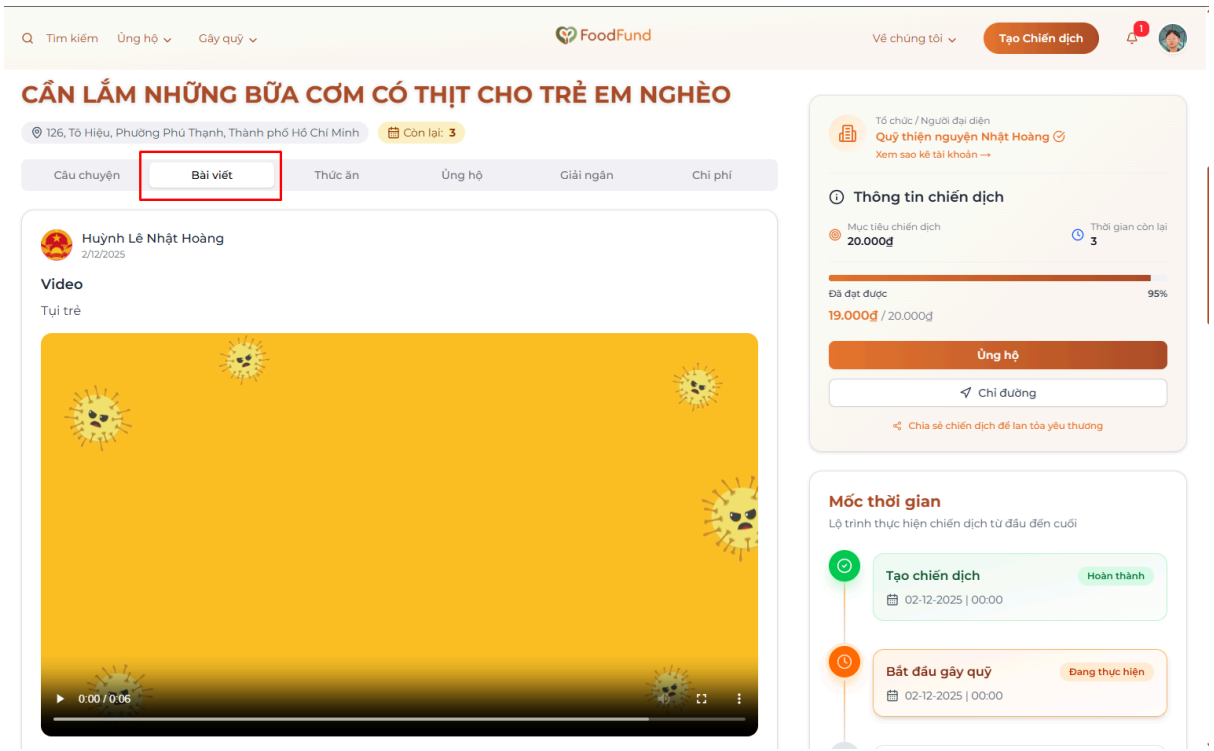


- **Function details:**
  - **Data requirement:** none
  - **Validation:** none
  - **Business rules:** BR-02, BR-19

3.5 Posts Of A Campaign

3.5.1 View All Posts Of A Campaign

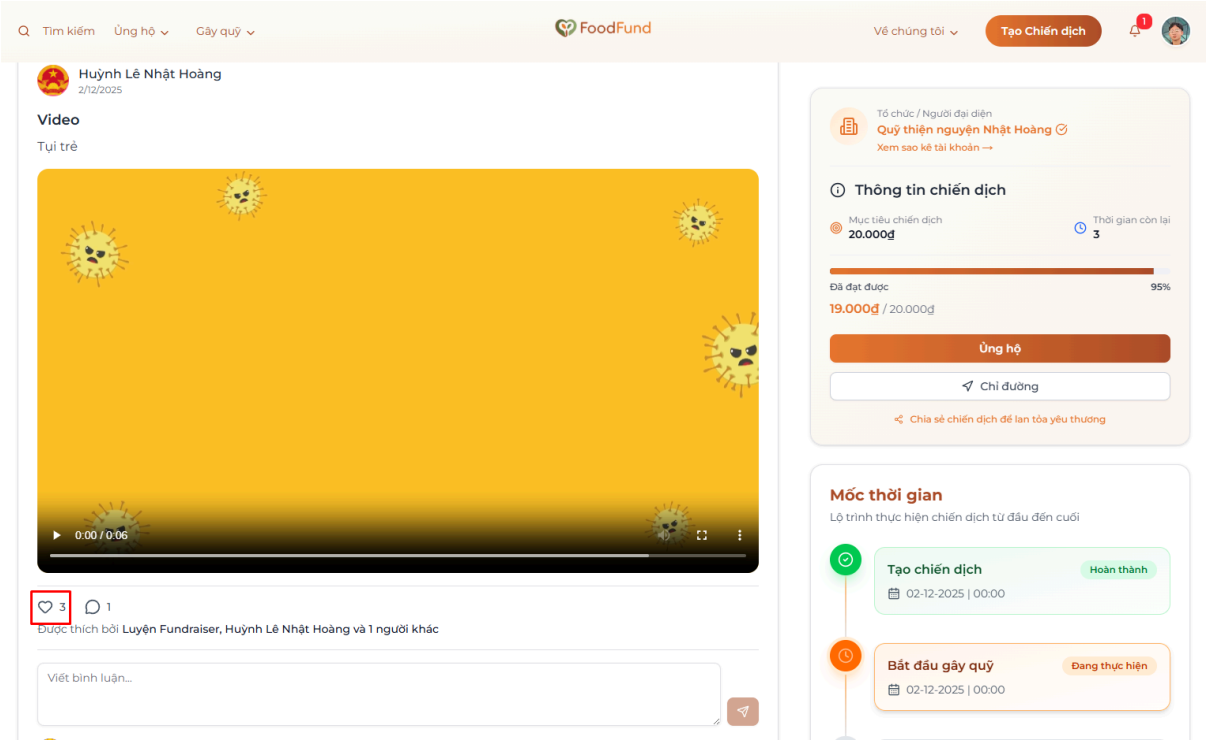
- **Function trigger:** Users access the post tab of the campaign detail page.
- **Function description:**
  - **Actors:** Guest, Donor, Fundraiser, Admin, Kitchen Staff, Delivery Staff
  - **Purpose:** Users view all posts of a campaign



- **Function details:**
  - **Data requirement:** campaign id, sort by, limit, offset
  - **Validation:**
    - Campaign id:
      - Type: string
      - Object id of campaign
    - Sort by:
      - Type: string
      - Optional
      - Default value: newest first
    - limit:
      - Type: int
      - Optional
      - Default value: 10
    - Offset:
      - Type: int
      - Optional
      - Default value: 0
  - **Business rules:** BR-02, BR-28

3.5.2 React on post

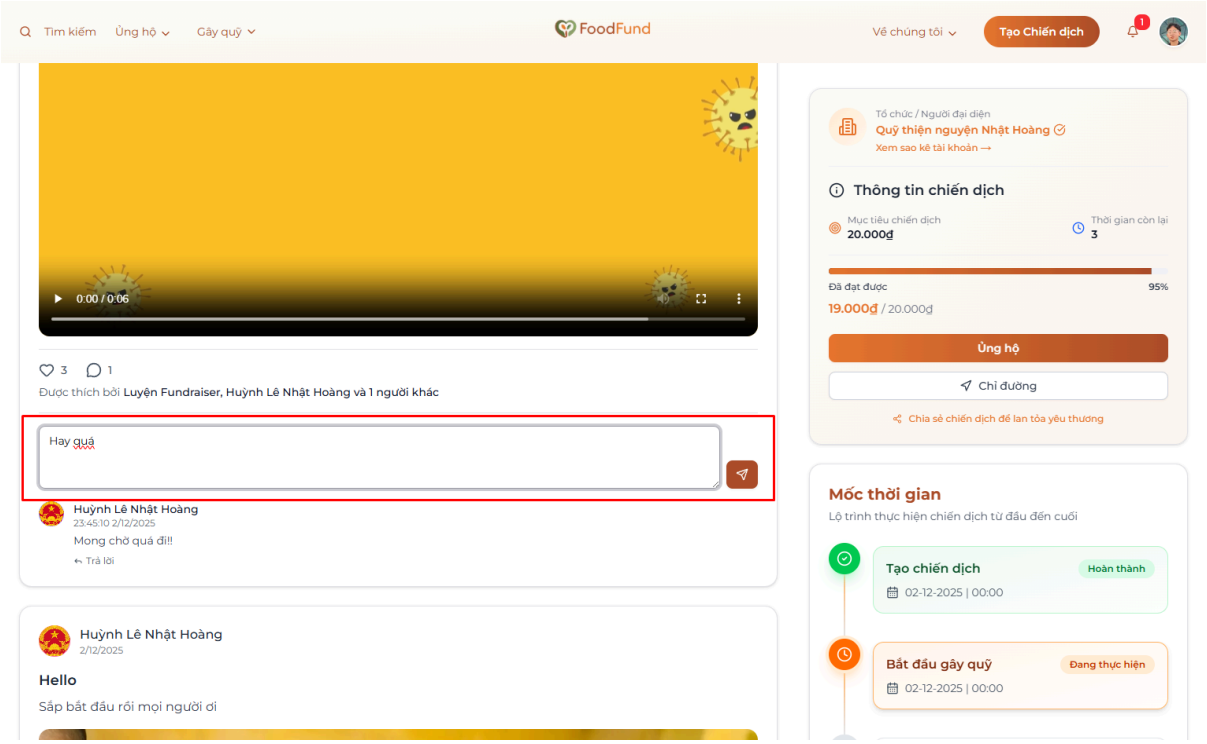
- **Function trigger:** User clicks “Heart icon” on the post tab of the campaign detail page.
- **Function description:**
  - **Actors:** Donor, Fundraiser, Admin, Kitchen Staff, Delivery Staff
  - **Purpose:** Users want to react on a post



- **Function details:**
  - **Data requirement:** post id
  - **Validation:**
    - post id:
      - Type: String
      - Object id of post item
  - **Business rules:** BR-02, BR-29

3.5.3 Comment Posts

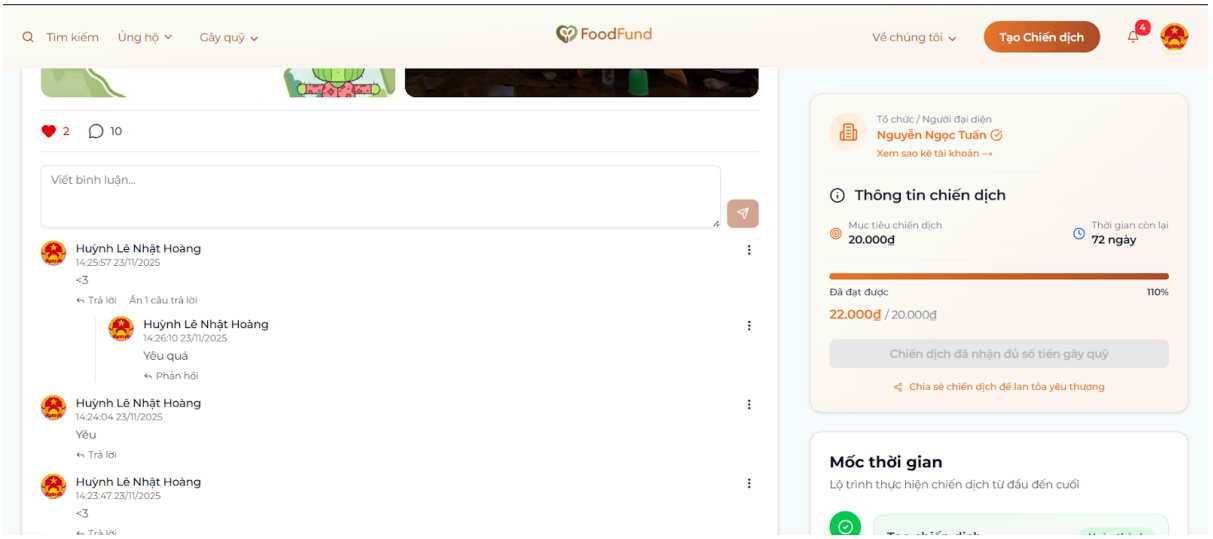
- **Function trigger:** User clicks “Comment icon” on the post tab of the campaign detail page.
- **Function description:**
  - **Actors:** Donor, Fundraiser, Admin, Kitchen Staff, Delivery Staff
  - **Purpose:** Users want to comment on a post



- **Function details:**
  - **Data requirement:** post id, content
  - **Validation:**
    - post id:
      - Type: String
      - Object id of post item
    - content:
      - Type: String
      - Min length: 1
  - **Business rules:** BR-02, BR-29

3.6.3 View Post Comments

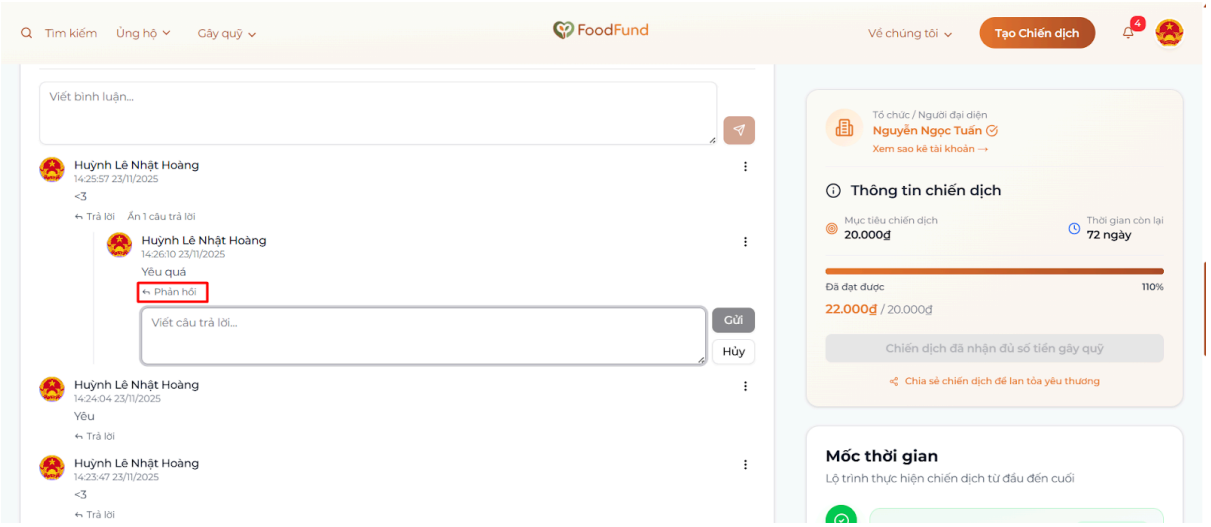
- **Function trigger:** Users view post comments on the post tab of the campaign detail page.
- **Function description:**
  - **Actors:** Guest, Donor, Fundraiser, Admin, Kitchen Staff, Delivery Staff
  - **Purpose:** Users want to view comments on a post



- **Function details:**
  - **Data requirement:** Id
  - **Validation:**
    - Id:
      - Type: String
      - Object Id of selected post
  - **Business rules:** BR-02, BR-29

3.6.4 Reply Post Comment

- **Function trigger:** Users click “Reply icon” on the post tab of the campaign detail page.
- **Function description:**
  - **Actors:** Donor, Fundraiser, Admin, Kitchen Staff, Delivery Staff
  - **Purpose:** Users want to reply comment on a post

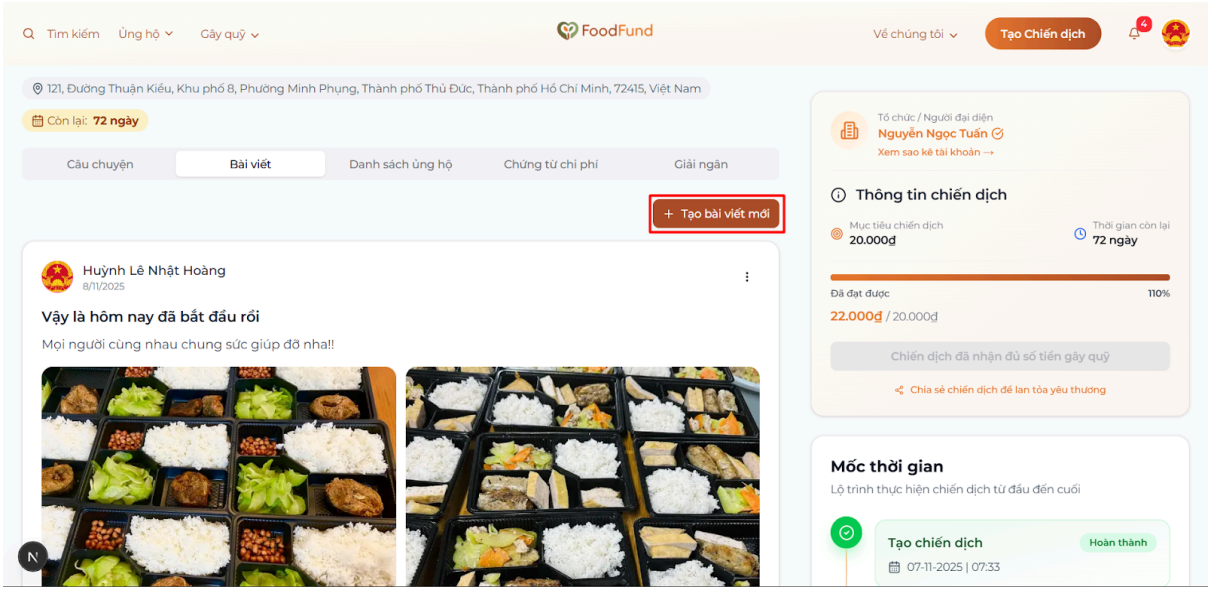


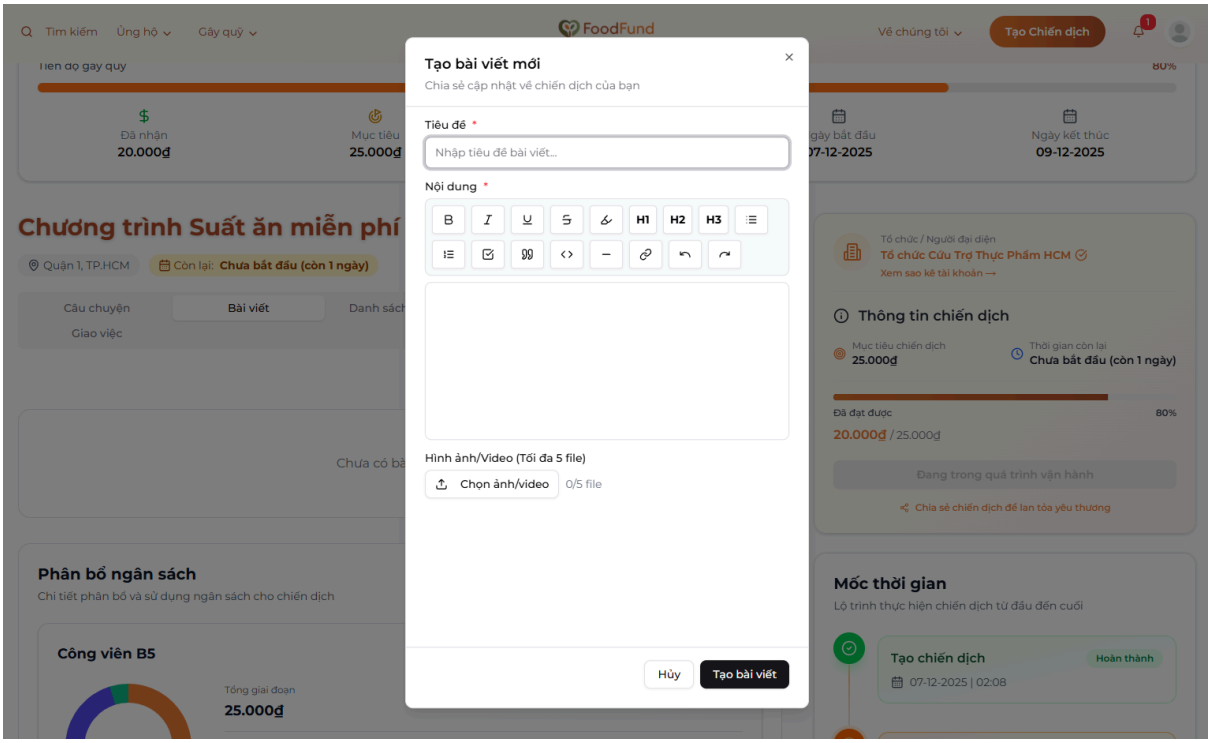
- **Function details:**
  - **Data requirement:** post id, content, parent comment id
  - **Validation:**
    - post id:

            - Type: String
            - Object id of selected post
        - parent comment id:
            - Type: String
            - Object id of comment
        - content:
            - Type: String
            - Min length: 1
    - **Business rules:** BR-02, BR-29

3.6.5 Create post of campaign

- **Function trigger:** Fundraiser clicks “Tạo bài viết” button on the post tab of the my campaign detail page.
- **Function description:**
    - **Actors:** Fundraiser
    - **Purpose:** Fundraiser wants to create a post

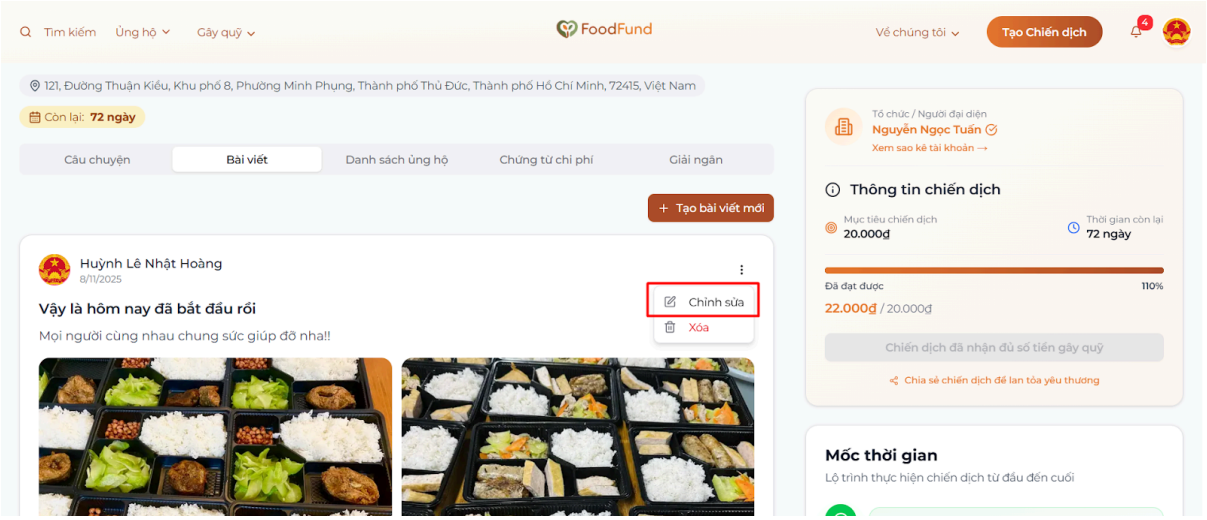


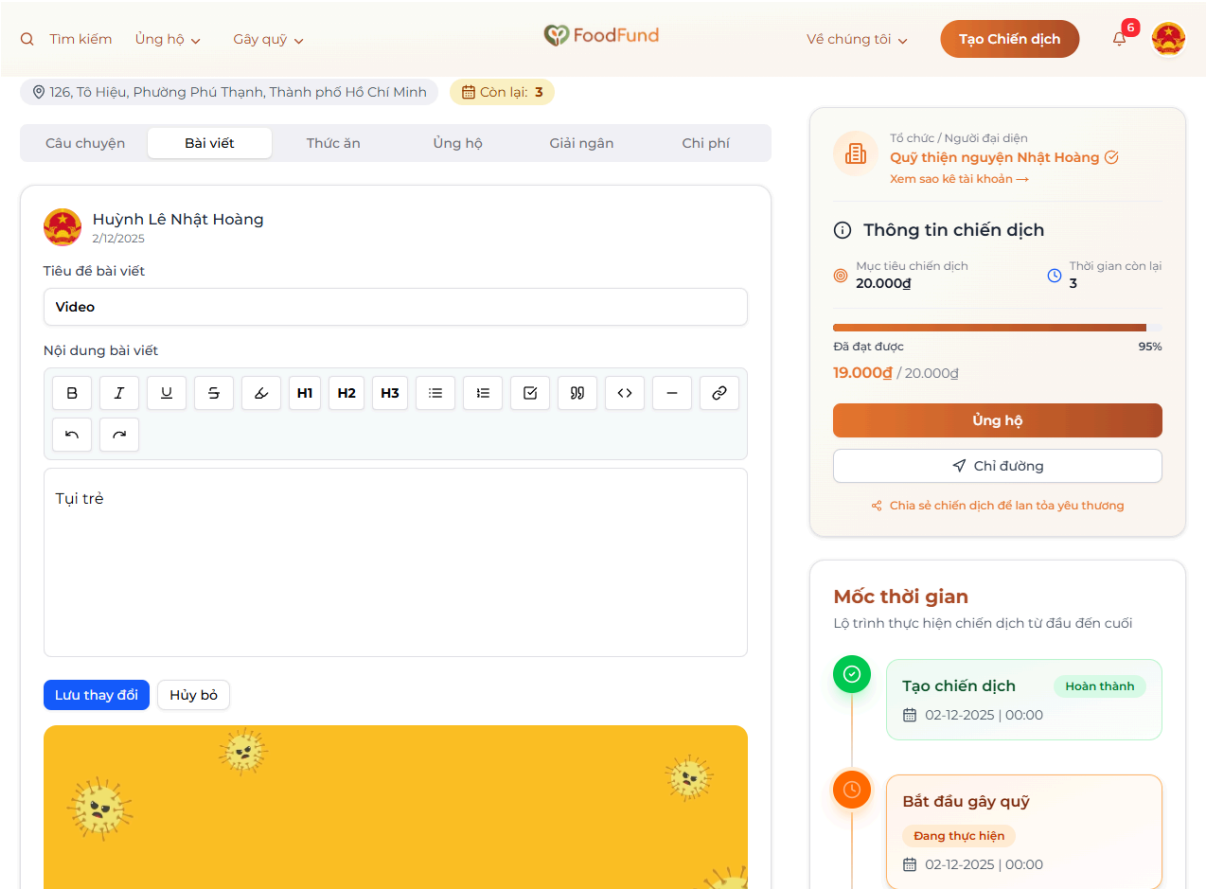


- **Function details:**
  - **Data requirement:** campaign id, title, content, media file keys
  - **Validation:**
    - campaign id:
      - Type: String
      - Object id of campaign
    - title:
      - Type: String
      - Length: 5 - 255
    - content:
      - Type: String
      - Min length: 10
    - media file keys:
      - Type: Array<String>
      - Array Size: 10
      - Optional
  - **Business rules:** BR-02, BR-27, BR-08

3.6.6 Edit post of campaign

- **Function trigger:** Fundraiser clicks “Chỉnh sửa” button on the post tab of the my campaign detail page.
- **Function description:**
  - **Actors:** Fundraiser
  - **Purpose:** Fundraiser wants to edit a post



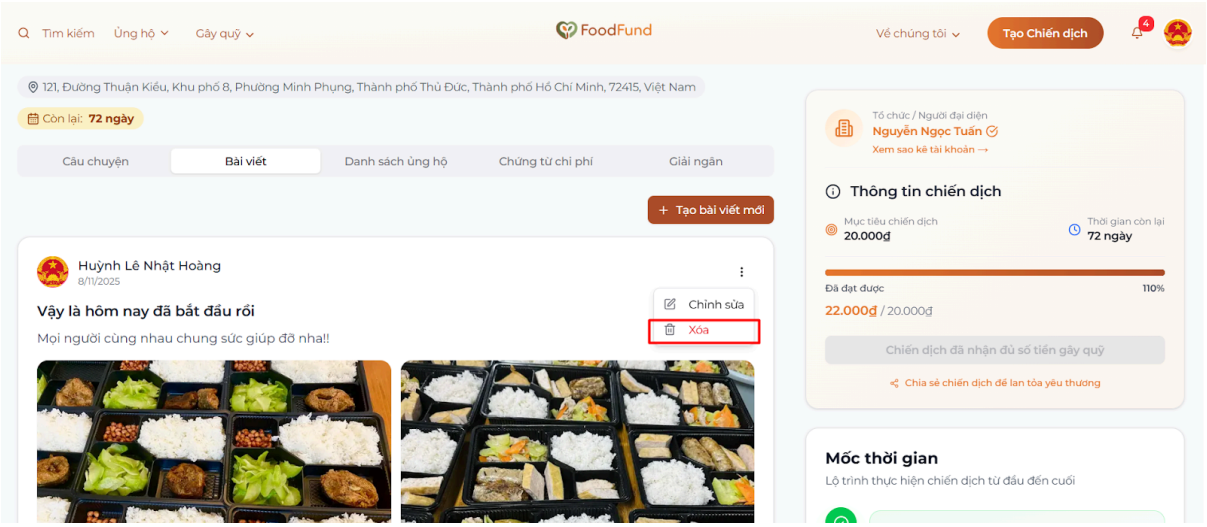


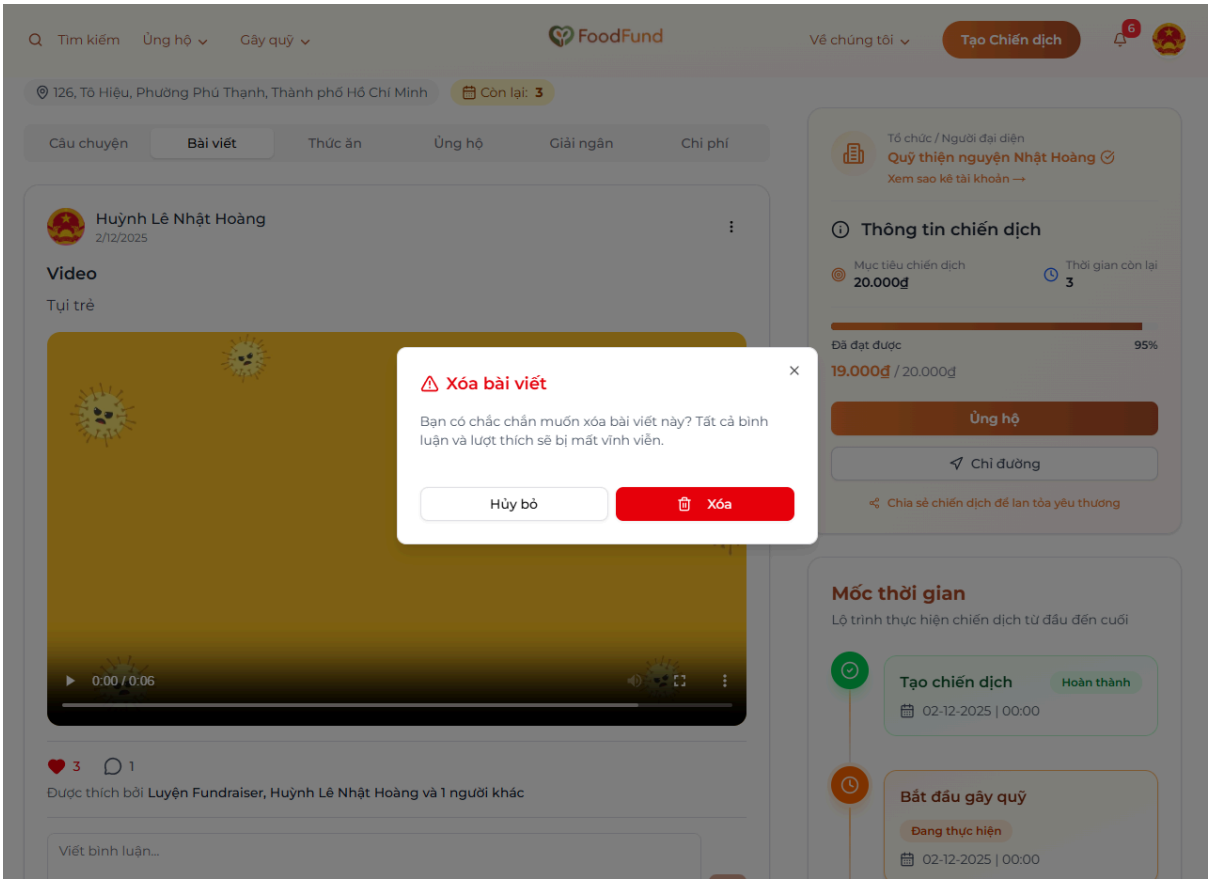
- **Function details:**
  - **Data requirement:** campaign id, title, content, media file keys
  - **Validation:**
    - campaign id:
      - Type: String
      - Object id of campaign
    - title:
      - Type: String
      - Length: 5 - 255
    - Content:

            - Type: String
            - Min length: 10
        - media file keys:
            - Type: Array<String>
            - Array Size: 10
            - Optional
    - **Business rules:** BR-02, BR-27, BR-08

3.6.7 Delete post of campaign

- **Function trigger:** Fundraiser clicks “Xóa” button on the post tab of the my campaign detail page.
- **Function description:**
    - **Actors:** Fundraiser
    - **Purpose:** Fundraiser wants to delete a post

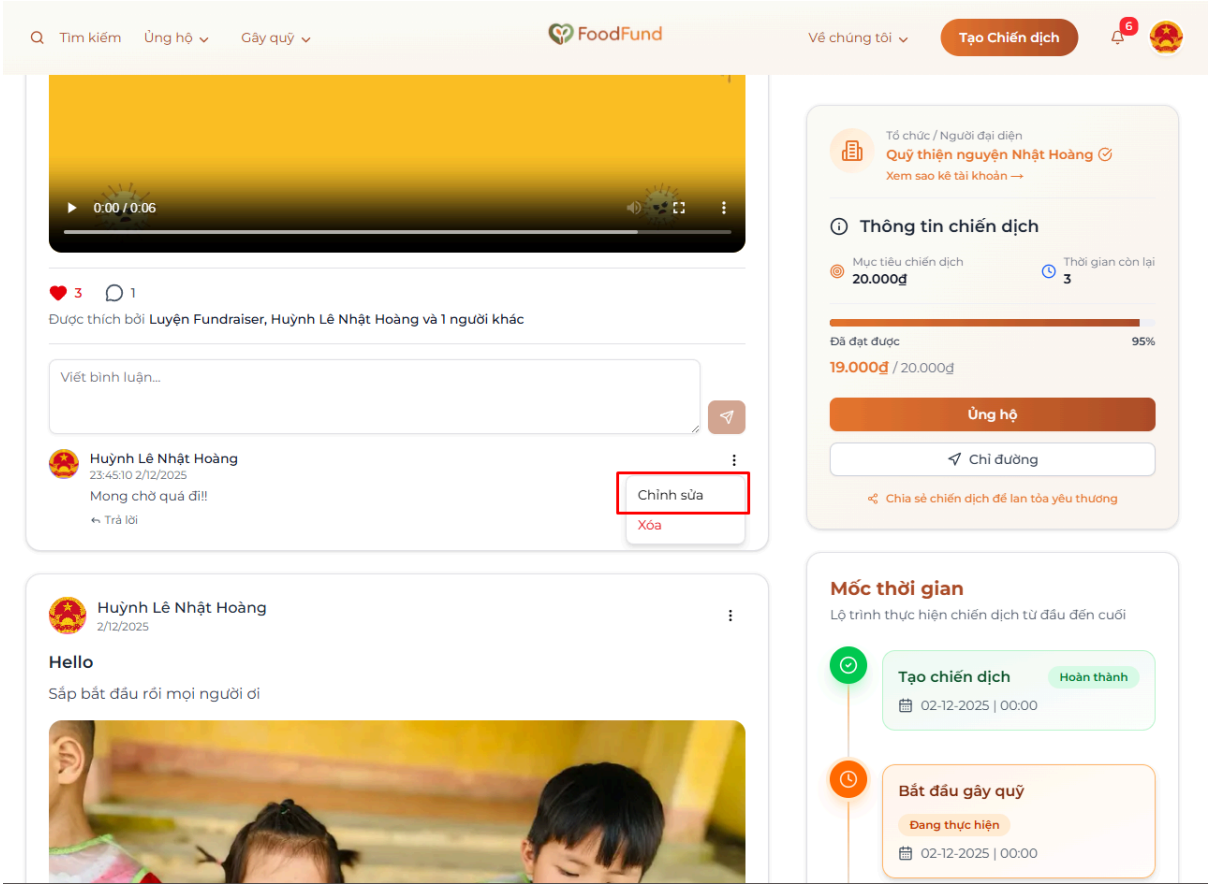


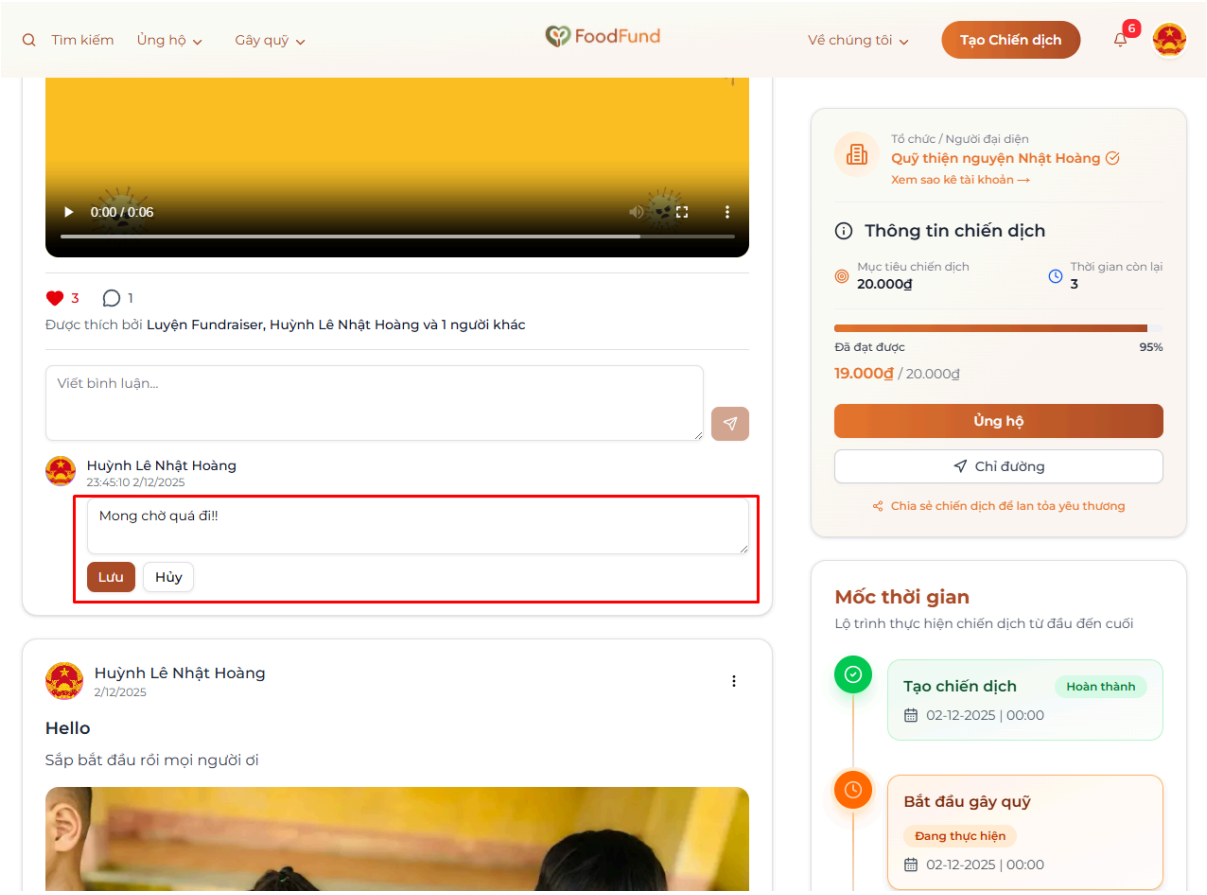


- **Function details:**
  - **Data requirement:** id
  - **Validation:** none
  - **Business rules:** BR-02, BR-27, BR-08

3.6.8 Update Comment

- **Function trigger:** Users click the “Chỉnh sửa” button on the post tab of the campaign detail page.
- **Function description:**
  - **Actors:** Donor, Fundraiser, Admin, Kitchen Staff, Delivery Staff
  - **Purpose:** Users want to update a comment



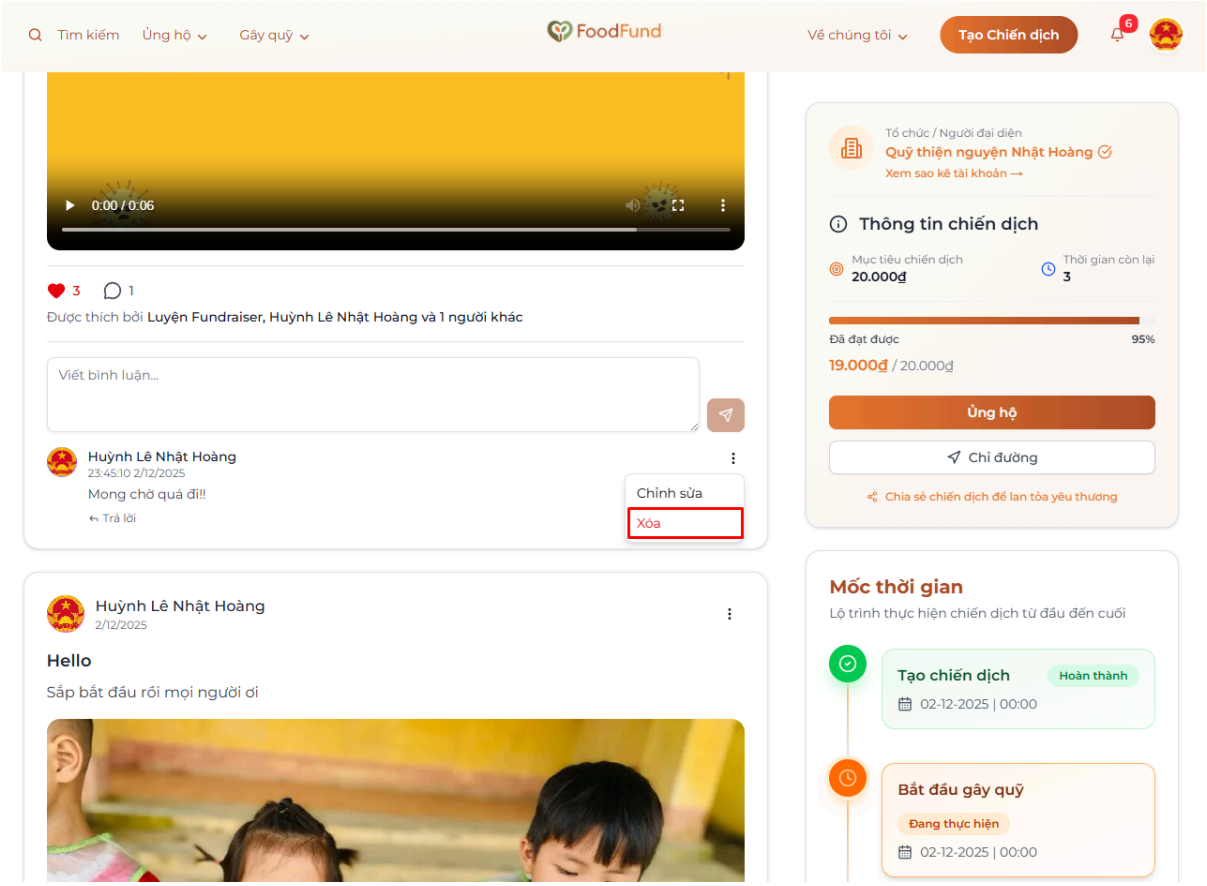


- **Function details:**

- **Data requirement:** commentId, content
- **Validation:**
  - CommentId:
    - Type: String
  - Content:
    - Type: String
- **Business rules:** BR-02

3.6.9 Delete Comment

- **Function trigger:** Users click the "Xóa" button on the post tab of the campaign detail page.
- **Function description:**
  - **Actors:** Donor, Fundraiser, Admin, Kitchen Staff, Delivery Staff
  - **Purpose:** Users want to delete a comment



- **Function details:**
  - **Data requirement:** commentId
  - **Validation:**
    - CommentId:
      - Type: String
  - **Business rules:** BR-02

3.7 Organisation

3.7.1 View all organizations

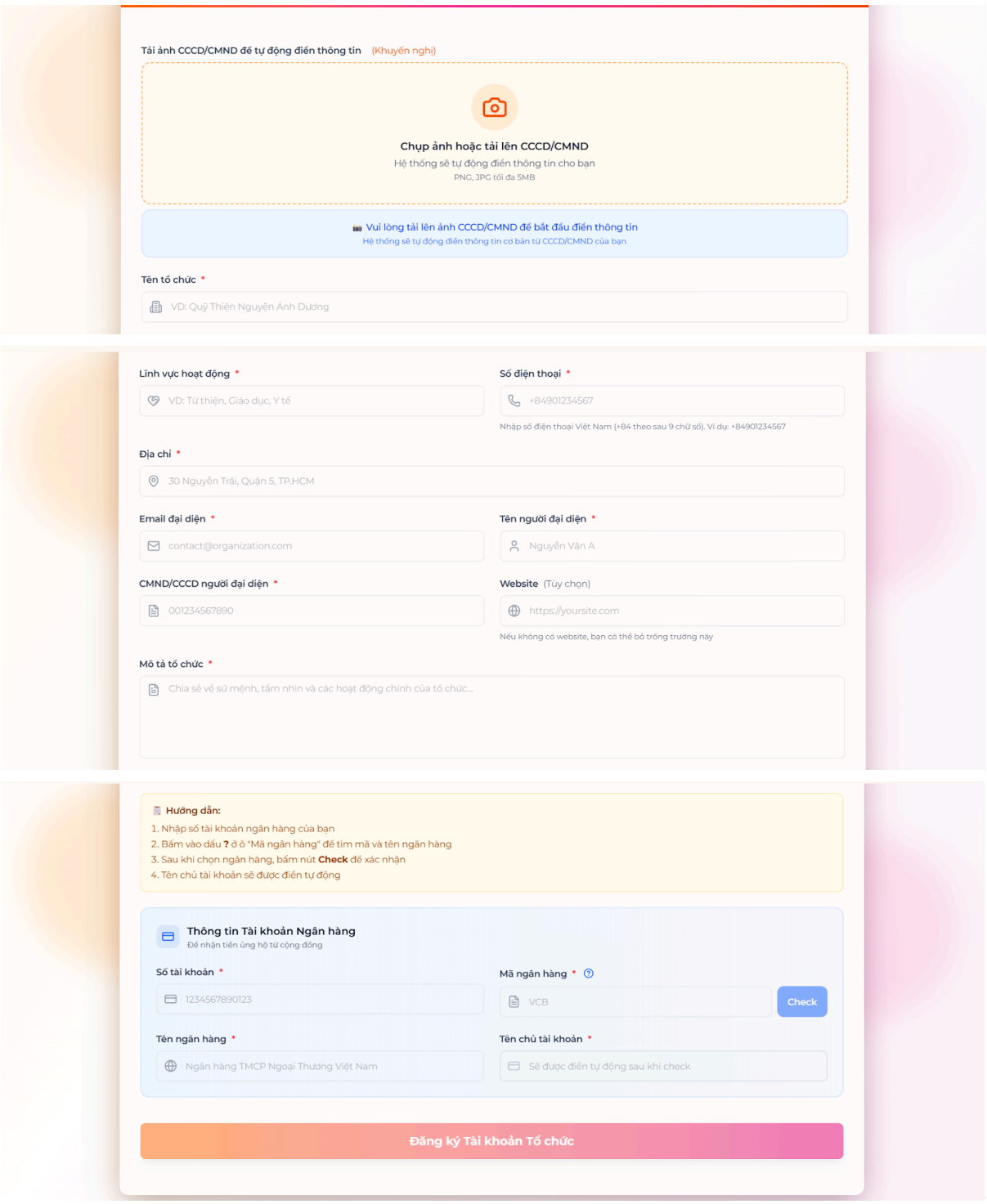
- **Function trigger:** User access organizations page.

- **Function description:**
  - **Actors:** Guest, Donor, Fundraiser, Admin, Kitchen Staff, Delivery Staff
  - **Purpose:** User want to view all organizations

- **Function details:**
  - **Data requirement:** none
  - **Validation:** none
  - **Business rules:** none

3.7.2 Create organization

- **Function trigger:** Fundraiser click "Tạo tổ chức" button on create organization page.
- **Function description:**
  - **Actors:** Fundraiser
  - **Purpose:** Fundraiser want to create an organization

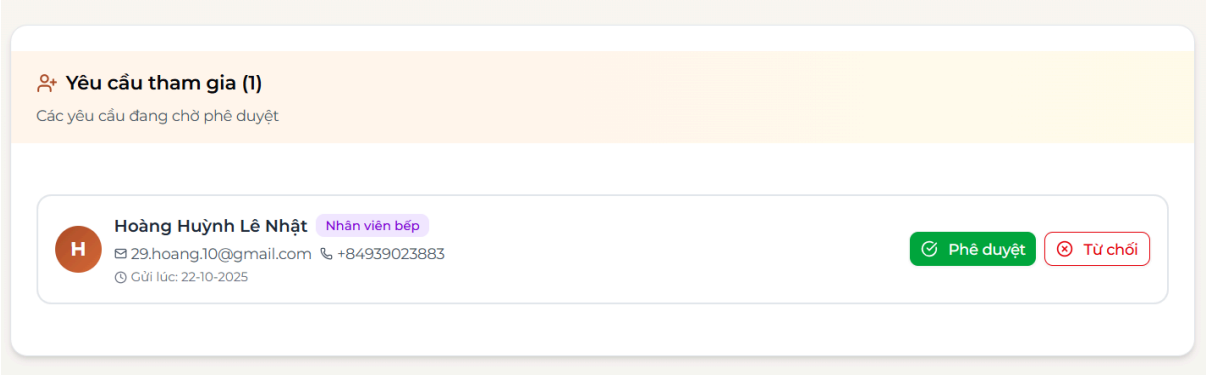


- **Function details:**
  - **Data requirement:** activity_field, address, description, email, name, phone_number, representative_identity_number, representative_name, website, bank_account_name, bank_account_number, bank_name, bank_short_name
  - **Validation:**
    - Activity_Field:
      - Type: String
    - Address:

            - Type: String
        - Description:
            - Type: String
        - Email:
            - Type: String
            - Must be valid email type, unique and not include blank.
        - Name:
            - Type: String
        - Phone_Number:
            - Type: String
            - Must be valid phone type
        - Representative_Identity_number:
            - Type: String
        - Representative_name:
            - Type: String
        - Website:
            - Type: String
            - Optional
        - Bank_Account_Name:
            - Type: String
        - Bank_Account_Number:
            - Type: String
        - Bank_Short_Name:
            - Type: String
    - **Business rules:** BR-02, BR-09, BR-06, BR-15, BR-11

3.7.3 Approve/Reject join organization request

- **Function trigger:** Fundraiser click "Duyệt/Từ chối" button on my organization detail page.
- **Function description:**
    - **Actors:** Fundraiser
    - **Purpose:** Fundraiser want to approve/ reject a join organization request



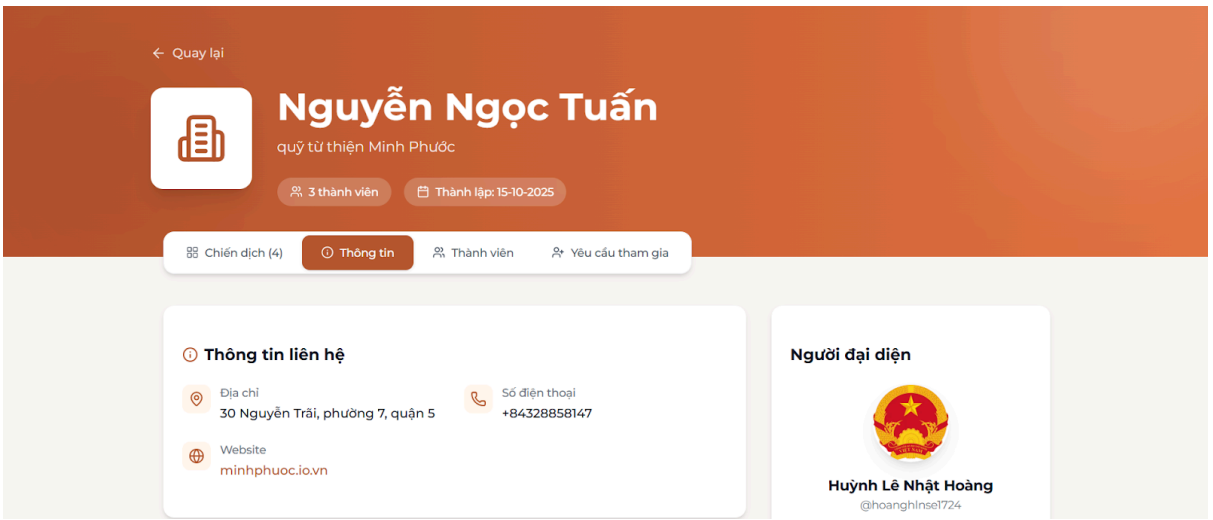
- **Function details:**
    - **Data requirement:** requestId
    - **Validation:**
        - requestId

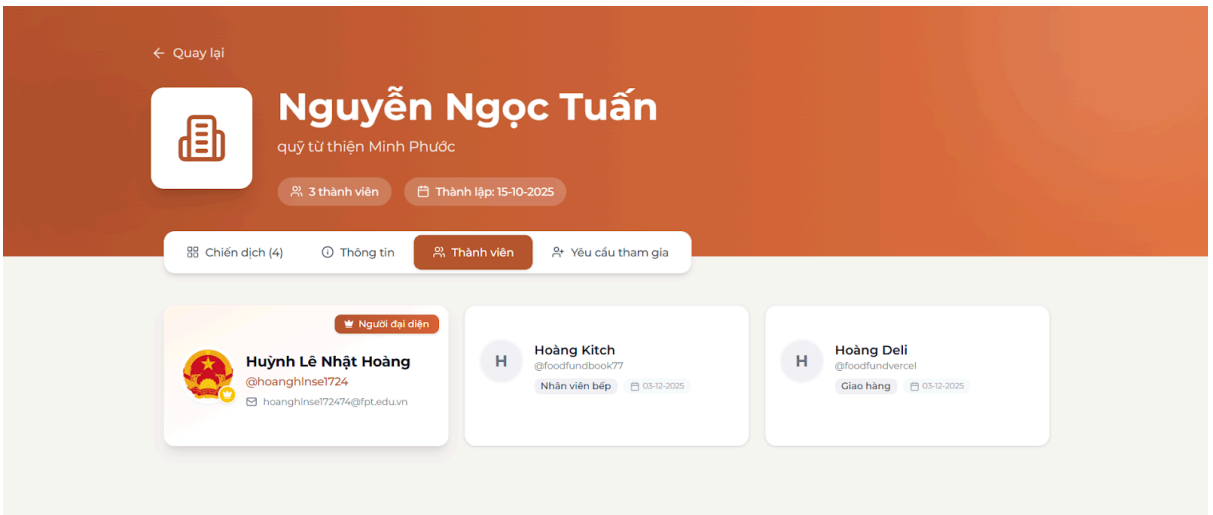
- Type: String
- requestId must be uuid

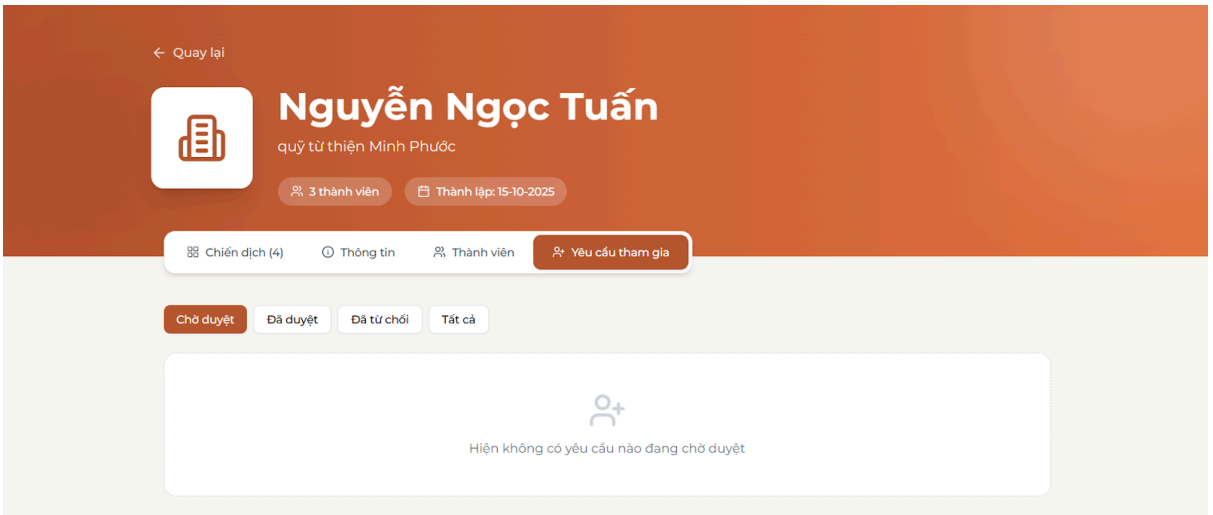
  - **Business rules:** BR-02, BR-07

3.7.4 View my organization

- **Function trigger:** Fundraiser click “Xem chi tiết” button on my organization tab of profile page.
- **Function description:**
  - **Actors:** Fundraiser
  - **Purpose:** Fundraiser want to view their organization



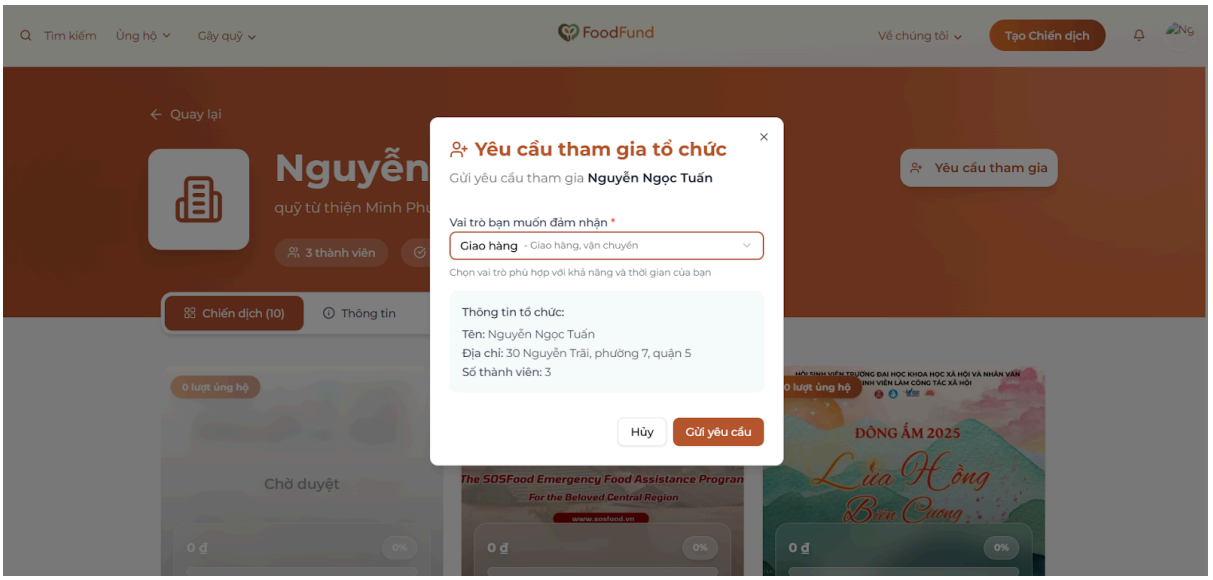




- **Function details:**
  - **Data requirement:** null
  - **Validation:** none
  - **Business rules:** BR-02, BR-08

3.7.5 Request join organization

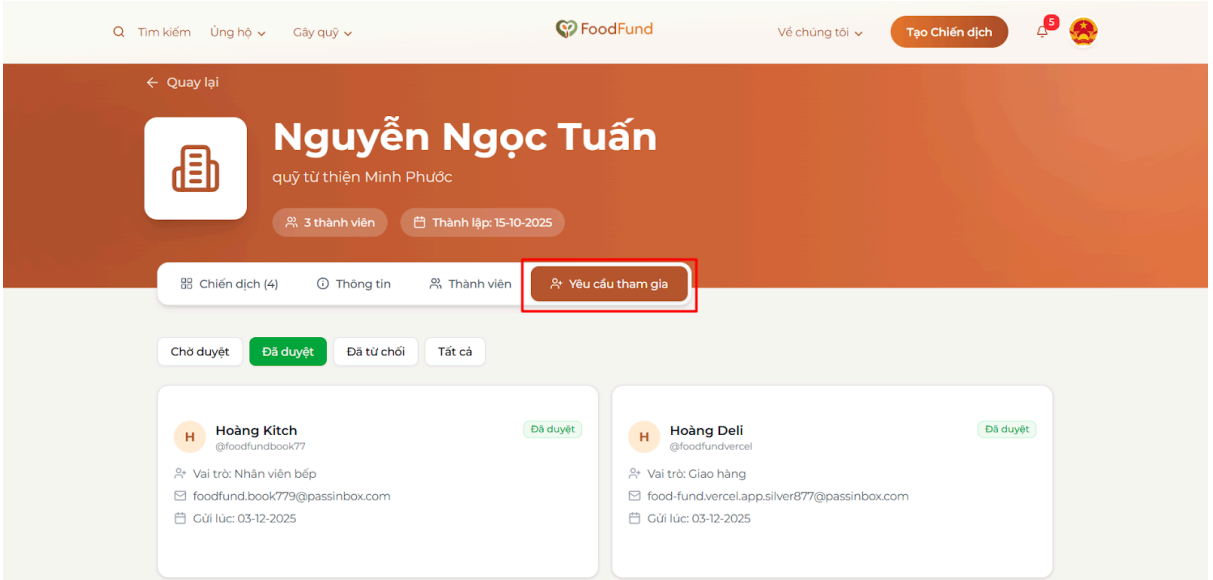
- **Function trigger:** Donor click “Yêu cầu tham gia” button on featured organization page.
- **Function description:**
  - **Actors:** Donor
  - **Purpose:** Donor want to join an organization



- **Function details:**
  - **Data requirement:** organization_id, requested_role
  - **Validation:**
    - organization_id:
      - Type: String
      - Organization_id must be uuid
    - requested_role:
      - Type: String
      - Role must be one of: KITCHEN_STAFF, DELIVERY_STAFF
  - **Business rules:** BR-02, BR-10

3.7.6 View organization join requests

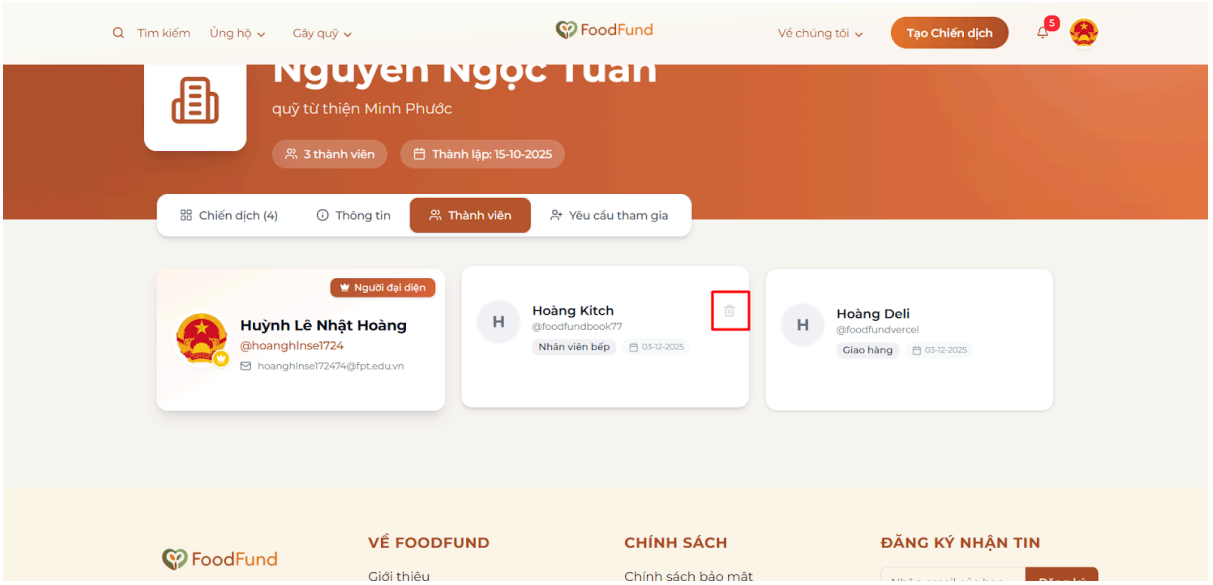
- **Function trigger:** Fundraiser click “Yêu cầu tham gia” tab on my organization page.



- **Function description:**
  - **Actors:** Fundraiser
  - **Purpose:** Fundraiser want to view all organization join requests
- **Function details:**
  - **Data requirement:** organization_id, requested_role
  - **Validation:**
    - Organization_id:
      - Type: String
  - **Business rules:** BR-02, BR-10

3.7.7 Remove staff

- **Function trigger:** Fundraiser click "Yêu cầu tham gia" button on featured organization page.

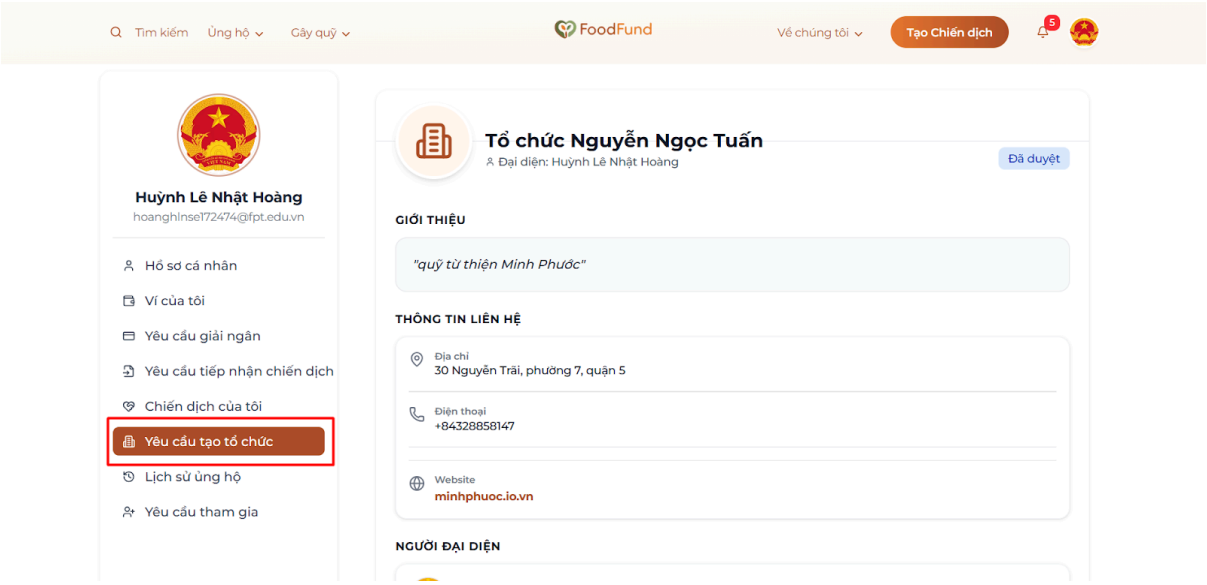


- **Function description:**
  - **Actors:** Fundraiser
  - **Purpose:** Fundraiser want to remove staff
- **Function details:**

- **Data requirement:** memberId
- **Validation:**
  - MemberId:
    - Type: String
- **Business rules:** BR-02, BR-18

3.7.8 View my create organization request

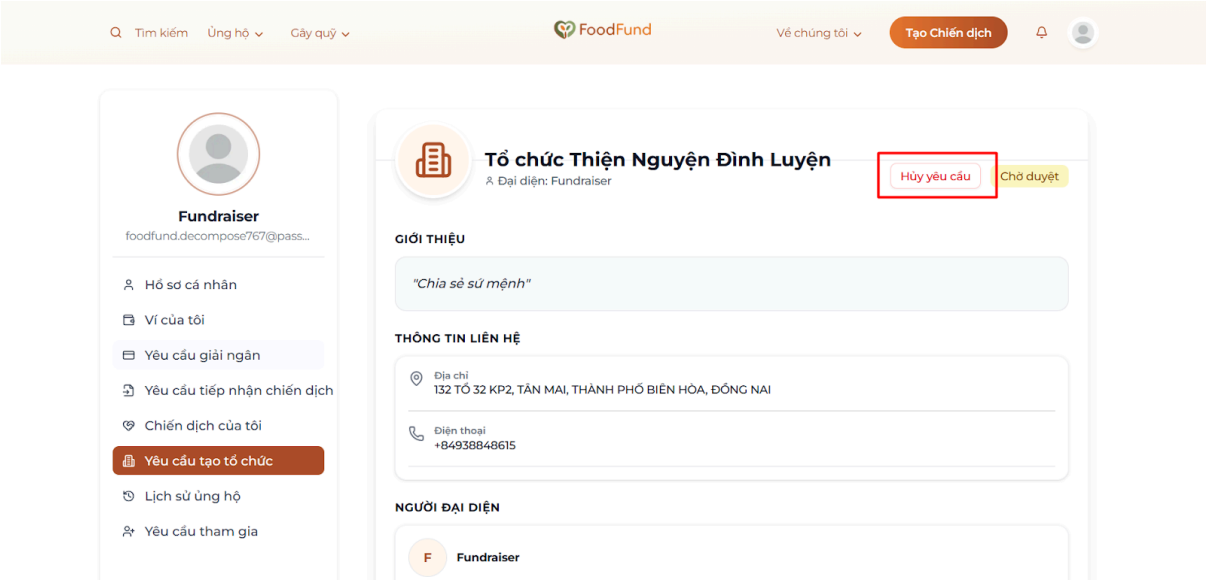
- **Function trigger:** Donor click “Yêu cầu tạo tổ chức” tab on profile page.



- **Function description:**
  - **Actors:** Donor
  - **Purpose:** Donor want to view their create organization request
- **Function details:**
  - **Data requirement:** null
  - **Validation:** null
  - **Business rules:** BR-02, BR-08

3.7.9 Cancel create organization request

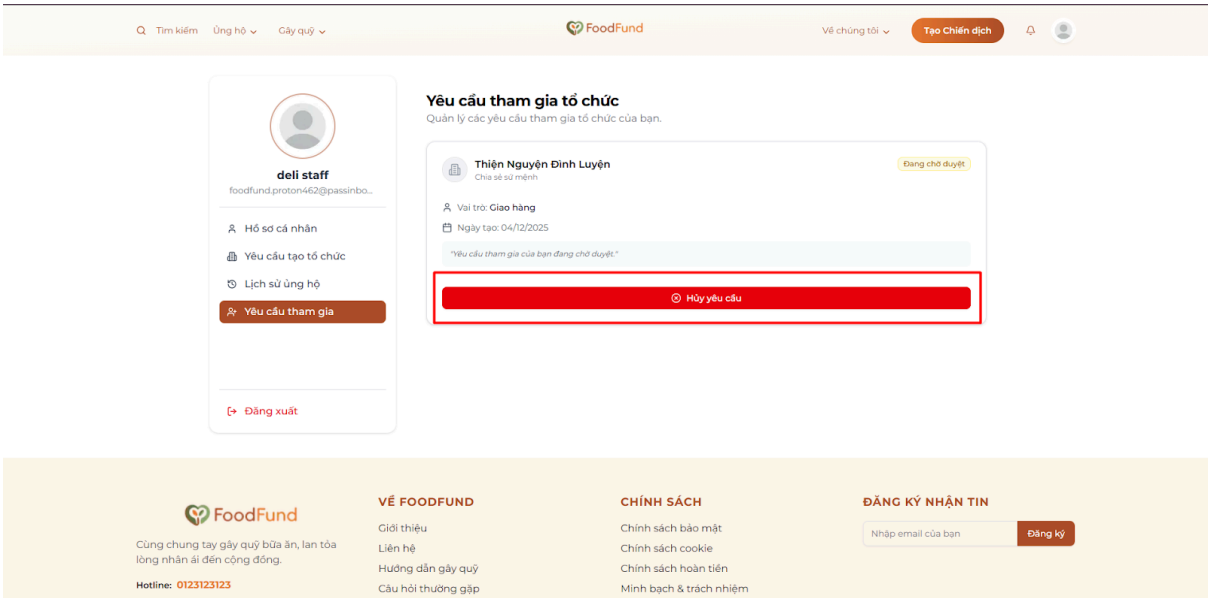
- **Function trigger:** Donor click “Hủy yêu cầu” button on tab “Yêu cầu tạo tổ chức” in profile page.
- **Function description:**
  - **Actors:** Donor
  - **Purpose:** Donor want to cancel create organization request



- **Function details:**
  - **Data requirement:** null
  - **Validation:** null
  - **Business rules:** BR-02

3.7.10 Cancel join organization request

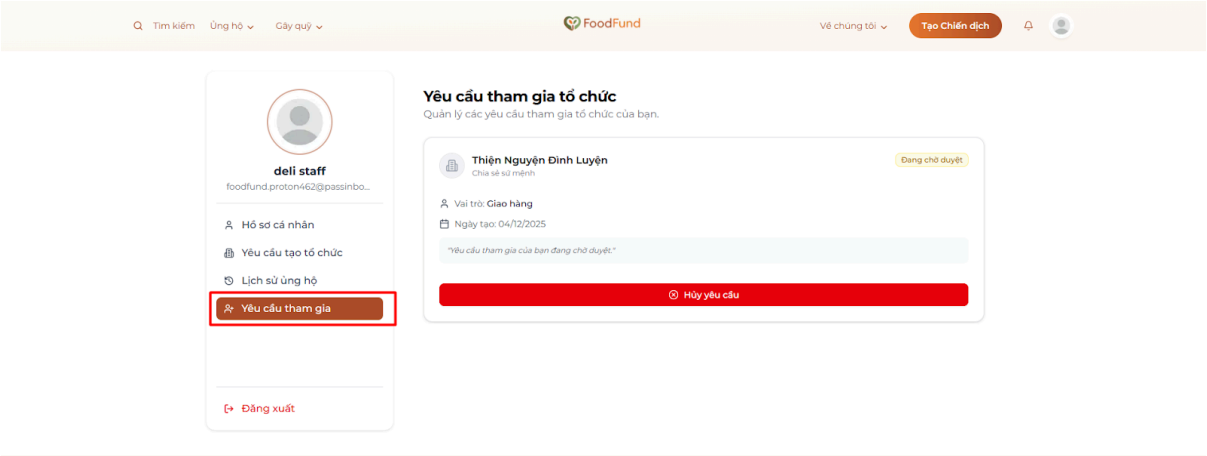
- **Function trigger:** Donor click "Hủy yêu cầu" button in "Yêu cầu tham gia" tab on profile page.
- **Function description:**
  - **Actors:** Donor
  - **Purpose:** Donor want cancel join organization request



- **Function details:**
    - **Data requirement:** requestId
    - **Validation:**
        - RequestId:
            - Type: String
    - **Business rules:** BR-02

3.7.11 View my join request

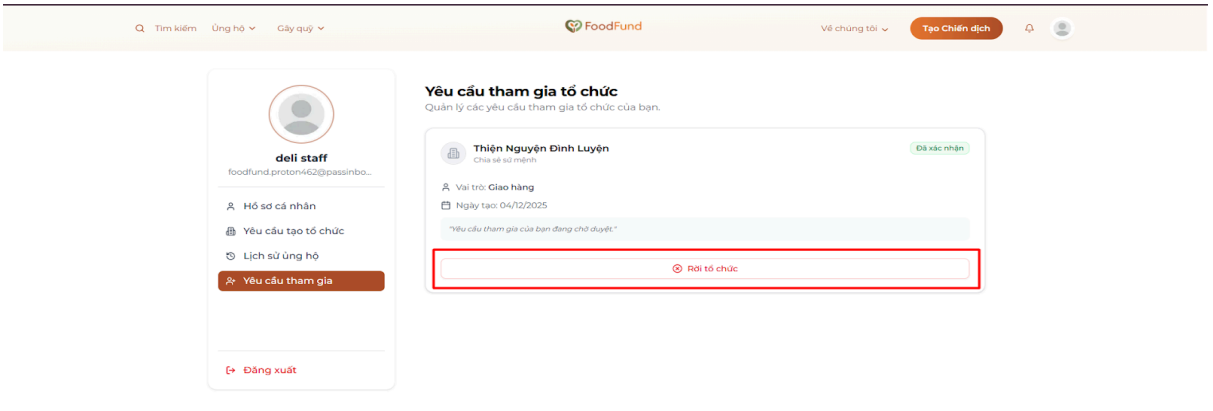
- **Function trigger:** Donor click “Yêu cầu tham gia” tab on profile page.
- **Function description:**
    - **Actors:** Donor
    - **Purpose:** Donor want to view their join request



- **Function details:**
    - **Data requirement:** null
    - **Validation:** null
    - **Business rules:** BR-02

3.7.12 Leave organization

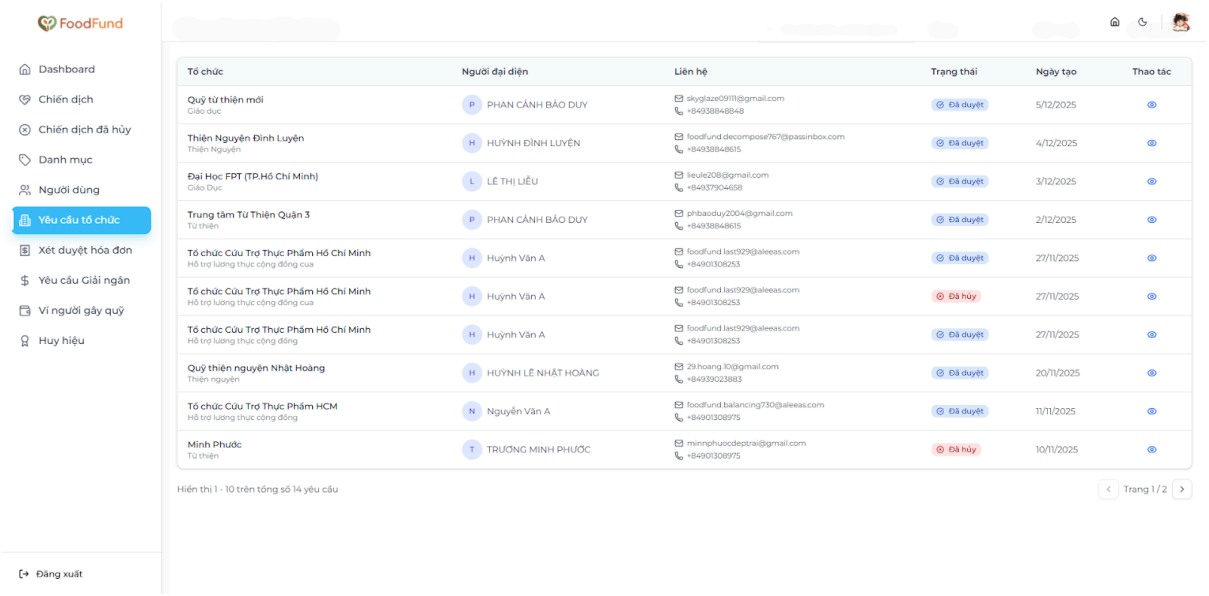
- **Function trigger:** Staff click “Rời tổ chức” button on “Yêu cầu tham gia” tab in profile page.
- **Function description:**
    - **Actors:** Staff
    - **Purpose:** Staff want to leave their organization

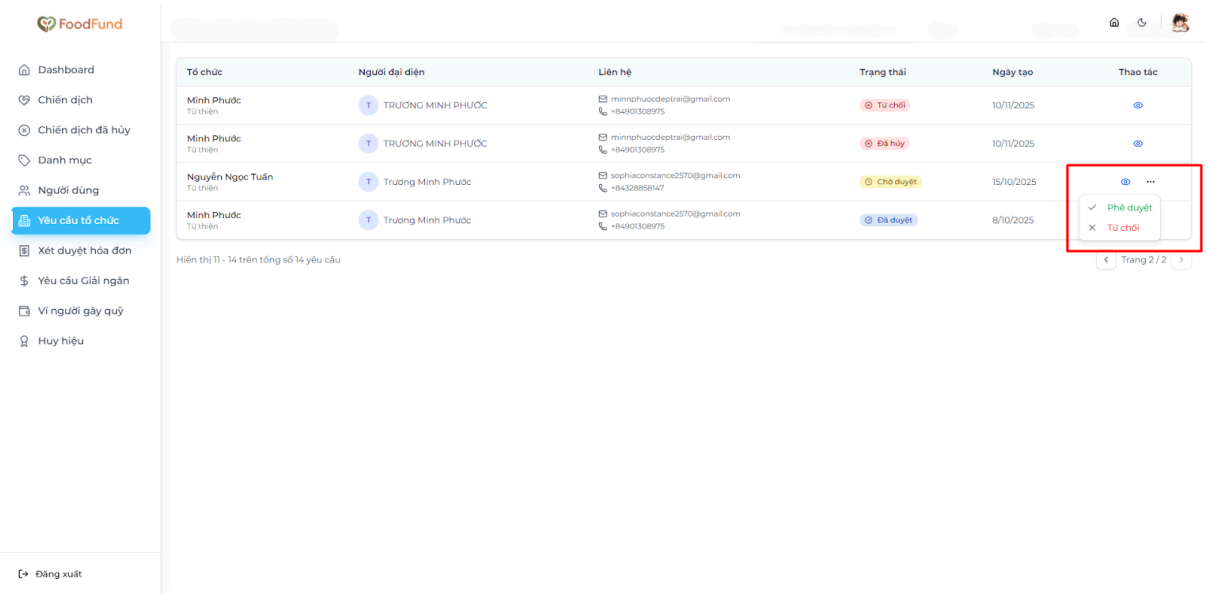


- **Function details:**
  - **Data requirement:** null
  - **Validation:** null
  - **Business rules:** BR-02, BR-10

3.7.13 Approval create organization request

- **Function trigger:** Admin clicks the “Yêu cầu tổ chức” tab on admin page.
- **Function description:**
  - **Actors:** Admin
  - **Purpose:** Admin want to approve/reject create organization request



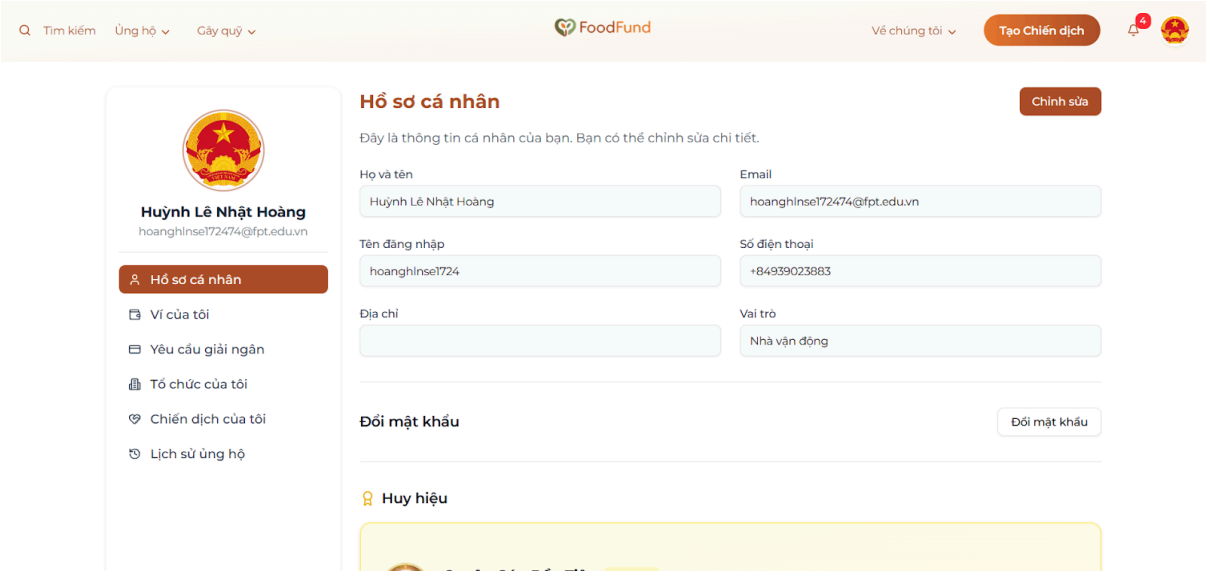


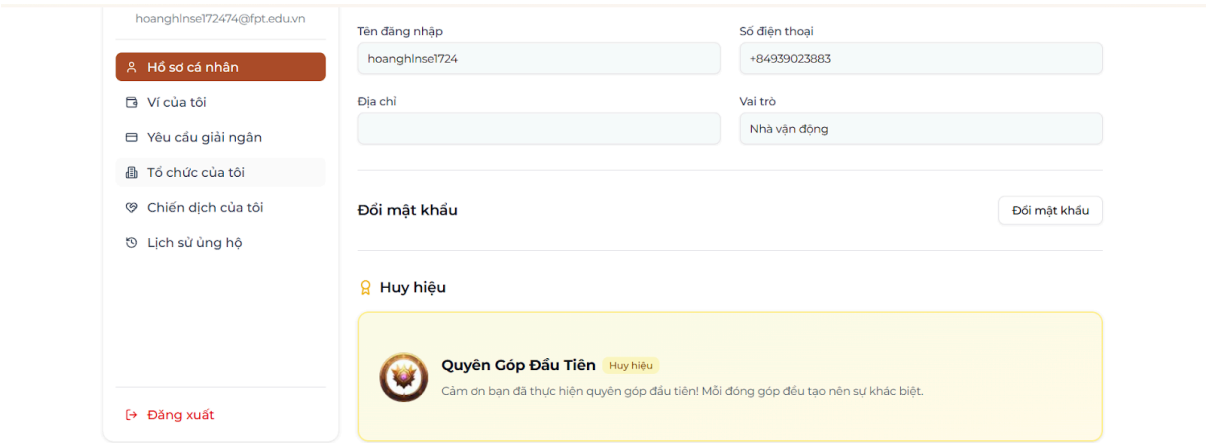
- **Function details:**
  - **Data requirement:** organizationId
  - **Validation:**
    - OrganizationId:
      - Type: String
  - **Business rules:** BR-02, BR-07

3.8 Profile

3.8.1 View Account Profile Detail

- **Function trigger:** Users access profile page.
- **Function description:**
  - **Actors:** Donor, Fundraiser, Kitchen Staff, Delivery Staff, Admin
  - **Purpose:** Users want to view their profile

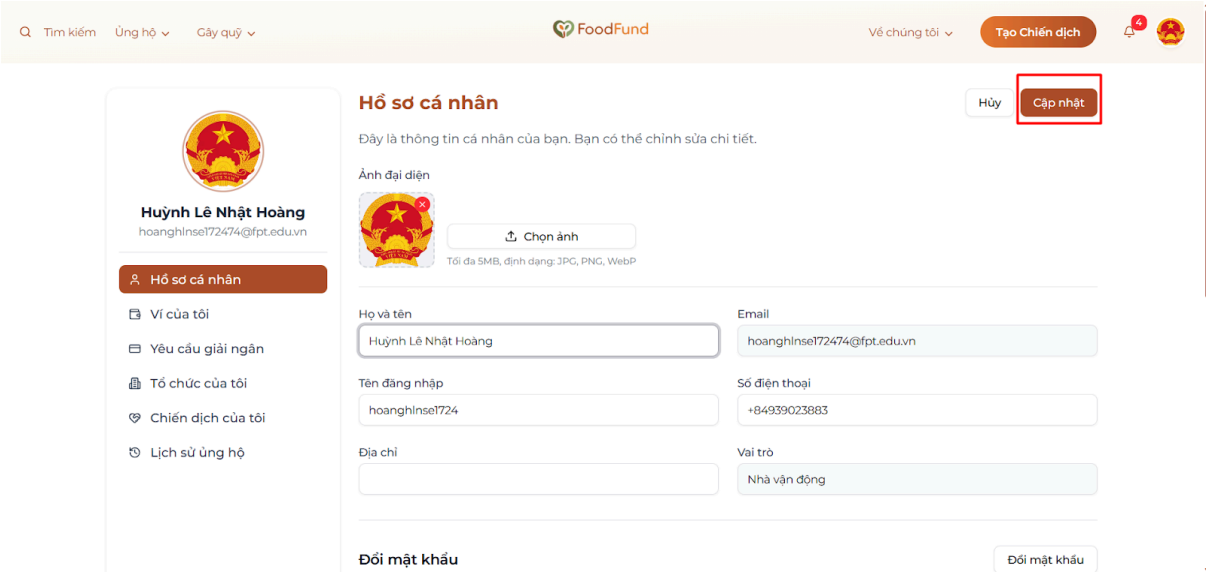




- **Function details:**
  - **Data requirement:** null
  - **Validation:** none
  - **Business rules:** BR-02

3.8.2 Edit account profile

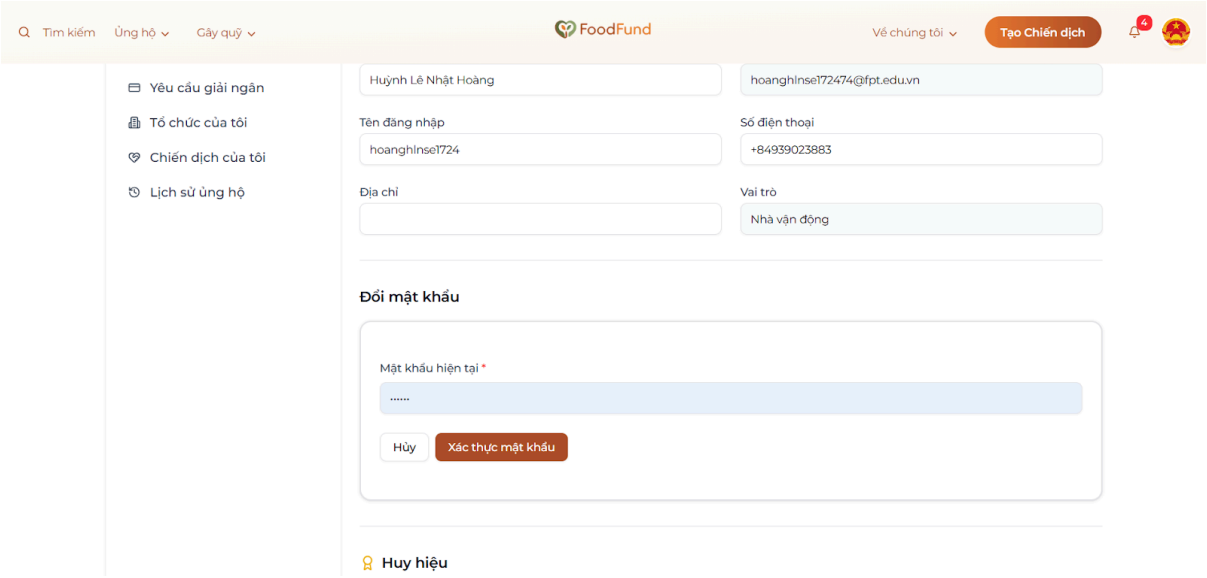
- **Function trigger:** User click "Chỉnh sửa hồ sơ" button on profile page.
- **Function description:**
  - **Actors:** Donor, Fundraiser, Kitchen Staff, Delivery Staff, Admin
  - **Purpose:** Donor, Fundraiser, Kitchen Staff, Delivery Staff, Admin want to edit their profile

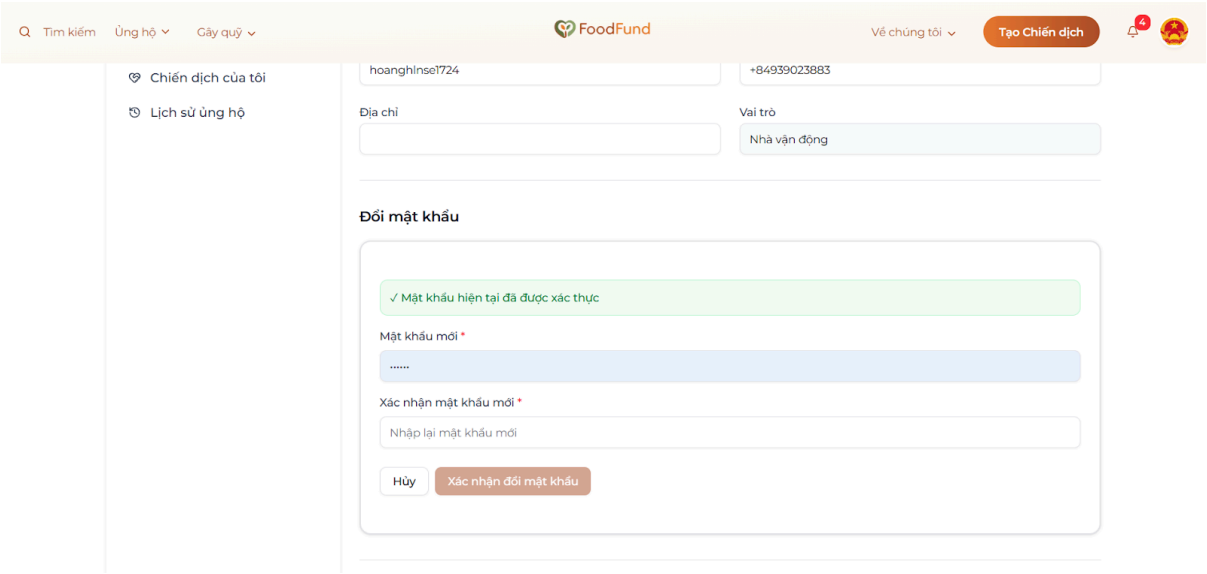


- **Function details:**
  - **Data requirement:** address, avatar_url, full_name, phone_number
  - **Validation:**
    - address:
      - Type: String
      - Address cannot exceed 500 characters
    - avatar_url:
      - Type: String
    - full_name:
      - Type: String
      - Full name must be at least 2 characters
      - Full name cannot exceed 100 characters
    - phone_number:
      - Type: String
      - Phone number must be a valid Vietnamese phone number
  - **Business rules:** BR-02

3.8.3 Change Password

- **Function trigger:** User click “Đổi mật khẩu” button on profile page.
- **Function description:**
  - **Actors:** Donor, Fundraiser, Kitchen Staff, Delivery Staff, Admin
  - **Purpose:** Donor, Fundraiser, Kitchen Staff, Delivery Staff, Admin want to change password



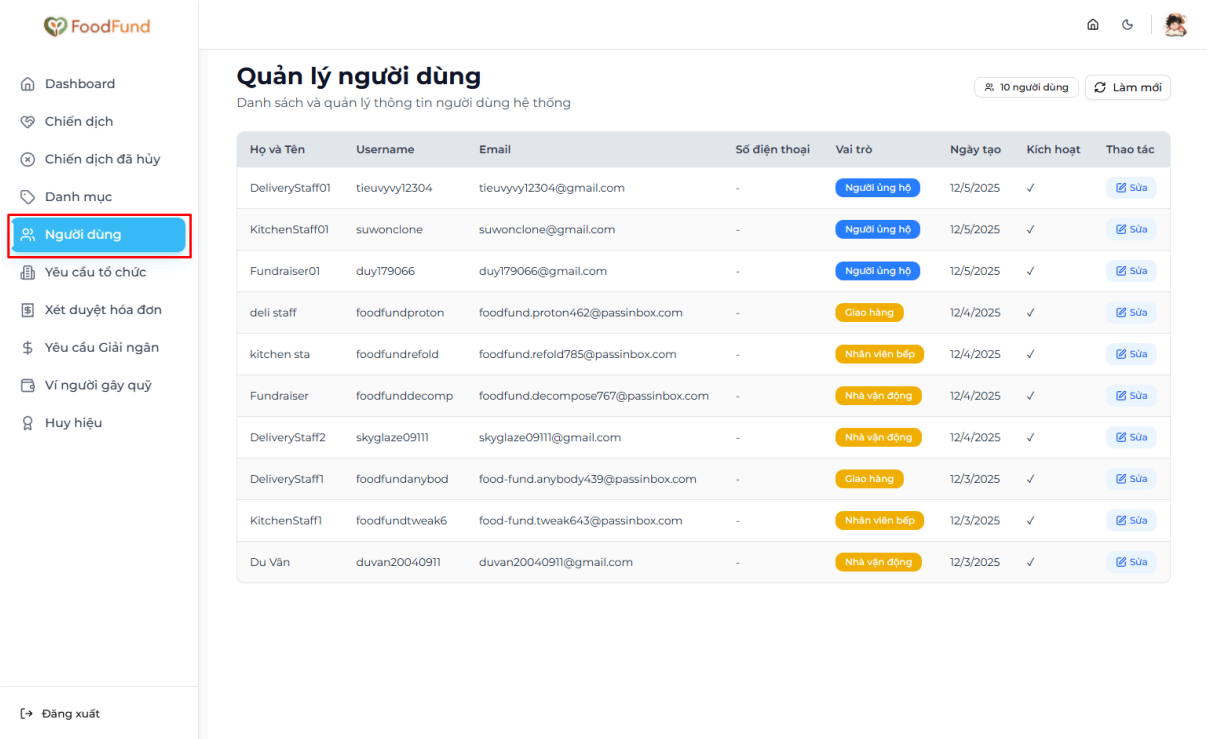


- **Function details:**
  - **Data requirement:** currentPassword, confirmNewPassword, newPassword
  - **Validation:**
    - currentPassword:
      - Type: String
      - Current password must be a string
    - newPassword:
      - Type: String
      - Password must be at least 6 characters long
    - confirmNewPassword
      - Type: String
      - Confirm new password must be a string
  - **Business rules:** BR-02, BR-03, BR-04, BR-05

3.9 Manage Users

3.9.1 View All Users

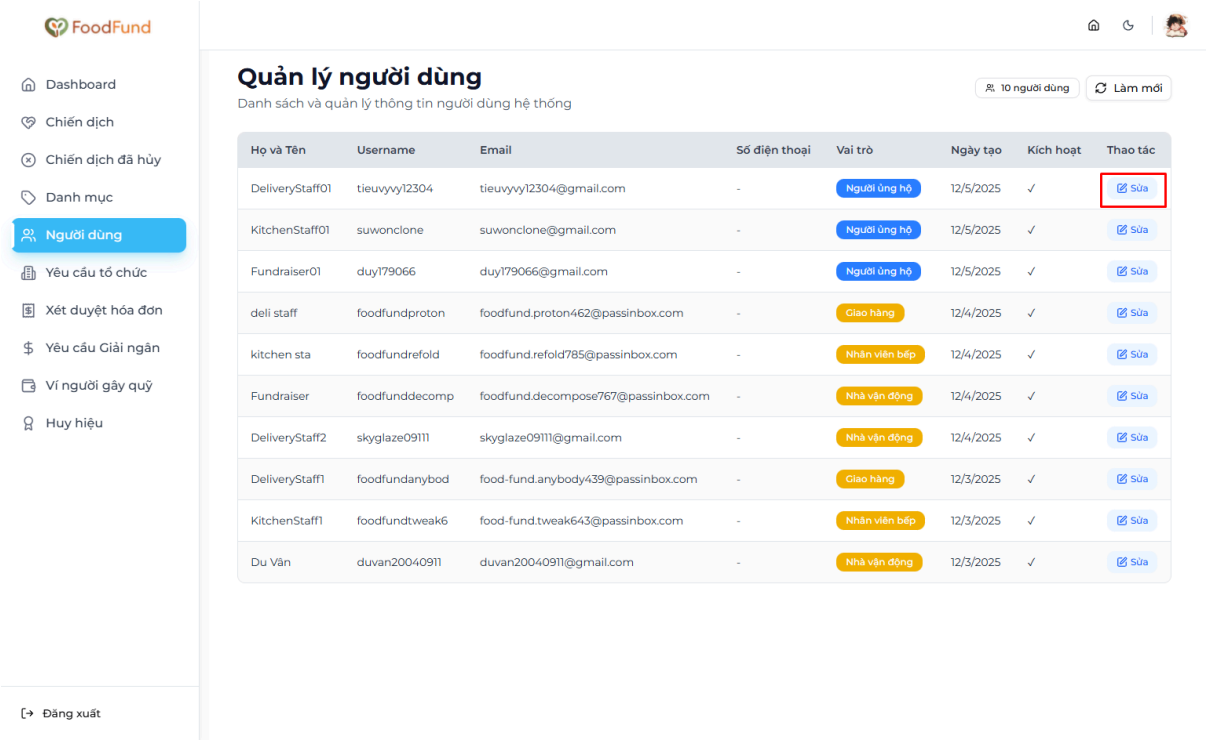
- **Function trigger:** Admin clicks “Người dùng” tab on the admin page.
- **Function description:**
  - **Actors:** Admin
  - **Purpose:** Admin wants to view all users of the system

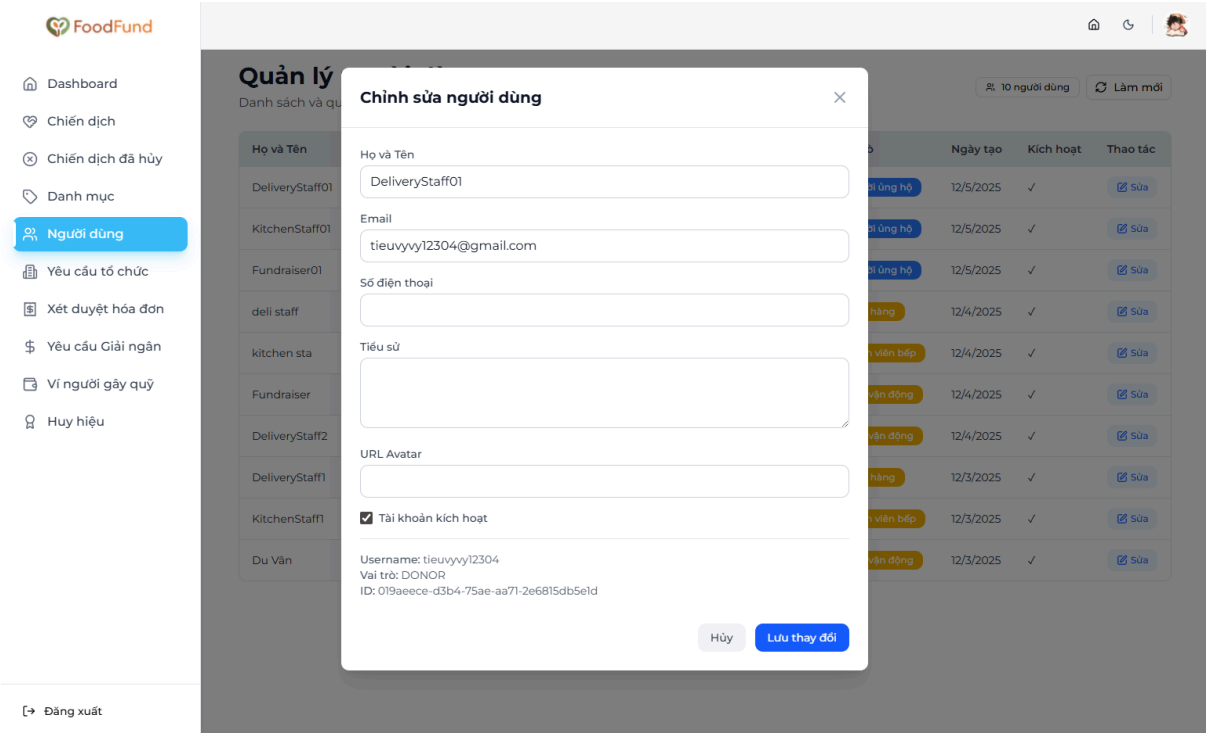


- **Function details:**
  - **Data requirement:** none
  - **Validation:** none
  - **Business rules:** BR-02

3.9.2 Update Users

- **Function trigger:** Admin clicks “Sửa” button in the manage users page.
- **Function description:**
  - **Actors:** Admin
  - **Purpose:** Admin wants to edit users of the system





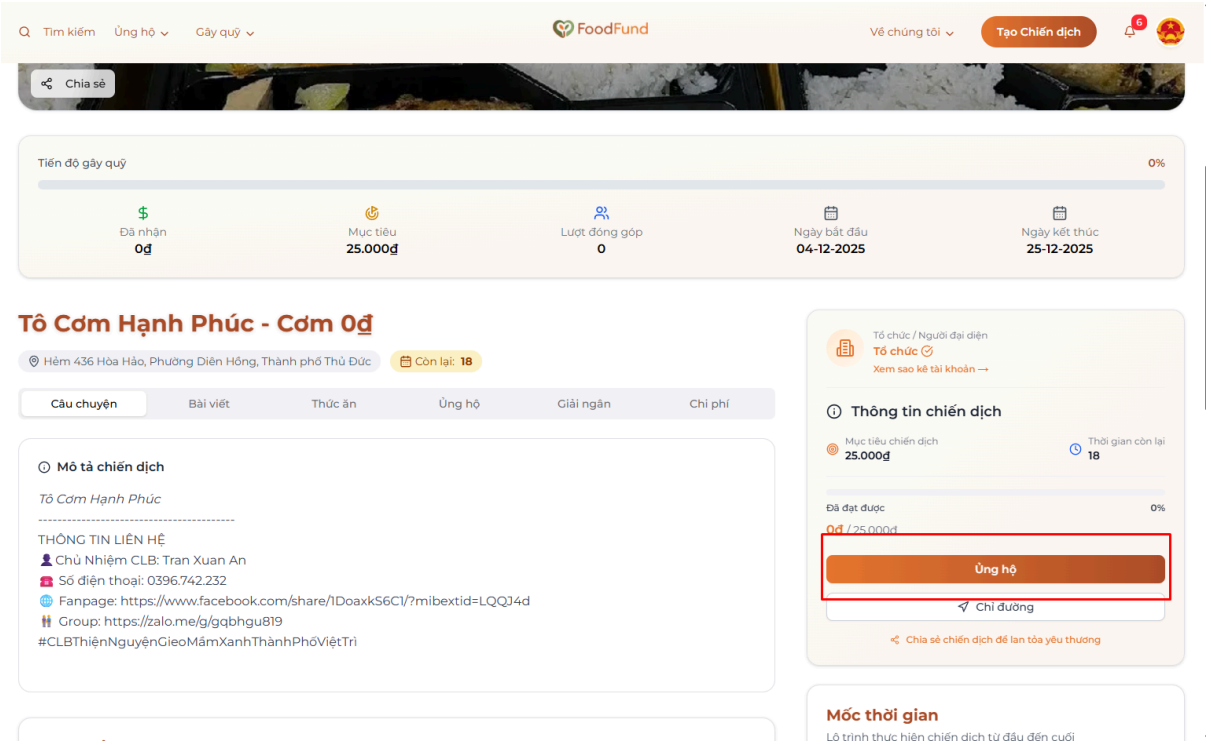
- **Function details:**
  - **Data requirement:** avatar_url, email, full_name, is_active, phone_number
  - **Validation:**
    - Avatar_Url:
      - Type: String
    - Email:

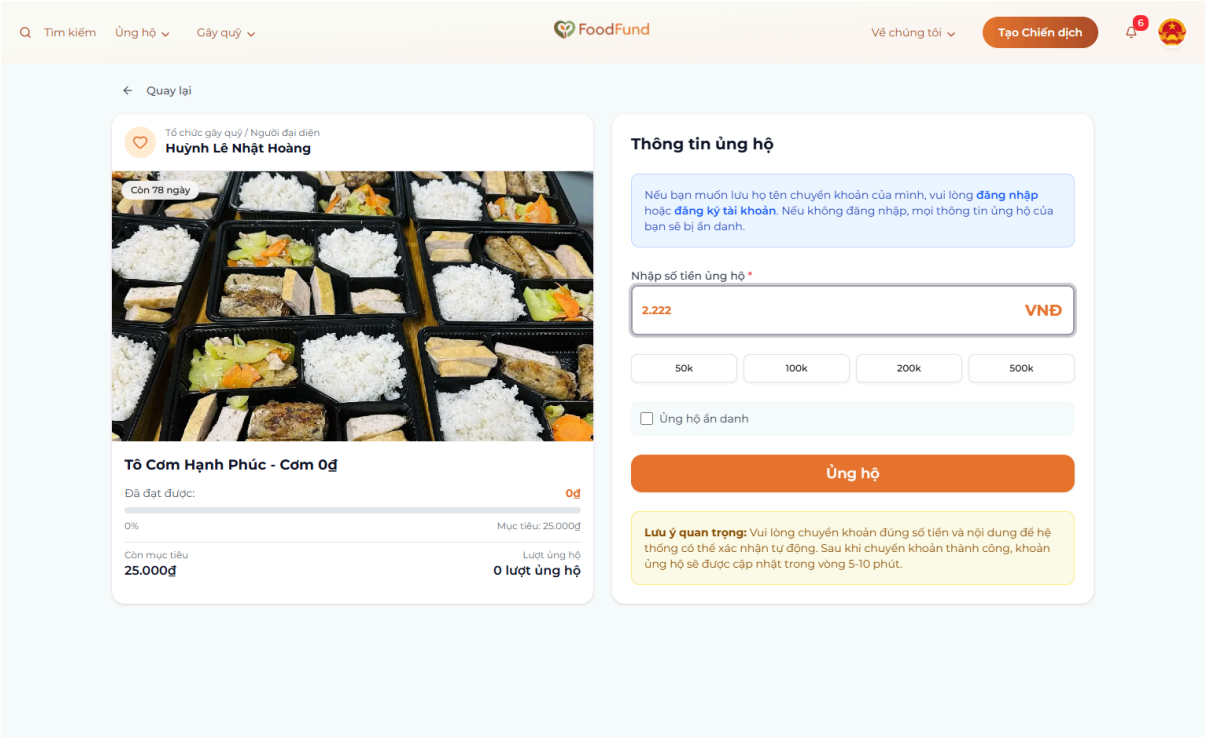
            - Type: String
            - Must be valid email type, unique and not include blank.
        - Full_Name:
            - Type: String
        - Is_Active:
            - Type: Boolean
        - Phone_Number:
            - Type: String
            - Must be valid email phone
    - **Business rules:** BR-02, BR-03, BR-04, BR-06

3.10 Donation

3.10.1 Donation

- **Function trigger:** Users click the "Ủng hộ" button in the campaign detail page.
- **Function description:**
    - **Actors:** Admin
    - **Purpose:** Admin wants to donate a campaign

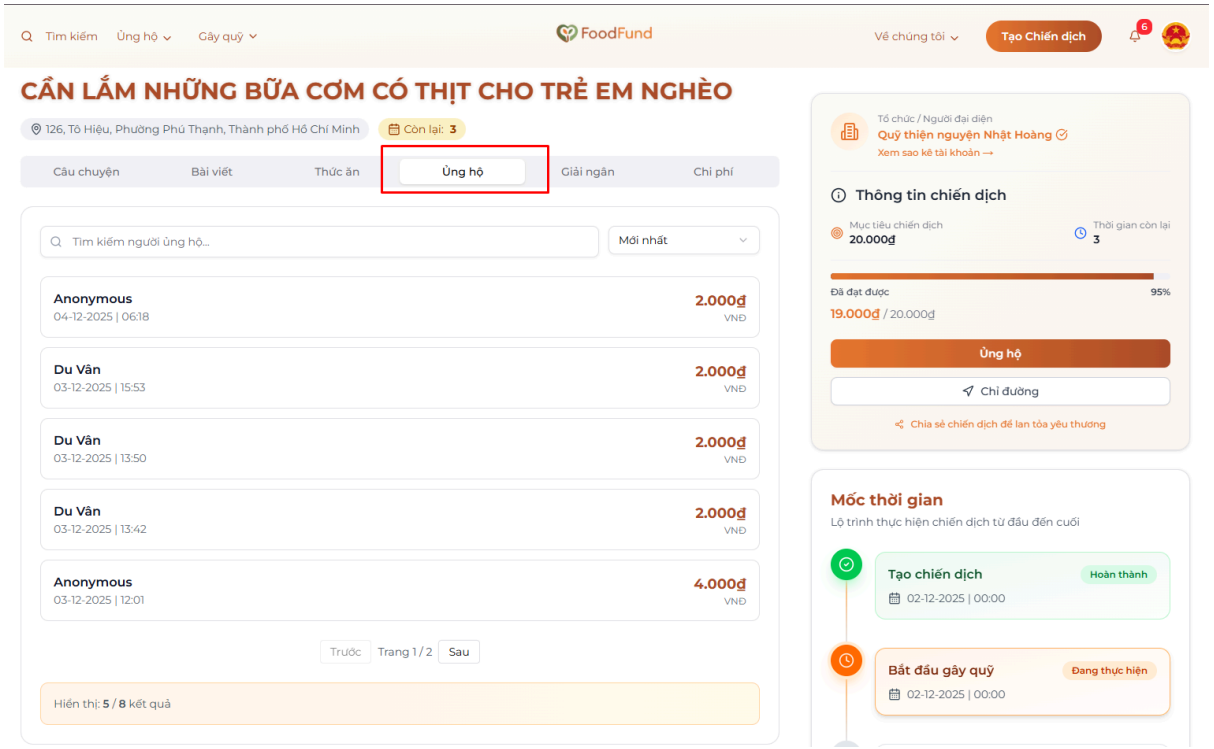




- **Function details:**
  - **Data requirement:** amount, campaignId, isAnonymous
  - **Validation:**
    - Amount:
      - Type: BigInt
    - CampaignId:
      - Type: String
    - IsAnonymous:
      - Type: Boolean
  - **Business rules:** BR-33, BR-35, BR-34, BR-38, BR-38

3.10.2 View Campaign Donation List

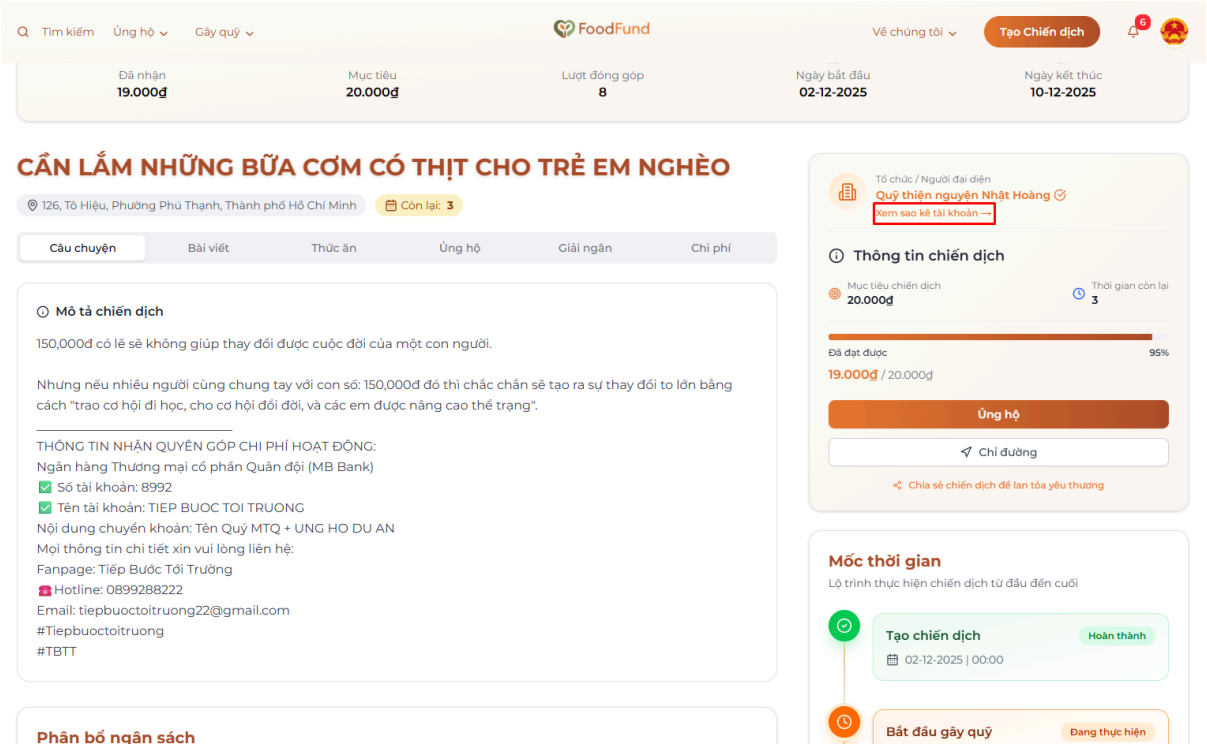
- **Function trigger:** Users click the “Ủng hộ” tab in the campaign detail page.
- **Function description:**
  - **Actors:** Guest, Donor, Fundraiser, Admin, Kitchen Staff, Delivery Staff
  - **Purpose:** Users want to view all donation of the campaign

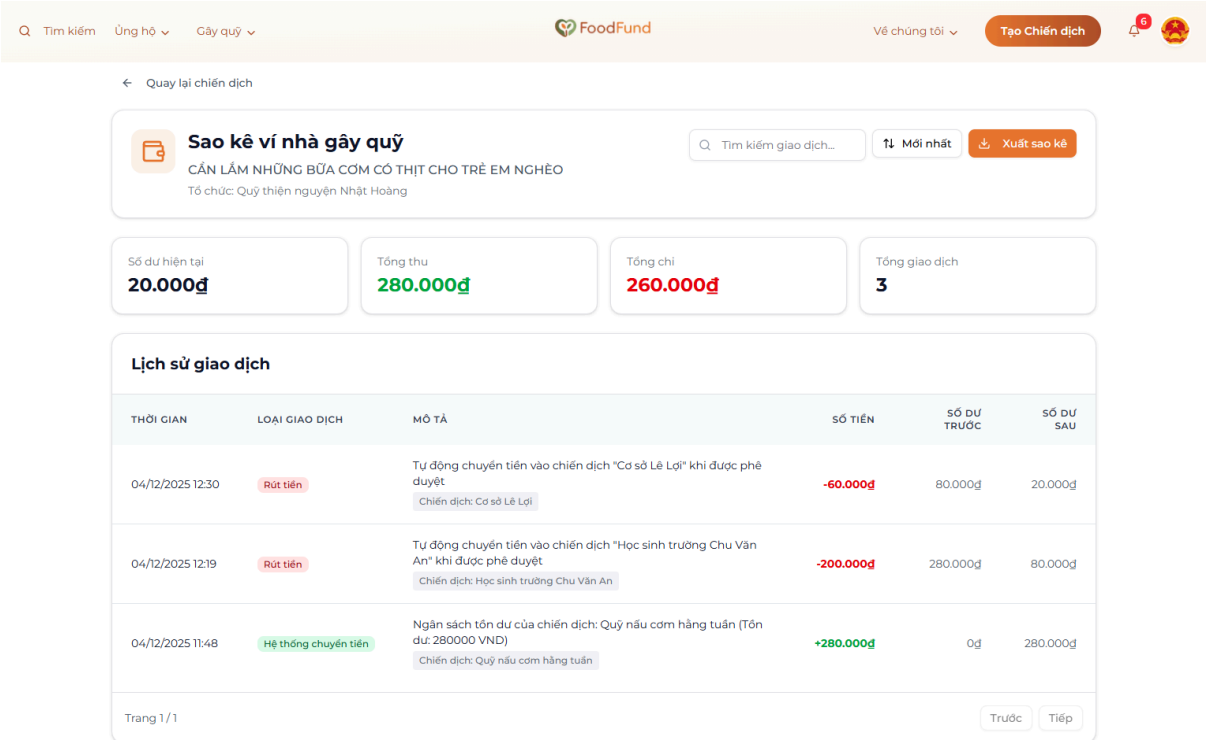


- **Function details:**
  - **Data requirement:** campaignId
  - **Validation:**
    - CampaignId:
      - Type: String
  - **Business rules:** none

3.10.3 View Campaign Donation Statement

- **Function trigger:** Users click the “Xem sao kê tài khoản” in the campaign detail page.
- **Function description:**
  - **Actors:** Guest, Donor, Fundraiser, Admin, Kitchen Staff, Delivery Staff
  - **Purpose:** Users want to view campaign donation statement



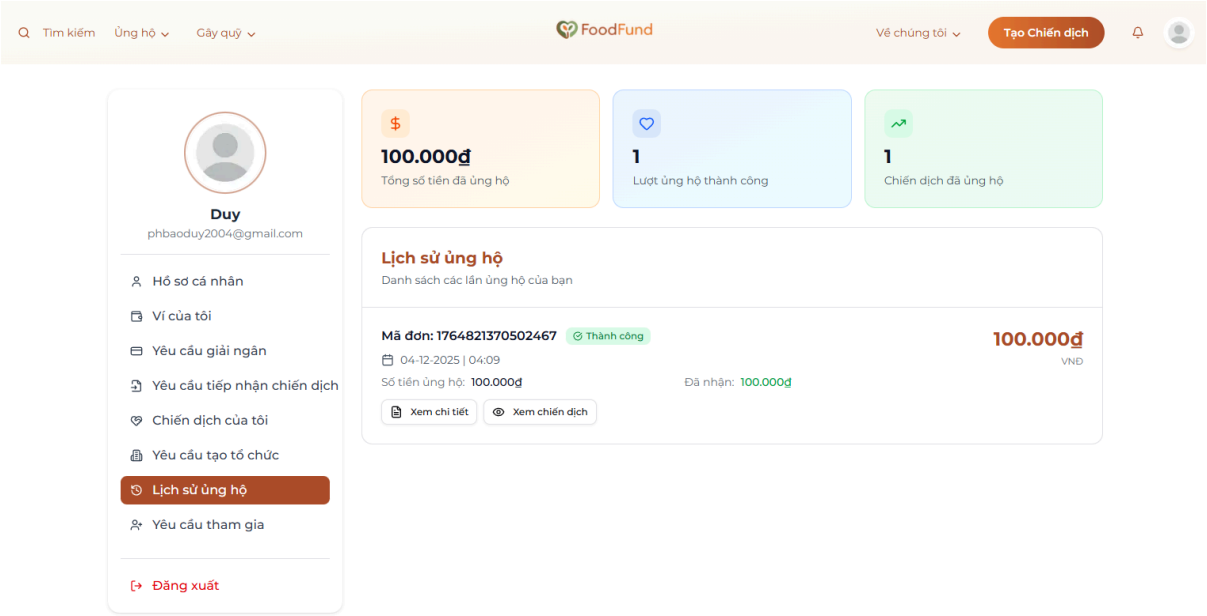


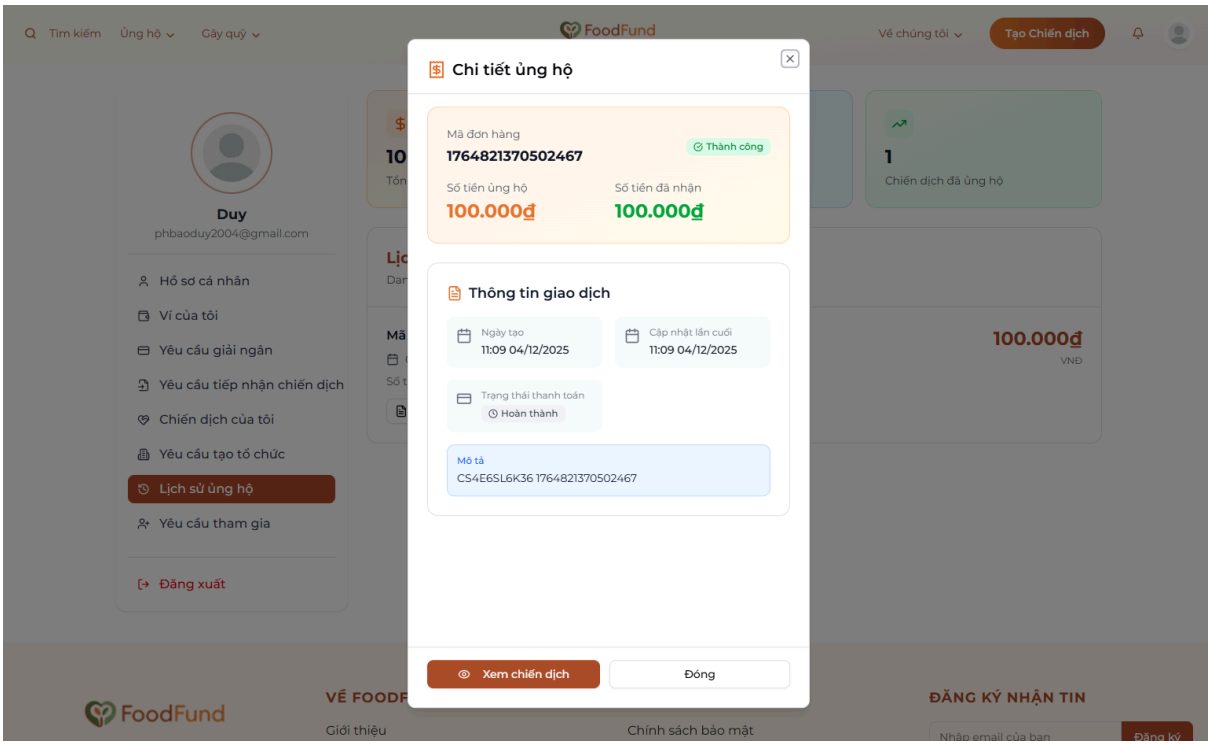
- **Function details:**
  - **Data requirement:** campaignId
  - **Validation:**
    - CampaignId:
      - Type: String
  - **Business rules:** BR-39

3.10.4 View History Donation

- **Function trigger:** Users click the “Lịch sử ủng hộ” tab in the profile page.

- **Function description:**
  - **Actors:** Donor, Fundraiser, Admin, Kitchen Staff, Delivery Staff
  - **Purpose:** Users want to view history donation



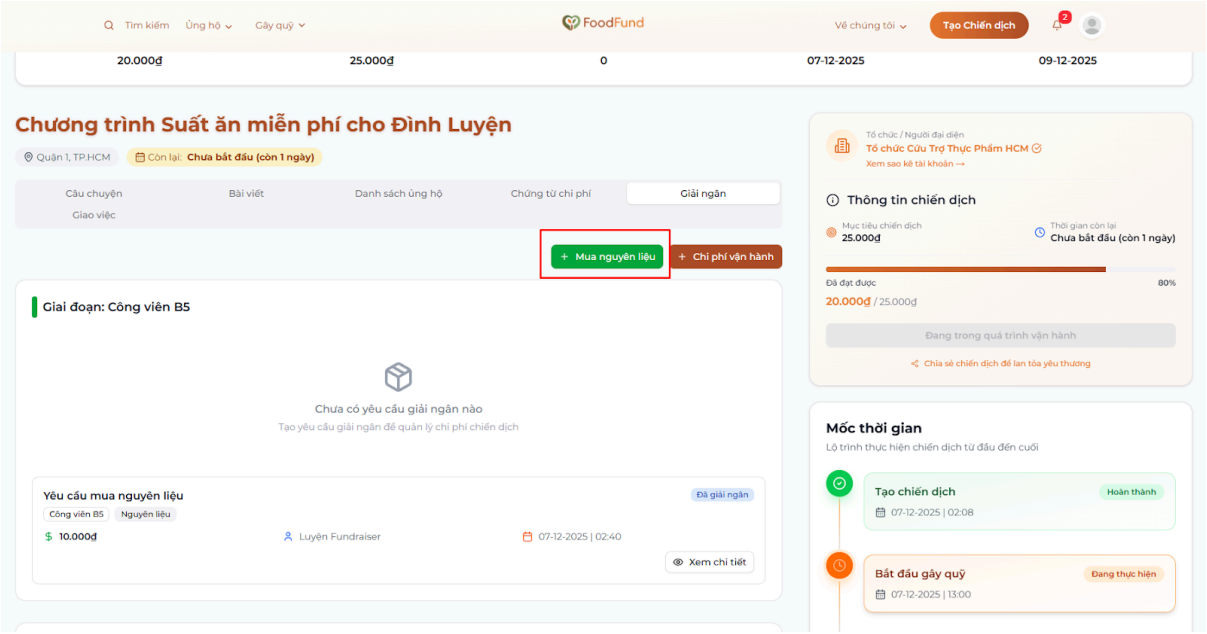


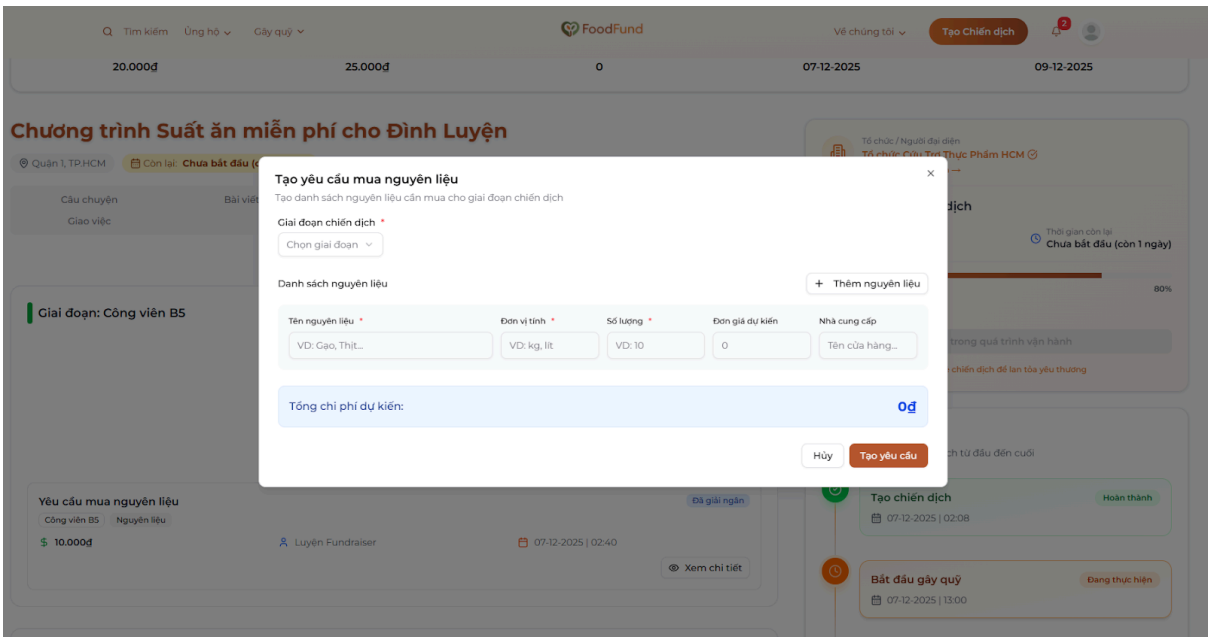
- **Function details:**
  - **Data requirement:** none
  - **Validation:** none
  - **Business rules:** BR-02, BR-20

3.11 Ingredient Disbursement Request

3.11.1 Create Ingredient Disbursement Request

- **Function trigger:** Fundraiser clicks the “Mua nguyên liệu” button in the my campaign detail page.
- **Function description:**
  - **Actors:** Fundraiser
  - **Purpose:** Fundraiser wants to create ingredient disbursement request





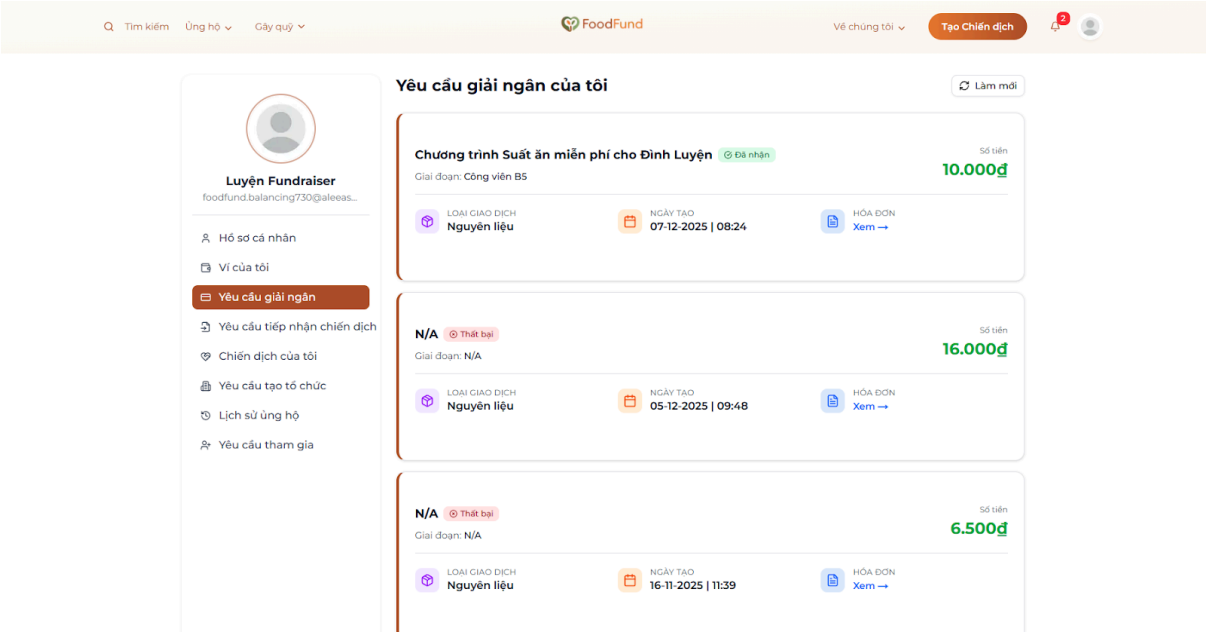
- **Function details:**
  - **Data requirement:** campaignPhaseId, totalCost, items, ingredientName, quantity, unit, estimatedUnitPrice, supplier, plannedIngredientId
  - **Validation:**

- CampaignPhaseId:
  - Type: String
- TotalCost:
  - Type: Number
- Items:
  - Type: Array
- IngredientName:
  - Type: String
- Quantity:
  - Type: Quantity
- Unit:
  - Type: String
- EstimatedUnitPrice:
  - Type: BigInt
- Supplier:
  - Type: String
- PlannedIngredientId:
  - Type: String

- **Business rules:** BR-02, BR-36, BR-41, BR-42, BR-18

3.11.2 View My Ingredient Request

- **Function trigger:** Fundraiser clicks the "Yêu cầu giải ngân" tab in the profile page.
- **Function description:**
  - **Actors:** Fundraiser
  - **Purpose:** Fundraiser wants to view their ingredient request

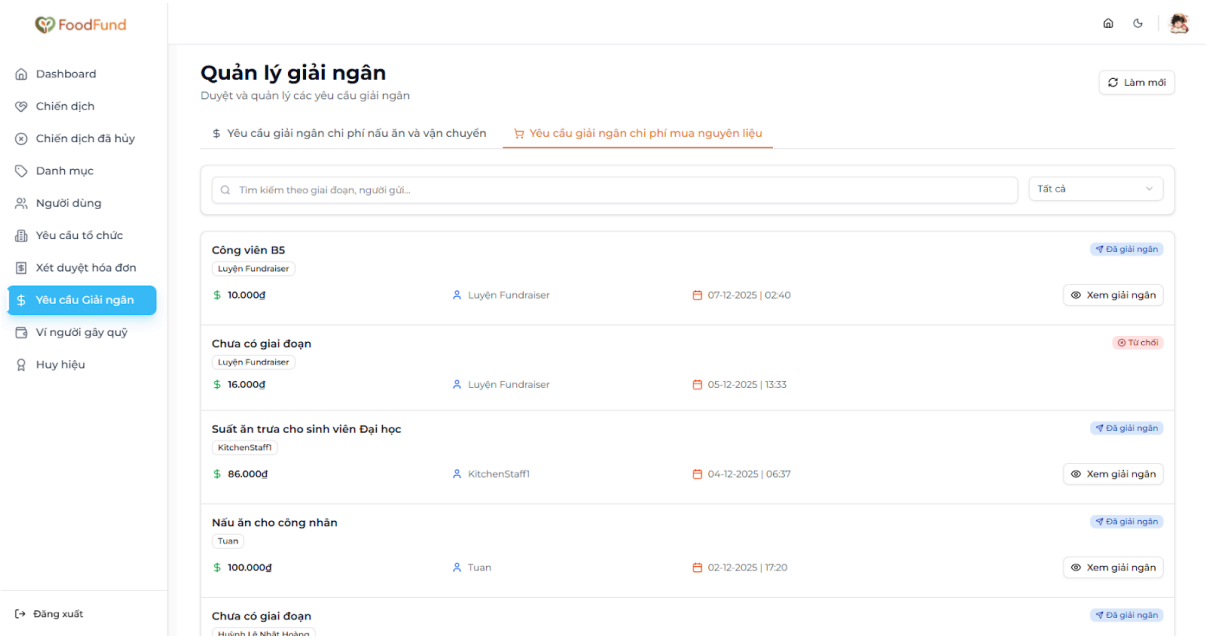


- **Function details:**

- **Data requirement:** none
- **Validation:** none
- **Business rules:** BR-02, BR-08

3.11.3 View Ingredient Request

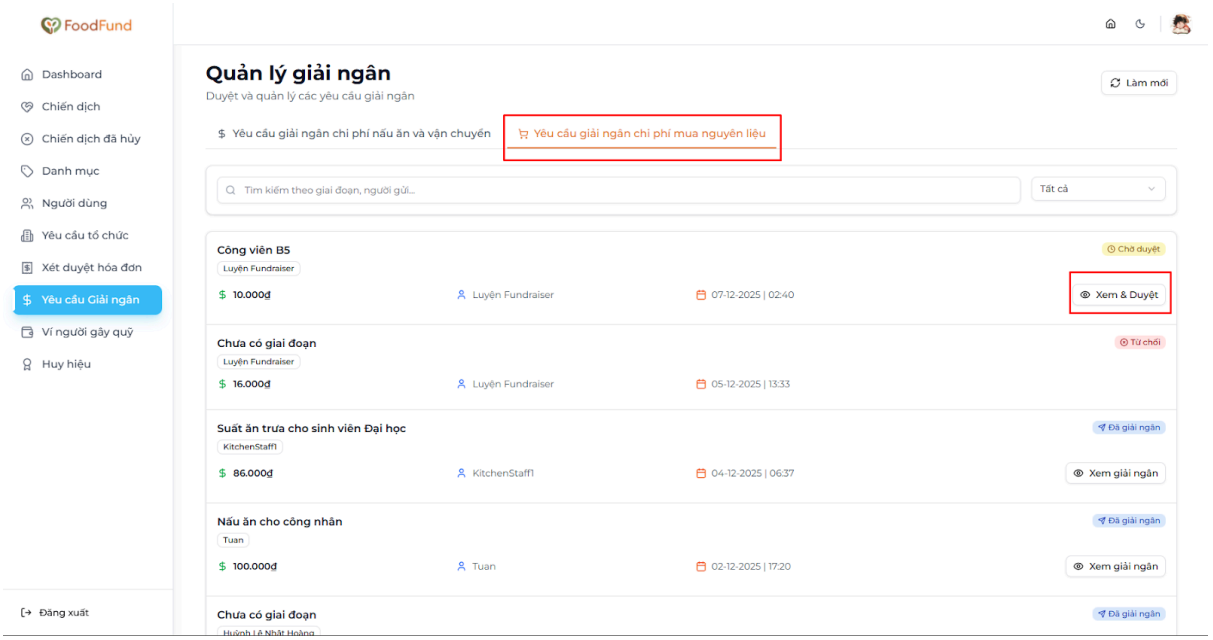
- **Function trigger:** Admin clicks the “Yêu cầu giải ngân chi phí mua nguyên liệu” tab in the admin page.
- **Function description:**
  - **Actors:** Admin
  - **Purpose:** Admin wants to view all ingredient request

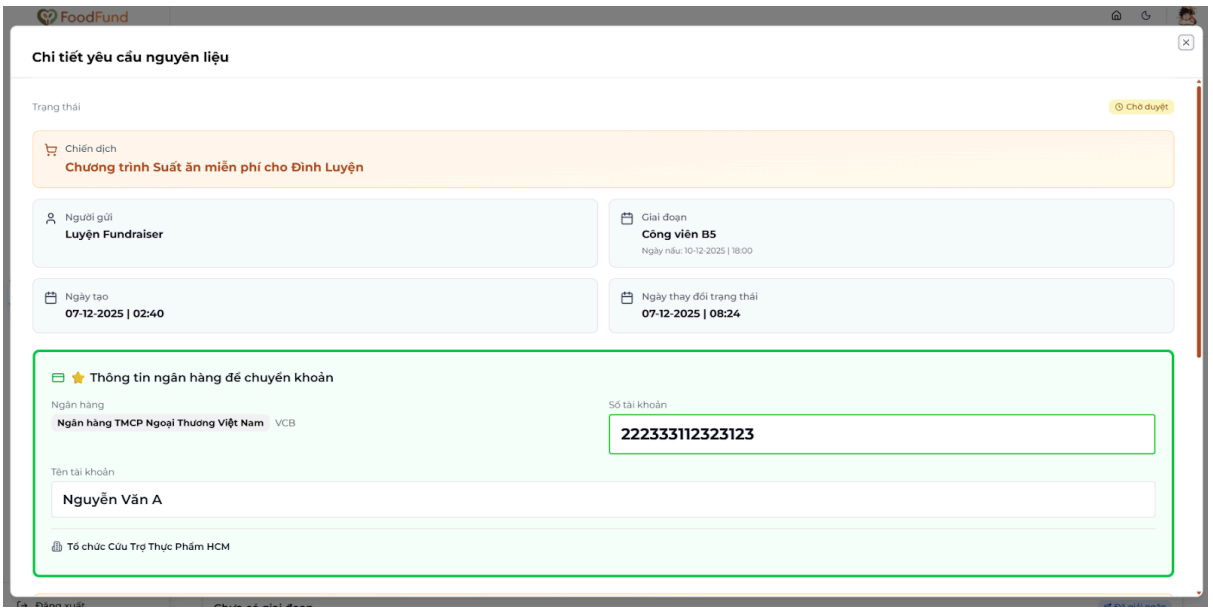


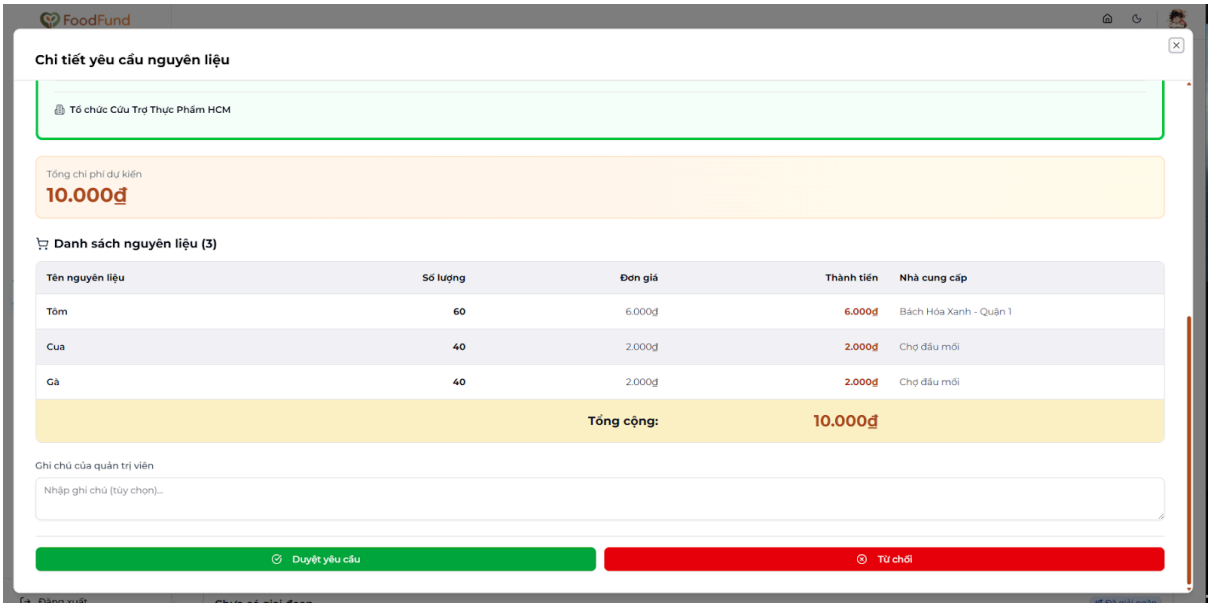
- **Function details:**
  - **Data requirement:** none
  - **Validation:** none
  - **Business rules:** BR-02

3.11.4 Ingredient Request Approval

- **Function trigger:** Admin clicks the “Xử lý” button in “Yêu cầu giải ngân chi phí mua nguyên liệu” tab.
- **Function description:**
  - **Actors:** Admin
  - **Purpose:** Admin wants to approve/reject an ingredient request


FoodFund
Dashboard
Chiến dịch
Chiến dịch đã hủy
Danh mục
Người dùng
Yêu cầu tổ chức
Xét duyệt hóa đơn
Yêu cầu Giải ngân
Ví người gây quỹ
Huy hiệu
Đăng xuất
Quản lý giải ngân
Duyệt và quản lý các yêu cầu giải ngân
Làm mới
Yêu cầu giải ngân chi phí nấu ăn và vận chuyển
Yêu cầu giải ngân chi phí mua nguyên liệu
Tìm kiếm theo giai đoạn, người gửi...
Tất cả
Công viên B5
Chờ duyệt
Luyện Fundraiser
10.000đ
Luyện Fundraiser
07-12-2025 | 02:40
Xem & Duyệt
Chưa có giai đoạn
Từ chối
Luyện Fundraiser
16.000đ
Luyện Fundraiser
05-12-2025 | 13:33
Suất ăn trưa cho sinh viên Đại học
Đã giải ngân
KitchenStaff1
86.000đ
KitchenStaff1
04-12-2025 | 06:37
Xem giải ngân
Nấu ăn cho công nhân
Đã giải ngân
Tuan
100.000đ
Tuan
02-12-2025 | 17:20
Xem giải ngân
Chưa có giai đoạn
Đã giải ngân


FoodFund
Chi tiết yêu cầu nguyên liệu
Trạng thái
Chờ duyệt
Chiến dịch
Chương trình Suất ăn miễn phí cho Đình Luyện
Người gửi
Luyện Fundraiser
Giai đoạn
Công viên B5
Ngày nấu: 10-12-2025 | 18:00
Ngày tạo
07-12-2025 | 02:40
Ngày thay đổi trạng thái
07-12-2025 | 08:24
Thông tin ngân hàng để chuyển khoản
Ngân hàng
Ngân hàng TMCP Ngoại Thương Việt Nam VCB
Số tài khoản
222333112323123
Tên tài khoản
Nguyễn Văn A
Tổ chức Cứu Trợ Thực Phẩm HCM

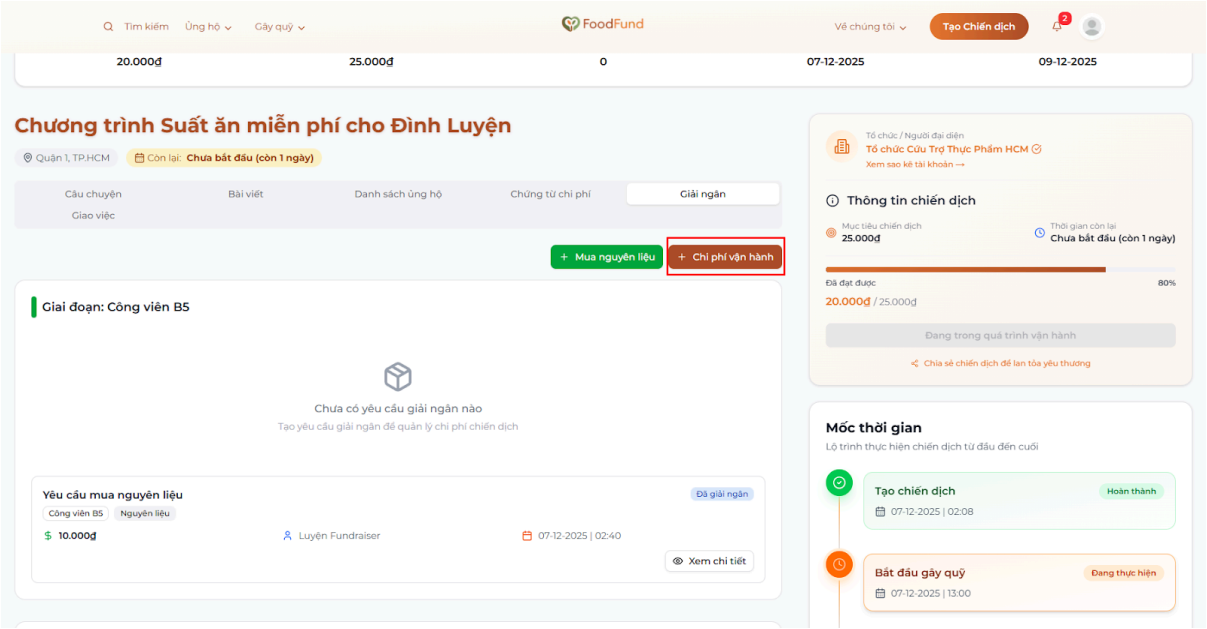


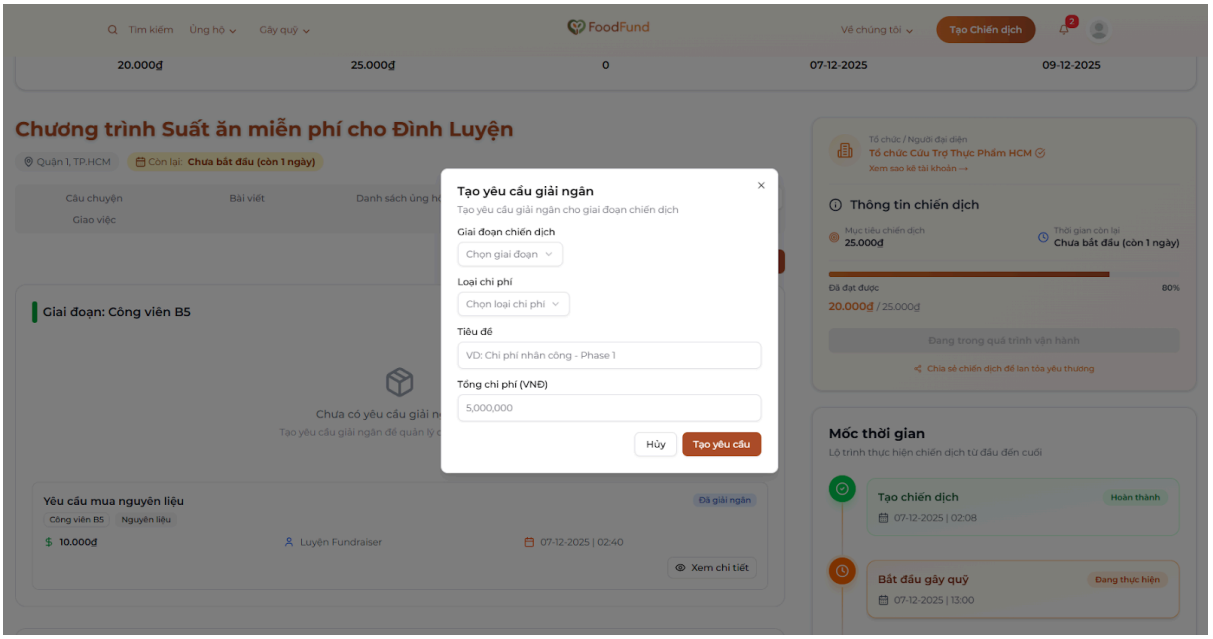
- **Function details:**
  - **Data requirement:** id, status, adminNote
  - **Validation:**
    - Id:
      - Type: String
    - Status:
      - Type: String
      - Must be PENDING, APPROVED, REJECTED
    - AdminNote:
      - Type: String
      - Optional
  - **Business rules:** BR-02, BR-36, BR-37

3.12 Operation Disbursement Request

3.12.1 Create Operation Disbursement Request

- **Function trigger:** Fundraiser clicks the “Chi phí vận hành” button in the my campaign detail page.
- **Function description:**
  - **Actors:** Fundraiser
  - **Purpose:** Fundraiser wants to create operation disbursement request

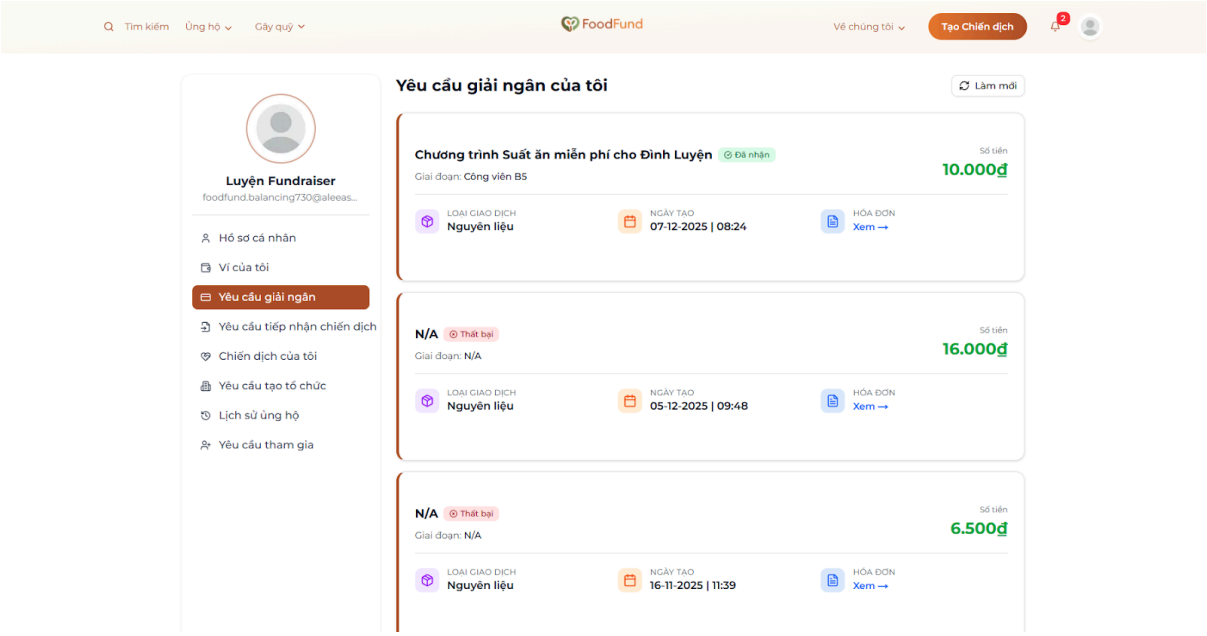




- **Function details:**
  - **Data requirement:** campaignPhaseId, expenseType, title, totalCost
  - **Validation:**
    - CampaignPhaseId:
      - Type: String
    - ExpenseType:
      - Type: String
      - Must be INGREDIENT, COOKING, DELIVERY
    - Title:
      - Type: String
    - TotalCost:
      - Type: BigInt
  - **Business rules:** BR-37, BR-36, BR-40

3.12.2 View My Operation Request

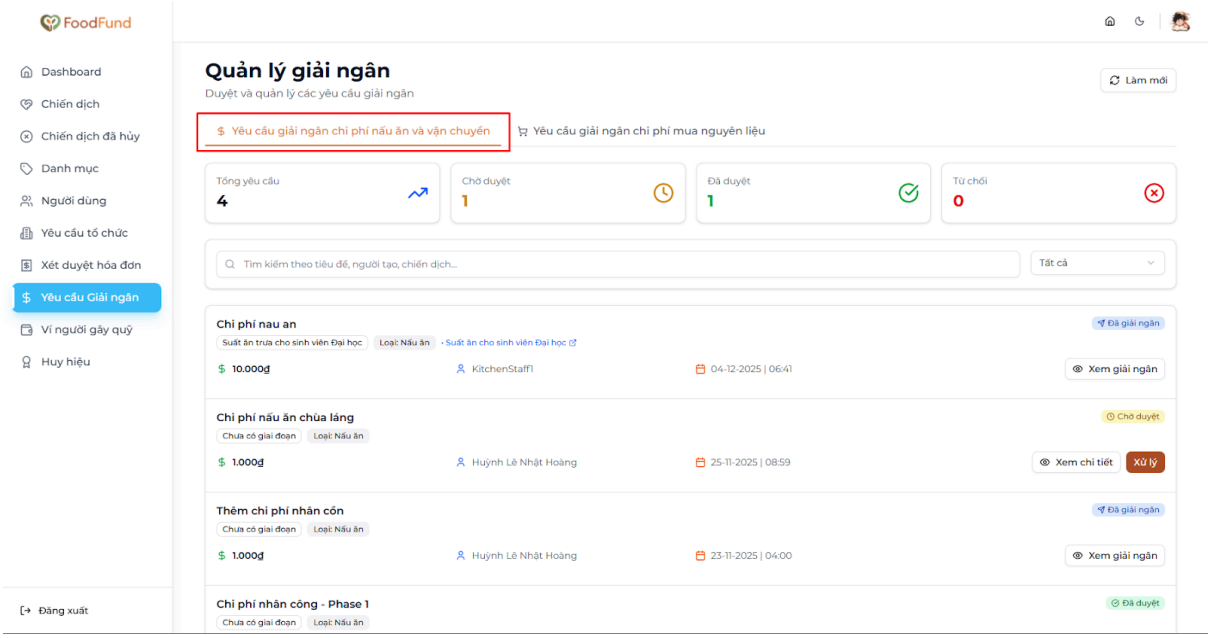
- **Function trigger:** Fundraiser clicks the “Yêu cầu giải ngân” tab in the profile page.
- **Function description:**
  - **Actors:** Fundraiser
  - **Purpose:** Fundraiser wants to view their operation request



- **Function details:**
  - **Data requirement:** none
  - **Validation:** none
  - **Business rules:** BR-02, BR-08

3.12.3 View Operation Request

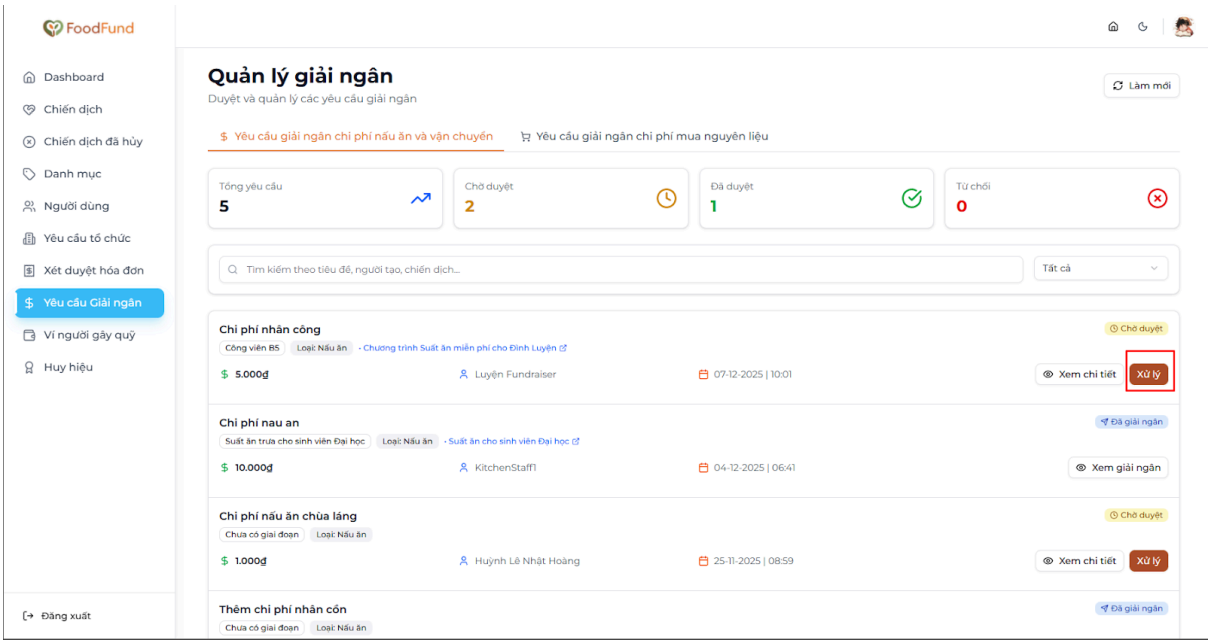
- **Function trigger:** Admin clicks the “Yêu cầu giải ngân chi phí nấu ăn và vận chuyển” tab in the admin page.
- **Function description:**
  - **Actors:** Admin
  - **Purpose:** Admin wants to view all operation request

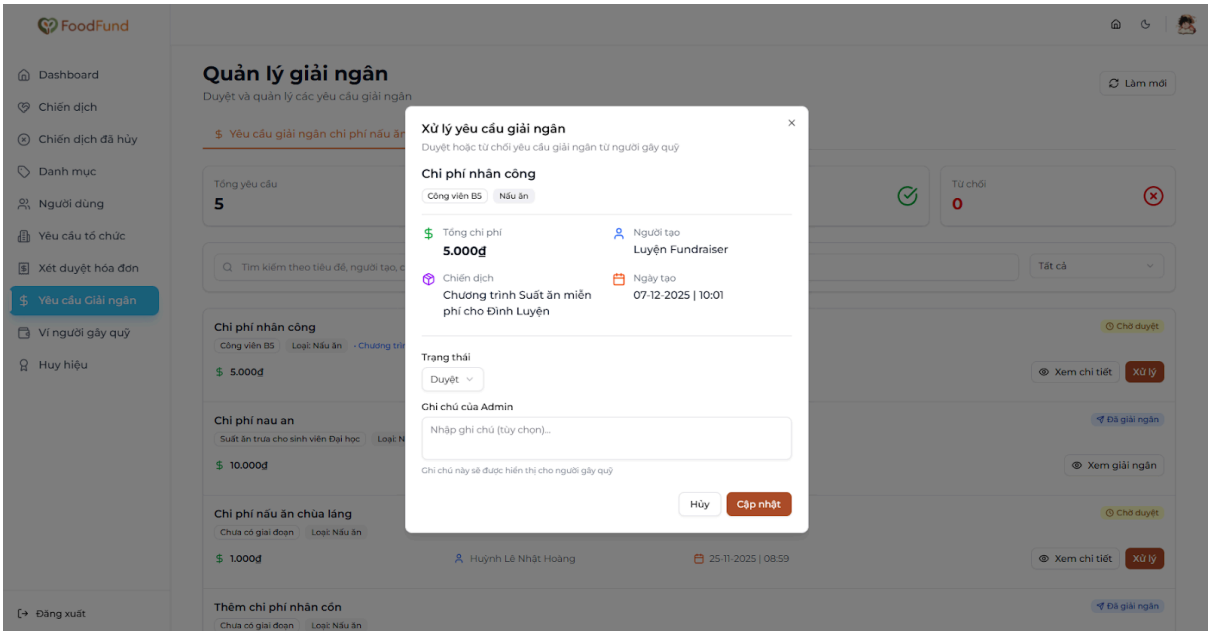


- **Function details:**
  - **Data requirement:** none
  - **Validation:** none
  - **Business rules:** BR-02

3.12.4 Operation Request Approval

- **Function trigger:** Admin clicks the "Xử lý" button in "Yêu cầu giải ngân chi phí nấu ăn và vận chuyển" tab.
- **Function description:**
  - **Actors:** Admin
  - **Purpose:** Admin wants to approve/reject an operation request



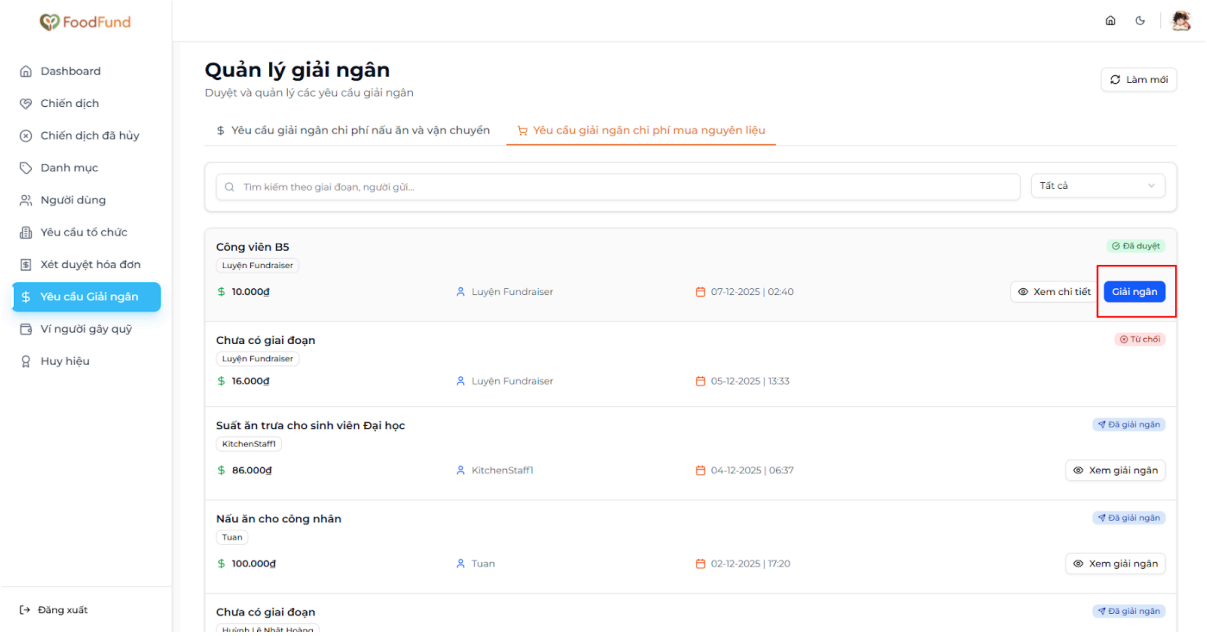


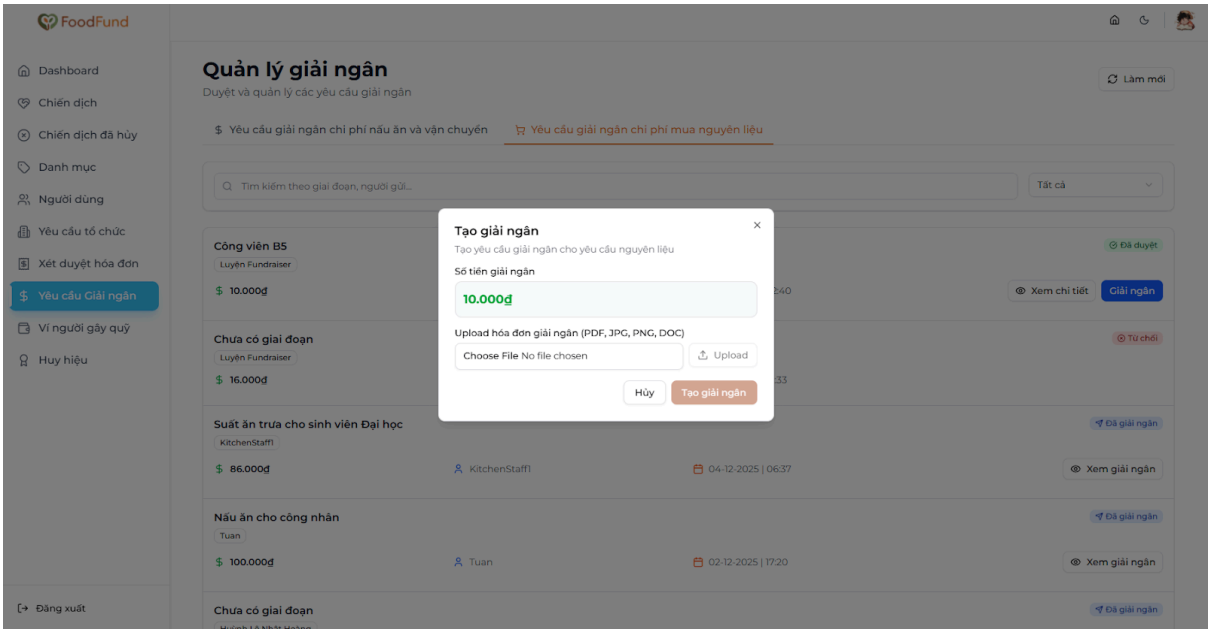
- **Function details:**
  - **Data requirement:** requestId, status, adminNote
  - **Validation:**
    - RequestId:
      - Type: String
      - Status:
        - Type: String
        - Must be PENDING, APPROVED, REJECTED
      - AdminNote:
        - Type: String
  - **Business rules:** BR-02, BR-36, BR-43

3.13 Inflow Transaction

3.13.1 Create Inflow Transaction

- **Function trigger:** Admin clicks the “Giải ngân” button in the operation request tab or ingredient request tab
- **Function description:**
  - **Actors:** Admin
  - **Purpose:** Admin wants to create inflow transaction



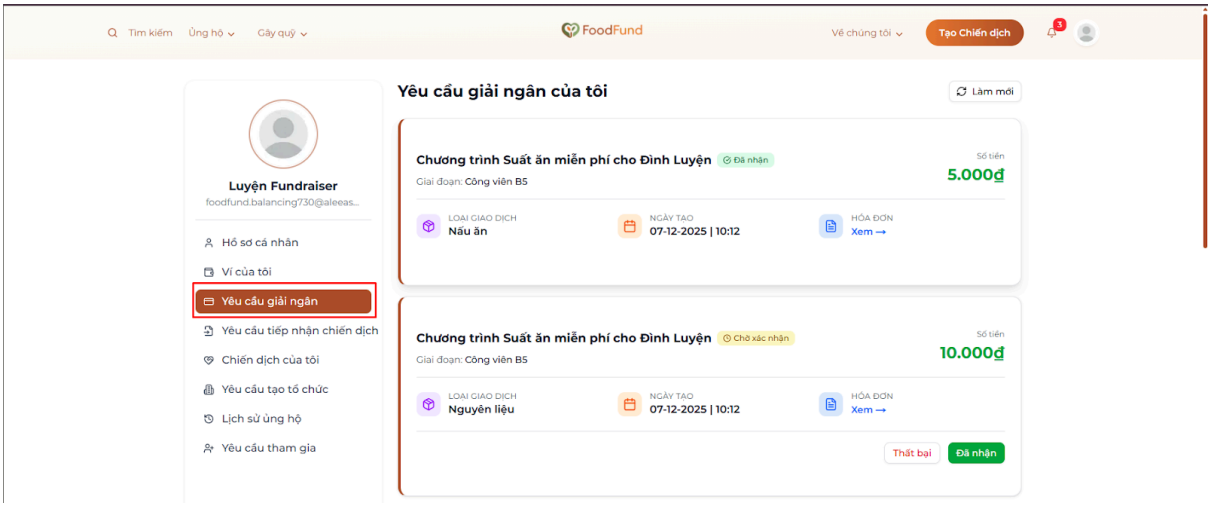


- **Function details:**
  - **Data requirement:** amount, campaignPhaseId, proof, ingredientRequestId/operationRequestId
  - **Validation:**
    - Amount:
      - Type: BigInt
    - CampaignPhaseId:
      - Type: String
    - Proof:
      - Type: String
    - IngredientRequestId:
      - Type: String
    - OperationRequestId:

- Type: String
  - **Business rules:** BR-02, BR-38

3.13.2 Get My Disbursement

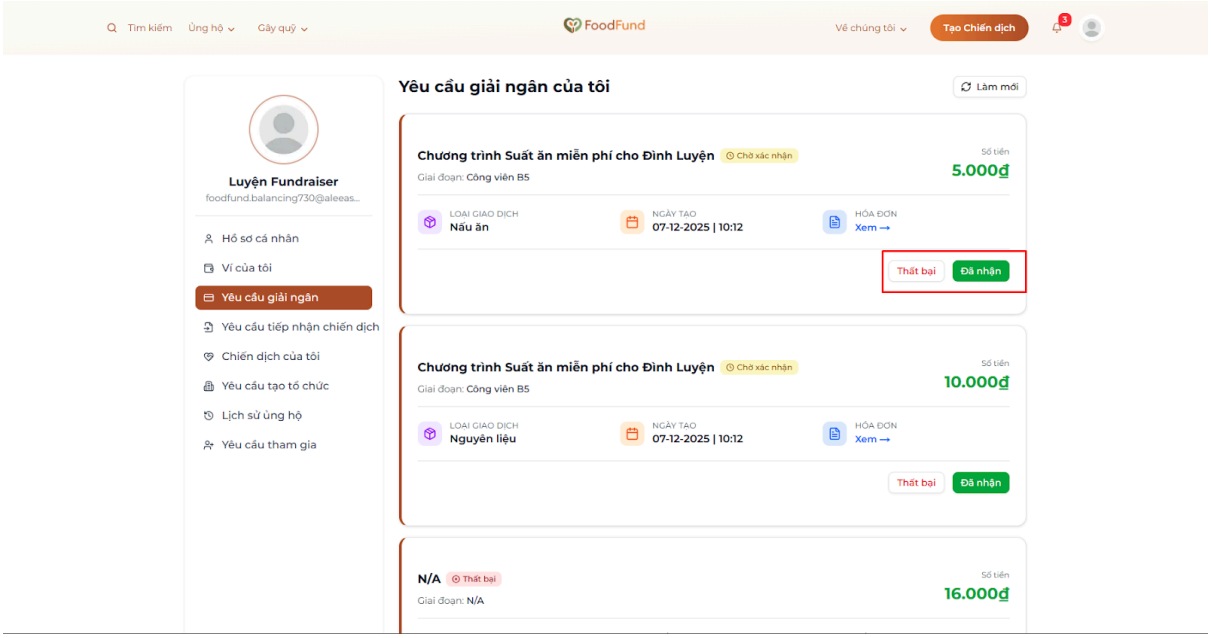
- **Function trigger:** Fundraiser clicks “Yêu cầu giải ngân” tab in the profile page
- **Function description:**
  - **Actors:** Fundraiser
  - **Purpose:** Fundraiser wants to confirm inflow transaction



- **Function details:**
  - **Data requirement:** none
  - **Validation:** none
  - **Business rules:** BR-08, BR-11

3.13.3 Confirm Disbursement

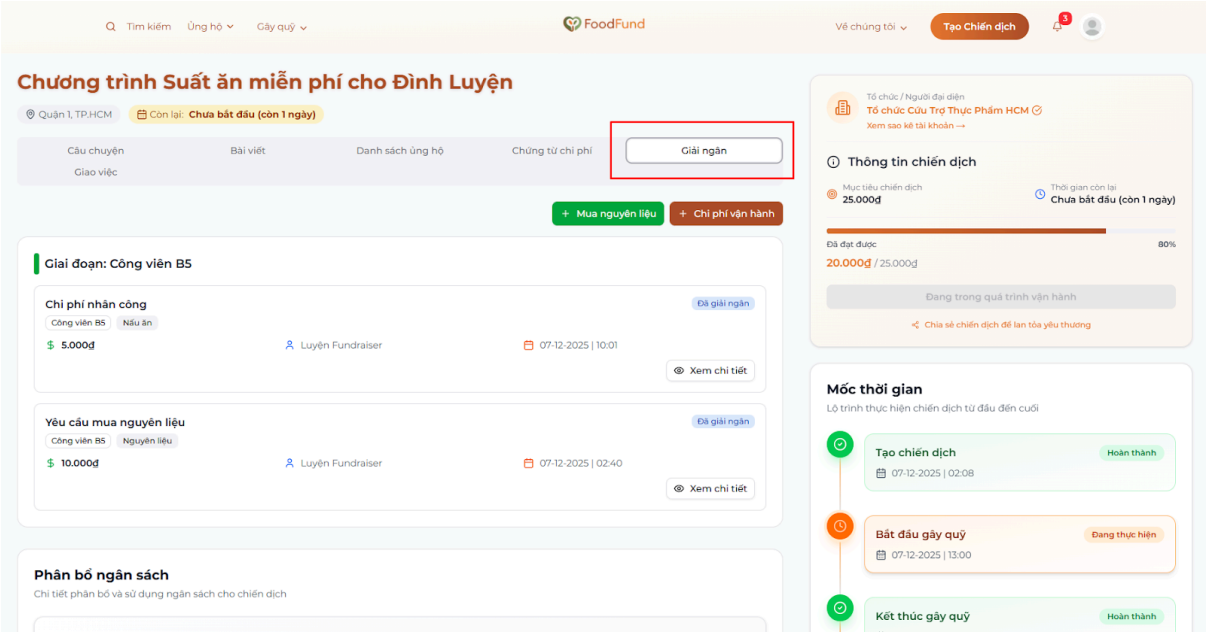
- **Function trigger:** Fundraiser clicks the “Thất bại/ Đã nhận” in “Yêu cầu giải ngân” tab in the profile page
- **Function description:**
  - **Actors:** Fundraiser
  - **Purpose:** Fundraiser wants to confirm inflow transaction



- **Function details:**
  - **Data requirement:** id, status
  - **Validation:**
    - Id:
      - Type: String
    - Status:
      - Type: String
  - **Business rules:** BR-02, BR-39

3.13.4 View Disbursement Of Campaign

- **Function trigger:** Users click “Giải ngân” tab in the campaign detail page
- **Function description:**
  - **Actors:** Guest, Donor, Fundraiser, Admin, Kitchen Staff, Delivery Staff
  - **Purpose:** Users want to view disbursement of campaign

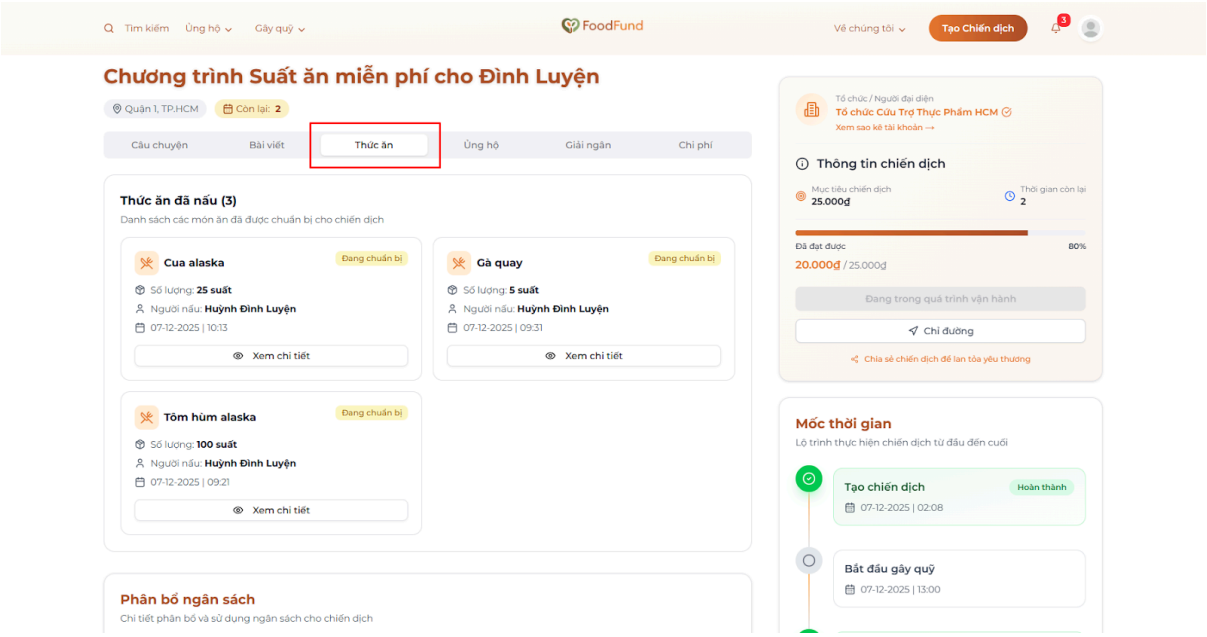


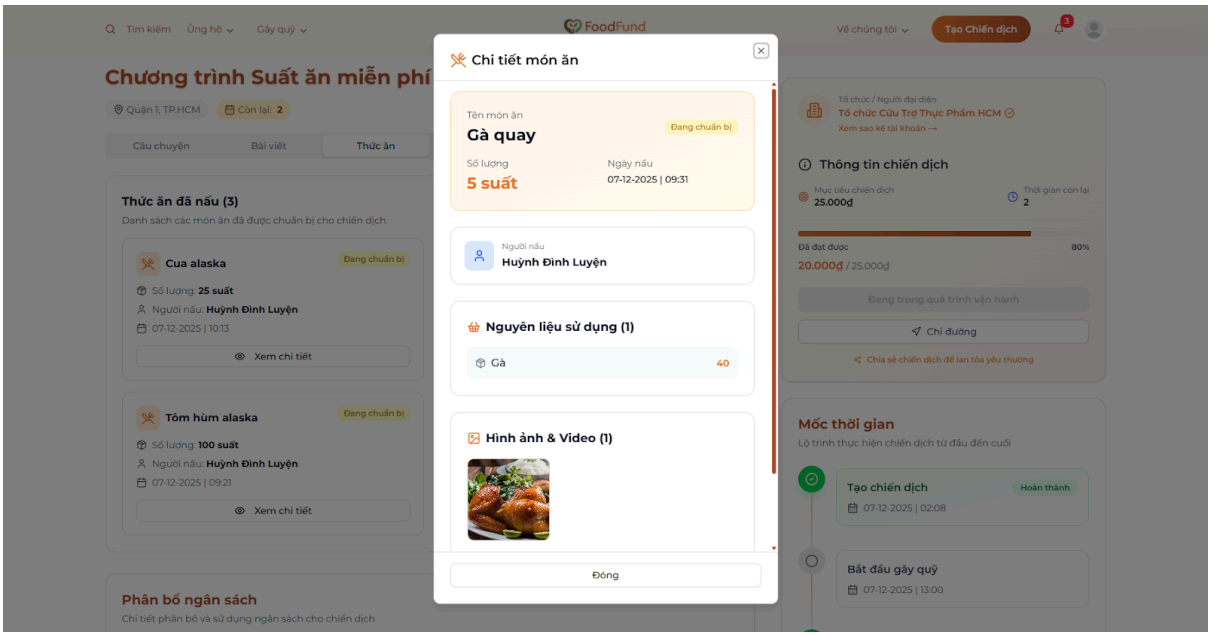
- **Function details:**
  - **Data requirement:** campaignPhaseId
  - **Validation:**
    - CampaignPhaseId
      - Type: String
  - **Business rules:** BR-36, BR-40

3.14 Meal Batch

3.14.1 View Meal Batches

- **Function trigger:** Users click the “Thức ăn” tab in the campaign detail page
- **Function description:**
  - **Actors:** Guest, Donor, Fundraiser, Admin, Kitchen Staff, Delivery Staff
  - **Purpose:** Users want to view meal batches of campaign



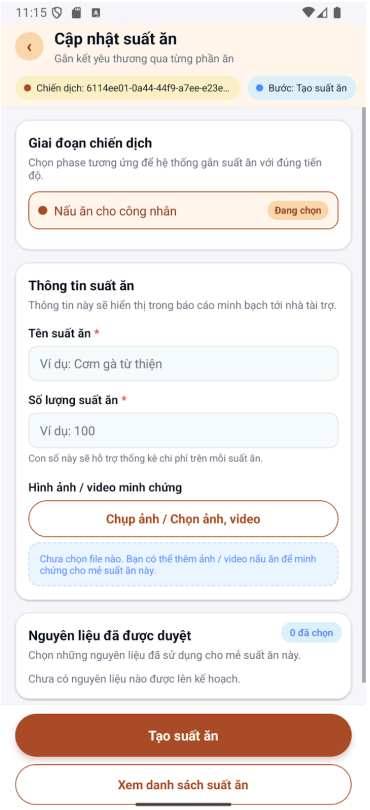


- **Function details:**

- **Data requirement:** campaignId
- **Validation:**
  - CampaignId
    - Type: String
- **Business rules:** none

3.14.2 Create Meal Batches

- **Function trigger:** Users click the "Tạo suất ăn" button in the Meal Batches creation page.
- **Function description:**
  - **Actors:** Kitchen Staff

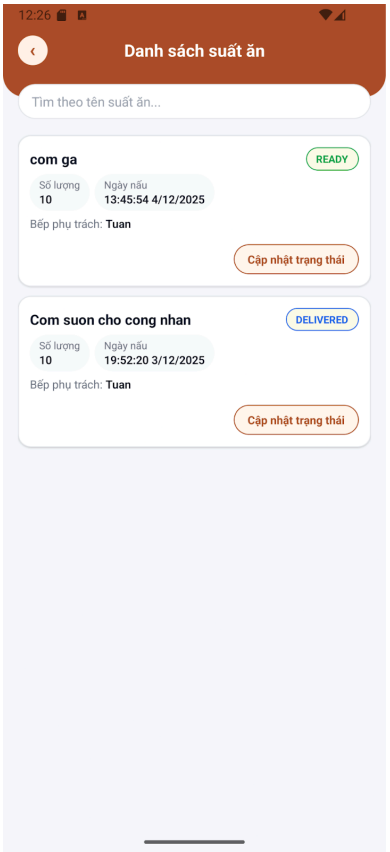


  - **Purpose:** Users want to create new meal batches for the campaign
- **Function details:**
  - **Data requirement:** campaignPhaseId, foodName, quantity, ingredientIds
  - **Validation:**
    - campaignPhaseId
      - Type: String
    - foodName
      - Type: String
    - quantity
      - Type: Number
    - ingredientIds
      - Type: Array of String

- **Business rules:** none

3.14.3 Update Meal Batches

- **Function trigger:** Users click "Hoàn thành/Hủy" button on meal batch detail page
- **Function description:**
  - **Actors:** Kitchen Staff
  - **Purpose:** Users want to update meal batches of a campaign

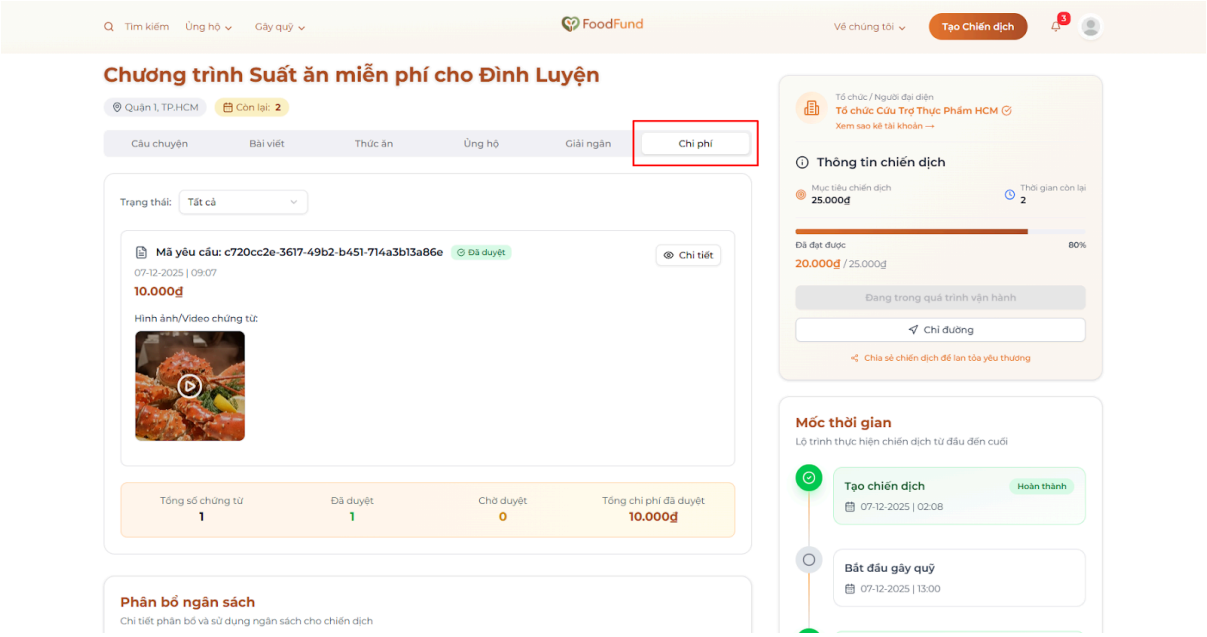


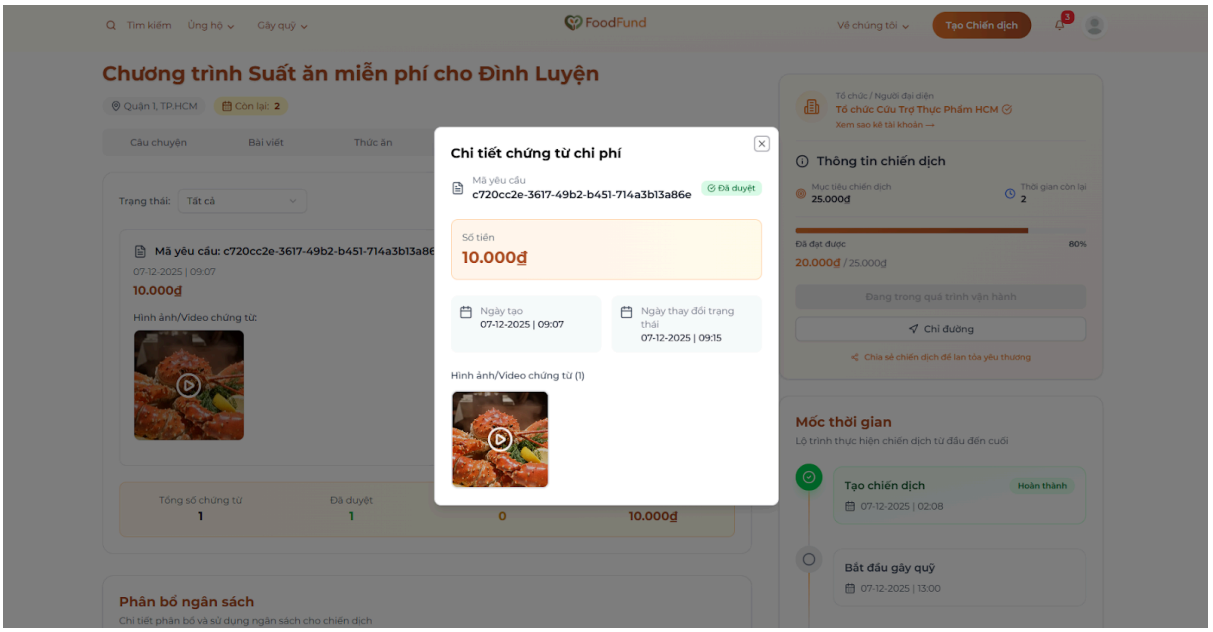
- **Function details:**
  - **Data requirement:** mealBatchId, status
  - **Validation:**
    - mealBatchId
      - Type: String
    - status
      - Type: String (PENDING | COOKING | READY)
    - cookedDate
      - Type: String
    - media
      - Type: Array of String
  - **Business rules:** none

3.15 Expense Proof

3.15.1 View Expense Proof

- **Function trigger:** Users click the “Chi phí” tab in the campaign detail page
- **Function description:**
  - **Actors:** Guest, Donor, Fundraiser, Admin, Kitchen Staff, Delivery Staff
  - **Purpose:** Users want to view expense proof of campaign

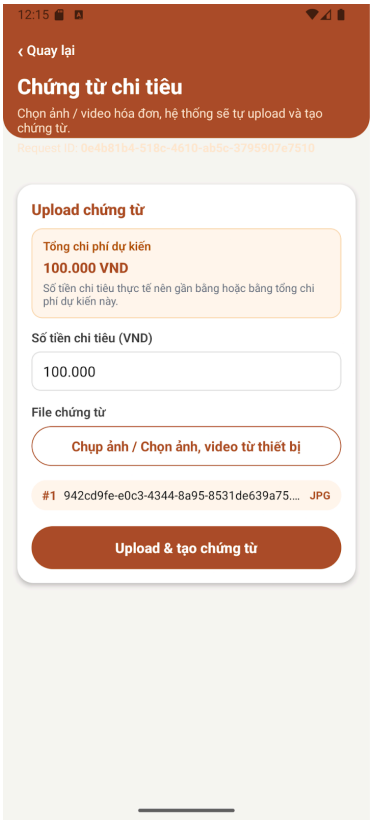




- **Function details:**
  - **Data requirement:** campaignId
  - **Validation:**
    - CampaignId
      - Type: String
  - **Business rules:** none

3.15.2 Create Expense Proof

- **Function trigger:** Users click "Tạo chứng từ" button on expense proof form page
- **Function description:**
  - **Actors:** Kitchen Staff
  - **Purpose:** Users want to submit expense proof with receipts/invoices for an ingredient request to verify actual spending

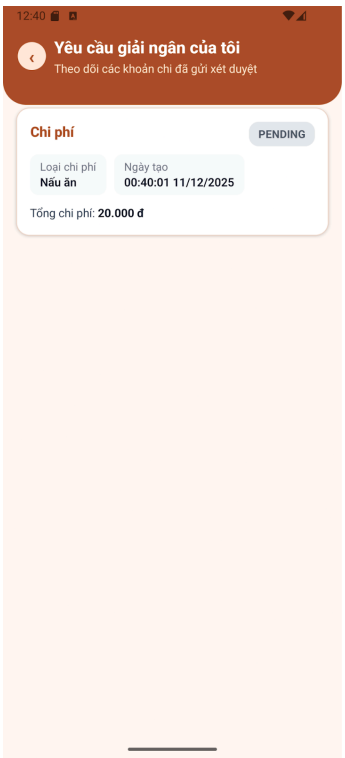


- **Function details:**
  - **Data requirement:** requestId, amount, mediaFileKeys
  - **Validation:**
    - requestId
      - Type: String
    - amount
      - Type: String
    - media file keys
      - Type: Array<String>
      - Array Size: 10
      - Optional
  - **Business rules:** none

3.15.3 Get My Expense Proof

- **Function trigger:** Users click the “Yêu cầu giải ngân của tôi” tab in the Giải ngân page

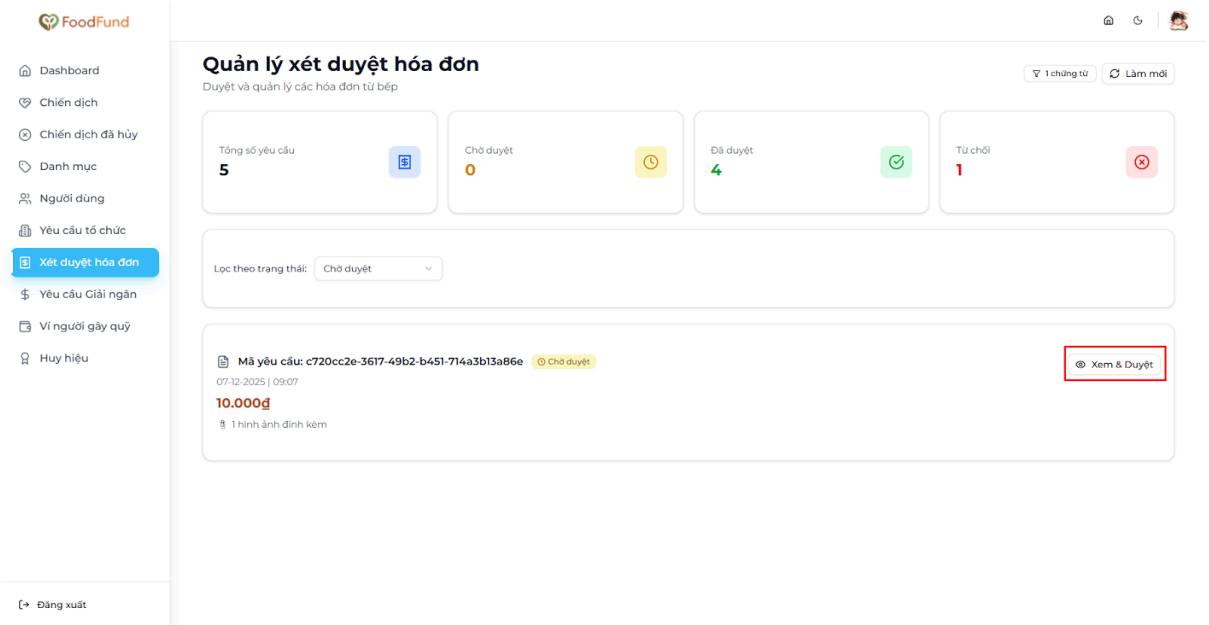
- **Function description:**
    - **Actors:** Kitchen Staff, Delivery Staff
    - **Purpose:** Users want to view their expense proof

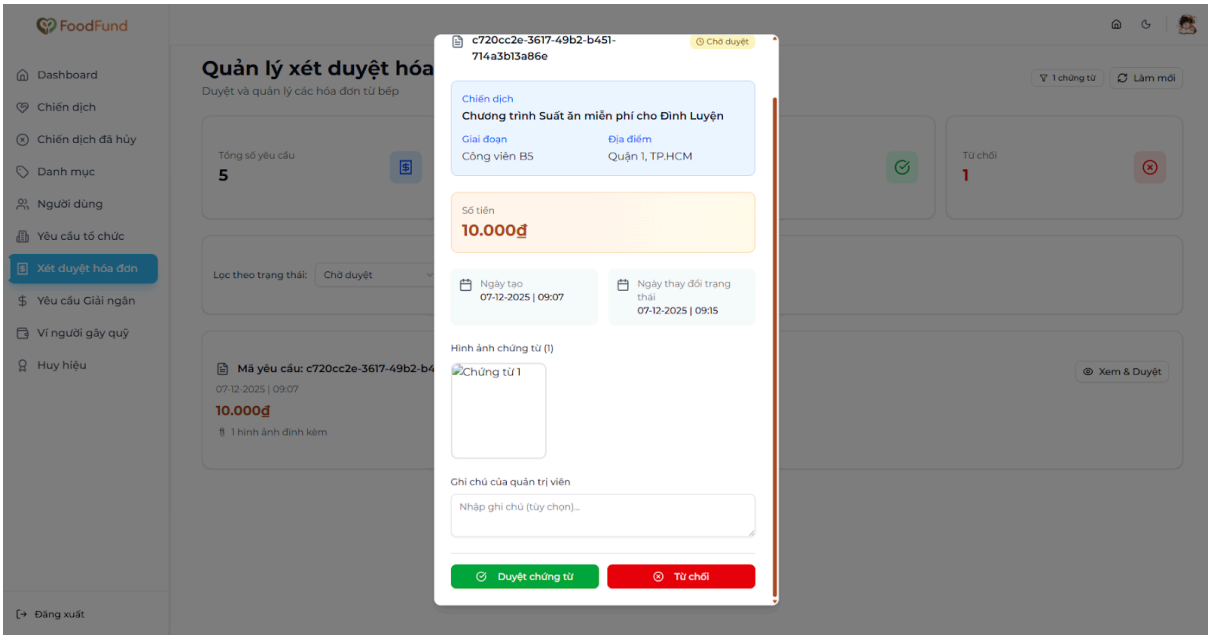


- **Function details:**
    - **Data requirement:** requestId
    - **Validation:**
        - requestId
            - Type: String
    - **Business rules:** none

3.15.4 Expense Proof Approval

- **Function trigger:** Admin clicks the “Duyệt/ Từ chối” button in “Xét duyệt hóa đơn” tab in the admin page
- **Function description:**
    - **Actors:** Admin
    - **Purpose:** Admin wants to approve/reject expense proof of a campaign



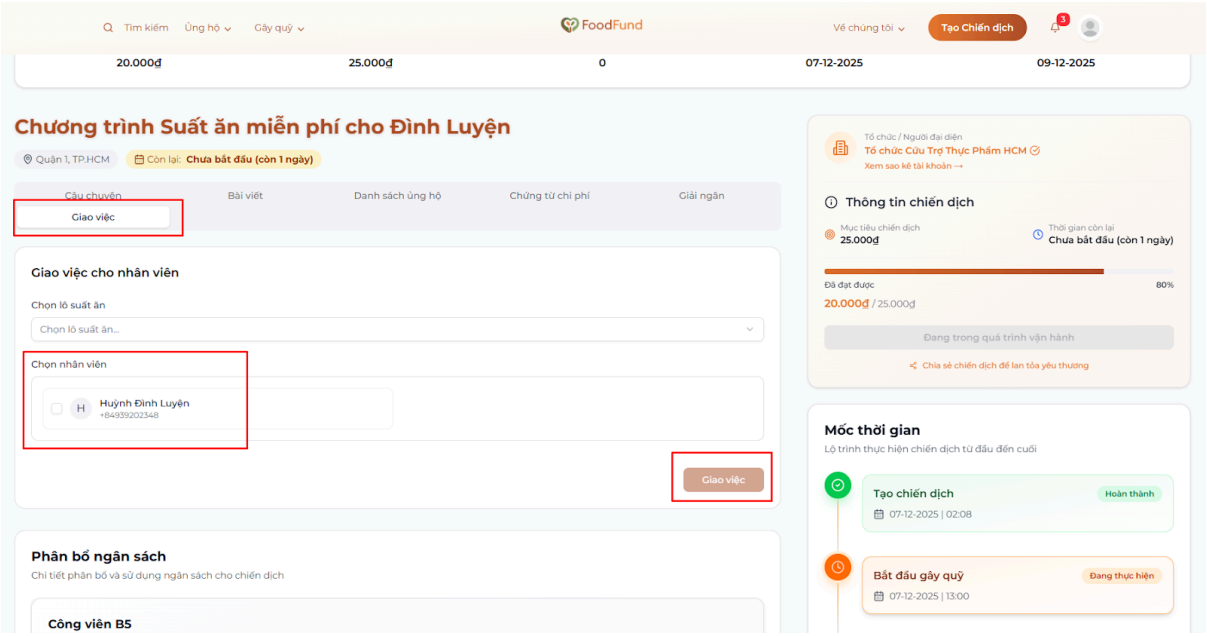


- **Function details:**
  - **Data requirement:** id, status, adminNote
  - **Validation:**
    - Id:
      - Type: String
    - Status:
      - Type: String
      - Must be PENDING, APPROVED, REJECTED
    - AdminNote:
      - String
      - Optional
  - **Business rules:** BR-02, BR-44, BR-42, BR-43, BR-40

3.16 Delivery Task

3.16.1 Assign Task To Delivery Staff

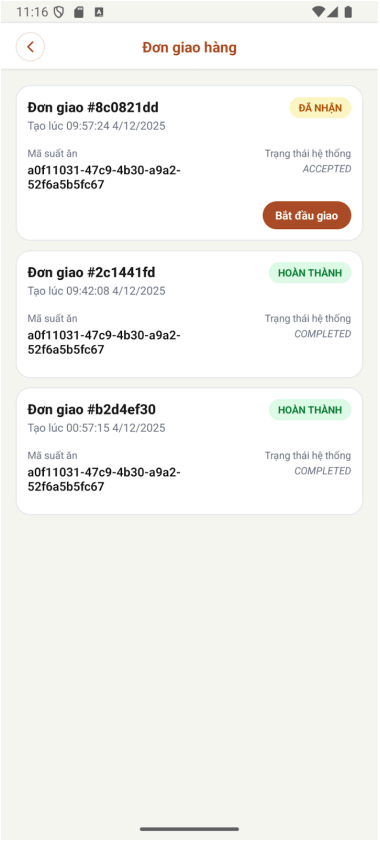
- **Function trigger:** Fundraiser clicks the “Giao việc” tab in the my campaign detail page
- **Function description:**
  - **Actors:** Fundraiser
  - **Purpose:** Fundraiser wants to assign task to delivery staff



- **Function details:**
  - **Data requirement:** mealBatchId, deliveryStaffIds
  - **Validation:** none
  - **Business rules:** BR-02, BR-10, BR-47, BR-48

3.16.2 View Tasks

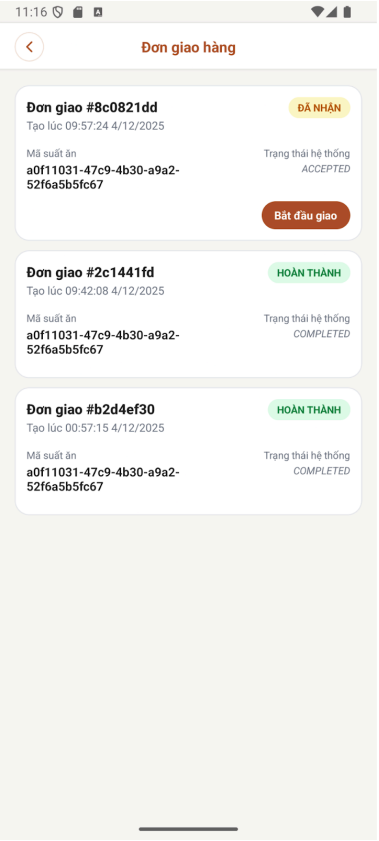
- **Function trigger:** Users click on a delivery task from the task list to view details
- **Function description:**
  - **Actors:** Delivery Staff
  - **Purpose:** Users want to view detailed information of a specific delivery task including status history



- **Function details:**
  - **Data requirement:** taskId
  - **Validation:**
    - taskId
      - Type: String
  - **Business rules:** BR-02

3.16.3 View My Tasks

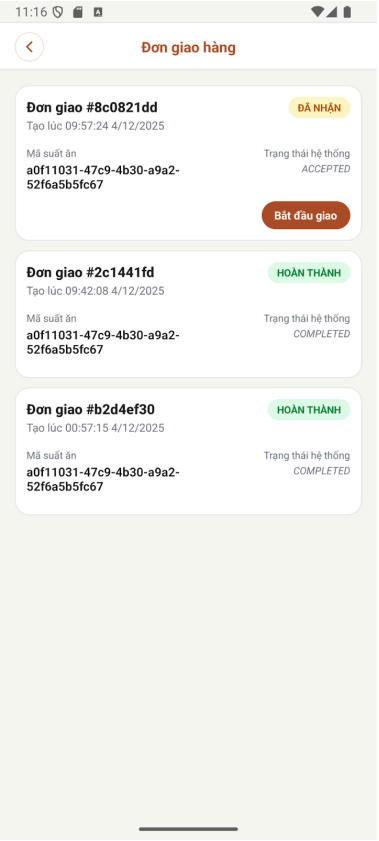
- **Function trigger:** Users navigate to delivery tasks page from delivery staff home screen
- **Function description:**
  - **Actors:** Delivery Staff
  - **Purpose:** Users want to view a list of delivery tasks assigned to them



- **Function details:**
  - **Data requirement:** limit, offset
  - **Validation:**
    - limit
      - Type: Number
    - offset
      - Type: Number
  - **Business rules:** BR-02

3.16.4 Change Status

- **Function trigger:** Users click status change button on delivery task detail page
- **Function description:**
  - **Actors:** Delivery Staff
  - **Purpose:** Users want to update the status of a delivery task (e.g., PENDING → IN_PROGRESS → DELIVERED)

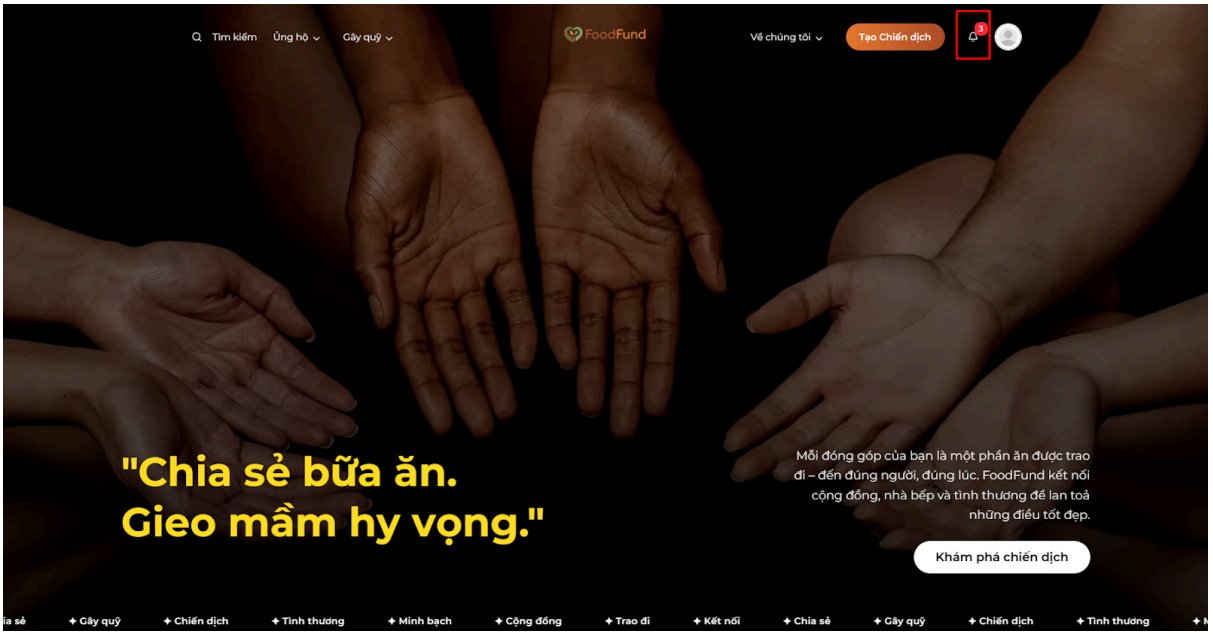


- **Function details:**
  - **Data requirement:** taskId, status
  - **Validation:**
    - taskId
      - Type: String
    - status
      - Type: String
  - **Business rules:** BR-02

3.17 Notification

3.17.1 View My Notifications

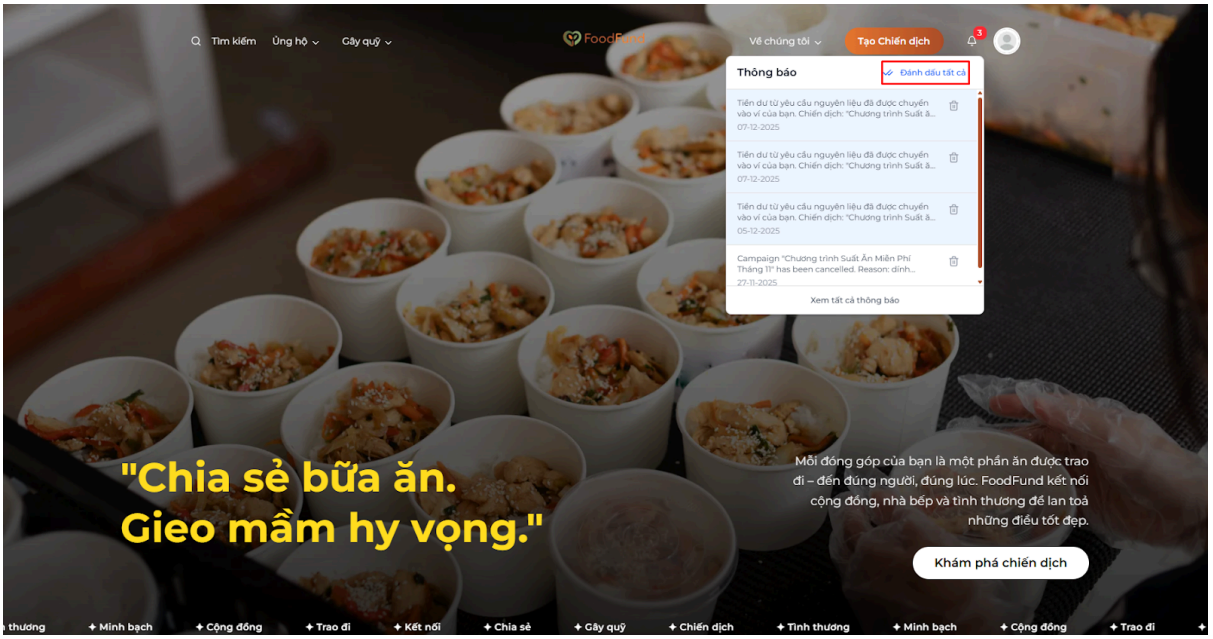
- **Function trigger:** Users click the bell icon in the navigation
- **Function description:**
  - **Actors:** Donor, Fundraiser, Admin, Kitchen Staff, Delivery Staff
  - **Purpose:** Users want to view their notifications



- **Function details:**
  - **Data requirement:** none
  - **Validation:** none
  - **Business rules:** BR-02

3.17.2 Mark All As Read

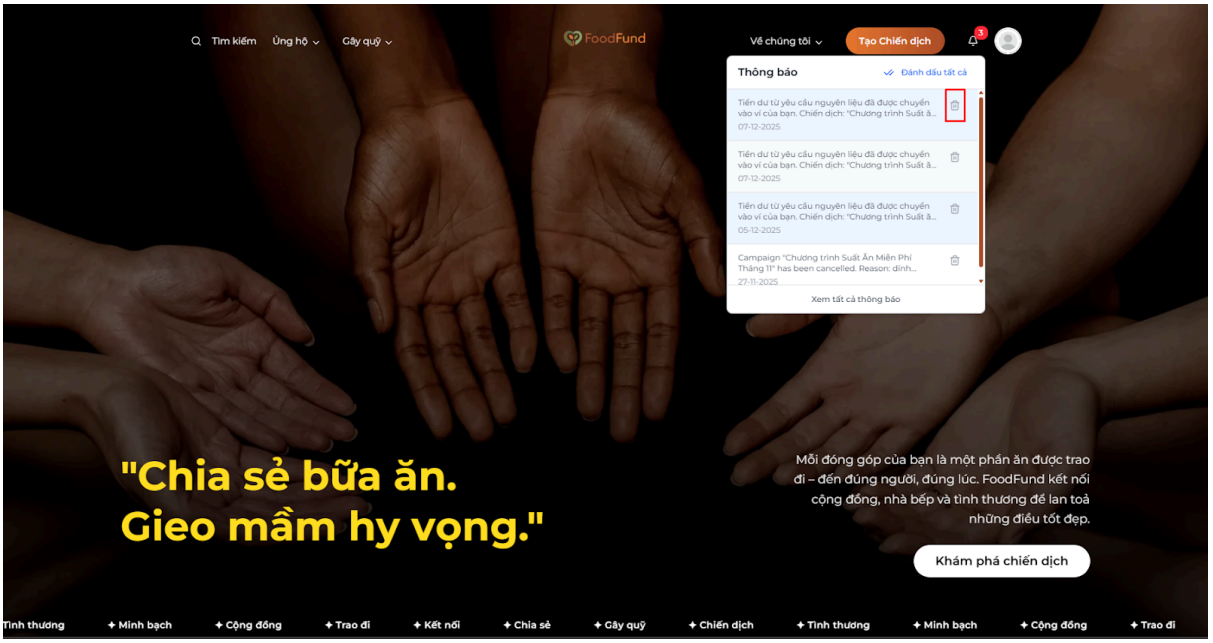
- **Function trigger:** Users click "Đánh dấu tất cả" in notification
- **Function description:**
  - **Actors:** Donor, Fundraiser, Admin, Kitchen Staff, Delivery Staff
  - **Purpose:** Users want to mark all as read their notifications



- **Function details:**
  - **Data requirement:** none
  - **Validation:** none
  - **Business rules:** BR-02, BR-34

3.17.3 Delete Notification

- **Function trigger:** Users click delete icon in notification
- **Function description:**
  - **Actors:** Donor, Fundraiser, Admin, Kitchen Staff, Delivery Staff
  - **Purpose:** Users want to delete their notifications

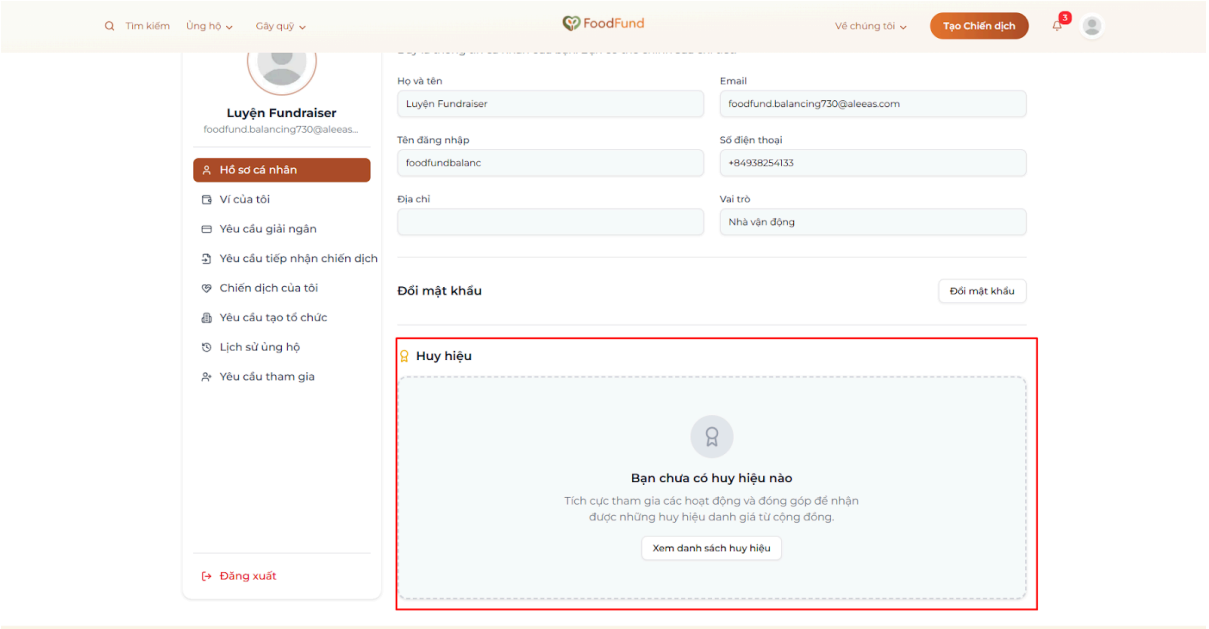


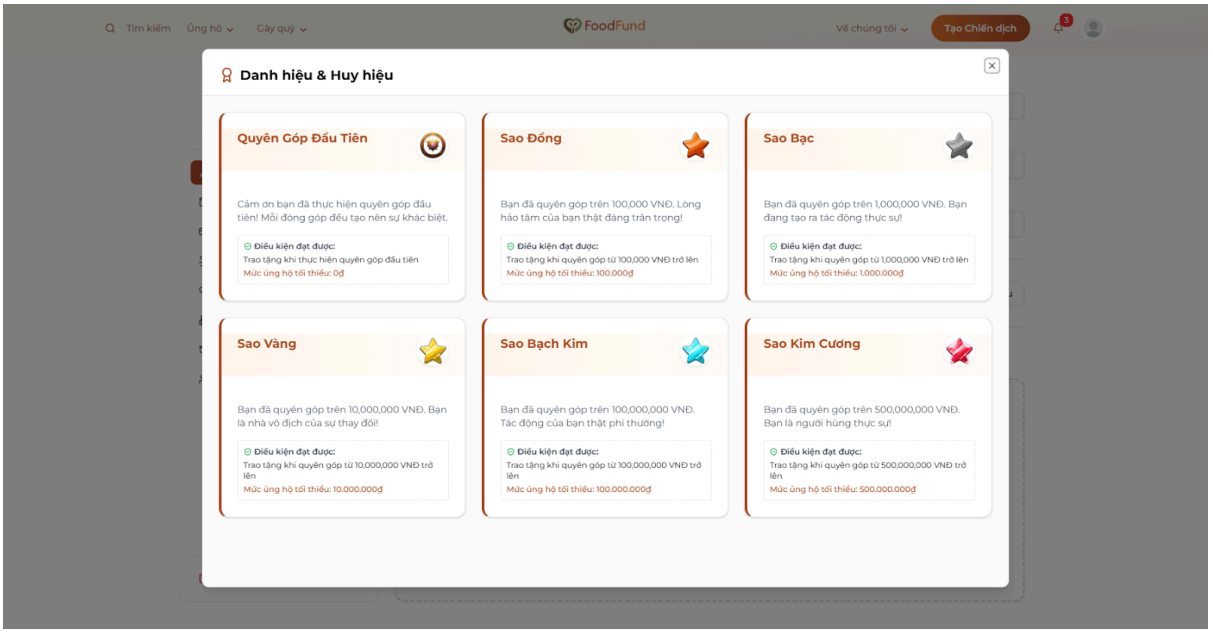
- **Function details:**
  - **Data requirement:** notificationId
  - **Validation:**
    - notificationId:
      - Type: String
  - **Business rules:** BR-02, BR-48

3.18 Badge

3.18.1 View All Badges

- **Function trigger:** Users click “Xem danh sách huy hiệu” in the profile page
- **Function description:**
  - **Actors:** Donor, Fundraiser, Admin, Kitchen Staff, Delivery Staff
  - **Purpose:** Users want to view all badges

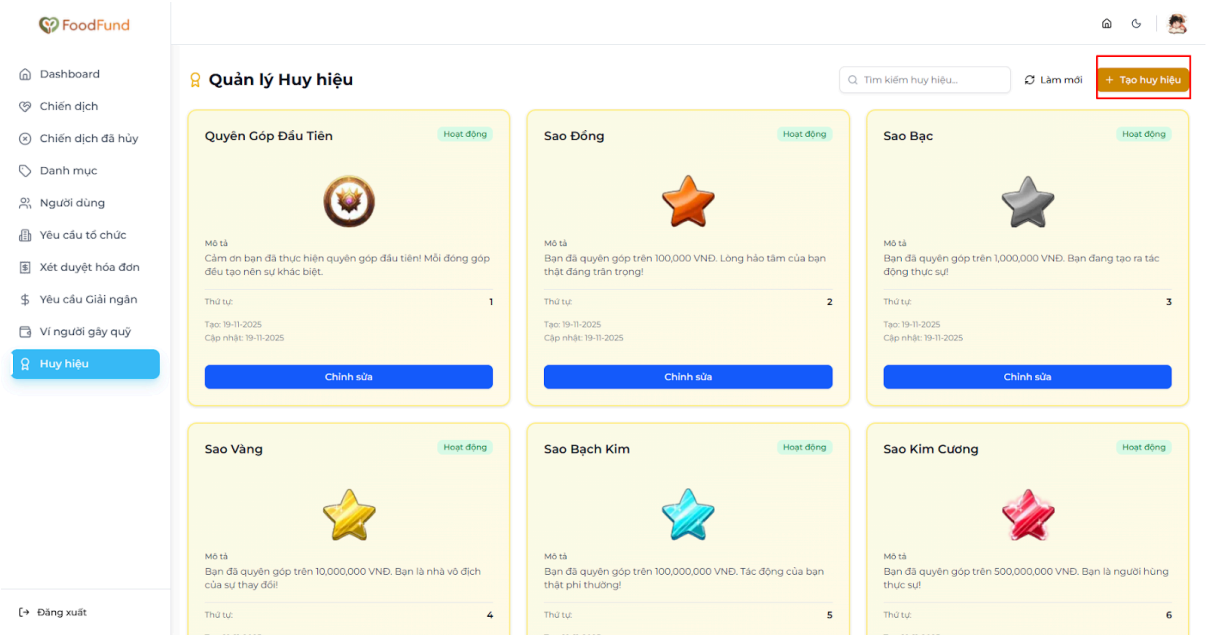

Tìm kiếm
Ủng hộ
Gây quỹ
FoodFund
Về chúng tôi
Tạo Chiến dịch
Luyện Fundraiser
foodfund.balancing730@aleeas...
Hồ sơ cá nhân
Ví của tôi
Yêu cầu giải ngân
Yêu cầu tiếp nhận chiến dịch
Chiến dịch của tôi
Yêu cầu tạo tổ chức
Lịch sử ủng hộ
Yêu cầu tham gia
Đăng xuất
Họ và tên
Luyện Fundraiser
Email
foodfund.balancing730@aleeas.com
Tên đăng nhập
foodfundbalanc
Số điện thoại
+84938254133
Địa chỉ
Vai trò
Nhà vận động
Đổi mật khẩu
Đổi mật khẩu
Huy hiệu
Bạn chưa có huy hiệu nào
Tích cực tham gia các hoạt động và đóng góp để nhận được những huy hiệu danh giá từ cộng đồng.
Xem danh sách huy hiệu

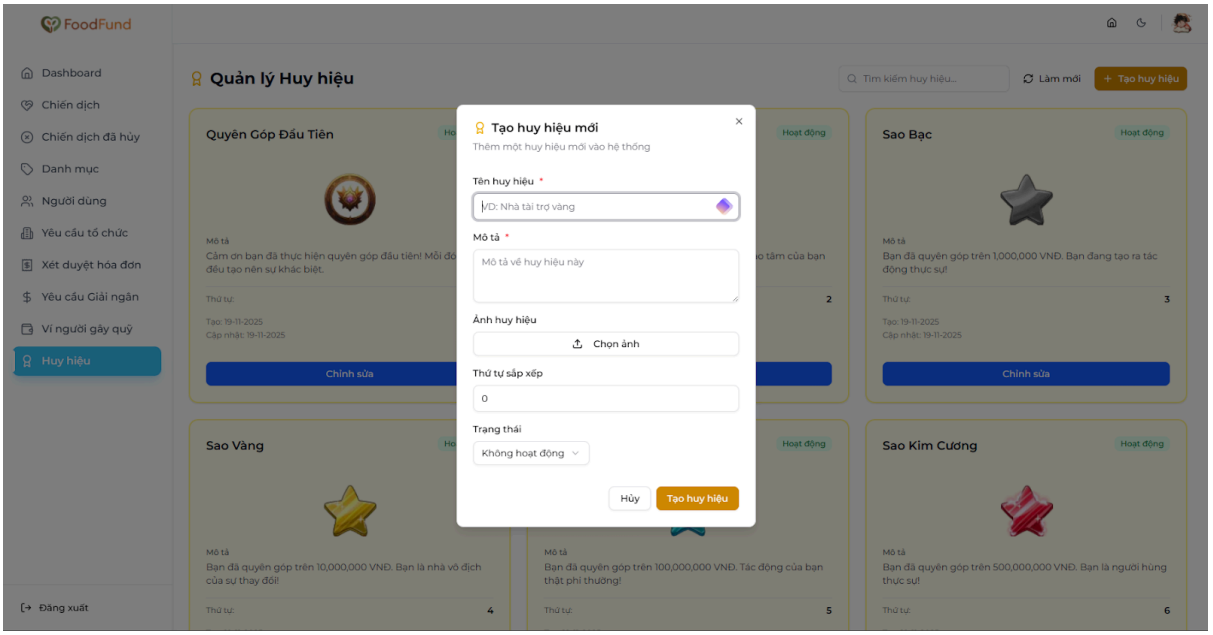

Danh hiệu & Huy hiệu
Quyên Góp Đầu Tiên
Cảm ơn bạn đã thực hiện quyên góp đầu tiên! Mỗi đóng góp đều tạo nên sự khác biệt.
Điều kiện đạt được:
Trao tặng khi thực hiện quyên góp đầu tiên
Mức ủng hộ tối thiểu: 0đ
Sao Đồng
Bạn đã quyên góp trên 100,000 VNĐ. Lòng hảo tâm của bạn thật đáng trân trọng!
Điều kiện đạt được:
Trao tặng khi quyên góp từ 100,000 VNĐ trở lên
Mức ủng hộ tối thiểu: 100.000đ
Sao Bạc
Bạn đã quyên góp trên 1,000,000 VNĐ. Bạn đang tạo ra tác động thực sự!
Điều kiện đạt được:
Trao tặng khi quyên góp từ 1,000,000 VNĐ trở lên
Mức ủng hộ tối thiểu: 1.000.000đ
Sao Vàng
Bạn đã quyên góp trên 10,000,000 VNĐ. Bạn là nhà vô địch của sự thay đổi!
Điều kiện đạt được:
Trao tặng khi quyên góp từ 10,000,000 VNĐ trở lên
Mức ủng hộ tối thiểu: 10.000.000đ
Sao Bạch Kim
Bạn đã quyên góp trên 100,000,000 VNĐ. Tác động của bạn thật phi thường!
Điều kiện đạt được:
Trao tặng khi quyên góp từ 100,000,000 VNĐ trở lên
Mức ủng hộ tối thiểu: 100.000.000đ
Sao Kim Cương
Bạn đã quyên góp trên 500,000,000 VNĐ. Bạn là người hùng thực sự!
Điều kiện đạt được:
Trao tặng khi quyên góp từ 500,000,000 VNĐ trở lên
Mức ủng hộ tối thiểu: 500.000.000đ

- **Function details:**
  - **Data requirement:** none
  - **Validation:** none
  - **Business rules:** none

3.18.2 Create Badge

- **Function trigger:** Admin clicks “Tạo huy hiệu” in the admin page
- **Function description:**
  - **Actors:** Admin
  - **Purpose:** Admin wants to create badge

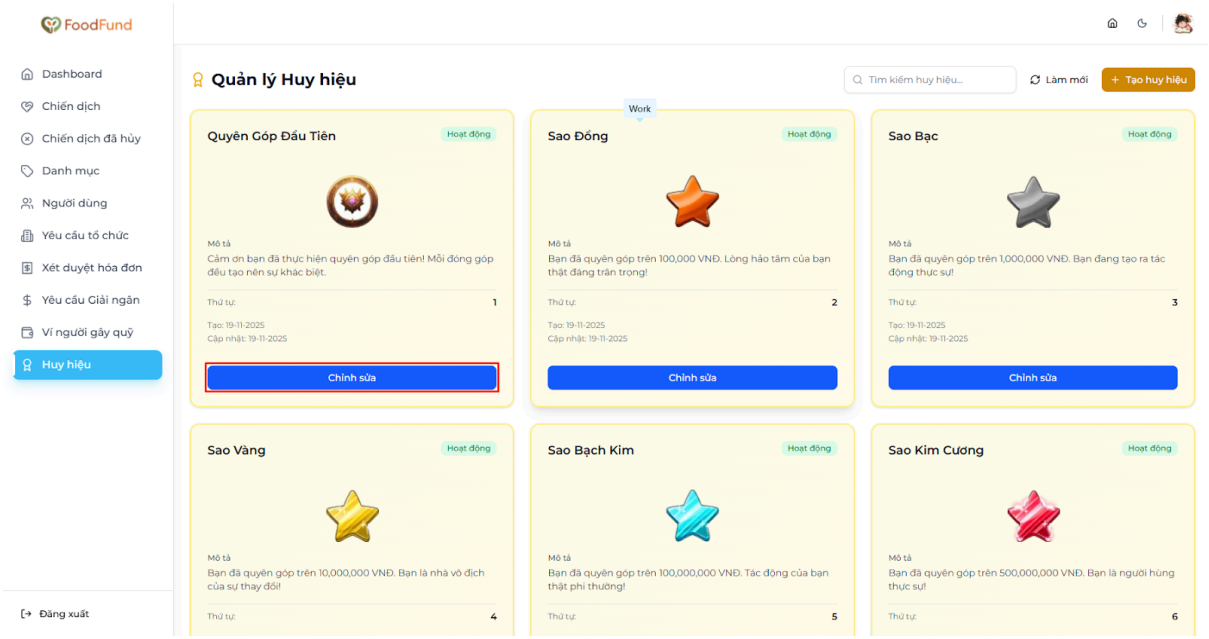


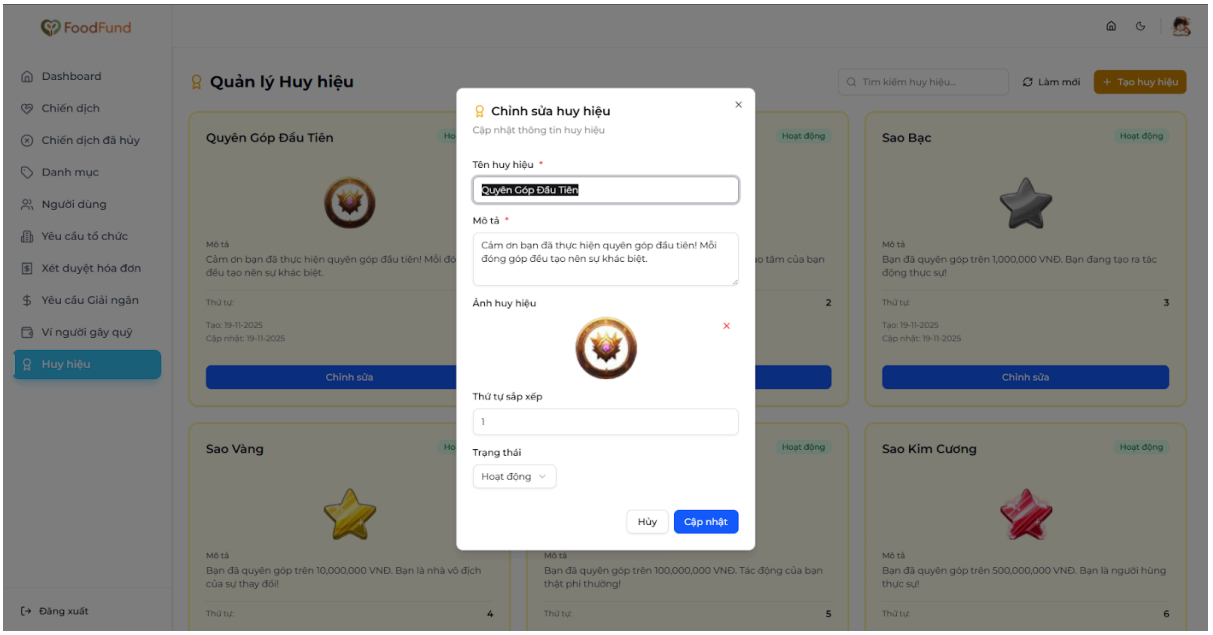


- **Function details:**
  - **Data requirement:** description, icon_url, is_active, name, sort_order
  - **Validation:**
    - Description:
      - Type: String
      - Length: 5-100 characters
    - Icon_Url:
      - Type: String
    - Is_Active:
      - Type: Boolean
    - Name:
      - Type: String
    - Sort_Order:
      - Type: Number
  - **Business rules:** none

3.18.3 Update Badge

- **Function trigger:** Admin clicks “Chỉnh sửa” in the admin page
- **Function description:**
  - **Actors:** Admin
  - **Purpose:** Admin wants to update badge



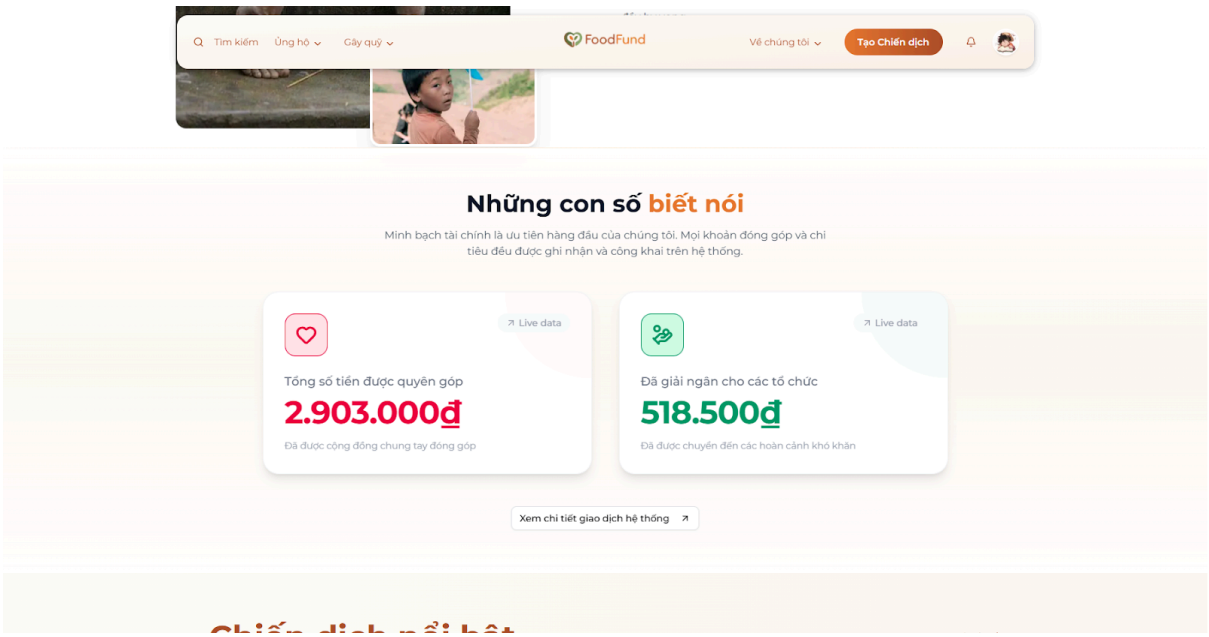


- **Function details:**
  - **Data requirement:** updateBadgeId, description, icon_url, is_active, name, sort_order
  - **Validation:**
    - UpdateBadgeId:
      - Type: String
    - Description:
      - Type: String
      - Length: 5-100 characters
    - Icon_Url:
      - Type: String
    - Is_Active:
      - Type: Boolean
    - Name:
      - Type: String
    - Sort_Order:
      - Type: Number
  - **Business rules:** none

3.19 Wallet

3.19.1 View System Wallet

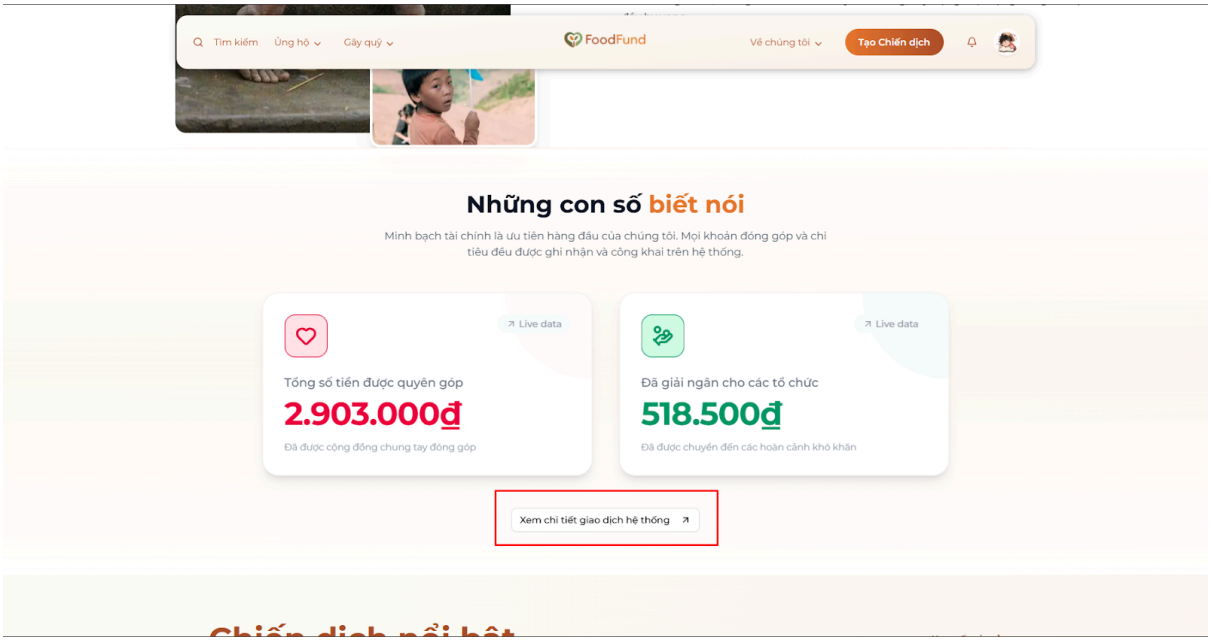
- **Function trigger:** Users access home page
- **Function description:**
  - **Actors:** Guest, Donor, Fundraiser, Admin, Kitchen Staff, Delivery Staff
  - **Purpose:** Users want to view system wallet

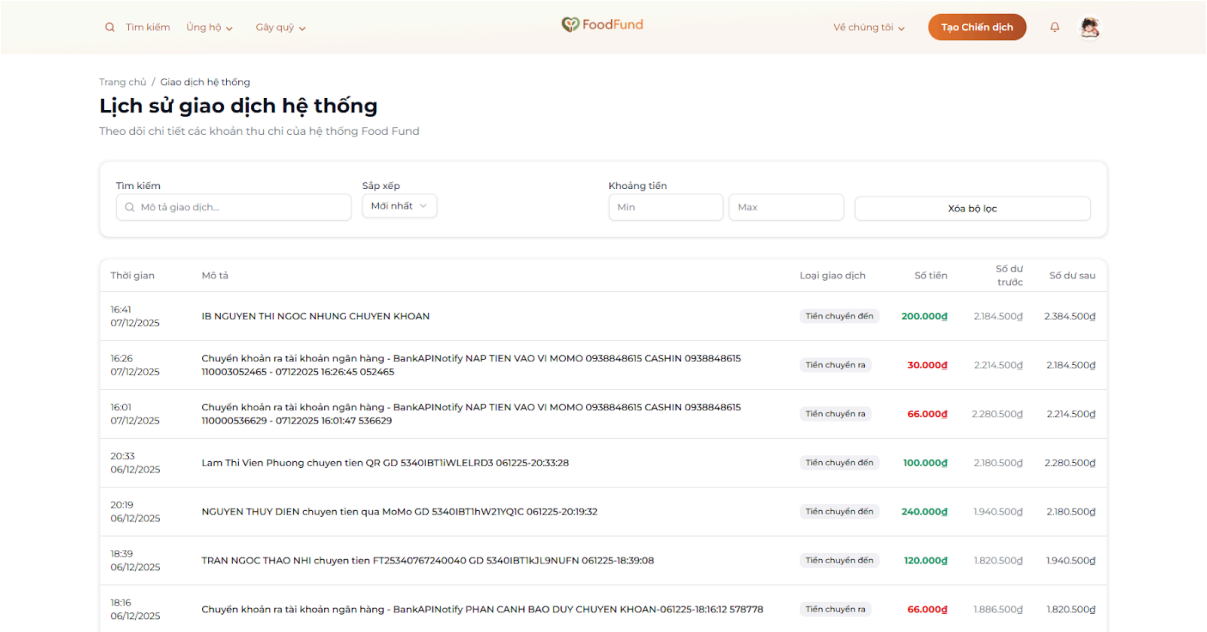


- **Function details:**
  - **Data requirement:** none
  - **Validation:** none
  - **Business rules:** none

3.19.2 View System Wallet Transaction

- **Function trigger:** Users click “Xem chi tiết giao dịch hệ thống” button
- **Function description:**
  - **Actors:** Guest, Donor, Fundraiser, Admin, Kitchen Staff, Delivery Staff
  - **Purpose:** Users want to view system wallet transaction

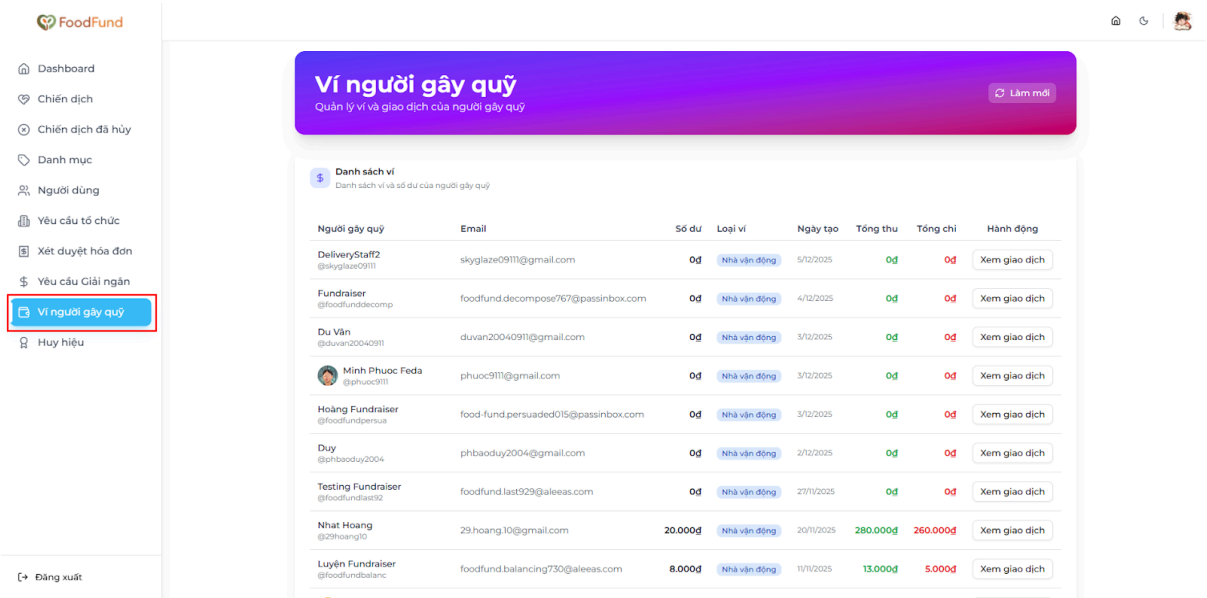




- **Function details:**
  - **Data requirement:** walletId
  - **Validation:**
    - WalletId:
      - Type: String
  - **Business rules:** none

3.19.3 View Fundraiser Wallet

- **Function trigger:** Admin clicks “Ví người gây quỹ” tab in admin page
- **Function description:**
  - **Actors:** Admin
  - **Purpose:** Admin wants to view fundraiser wallet

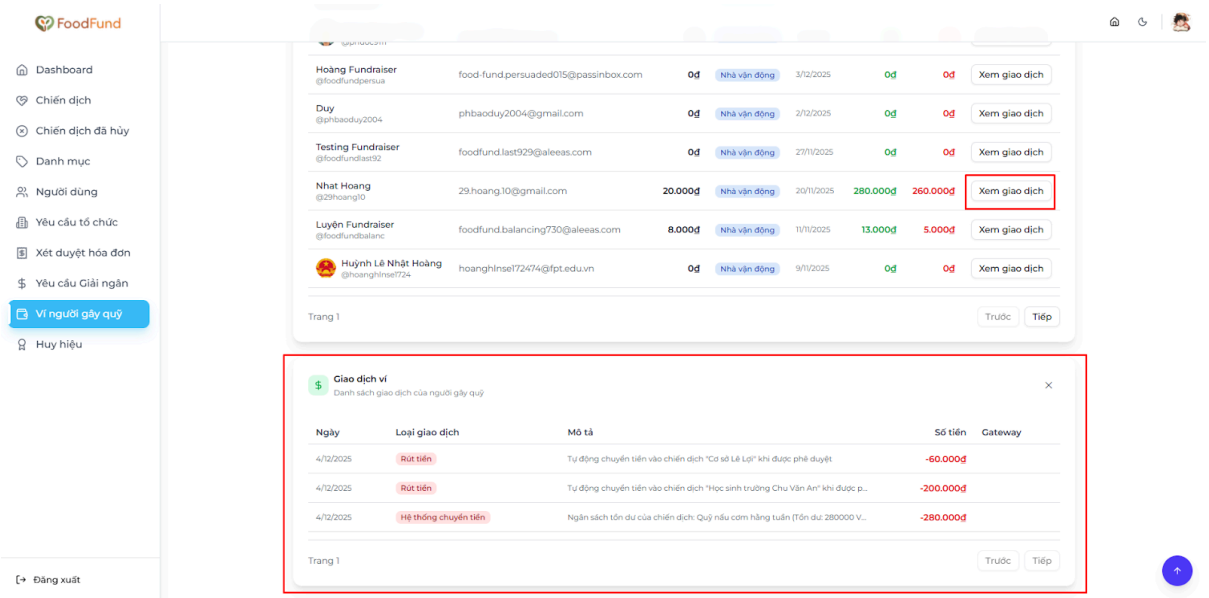


- **Function details:**
  - **Data requirement:** none
  - **Validation:** none

- **Business rules:** BR-02

3.19.4 View Fundraiser Wallet Transaction

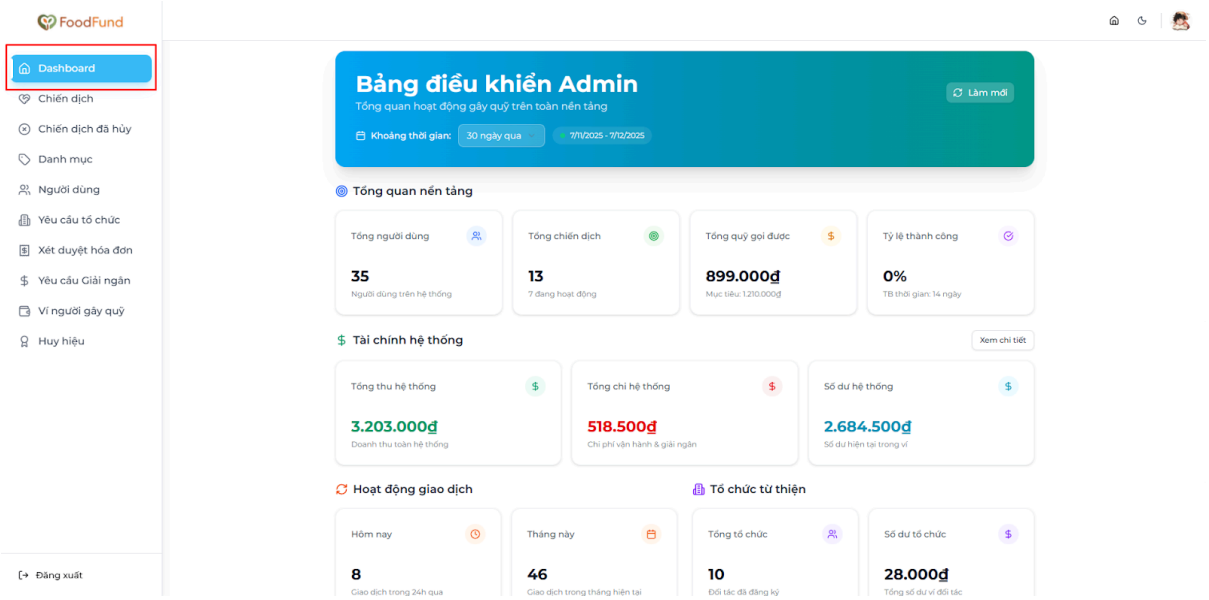
- **Function trigger:** Admin clicks “Xem giao dịch” button in fundraiser wall page
- **Function description:**
  - **Actors:** Admin
  - **Purpose:** Admin wants to view fundraiser wallet transaction



- **Function details:**
  - **Data requirement:** userId
  - **Validation:**
    - UserId:
      - Type: String··
  - **Business rules:** BR-02

3.19.5 Get Entire Platform Stats

- **Function trigger:** Admin accesses admin page
- **Function description:**
  - **Actors:** Admin
  - **Purpose:** Admin wants to view entire platform stats



- **Function details:**
  - **Data requirement:** none
  - **Validation:** none
  - **Business rules:** BR-02

4. Non-Functional Requirements

4.1 External Interfaces

- The application shall operate entirely in the Vietnamese language to ensure accessibility for local users.
- The system shall maintain compatibility with the latest stable releases of major browsers, including Chrome, Firefox, and Safari.
- The interface design shall be responsive, adapting smoothly to various screen sizes and device orientations.
- The system shall provide clear, concise, and user-friendly error messages whenever issues occur during external service interactions.
- All icons shall be easily recognizable and accurately represent their intended functions.
- Text elements must use legible fonts that support readability across different display resolutions.

4.2 Quality Attributes

4.2.1 Usability

- The platform shall be designed to be user-friendly and easy to navigate.
- The user interface shall follow modern web design principles to ensure a consistent and intuitive experience.
- Navigation and layout shall follow familiar and predictable patterns to enhance usability.

4.2.2 Reliability

- The system shall maintain an uptime of at least **99%**, excluding scheduled maintenance periods.
- Data integrity must be ensured to prevent the loss or corruption of donation and transaction records.

4.2.3 Performance

- It shall be capable of processing multiple simultaneous donations or campaign submissions efficiently.
- The platform shall handle a high number of concurrent users during peak fundraising periods without performance degradation.

4.2.4 Security

- User authentication shall be required for accessing donor, fundraiser, and administrative features.
- All donation and fund transfer activities shall be fully logged and auditable to support transparency and traceability.
- The platform shall utilize trusted and verified payment gateways for processing financial transactions securely.

5. Requirement Appendix

5.1 Business Rules

ID	Rule Definition
BR-01	Each **user** must have a unique email address.
BR-02	An **organization** must be represented by exactly one registered **user**.
BR-03	An **organization** registration starts in **pending** status and requires administrative review to become **verified** or **rejected**.
BR-04	A fundraiser organization is defined as a non-personal, community-oriented entity. It is not the personal property of the representative user, and funds within the organization are tied to the entity, not the individual.
BR-05	**Fundraisers** only manage and are responsible for the **organization** they create.
BR-06	**Fundraisers** can only create one **organization** at a time.
BR-07	**Kitchen staff** and **delivery staff** can only join one **organization** at a time.
BR-08	Upon the creation of a new **organization**, a corresponding **wallet** record must be automatically created and linked to the **fundraiser**.
BR-09	**Fundraisers** only create a **campaign** when they have an **organization**.
BR-10	**Fundraisers** only edit a **campaign** when the status of the **campaign is pending**.
BR-11	**Fundraisers** only delete a **campaign** when the status of the **campaign is pending or rejected**.
BR-12	**Upon creation**, a **campaign** must be assigned a **pending** status and cannot be publicly visible or receive donations until administrative approval.
BR-13	The time difference between the fundraising start date and fundraising end date of any campaign must be at least 24 hours.
BR-14	A **pending campaign** must be manually reviewed and moved to **approved** or **rejected** status by a user with the **admin** role.
BR-15	An **approved campaign** shall automatically transition to **active** status when the system date reaches the **fundraising start date**.
BR-16	An **active campaign** shall automatically transition to **processing** status when the system date passes the **fundraising end date**.

ID	Rule Definition
BR-17	A **campaign** extension is limited to a single occurrence and must be for a predefined duration (e.g., 7, 14, or 21 days).
BR-18	If a **campaign** is extended, an automated **notification** must be sent to all **existing donors**, informing them of the extension and the new end date.
BR-19	A **campaign** can only be marked **completed** by a **fundraiser** after all its associated **campaign phases** are marked **completed**.
BR-20	Admin can only change the status of a campaign from pending to approved or rejected**.**
BR-21	If a campaign's **received amount** exceeds its **target amount**, the surplus funds shall be transferred to the associated **wallet** upon the **campaign's completion** (before the next campaign starts).
BR-22	If a **campaign's received amount** is less than its **target amount** by its end date, the **fundraiser** may request a one-time extension.
BR-23	The system will not show **rejected campaigns** for **donors**.
BR-24	If a **campaign** reaches its final deadline and transitions to the **ended** status with a received amount of less than system config percentage of the target goal, the collected funds are **automatically transferred** to the organization wallet to support the future campaigns.
BR-25	Fundraisers only manage and are responsible for the **posts of the campaigns** they create.
BR-26	Only authenticated users can **read posts of campaigns**.
BR-27	Only authenticated users can **react and comment** on posts.
BR-28	In the event of a campaign cancellation due to administrative intervention (e.g., fraud or violation), the system must adhere to the following hierarchy of actions: 1. **Priority Action (Re-assignment):** The system first attempts to transfer the campaign and its collected funds to another eligible Fundraiser Organization to continue operations. 2. **Fallback Action (Refund):** A financial refund to individual donors is triggered **only if** no suitable organization can be found or accepts the re-assignment within a predefined timeframe.
BR-29	A campaign reassignment request sent to an organization is valid only for a predefined period (e.g., 7 days). If the organization does not respond within this timeframe, the system will automatically expire the request.
BR-30	To accept a re-assigned campaign(BR-28), a target organization must satisfy two strict conditions: 1. The organization must be fully **verified**. 2. The organization must **not** have any campaign currently in the **active** or **processing** state (to ensure they have sufficient capacity and focus to rescue the cancelled campaign).
BR-31	Upon the successful acceptance of a re-assignment(BR-28), the ownership of the campaign and the management rights of the undisbursed funds are legally and technically transferred to the **new organization**. The campaign status shall be preserved or reset to **active** based on the new owner's revised plan, subject to admin approval.

ID	Rule Definition
BR-32	The system must only allow monetary donations. In-kind donations (e.g., physical food) are explicitly excluded.
BR-33	The platform's Terms of Use, which must be accepted by the **donor** before payment, must clearly state **how surplus funds (over-funding)** and **failed-campaign funds (under-funding**) will be handled (i.e., transferred to the **organization wallet**).
BR-34	Donor/Guest can **donate** at least 1 thousand VND and at most 500 million thousand VND
BR-35	**Fund disbursement** shall be processed in distinct phases (e.g., Ingredients, Cooking, Delivery) based on **admin**-approved **expense requests**.
BR-36	Disbursement requests for a specific phase (e.g., Cooking) can only be created after the previous phase (e.g., Ingredients) implies readiness (e.g., Ingredients purchased).
BR-37	The **inflow transaction** must include a proof for the **admin** to upload the proof of the manual bank transfer.
BR-38	If the actual cost of an expense phase (e.g., ingredients) is less than the disbursed amount, the **fundraiser** is obligated to spend the remaining balance on items within the same approved category (e.g., additional ingredients).
BR-39	**Fundraisers** must purchase **ingredients** from verifiable suppliers (e.g., "Bách Hóa Xanh").
BR-40	For any single campaign phase, only one ingredient request and one operation request can be created. If any request is rejected, the requests can be created again.
BR-41	If the actual cost of an expense phase (e.g., ingredients) is **greater** than the disbursed amount, the organization is obligated to cover the deficit using funds via a self-funding transaction.
BR-42	Each uploaded expense proof must be individually reviewed by the admin.
BR-43	The organization wallet is designated to hold: 1. Surplus funds transferred from completed campaigns. 2. Funds from failed campaigns (under system config percentage funding). 3. Surplus funds transferred from ingredient request amount is less than ingredient budget amount of campaign.
BR-44	If a **delivery task** (e.g., 500 meals) experiences a partial loss (e.g., 100 meals destroyed), the delivery task status must be marked as **failed** and logged for auditing.

Table 17. Business Rules

5.2 Common Requirements

- CR-01: All user actions must be authenticated and authorized based on their assigned roles (Donor, Fundraiser, Delivery Staff, Kitchen Staff, Admin).
- CR-02: The user interface must be compatible across major web browsers and adhere to established usability guidelines to ensure a consistent user experience.
- CR-03: All financial activities shall be processed through secure and trusted payment gateways.
- CR-04: All donation and fund-related transactions (e.g., donations, fund disbursements) must be fully logged and auditable.

5.3 Application Messages List

#	Message code	Message Type	Context	Content
1	MSG01	In line	There is not any search result	*No search results.*
2	MSG02	In red, under the text box	Input-required fields are empty	*The * field is required.*
3	MSG03	Toast message	Updating asset(s) information successfully	*Update asset(s) successfully.*
4	MSG04	Toast message	The user successfully logs in.	*Login successful. Welcome back!*
5	MSG05	Toast message	The user enters the wrong email or password.	*Invalid email or password. Please try again.*
6	MSG06	Toast message	The user successfully registers an account.	*Registration successful. Please check your email to verify.*
7	MSG07	Toast message	The user tries to register with an existing email.	*Delete asset(s) successfully.*
8	MSG08	Toast message	**Donor** makes a successful donation.	*Exceed max length of {max_length}.*
9	MSG09	Toast message	**Donor's** donation attempt fails.	*Incorrect user name or password. Please check again.*
10	MSG10	Toast message	**Donor** successfully follows a campaign.	*You are now following this campaign.*
11	MSG11	Toast message	**Fundraiser** submits a new campaign for review.	*Campaign submitted successfully for approval.*
12	MSG12	Toast message	**Fundraiser** submits a disbursement proposal.	*Your disbursement proposal has been submitted.*
13	MSG13	Toast message	**Fundraiser** successfully updates campaign details.	*Campaign details have been updated.*
14	MSG14	Toast message	**The fundraiser** tries to submit an incomplete form.	*Please fill out all required fields before submitting.*
15	MSG15	Toast message	**Staff** sends a request to join an organization.	*Your request to join has been sent.*
16	MSG16	Toast message	**Fundraiser** approves a staff member's join request.	*A new member has been approved and added to your organization.*
17	MSG17	Toast message	**Kitchen Staff** submits a receipt.	*Your receipt has been uploaded successfully.*
18	MSG18	Toast message	**Kitchen Staff** updates a meal batch status.	*Meal batch status has been updated.*

#	Message code	Message Type	Context	Content
19	MSG19	Toast message	**Delivery Staff** updates a delivery status.	*Delivery status has been updated successfully.*
20	MSG20	Toast message	**Kitchen Staff** submits a withdrawal request.	*Your withdrawal request for ingredients has been submitted.*
21	MSG21	Toast message	The user uploads a file that is too large.	*Upload failed. File size must be under 5MB.*
22	MSG22	Toast message	The user uploads an invalid file type for a receipt.	*Upload failed. Please use a valid file type (JPG, PNG).*
23	MSG23	Toast message	**Admin** approves an organization request.	*The organization has been approved.*
24	MSG24	Toast message	**Admin** approves a campaign request.	*The campaign has been approved and is now live.*
25	MSG25	Toast message	**Admin** approves a disbursement request.	*The disbursement proposal has been approved.*
26	MSG26	Toast message	**Admin** rejects any pending request.	*The request has been successfully rejected.*
27	MSG27	Toast message	**Admin** bans a user account.	*User accounts have been banned.*
28	MSG28	Toast message	**Admin** updates a campaign category.	*Campaign categories have been updated.*
29	MSG29	Toast message	Any user's session is about to expire.	*Your session is about to expire. You will be logged out soon.*
30	MSG30	Toast message	Any user's profile information is updated.	*Your profile has been updated successfully.*
31	MSG31	Toast message	An action is completed but there's no internet.	*Action failed. Please check your internet connection.*
32	MSG32	Toast message	A generic server error occurs.	*An unexpected error occurred. Please try again later.*

Table 18. Application Messages List

5.4 Other Requirements

Additional requirement might include:

- Audit logging: The system shall record key user and financial activities for transparency and compliance.
- Data Retention: User and transaction data shall be stored securely for a defined period and deleted when no longer needed.
- Scalability & Maintenance: The system shall support easy expansion and updates with minimal downtime.
- Accessibility: The platform must conform to accessibility standards (e.g., WCAG 2.1) to ensure usability for all users

IV. Software Design Description

1. System Design

1.1 System Architecture

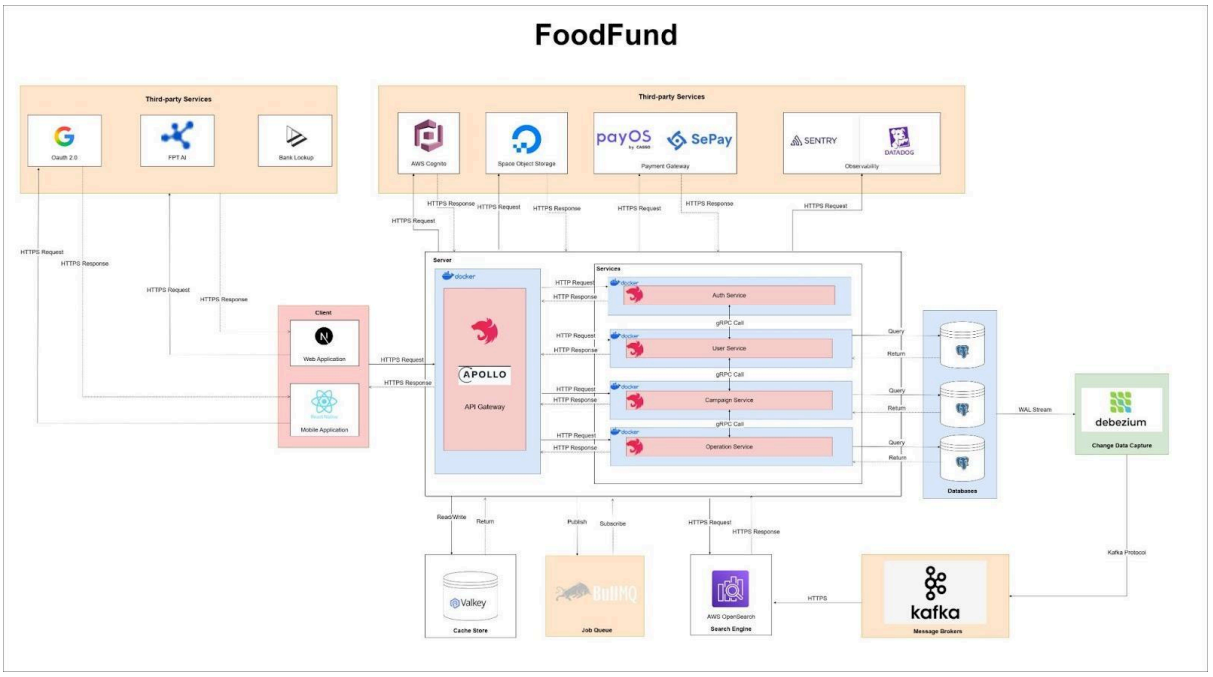


Figure 11. System Architecture Diagram

1.1.1 Overview

FoodFund is composed of a frontend layer (featuring both Web and Mobile applications) and a microservices-based backend that communicates via an API Gateway. The backend integrates with a range of external providers, including services for Payments(PayOS and Sepay), Storage(Digital Ocean Space), Search engines, and Observability(DataDog). Internally, the system utilizes gRPC for fast service-to-service communication and relies on a message broker and caching layer to handle data flow. Each service manages its own distinct database and responsibilities that, together, form the complete FoodFund ecosystem.

1.1.2 Major Components

1.1.2.1 FoodFund Client

Description: The user-facing interfaces comprising the FoodFund Web Application and Mobile Application.

Responsibility: Handles user interactions, displays data, and initiates secure HTTPS requests to the backend.

1.1.2.2 FoodFund Server

1.1.2.2.1 API Gateway

Description: The intermediate layer functioning as the single entry point between client applications and the internal backend system.

Responsibility: Handles request routing to appropriate services, enforces security policies, manages load balancing, and abstracts internal system complexity to present a unified API surface.

1.1.2.2.2 Microservices

Description: Independent functional modules that house the core business logic of the application.

Responsibilities: Processes business rules, manages dedicated databases (Database per service), and communicates internally using high-performance gRPC calls.

1.1.2.2.3 Managed Database (PostgreSQL)

Description: Fully-managed relational database service providing persistent data storage for each microservice.

Responsibilities: Stores structured business data with ACID compliance, enforces data integrity, and integrates with Prisma ORM for type-safe database access.

1.1.2.3 External Systems

1.1.2.3.1 Digital Ocean Managed Valkey

Description: High-performance in-memory data store used for caching and session management.

Responsibilities: Caches frequently accessed data, stores user sessions, manages distributed locks, and implements rate limiting mechanisms.

1.1.2.3.2 BullMQ

Description: Redis-backed job queue system for managing asynchronous background tasks and workflows.

Responsibilities: Processes long-running tasks asynchronously, handles email notifications and badge awards, implements retry logic, and integrates with Datadog for monitoring.

1.1.2.3.3 Digital Ocean Spaces

Description: S3-compatible object storage service for managing user-generated content and static assets.

Responsibilities: Stores campaign images and user avatars, provides CDN-backed content delivery, and generates pre-signed URLs for secure uploads.

1.1.2.3.4 PayOS

Description: Vietnamese payment gateway integration for processing online donations and transactions.

Responsibilities: Creates payment links and QR codes, processes multiple payment methods, handles webhooks for real-time status updates, and implements payment security.

1.1.2.3.5 Sepay

Description: Bank transfer monitoring service that detects and verifies incoming bank transactions.

Responsibilities: Monitors bank accounts in real-time, matches transactions with donation codes, sends webhooks for confirmed transfers, and enables VietQR integration.

1.1.2.3.6 AWS Cognito

Description: Managed authentication and user identity service providing secure user management.

Responsibilities: Handles user registration and login, issues JWT tokens, manages user attributes, implements MFA, and syncs with application database.

1.1.2.3.7 Google OAuth2

Description: Third-party authentication provider enabling users to sign in with their Google accounts.

Responsibilities: Provides social login functionality, authenticates through OAuth2 flow, retrieves user profile information, and integrates with AWS Cognito.

1.1.2.3.8 Sentry

Description: Application monitoring and error tracking platform for identifying and debugging production issues.

Responsibilities: Captures runtime errors across microservices, provides detailed stack traces, tracks error frequency, and sends real-time alerts for critical issues.

1.1.2.3.9 Datadog

Description: Comprehensive observability platform for monitoring application performance, infrastructure, and business metrics.

Responsibilities: Collects metrics from all services, monitors API performance and database queries, provides distributed tracing, creates custom dashboards, and integrates with BullMQ for queue monitoring.

1.1.2.3.10 FPT AI API

Description: AI-powered identity verification service for authenticating Vietnamese national ID cards (CCCD).

Responsibilities: Extracts and validates information from ID card images using OCR, verifies document authenticity, performs face matching between ID photo and selfie, and returns structured citizen data.

1.1.2.3.11 Bank Lookup API

Description: Banking information lookup service for validating and retrieving bank account details.

Responsibilities: Verifies bank account numbers, retrieves account holder names, validates bank codes and branches, and provides real-time account status checks.

1.1.2.3.12 AWS OpenSearch

Description: Distributed search and analytics engine for full-text search and log analysis.

Responsibilities: Indexes campaign and user data for fast full-text search, provides advanced filtering and aggregation queries, enables fuzzy matching

1.1.2.3.13 Apache Kafka

Description: High-performance distributed event streaming platform serving as the backbone for event-driven architecture and real-time data pipelines.

Responsibilities: Acts as a central message broker, captures and persists data change events (CDC) from Debezium, ensures message ordering and durability, and distributes data to consumer services for asynchronous processing and synchronization.

1.1.2.3.14 Debezium

Description: Open-source distributed platform for Change Data Capture (CDC), built on top of Kafka Connect to turn existing databases into real-time event streams.

Responsibilities: Monitors the PostgreSQL Write-Ahead Log (WAL), captures row-level changes (INSERT, UPDATE, DELETE) in real-time, and streams these changes as standardized events into Apache Kafka topics with minimal impact on primary database performance.

1.2 Package Diagram

1.2.1 Backend Package Diagram

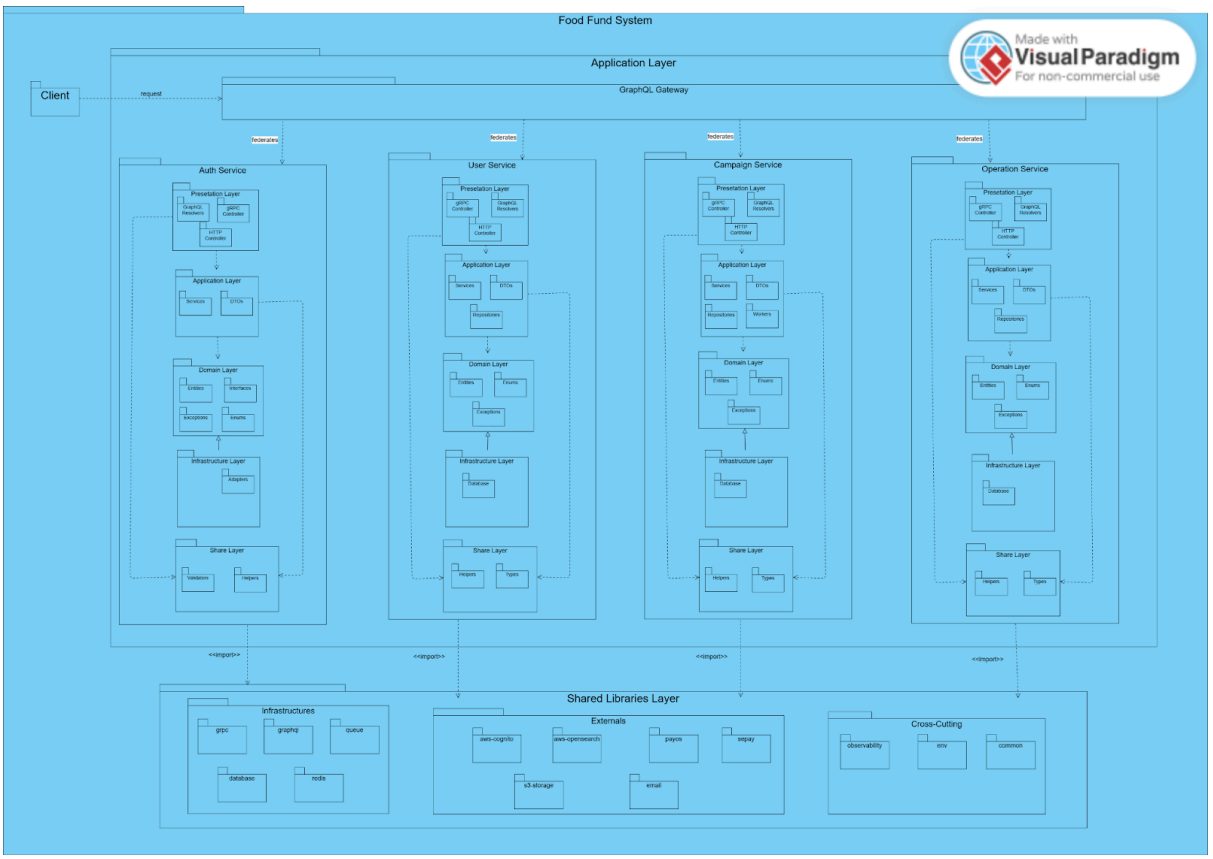


Figure 12. Backend Package Diagram

Package Descriptions

No	Package	Description
01	GraphQL Gateway	API Gateway that provides a unified GraphQL interface for client applications.
02	Auth Service	Authentication and authorization microservice managing user identity and access control.
03	User Service	User management microservice handling user profiles, badges, and wallet operations.
04	Campaign Service	Campaign and donation management microservice for fundraising operations.
05	Operation Service	Operational management microservice for kitchen and delivery operations.

No	Package	Description
06	Presentation Layer	External interface layer exposing APIs and handling client communication through GraphQL Resolvers, gRPC Controllers, and HTTP Controller
07	Application Layer	Business logic layer orchestrating use cases and workflows through Services, Repositories, DTOs, and Background Processors.
08	Domain Layer	Core business domain layer containing pure business entities, rules, and domain logic independent of technical concerns.
09	Infrastructure Layer	Technical implementation layer handling external integrations, data persistence, and third-party service adapters.
10	Share Layer	Service-specific shared utilities, validators, helpers, and cross-cutting concerns within a microservice.
11	Shared Libraries Layer	Reusable libraries provide common functionality across all microservices.
12	Infrastructures	Foundational libraries provide essential infrastructure for communication, data management, and asynchronous processing.
13	Externals	Third-party service integration libraries wrapping external APIs and cloud services.
14	Cross-Cutting	Shared utilities and cross-cutting functionality used across all layers and services.

Table 19. Backend Package Description

1.2.2 Frontend Package Diagram

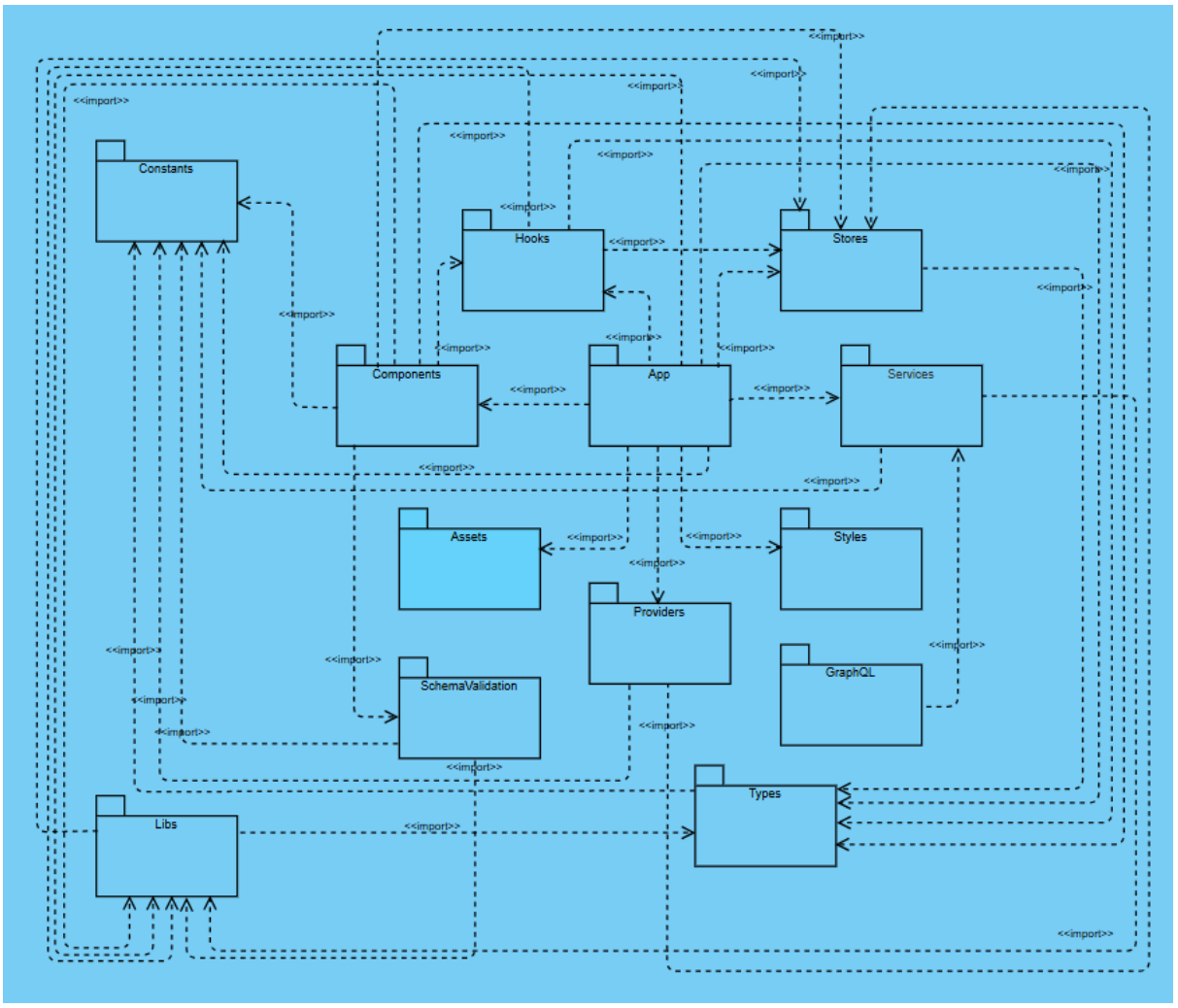


Figure 13. Frontend Package Diagram

Package Descriptions

No	Package	Description
01	App	Core package for application routing, page components, and global layouts.
02	Assets	Stores static resources such as images, fonts.
03	Styles	Manages global styling, design tokens, and theme configuration.
04	Providers	Wraps the application with global context providers for state management, theming, and API connectivity.
05	Hooks	Custom React hooks for encapsulating reusable logic like auth guards, form validation, and animations.
06	Stores	Manages global state using Redux slices.
07	Components	Reusable UI building blocks, organized by feature and shared UI elements.

No	Package	Description
08	Services	Encapsulates business logic and API interaction methods for each domain entity.
09	Constants	Defines static application values, configuration settings, routes, and validation messages.
10	Libs	Core utilities and configurations, including Apollo Client setup, helpers, and third-party integrations.
11	Types	TypeScript definitions and interfaces for API responses and data models.
12	GraphQL	Contains all GraphQL query and mutation definitions used to communicate with the backend.
13	SchemaValidation	Zod schemas for form validation and data integrity checks.

Table 20. Frontend Package Description

1.2.3 Mobile Package Diagram

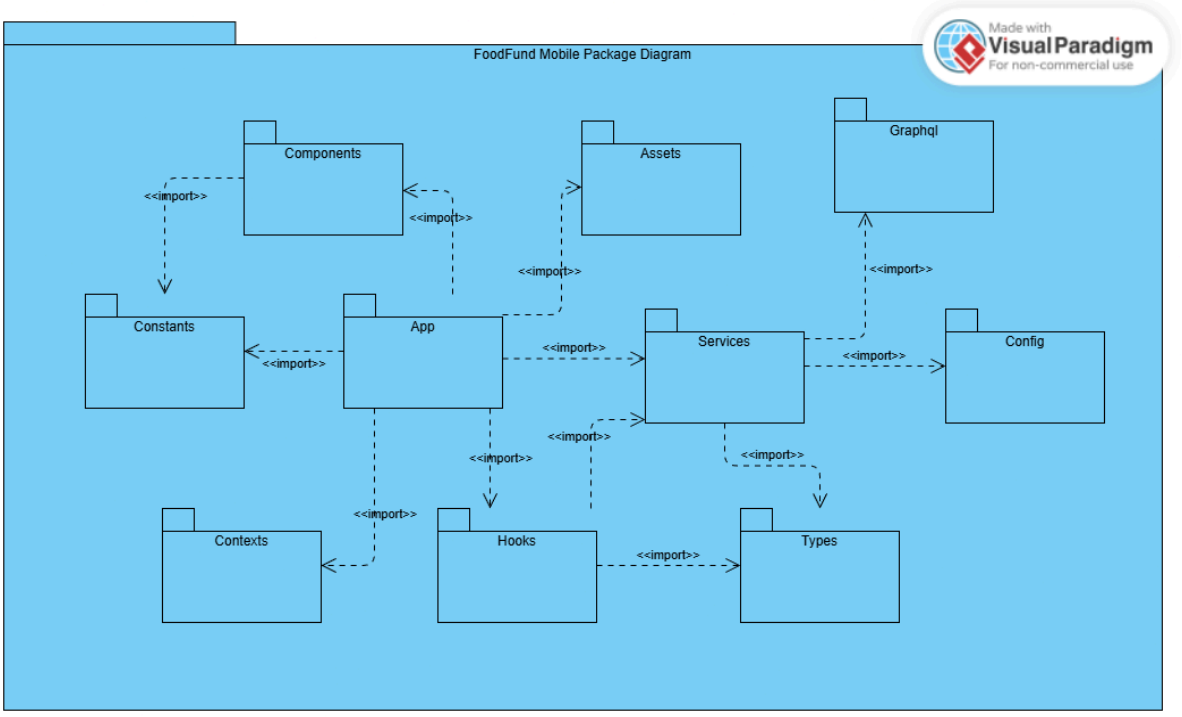


Figure 14. Mobile Package Diagram

Package Descriptions

No	Package	Description
1	App	Core package for application routing, page components, and global layouts. Contains screens for all user roles (Donor, Kitchen Staff, Delivery Staff).
2	Components	Reusable UI components such as buttons, cards, modals, loading indicators, and shared widgets across screens.
3	Assets	Static resources including images, icons, fonts, and other media files used throughout the application.
4	GraphQL	GraphQL queries, mutations, and subscriptions for communication with the backend API.
5	Services	Business logic and API interaction layer. Handles data fetching, authentication, and domain-specific operations.
6	Config	Application configuration including API endpoints, environment variables, and external service settings.
7	Constants	Application-wide constants such as color palettes, theme values, and static configuration values.
8	Contexts	React Context providers for global state management (e.g., PhaseContext for campaign phases).
9	Hooks	Custom React hooks that encapsulate reusable stateful logic and bridge UI components with services.
10	Types	TypeScript type definitions for API responses, component props, and shared data structures.

Table 21. Mobile Package Diagram Description

2. Database Design

2.1 FoodFund Database Design

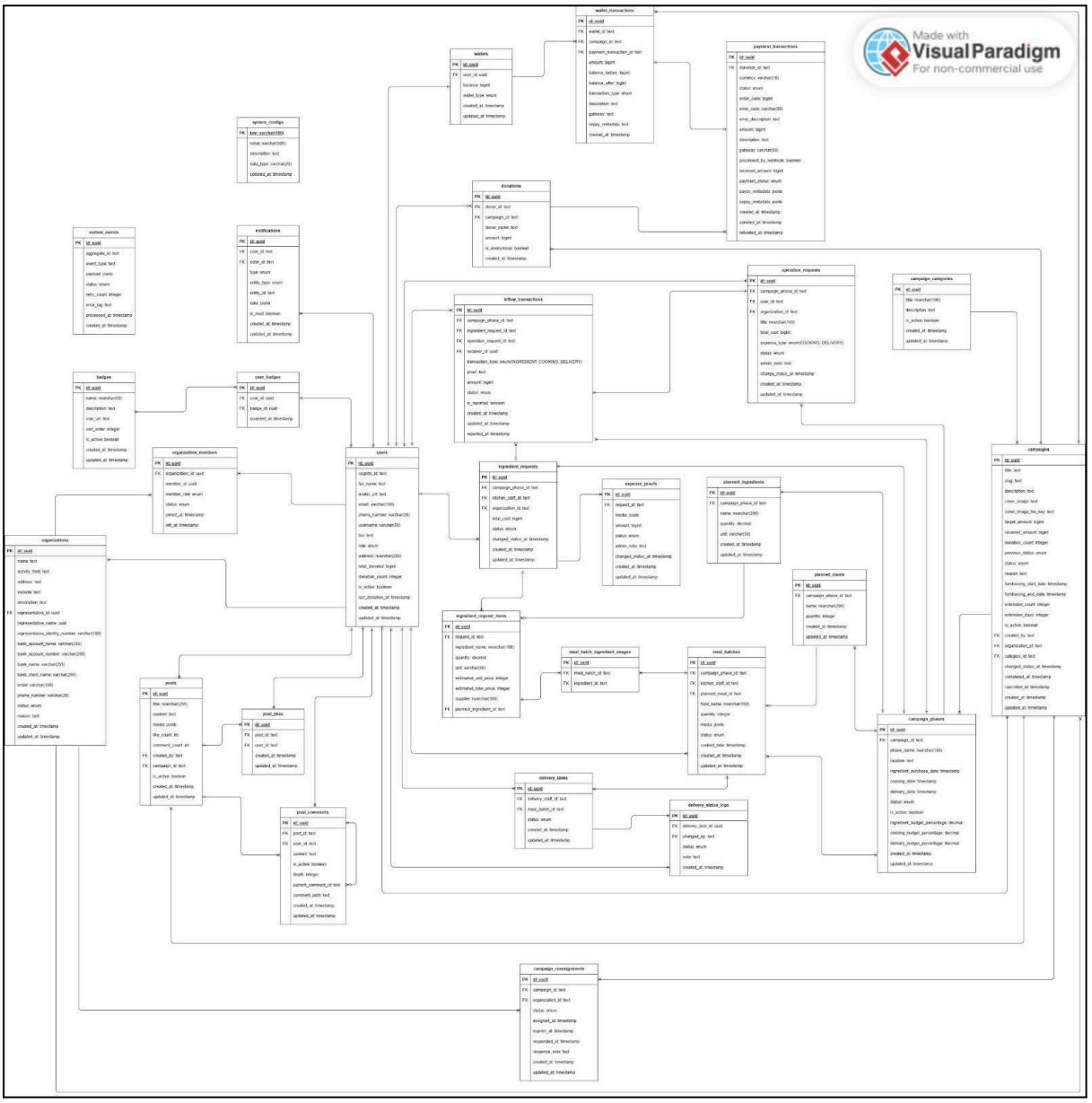


Figure 15. FoodFund Database Design

Attribute Data Dictionary

No	Table Name	Attribute Name	Data Type	Not Null	PK/ FK
1	users	id	uuid	Yes	PK
		cognito_id	varchar	Yes	
		full_name	text	Yes	
		avatar_url	varchar	No	
		email	varchar	Yes	
		phone_number	varchar	No	
		role	enum: DONOR, FUNDRAISER, KITCHEN_STAFF, DELIVERY_STAFF, ADMIN	Yes	
		is_active	boolean	Yes	
		address	text	No	
		user_name	varchar	Yes	
		bio	text	No	
		total_donated	bigint	No	
		donation_count	integer	No	
		last_donation_at	timestamp	No	
		created_at	timestamp	Yes	
		updated_at	timestamp	Yes	
2	organizations	id	uuid	Yes	PK
		name	text	Yes	
		activity_field	text	Yes	
		address	text	Yes	
		website	text	Yes	
		description	text	Yes	
		representative_id	uuid	Yes	FK
		representative_name	text	Yes	
		representative_identity_number	text	Yes	
		bank_account_name	varchar	No	
		bank_account_number	varchar	No	
		bank_name	varchar	No	
		bank_short_name	varchar	No	
		email	varchar	Yes	
		phone_number	varchar	Yes	
		status	enum: PENDING, VERIFIED, REJECTED, CANCELLED, INACTIVE	Yes	
		reason	text	No	
		created_at	timestamp	Yes	
		updated_at	timestamp	Yes	
3	organization_members	id	uuid	Yes	PK
		organization_id	uuid	Yes	FK

No	Table Name	Attribute Name	Data Type	Not Null	PK/ FK
		member_id	uuid	Yes	FK
		status	enum: PENDING VERIFIED REJECTED CANCELLED INACTIVE	Yes	
		member_role	enum: KITCHEN_STAFF, DELIVERY_STAFF, FUNDRAISER	Yes	
		joined_at	timestamp	Yes	
		left_at	timestamp	No	
4	badges	id	uuid	Yes	PK
		name	text	Yes	
		description	text	Yes	
		icon_url	text	Yes	
		sort_order	integer	Yes	
		is_active	boolean	Yes	
		created_at	timestamp	Yes	
		updated_at	timestamp	Yes	
5	user_badges	id	uuid	Yes	PK
		user_id	uuid	Yes	FK
		badge_id	uuid	Yes	FK
		awarded_at	timestamp	Yes	
6	wallets	id	uuid	Yes	PK
		user_id	uuid	Yes	FK
		wallet_type	enum: FUNDRAISER, ADMIN	Yes	
		balance	bigint	Yes	
		created_at	timestamp	Yes	
		updated_at	timestamp	Yes	
7	wallet_transactions	id	uuid	Yes	PK
		wallet_id	uuid	Yes	FK
		campaign_id	text	No	
		payment_transaction_id	text	No	
		amount	bigint	Yes	
		balance_before	bigint	Yes	
		balance_after	bigint	Yes	
		transaction_type	Enum: INCOMING_TRANSFER, WITHDRAWAL, ADMIN_ADJUSTMENT	Yes	
		description	text	No	
		gateway	varchar	Yes	
		sepay_metadata	jsonb	No	
		created_at	timestamp	Yes	
8	campaigns	id	uuid	Yes	PK
		title	text	Yes	
		slug	text	Yes	

No	Table Name	Attribute Name	Data Type	Not Null	PK/ FK
		description	text	Yes	
		cover_image	text	Yes	
		target_amount	bigint	Yes	
		received_amount	bigint	Yes	
		status	enum: PENDING, APPROVED, REJECTED, ACTIVE, PROCESSING, COMPLETED, CANCELLED, ENDED	Yes	
		previous_status	enum: PENDING, APPROVED, REJECTED, ACTIVE, PROCESSING, COMPLETED, CANCELLED, ENDED	No	
		is_active	boolean	Yes	
		created_by	text	Yes	
		organization_id	text	Yes	
		created_at	timestamp	Yes	
		updated_at	timestamp	Yes	
		cover_image_file_key	text	No	
		category_id	uuid	Yes	FK
		completed_at	timestamp	No	
		fundraising_end_date	timestamp	Yes	
		fundraising_start_date	timestamp	Yes	
		changed_status_at	timestamp	No	
		extension_count	integer	Yes	
		extension_days	integer	Yes	
		donation_count	integer	Yes	
		reason	text	No	
		cancelled_at	timestamp	No	
9	campaign_phases	id	uuid	Yes	PK
		campaign_id	uuid	Yes	FK
		phase_name	varchar	Yes	
		location	text	Yes	
		ingredient_purchase_date	timestamp	Yes	
		cooking_date	timestamp	Yes	
		delivery_date	timestamp	Yes	
		status	enum: PLANNING, AWAITING_INGREDIENT_DISBURSEMENT, INGREDIENT_PURCHASE,	Yes	

No	Table Name	Attribute Name	Data Type	Not Null	PK/ FK
			AWAITING_AUDIT, AWAITING_COOKING_DISBURSEMENT, COOKING, AWAITING_DELIVERY_DISBURSEMENT, DELIVERY, COMPLETED, CANCELLED, FAILED		
		created_at	timestamp	Yes	
		updated_at	timestamp	Yes	
		is_active	boolean	Yes	
		ingredient_budget_percentage	decimal	Yes	
		cooking_budget_percentage	decimal	Yes	
		delivery_budget_percentage	decimal	Yes	
10	campaign_reassignments	id	uuid	Yes	PK
		campaign_id	uuid	Yes	FK
		organization_id	text	Yes	
		status	enum: PENDING, APPROVED, REJECTED, EXPIRED, CANCELLED	Yes	
		assigned_at	timestamp	Yes	
		expires_at	timestamp	Yes	
		responded_at	timestamp	No	
		response_note	text	No	
		created_at	timestamp	Yes	
		updated_at	timestamp	Yes	
11	campaign_categories	id	uuid	Yes	PK
		title	varchar	Yes	
		description	text	Yes	
		is_active	boolean	Yes	
		created_at	timestamp	Yes	
		updated_at	timestamp	Yes	
12	donations	id	uuid	Yes	PK
		donor_id	text	Yes	
		campaign_id	uuid	Yes	FK
		amount	bigint	Yes	
		is_anonymous	boolean	Yes	
		created_at	timestamp	Yes	
		donor_name	text	Yes	
13	payment_transactions	id	uuid	Yes	PK
		donation_id	uuid	Yes	FK
		currency	varchar	Yes	
		status	enum: SUCCESS,	Yes	

No	Table Name	Attribute Name	Data Type	Not Null	PK/ FK
			FAILED, REFUNDED, PENDING		
		created_at	timestamp	Yes	
		updated_at	timestamp	Yes	
		refunded_at	timestamp	No	
		error_code	varchar	No	
		error_description	text	No	
		order_code	bigint	Yes	
		amount	bigint	Yes	
		description	text	Yes	
		gateway	varchar	Yes	
		processed_by_webhook	boolean	Yes	
		received_amount	bigint	Yes	
		payment_status	enum: PENDING, PARTIAL, COMPLETED, OVERPAID	Yes	
		payos_metadata	jsonb	No	
		sepay_metadata	jsonb	No	
14	posts	id	uuid	Yes	PK
		campaign_id	uuid	Yes	FK
		created_by	text	Yes	
		title	varchar	Yes	
		content	text	Yes	
		media	jsonb	No	
		like_count	integer	Yes	
		comment_count	integer	Yes	
		is_active	boolean	Yes	
		created_at	timestamp	Yes	
		updated_at	timestamp	Yes	
15	post_likes	id	uuid	Yes	PK
		post_id	uuid	Yes	FK
		user_id	text	Yes	
		created_at	timestamp	Yes	
		updated_at	timestamp	Yes	
16	post_comments	id	uuid	Yes	PK
		post_id	uuid	Yes	FK
		user_id	text	Yes	
		content	text	Yes	
		parent_comment_id	uuid	Yes	FK
		comment_path	text	No	
		depth	integer	Yes	
		created_at	timestamp	Yes	
		updated_at	timestamp	Yes	
		is_active	boolean	Yes	
17	notifications	id	text	Yes	PK

No	Table Name	Attribute Name	Data Type	Not Null	PK/ FK
		user_id	text	Yes	
		actor_id	text	No	
		type	enum: CAMPAIGN_APPROVED, CAMPAIGN_REJECTED, CAMPAIGN_COMPLETED, CAMPAIGN_CANCELLED, CAMPAIGN_DONATION_RECEIVED, CAMPAIGN_NEW_POST POST_LIKE, POST_COMMENT, POST_REPLY, INGREDIENT_REQUEST_APPROVED, DELIVERY_TASK_ASSIGNED SYSTEM_ANNOUNCEMENT	Yes	
		entity_type	enum: CAMPAIGN, POST, COMMENT, DONATION, INGREDIENT_REQUEST, DELIVERY_TASK	Yes	
		entity_id	text	No	
		data	jsonb	No	
		is_read	boolean	Yes	
		created_at	timestamptz	Yes	PK
		updated_at	timestamptz	Yes	
18	outbox_events	id	uuid	Yes	PK
		aggregate_id	text	Yes	
		event_type	text	Yes	
		payload	jsonb	Yes	
		status	enum: PENDING, PROCESSING, COMPLETED, FAILED	Yes	
		retry_count	integer	Yes	
		created_at	timestamp	Yes	
		processed_at	timestamp	No	
		error_log	text	No	
19	ingredient_requests	id	uuid	Yes	PK
		campaign_phase_id	text	Yes	
		kitchen_staff_id	text	Yes	
		organization_id	text	Yes	
		total_cost	bigint	Yes	
		status	enum: PENDING, APPROVED, REJECTED,	Yes	

No	Table Name	Attribute Name	Data Type	Not Null	PK/ FK
			DISBURSED		
		changed_status_at	timestamp	No	
		created_at	timestamp	Yes	
		updated_at	timestamp	Yes	
20	ingredient_request_items	id	uuid	Yes	PK
		request_id	uuid	Yes	FK
		ingredient_name	varchar	Yes	
		quantity	varchar	Yes	
		estimated_unit_price	integer	Yes	
		estimated_total_price	integer	Yes	
		supplier	varchar	No	
21	expense_proofs	id	uuid	Yes	PK
		request_id	uuid	Yes	FK
		media	jsonb	Yes	
		amount	bigint	Yes	
		status	enum: PENDING, APPROVED, REJECTED	Yes	
		admin_note	text	No	
		changed_status_at	timestamp	No	
		created_at	timestamp	Yes	
		updated_at	timestamp	Yes	
22	operation_requests	id	uuid	Yes	PK
		campaign_phase_id	text	Yes	
		user_id	text	Yes	
		organization_id	text	Yes	
		title	varchar	Yes	
		total_cost	bigint	Yes	
		expense_type	enum: COOKING, DELIVERY	Yes	
		status	enum: PENDING, APPROVED, REJECTED, DISBURSED	Yes	
		admin_note	text	No	
		changed_status_at	timestamp	No	
		created_at	timestamp	Yes	
		updated_at	timestamp	Yes	
23	meal_batches	id	uuid	Yes	PK
		campaign_phase_id	text	Yes	
		kitchen_staff_id	text	Yes	
		food_name	varchar	Yes	
		quantity	integer	Yes	
		media	jsonb	No	
		status	enum: PREPARING, READY, DELIVERED	Yes	

No	Table Name	Attribute Name	Data Type	Not Null	PK/ FK
		cooked_date	timestamp	Yes	
		created_at	timestamp	Yes	
		updated_at	timestamp	Yes	
24	meal_batch_ingredient_ usages	id	uuid	Yes	PK
		meal_batch_id	uuid	Yes	FK
		ingredient_id	uuid	Yes	FK
25	delivery_tasks	id	uuid	Yes	PK
		delivery_staff_id	text	Yes	
		meal_batch_id	uuid	Yes	FK
		status	enum: PENDING, ACCEPTED, REJECTED, OUT_FOR_DELIVERY, COMPLETED, FAILED	Yes	
		created_at	timestamp	Yes	
		updated_at	timestamp	Yes	
26	delivery_status_logs	id	uuid	Yes	PK
		delivery_task_id	uuid	Yes	FK
		status	enum: PENDING, ACCEPTED, REJECTED, OUT_FOR_DELIVERY, COMPLETED, FAILED	Yes	
		changed_by	text	Yes	
		note	text	No	
		created_at	timestamp	Yes	
27	inflow_transactions	id	uuid	Yes	PK
		campaign_phase_id	text	Yes	
		ingredient_request_id	text	No	FK
		operation_request_id	text	No	FK
		receiver_id	text	Yes	
		transaction_type	text	Yes	
		proof	text	Yes	
		amount	bigint	Yes	
		status	enum	Yes	
		is_reported	boolean	Yes	
		created_at	timestamp	Yes	
		updated_at	timestamp	Yes	
		reported_at	timestamp	Yes	
28	planned_meals	id	uuid	Yes	PK
		campaign_phase_id	text	Yes	FK
		name	varchar(200)	Yes	
		quantity	integer	Yes	
		created_at	timestamp	Yes	
		updated_at	timestamp	Yes	

No	Table Name	Attribute Name	Data Type	Not Null	PK/ FK
29	planned_ingredients	id	uuid	Yes	PK
		campaign_phase_id	text	Yes	FK
		name	nvarchar(100)	Yes	
		quantity	decimal	Yes	
		unit	varchar(50)	Yes	
		created_at	timestamp	Yes	
		updated_at	timestamp	Yes	
30	system_configs	key	varchar(100)	Yes	PK
		value	varchar(500)	Yes	
		description	text	Yes	
		data_type	varchar(20)	Yes	
		updated_at	timestamp	Yes	

Table 22 Attribute Data Dictionary

2.2 Database Design for User Service

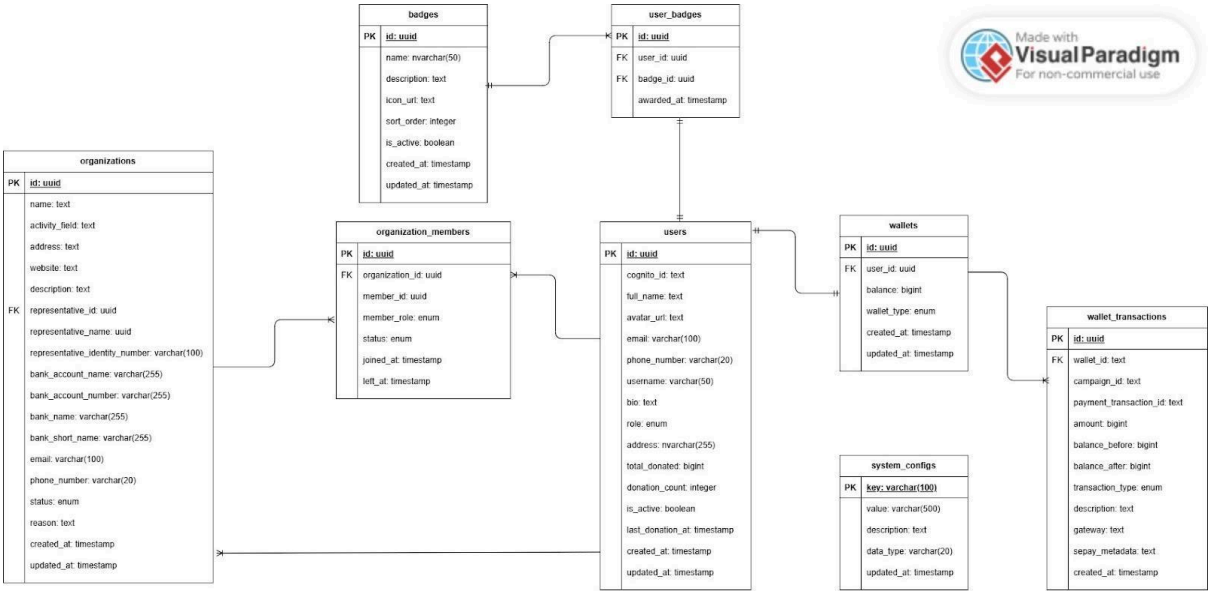


Figure 16. Database Design for User Service

Table Descriptions

No	Table	Description
01	users	Represents the central entity for all individuals interacting with the system, including donors, fundraisers, kitchen staffs, delivery staffs and administrators. This table stores authentication references (Cognito), personal profiles, and aggregated donation statistics.
02	organizations	Represents a registered legal entity or charity group managed by a fundraiser. This table contains verification details, banking information, and the legal representative's credentials required for conducting fundraising campaigns.
03	organization_members	Represents the association between a user and an organization. This junction table defines the specific role

No	Table	Description
		(e.g., Kitchen Staff, Delivery Staff) and the verification status of a member within a specific organization context.
04	badges	Defines the achievement elements available in the system. This table stores the metadata for achievements, including the badge name, icon, and display order.
05	user_badges	Represents the specific association of a badge awarded to a user. This table tracks which user possesses which badge and the timestamp of when the award was granted.
06	wallets	Stores the virtual financial balance associated with a specific user (typically a Fundraiser or Admin). This entity is used to manage internal funds, such as surplus from campaigns or operational budgets.
07	wallet_transactions	Represents an immutable audit log of all financial movements within a wallet. This table records incoming transfers, withdrawals, and adjustments, ensuring full traceability of fund allocation.
08	system_configs	Represents system configuration settings that can be dynamically modified by administrators without code deployment.

Table 23. Database Design for User Service Table Descriptions

2.3 Database Design for Campaign Service

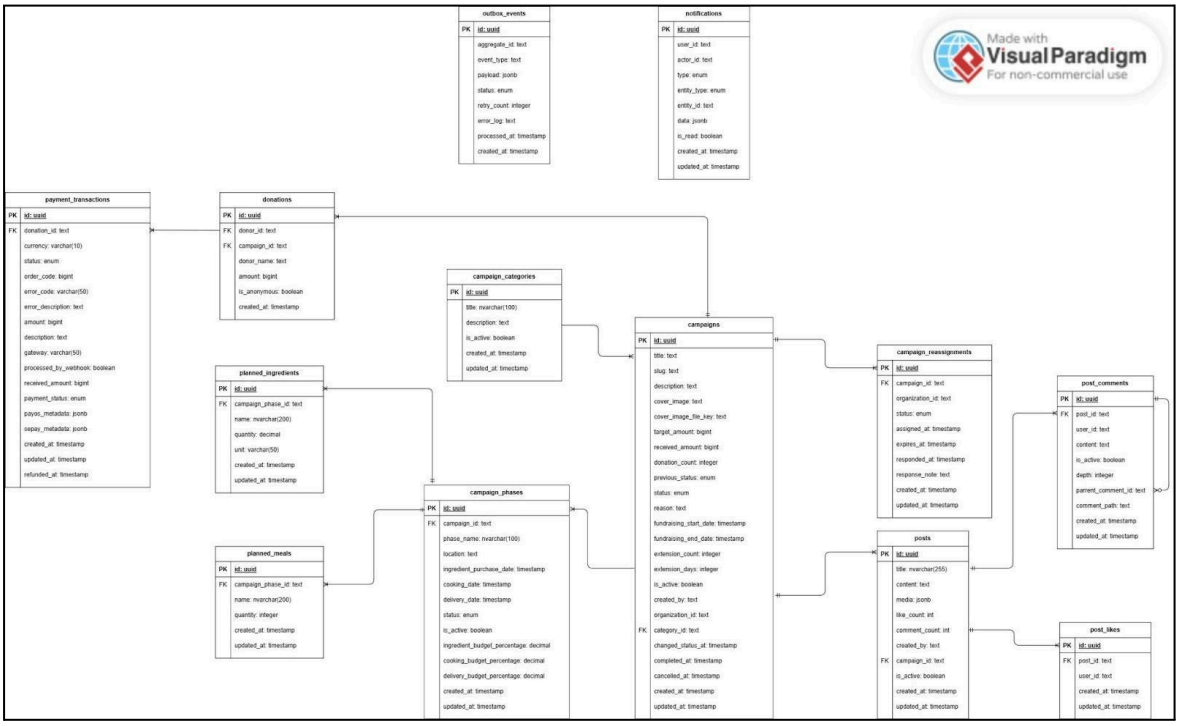


Figure 17. Database Design for Campaign Service

Table Descriptions

No	Table	Description
01	campaigns	Represents the core entity of the system, storing all essential information about a fundraising initiative. This includes the financial targets, current status, timelines, associated organization, and fundraising metrics.
02	campaign_phases	Defines the operational lifecycle of a campaign by breaking it down into distinct stages (e.g., Ingredient Purchase, Cooking, Delivery). This table tracks the timeline, status, and budget allocation for each phase.
03	campaign_reassignments	Manages the administrative process of transferring campaign ownership from one organization to another. This is used when a campaign is cancelled or the original fundraiser cannot fulfill their duties.
04	campaign_categories	Stores the classification data for campaigns (e.g., "Emergency Relief", "School Meals").
05	donations	Records individual financial contributions made by donors to specific campaigns. This table acts as the bridge between the user (donor) and the campaign.
06	payment_transactions	Store detailed audit trail of financial processing associated with a donation. It captures payment gateway responses, transaction statuses (success/fail), and reconciliation data.
07	posts	Represents updates, announcements, or "proof of work" content created by fundraisers within a campaign. This facilitates transparency and donor engagement.
08	post_likes	Tracks user engagement by recording which users have "liked" specific campaign posts.
09	post_comments	Supports a hierarchical discussion system allowing users to comment on posts and reply to other comments (nested threads).
10	notifications	Stores system-generated alerts triggered by various events (e.g., donation received, post liked) to inform users of relevant activities or required actions.
11	outbox_events	Stores events for the Transactional Outbox Pattern, ensuring reliable, asynchronous execution of side effects.
12	planned_meals	Represents a structured record of specific food items or full meals intended to be prepared and distributed during a designated campaign phase. This entity facilitates the planning of operational outputs by defining the target item names and their respective quantities.
13	planned_ingredients	Represents the detailed inventory of raw materials and supplies required to execute a specific campaign phase. This entity captures essential procurement data, including the ingredient name, precise quantity, and unit of measurement, ensuring accurate budgeting and resource allocation.

Table 24. Database Design for Campaign Service Table Descriptions

2.4 Database Design for Operation Service

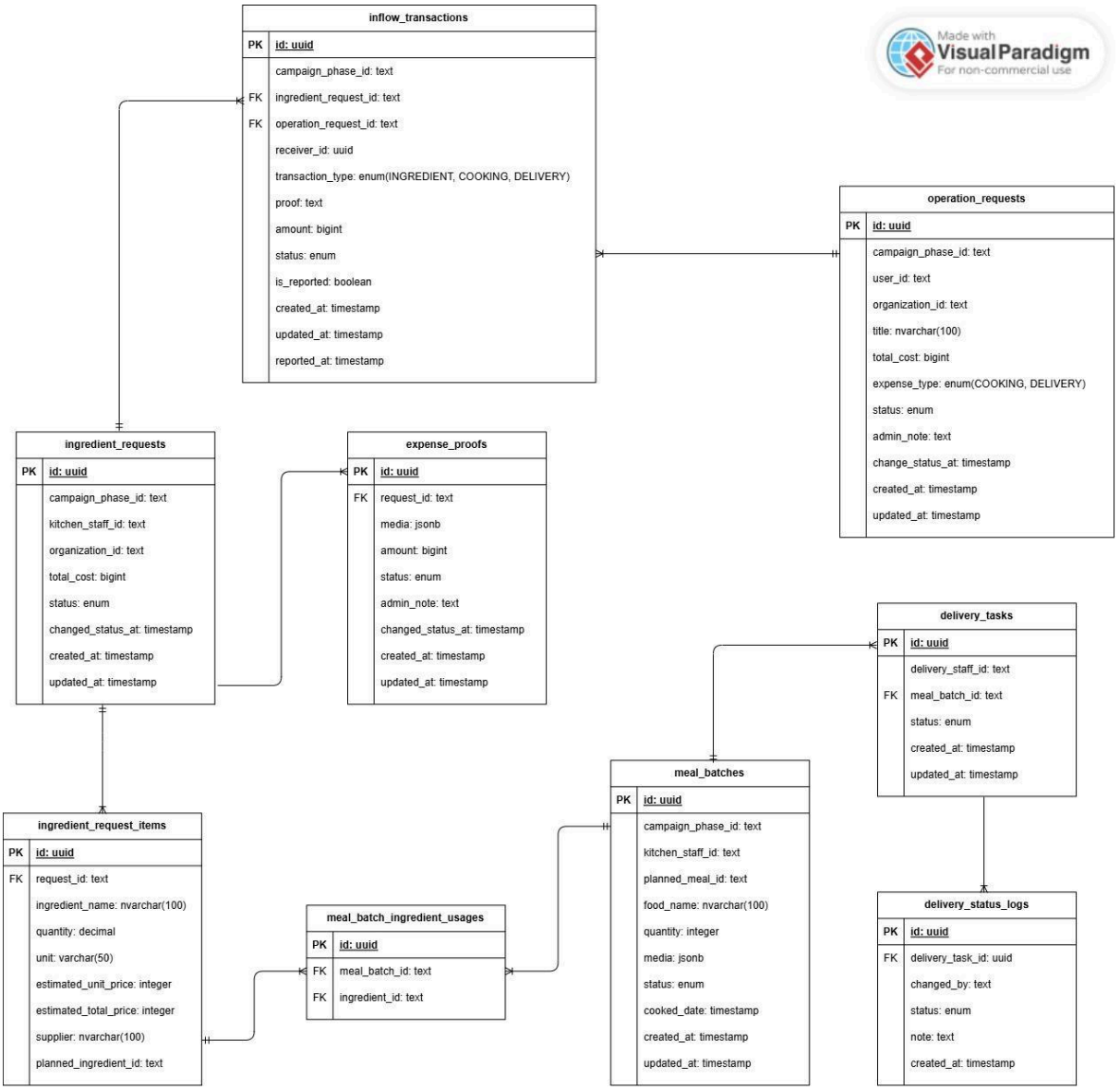


Figure 18. Database Design for Operation Service

Table Descriptions

No	Table	Description
01	ingredient_requests	Represents a formal requisition created to procure necessary ingredients for a specific campaign phase. This entity tracks the total estimated cost, approval status, and timestamps for auditing purposes.
02	ingredient_request_items	Details the specific line items within an ingredient request, including the ingredient name, quantity, estimated pricing, and potential suppliers.
03	expense_proofs	Stores evidentiary documentation (e.g., images of receipts, invoices) uploaded by staff. This entity is used to validate the actual expenditure against the approved ingredient request before or after disbursement.

No	Table	Description
04	operation_requests	Represents financial requests for non-ingredient operational needs, such as utility costs (cooking fuel) or logistics expenses (delivery gasoline). It links a specific user and expense type to a campaign phase.
05	meal_batches	Represents a specific production unit of food prepared by kitchen staff. This table tracks the output quantity, visual proof of the cooked food (media), and the current status of the batch (e.g., Ready, Delivered).
06	meal_batch_ingredient_usages	Acts as a junction table creating a many-to-many relationship between meal batches and ingredient items. It serves as an audit trail to track exactly which purchased ingredients were consumed to produce a specific batch of meals.
07	delivery_tasks	Represents an assignment given to a delivery staff member to transport a specific meal batch. It tracks the current status of the delivery lifecycle from assignment to completion.
08	delivery_status_logs	Maintains an immutable historical record of all status transitions for a delivery task. This ensures full accountability by recording who changed the status, when it happened, and any associated notes.
09	inflow_transactions	Records the internal disbursement of funds from FoodFund to fundraisers. It links financial transfers to specific ingredient or operational requests and tracks the transaction status and proof of transfer.

Table 25. Database Design for Operation Service Table Descriptions

3. Detailed Design

3.1 Authentication

3.1.1 Class Diagram

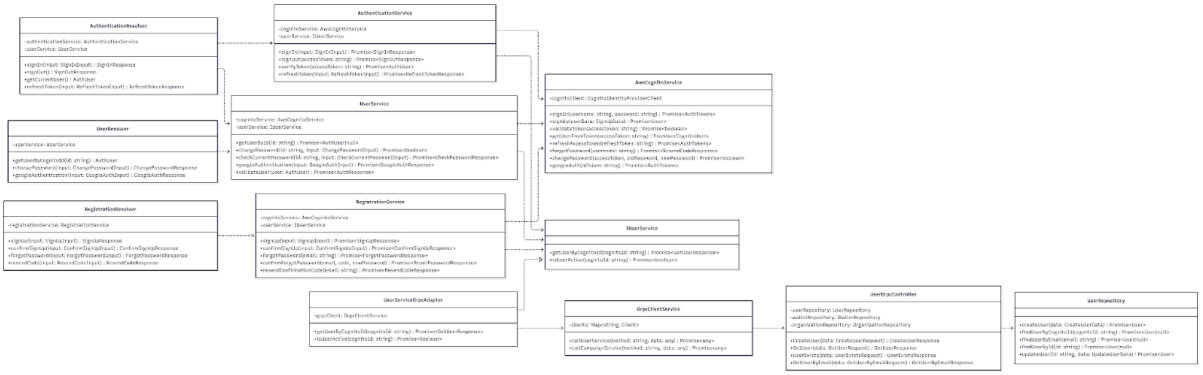


Figure 19. Authentication Class Diagram

3.1.2 Class Specification

AuthenticationResolver

No	Method	Description
1	**signIn(input)**	Authenticate user with email and password.
2	**signOut()**	Sign out current user globally.
3	**getCurrentUser()**	Get current authenticated user info.
4	**verifyToken()**	Verify and validate access token.
5	**refreshToken(input)**	Refresh expired access token.

Table 26. AuthenticationResolver Class Specification

RegistrationResolver

No	Method	Description
1	**signUp(input)**	Register new user with email verification.
2	**confirmSignUp(input)**	Confirm email with verification code
3	**resendConfirmationCode(input)**	Resend email verification code.
4	**forgotPassword(input)**	Initiate password reset flow.
5	**confirmForgotPassword(input)**	Reset password with confirmation code.

Table 27. Registration Class Specification

UserResolver

No	Method	Description
1	**getUserByCognitoId(cognitoId)**	Get user details by Cognito ID.
2	**changePassword(input)**	Change user password.
3	**checkCurrentPassword(input)**	Verify current password.
4	**googleAuthentication(input)**	Sign in or sign up with Google OAuth.

Table 28. UserResolver Class Specification

3.1.3 Sign In Sequence Diagram

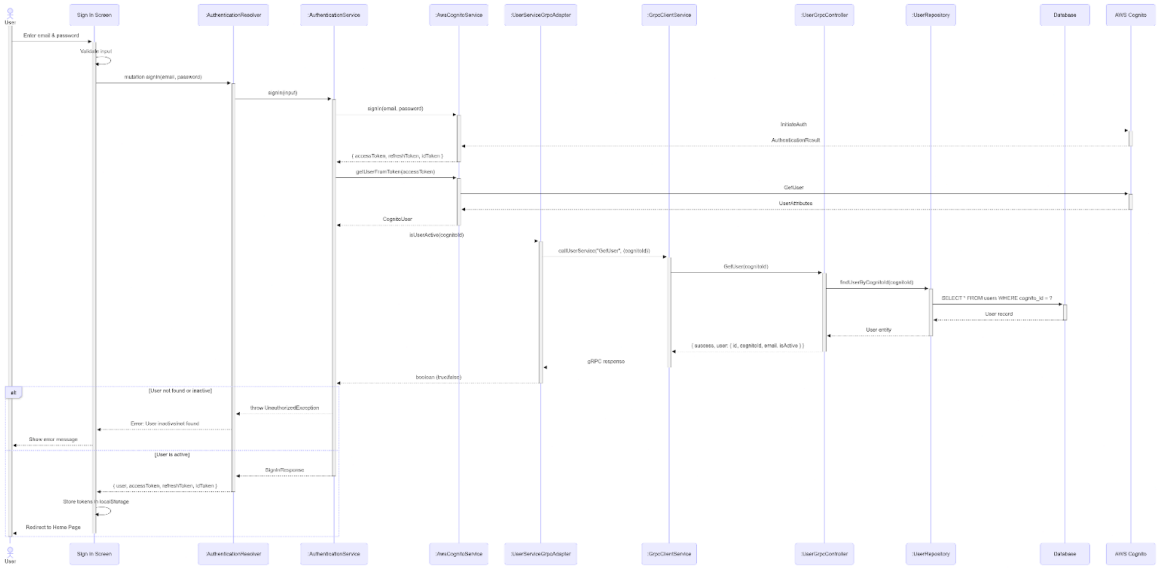


Figure 20. Sign In Sequence Diagram

3.1.4 Sign Up Sequence Diagram

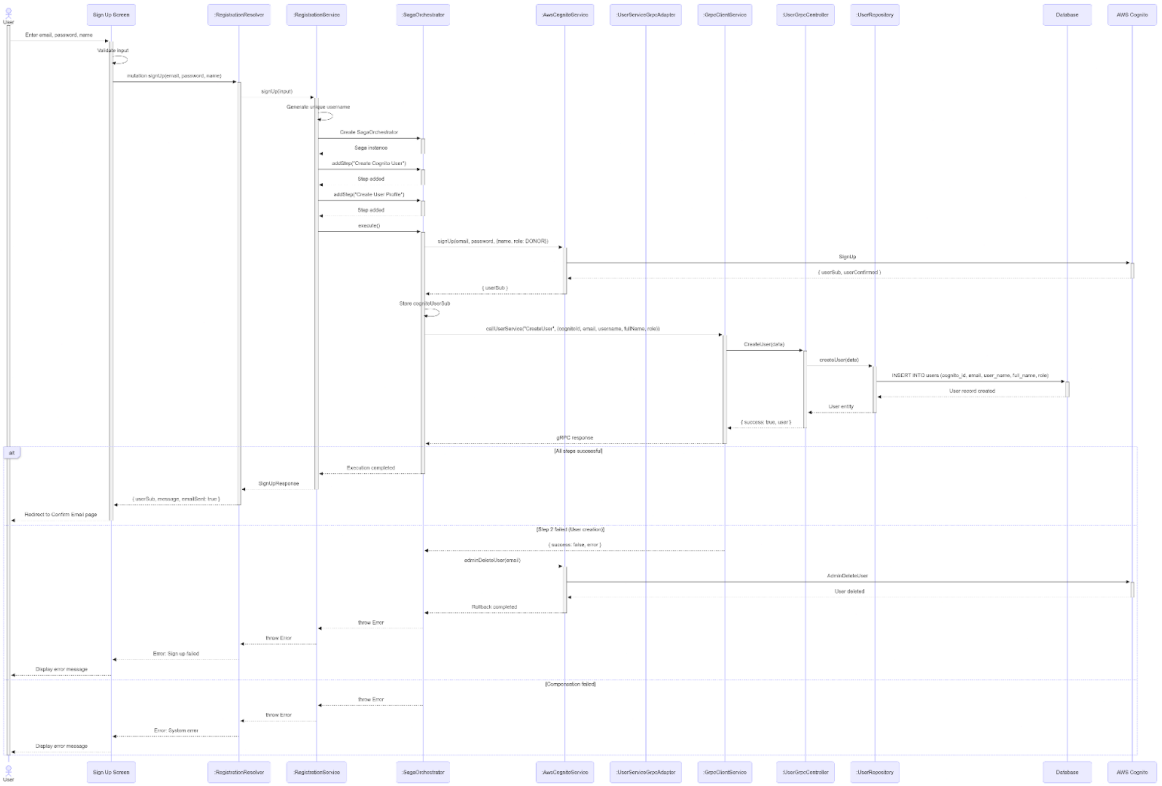


Figure 21. Sign Up Sequence Diagram

3.1.5 Sign Out Sequence Diagram

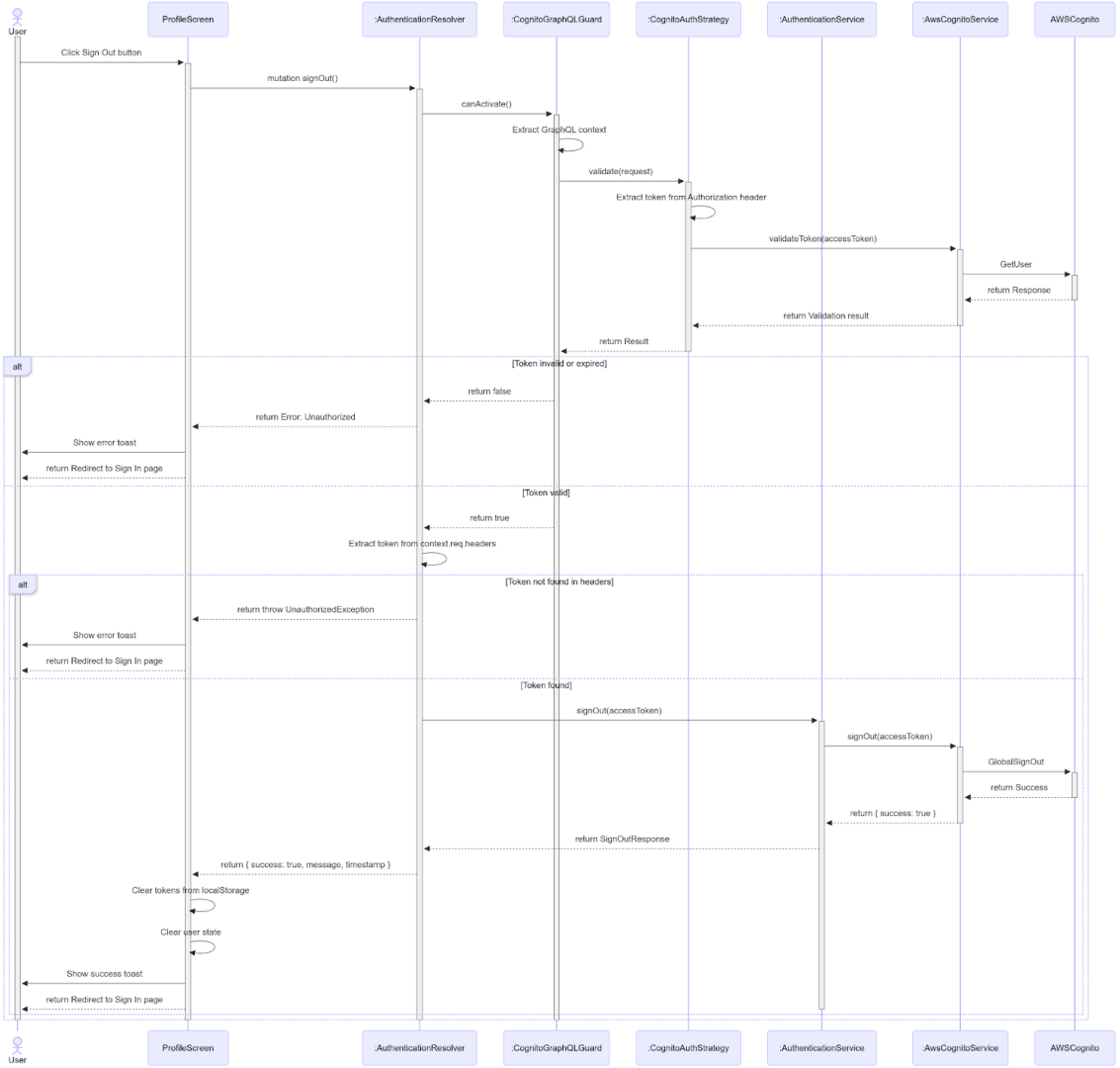


Figure 22. Sign Up Sequence Diagram

3.2 Campaign Category Management

3.2.1 Class Diagram

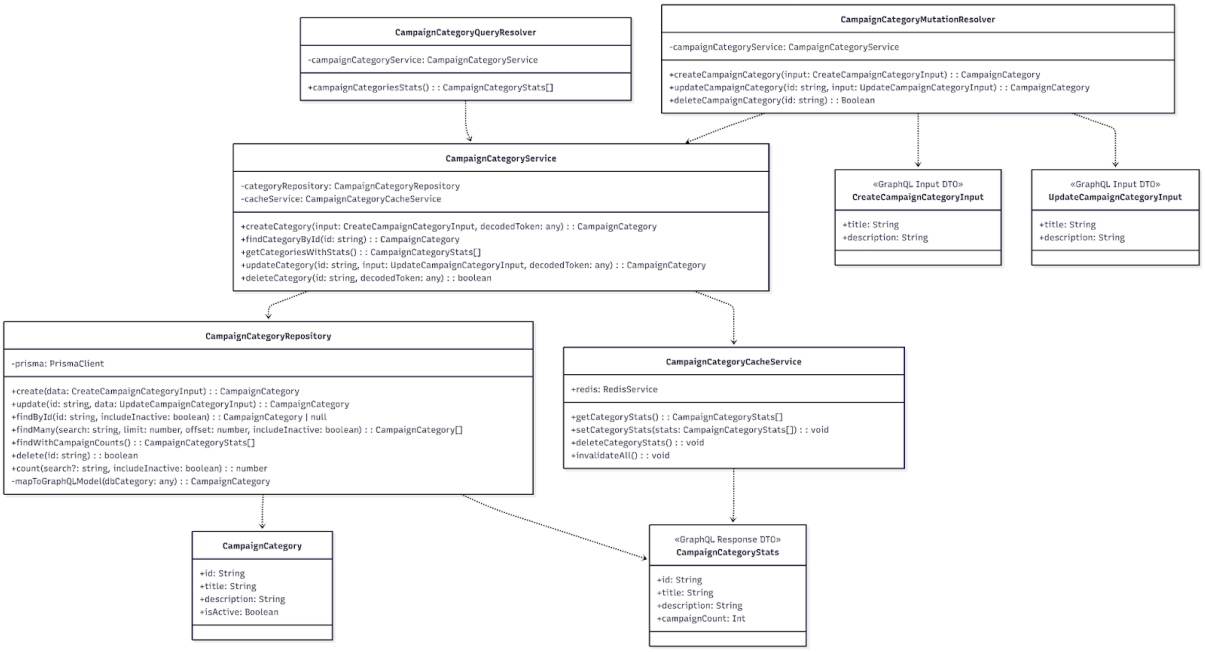


Figure 23. Campaign Category Class Diagram

3.2.2 Class Specifications

CampaignCategoryQueryResolver

No	Method	Description
1	**campaignCategoriesStats()**	Handles the GraphQL query to retrieve a list of campaign categories enriched with statistical data (specifically the count of active campaigns per category). It delegates the business logic to the CampaignCategoryService.

Table 29. CampaignCategoryQueryResolver Class Specification

CampaignCategoryMutationResolver

No	Method	Description
1	**createCampaignCategory(input)**	Handles the GraphQL mutation to create a new campaign category. It accepts CreateCampaignCategoryInput containing the title and description, then delegates execution to the service layer.
2	**updateCampaignCategory(id, input)**	Handles the GraphQL mutation to modify an existing campaign category identified by id. It passes the update payload (UpdateCampaignCategoryInput) to the service layer.
3	**deleteCampaignCategory(id)**	Handles the GraphQL mutation to soft-delete a campaign category. It delegates the request to the service layer to perform validation and status updates.

Table 30. CampaignCategoryMutationResolver Class Specification

3.2.3 View Campaign Categories Sequence Diagram

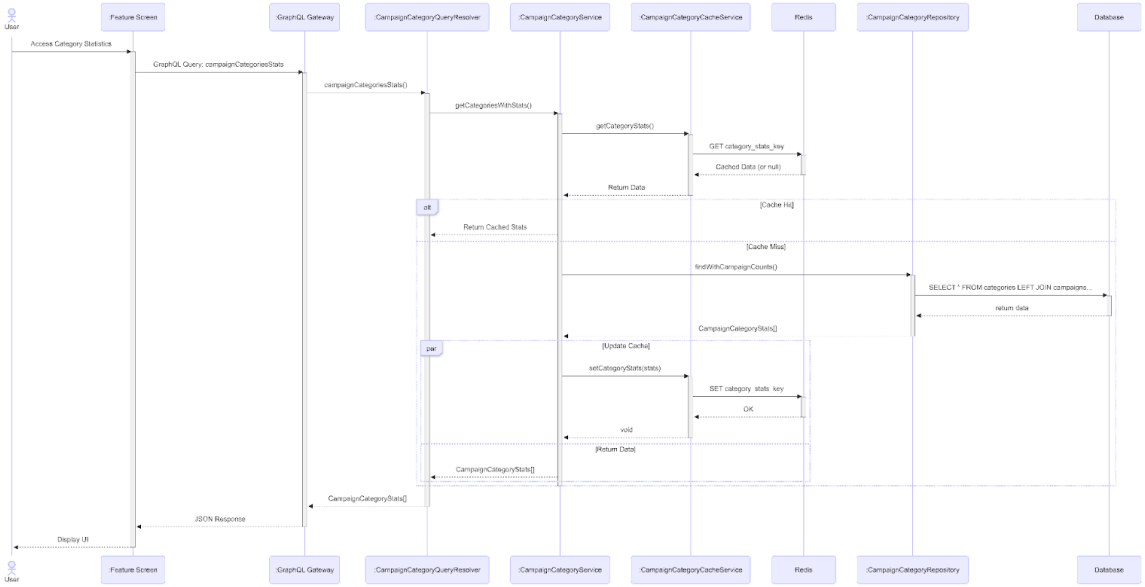


Figure 24. View Campaign Categories Sequence Diagram

3.2.4 Create Campaign Category Sequence Diagram

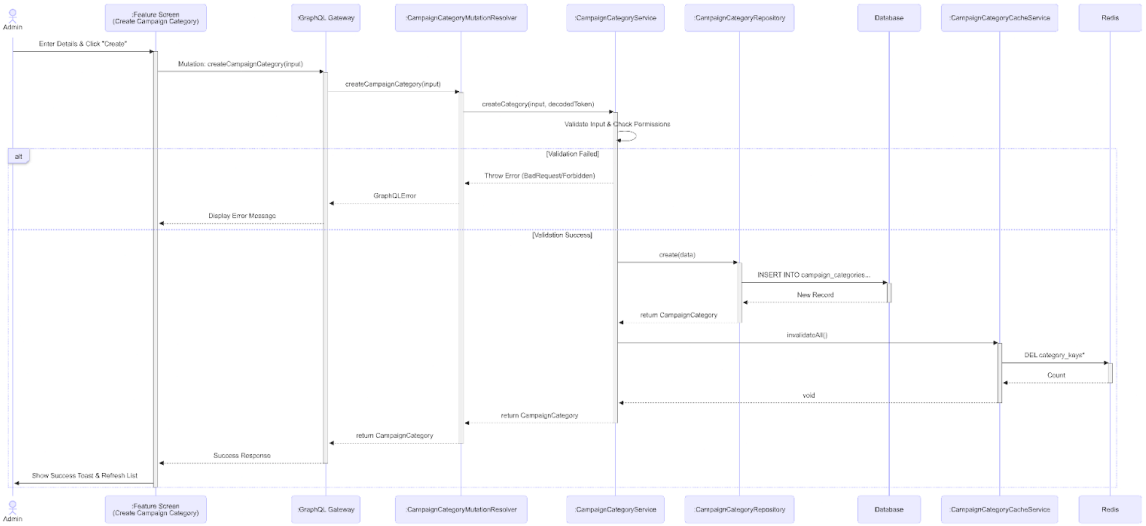


Figure 25. Create Campaign Category Sequence Diagram

3.2.5 Update Campaign Category Sequence Diagram

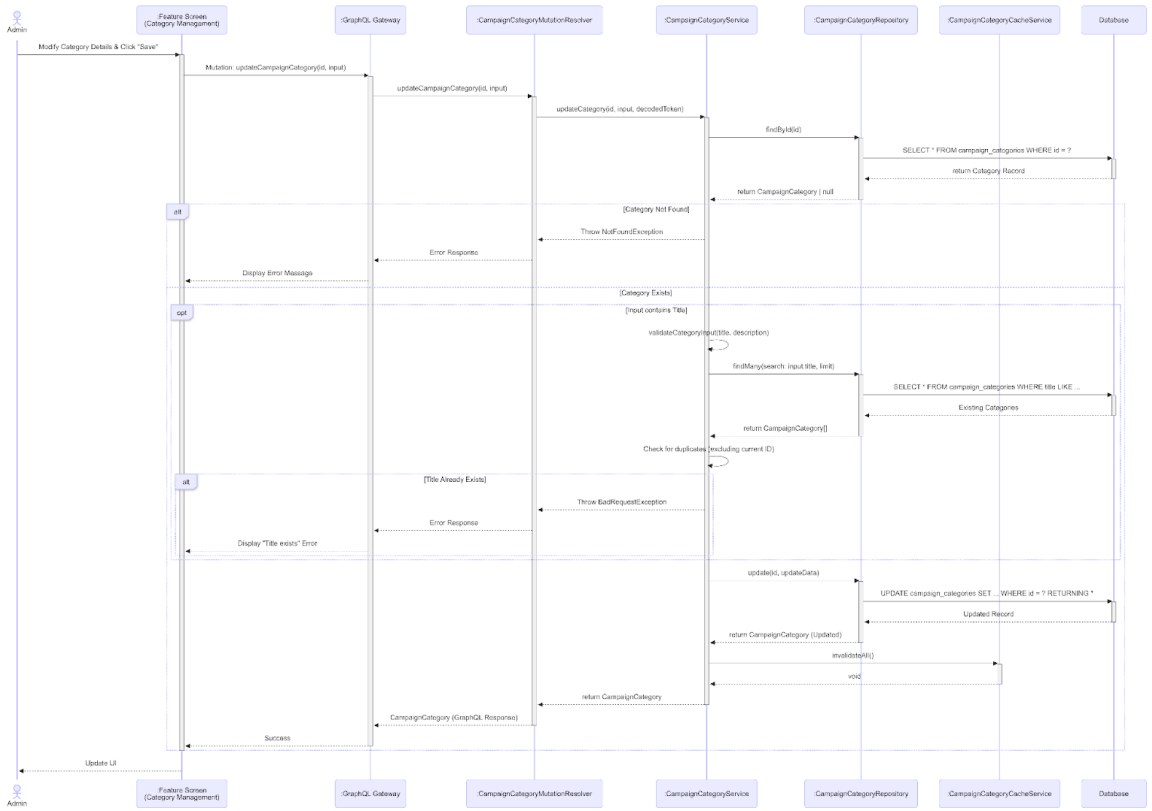


Figure 26. Update Campaign Category Sequence Diagram

3.2.6 Delete Campaign Category Sequence Diagram

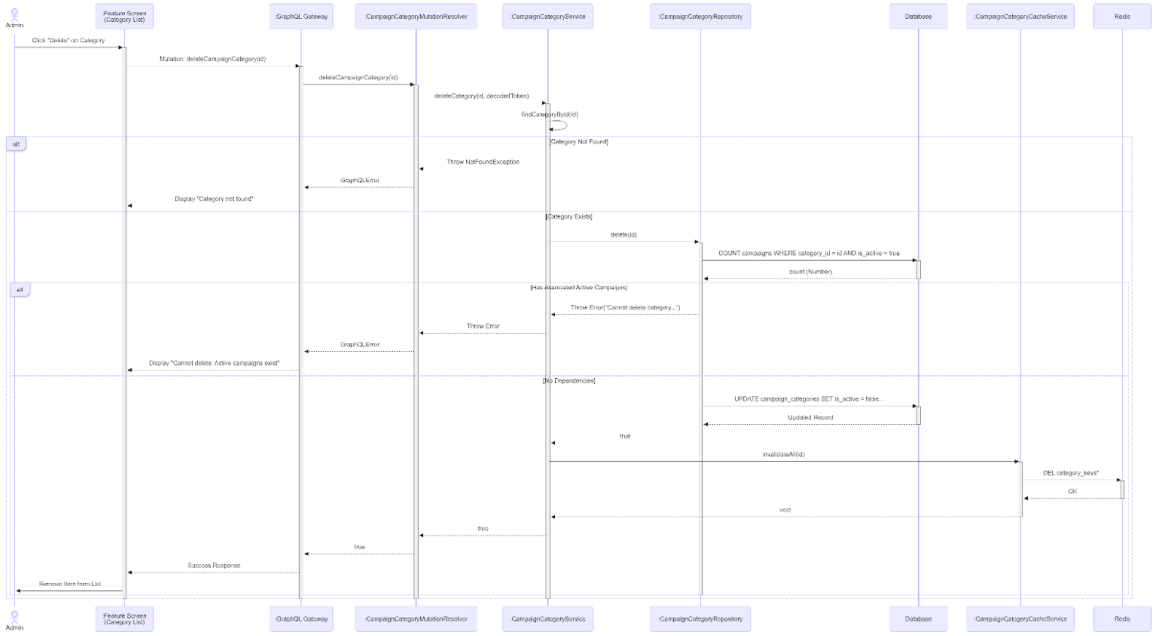


Figure 27. Delete Campaign Category Sequence Diagram

3.3 User Management

3.3.1 Class Diagram

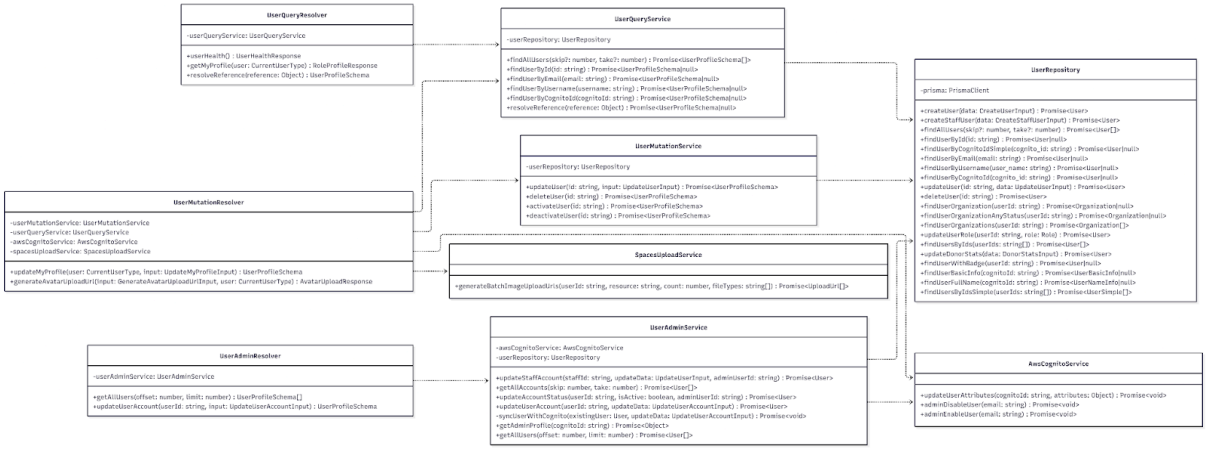


Figure 28. User Management Class Diagram

3.3.2 Class Specification

UserQueryResolver

No	Method	Description
1	**getMyProfile()**	Get authenticated user's profile with role-based information

Table 31. UserQueryResolver Class Specification

UserMutationResolver

No	Method	Description
1	**updateMyProfile(input)**	Update user profile

Table 32. UserMutationResolver Class Specification

UserAdminResolver

No	Method	Description
1	**getAllUsers(offset, limit)**	Get all users with pagination
2	**updateUserAccount(userId, input)**	Update user account information

Table 33. UserAdminResolver Class Specification

3.3.3 Get My Profile Sequence Diagram

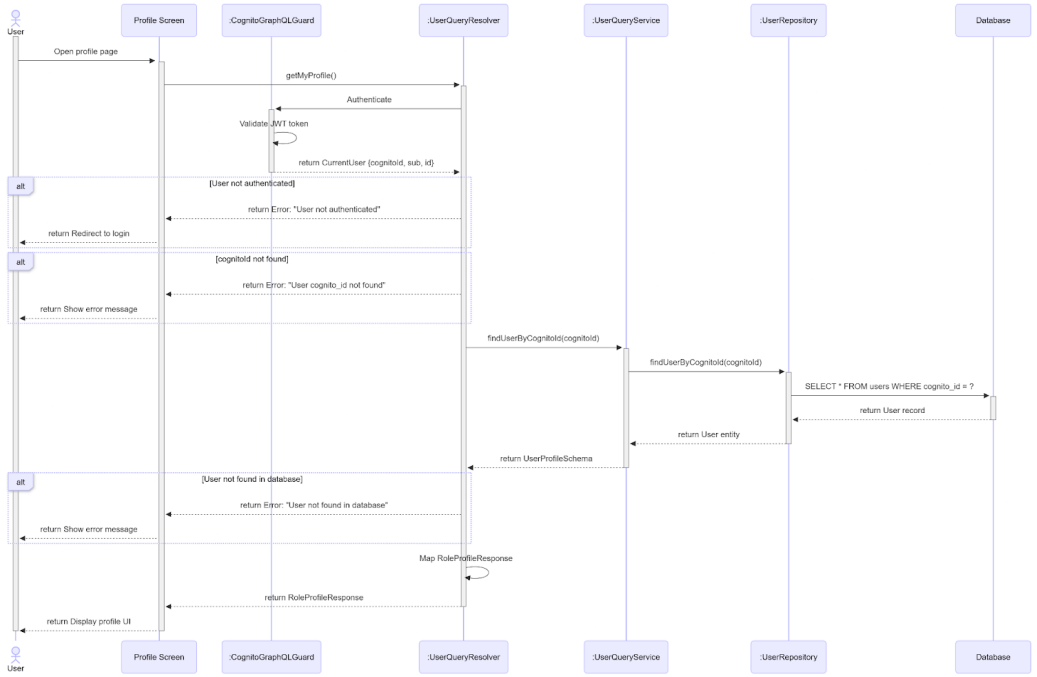


Figure 29. Get My Profile Sequence Diagram

3.3.4 Update Profile Sequence Diagram

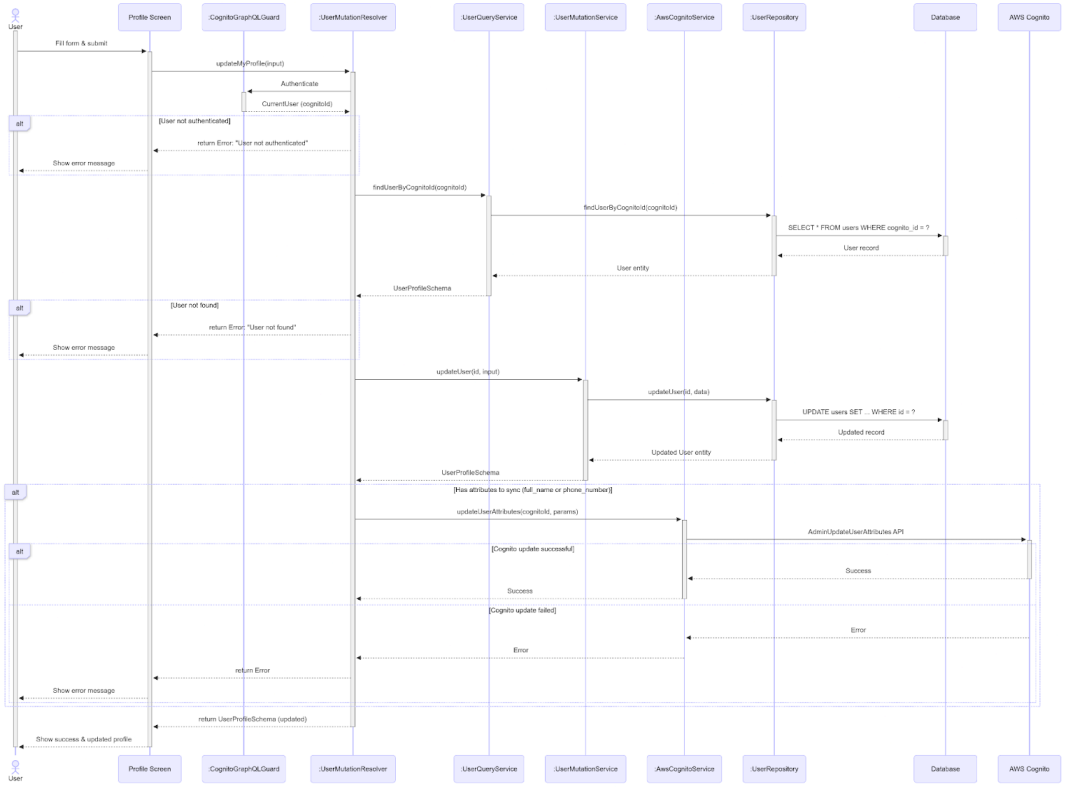


Figure 30. Update Profile Sequence Diagram

3.3.5 Get All Users Sequence Diagram

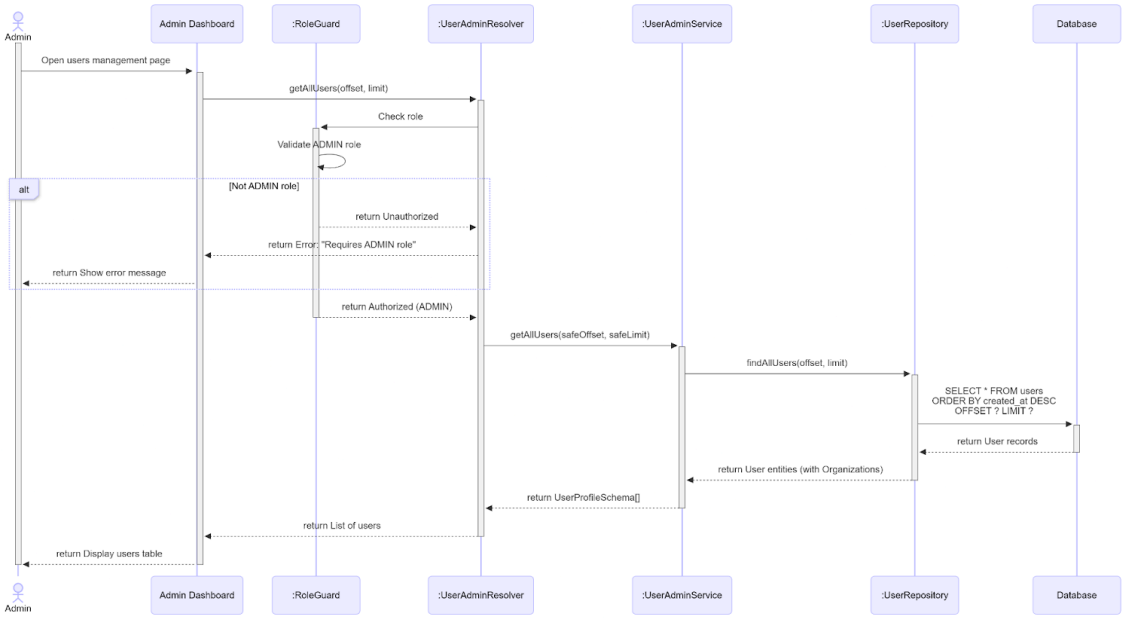


Figure 31. Get All Users Sequence Diagram

3.3.6 Update User Account Sequence Diagram

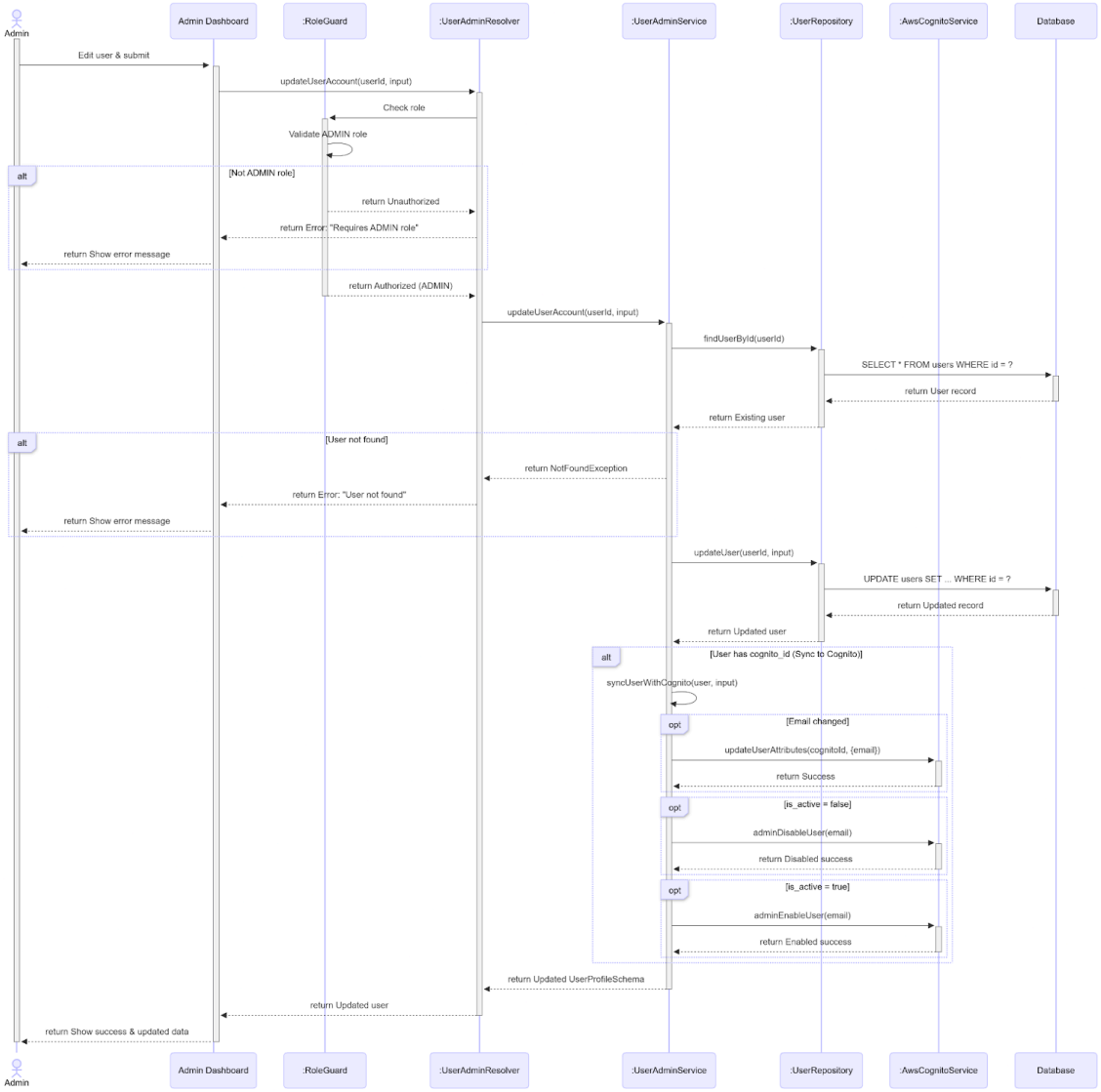


Figure 32. Update User Account Sequence Diagram

3.4 Organization Management

3.4.1 Class Diagram

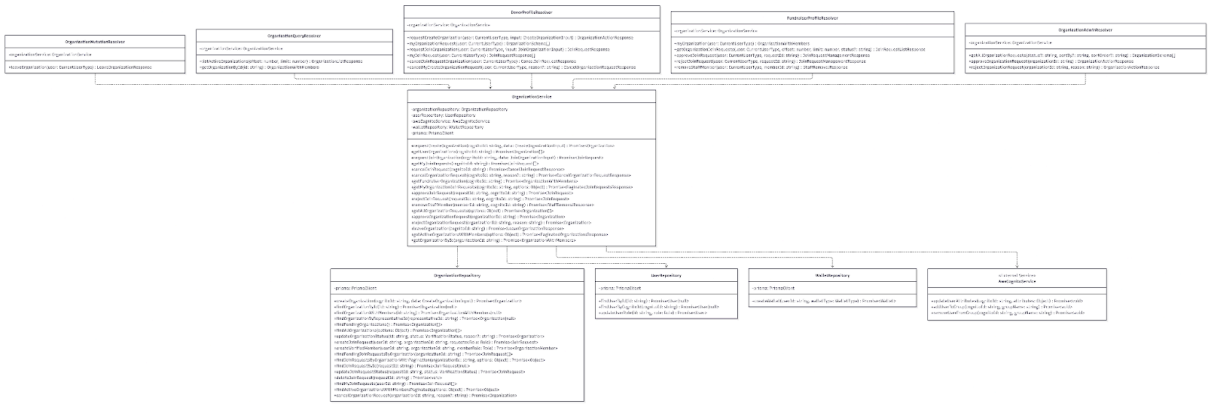


Figure 33. Organization Management Class Diagram

3.4.2 Class Specification

DonorProfileResolver

No	Method	Description
1	**requestCreateOrganization(input)**	Submit organization creation request.
2	**myOrganizationRequest()**	View own organization requests.
3	**cancelMyCreateOrganizationRequest(reason)**	Cancel pending request.
4	**requestJoinOrganization(input)**	Submit a join request (choose KITCHEN_STAFF or DELIVERY_STAFF).
5	**myJoinRequest()**	View own join requests.
6	**cancelJoinRequestOrganization()**	Cancel pending join request.

Table 34. DonorProfileResolver Class Specification

OrganizationAdminResolver

No	Method	Description
1	**getAllOrganizationRequests(status, sortBy, sortOrder)**	View all requests with filtering and sorting.
2	**approveOrganizationRequest(organizationId)**	Approve Donor request
3	**rejectOrganizationRequest(organizationId, reason)**	Reject request with reason.

Table 35. OrganizationAdminResolver Class Specification

FundraiserProfileResolver

No	Method	Description
1	**getOrganizationJoinRequests(offset, limit, status)**	View join requests for own organization
2	**approveJoinRequest(requestId)**	Approve join request
3	**rejectJoinRequest(requestId)**	Reject join request
4	**removeStaffMember(memberId)**	Force remove staff from organization
5	**myOrganization()**	View own organization with all members

Table 36. FundraiserProfileResolver Class Specification

OrganizationMutationResolver

No	Method	Description
1	**leaveOrganization()**	Voluntary leave

Table 37. OrganizationMutationResolver Class Specification

OrganizationQueryResolver

No	Method	Description
1	**listActiveOrganizations(offset, limit)**	Get list of active organization with pagination
2	**getOrganizationById(id)**	Get organization information with specific ID

Table 38. OrganizationQueryResolver Class Specification

3.4.3 Request Create Organization Sequence Diagram

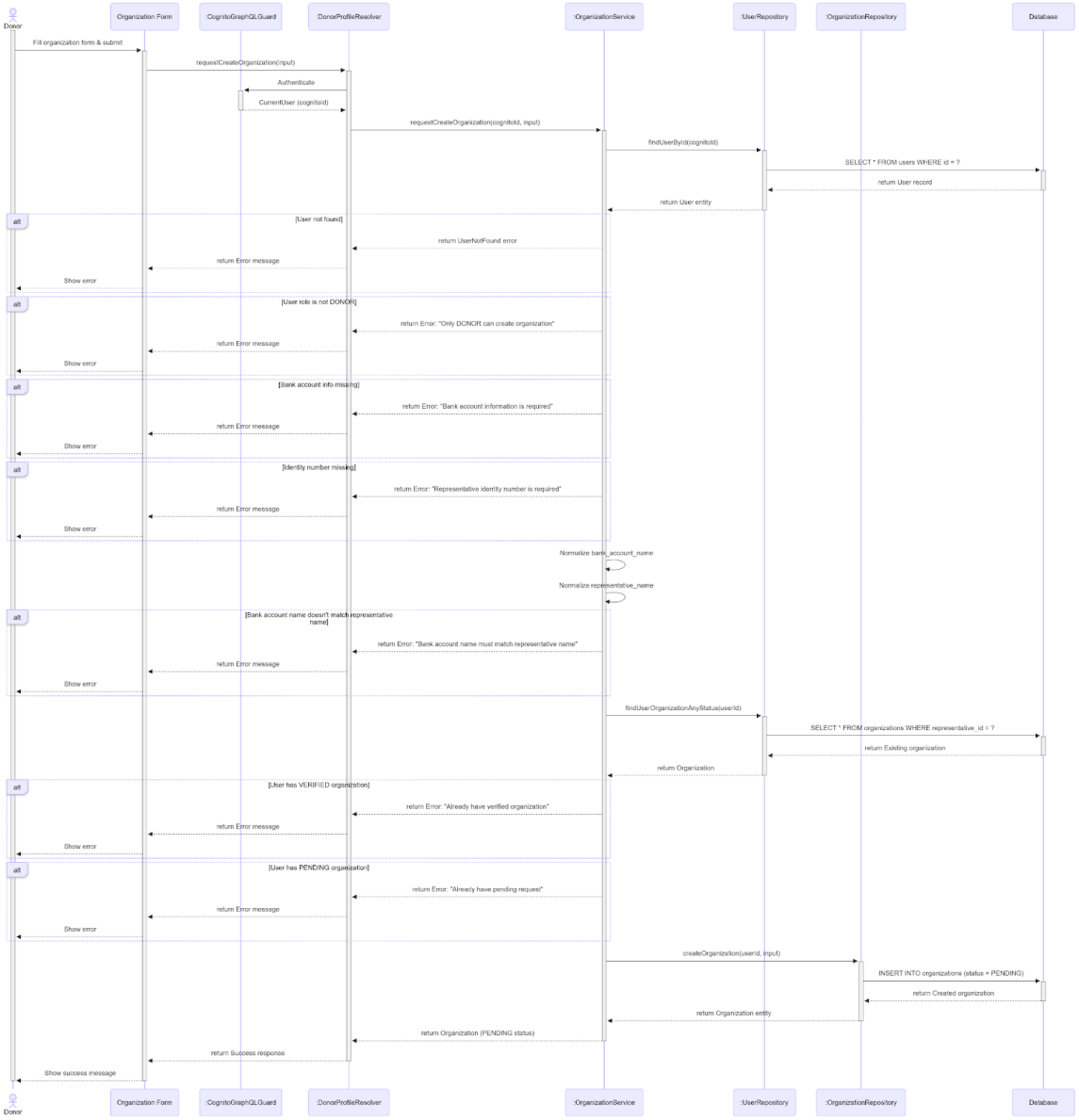


Figure 34. Request Create Organization Sequence Diagram

3.4.4 Approve Request Create Organization Sequence Diagram

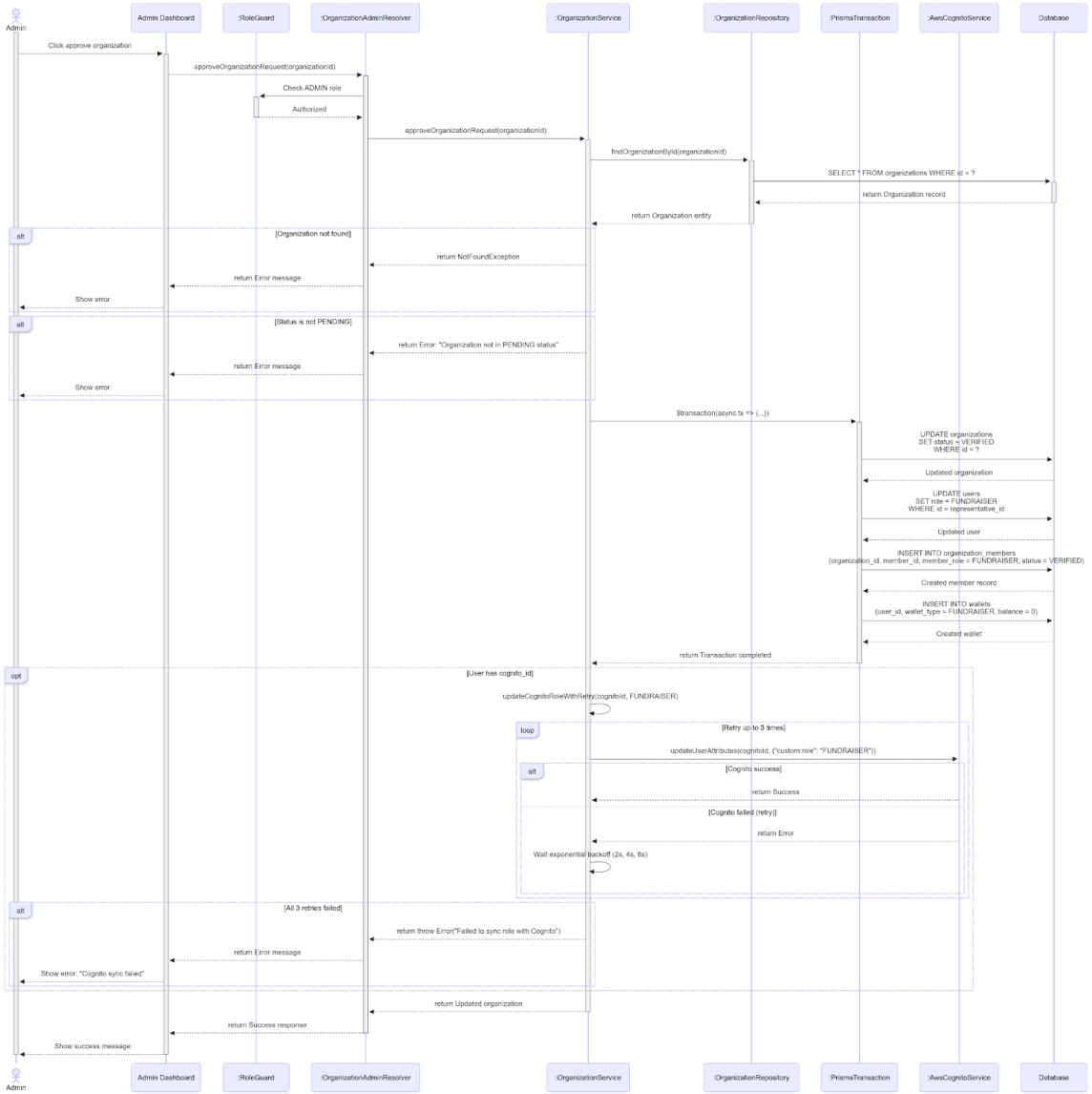


Figure 35. Approve Request Create Organization Sequence Diagram

3.4.5 Approve Join Organization Request

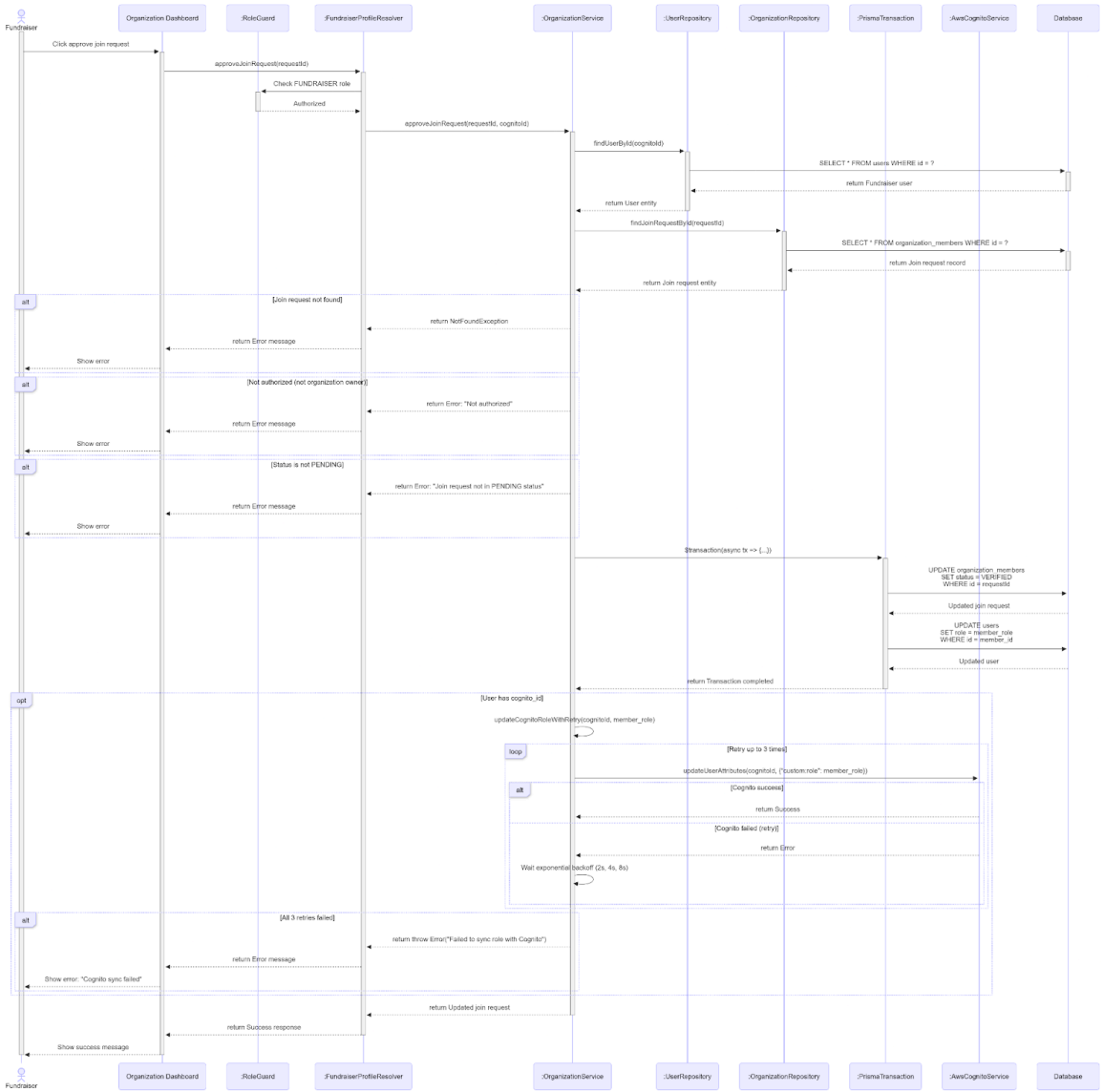


Figure 36. Approve Join Organization Request

3.4.6 Leave Organization Sequence Diagram

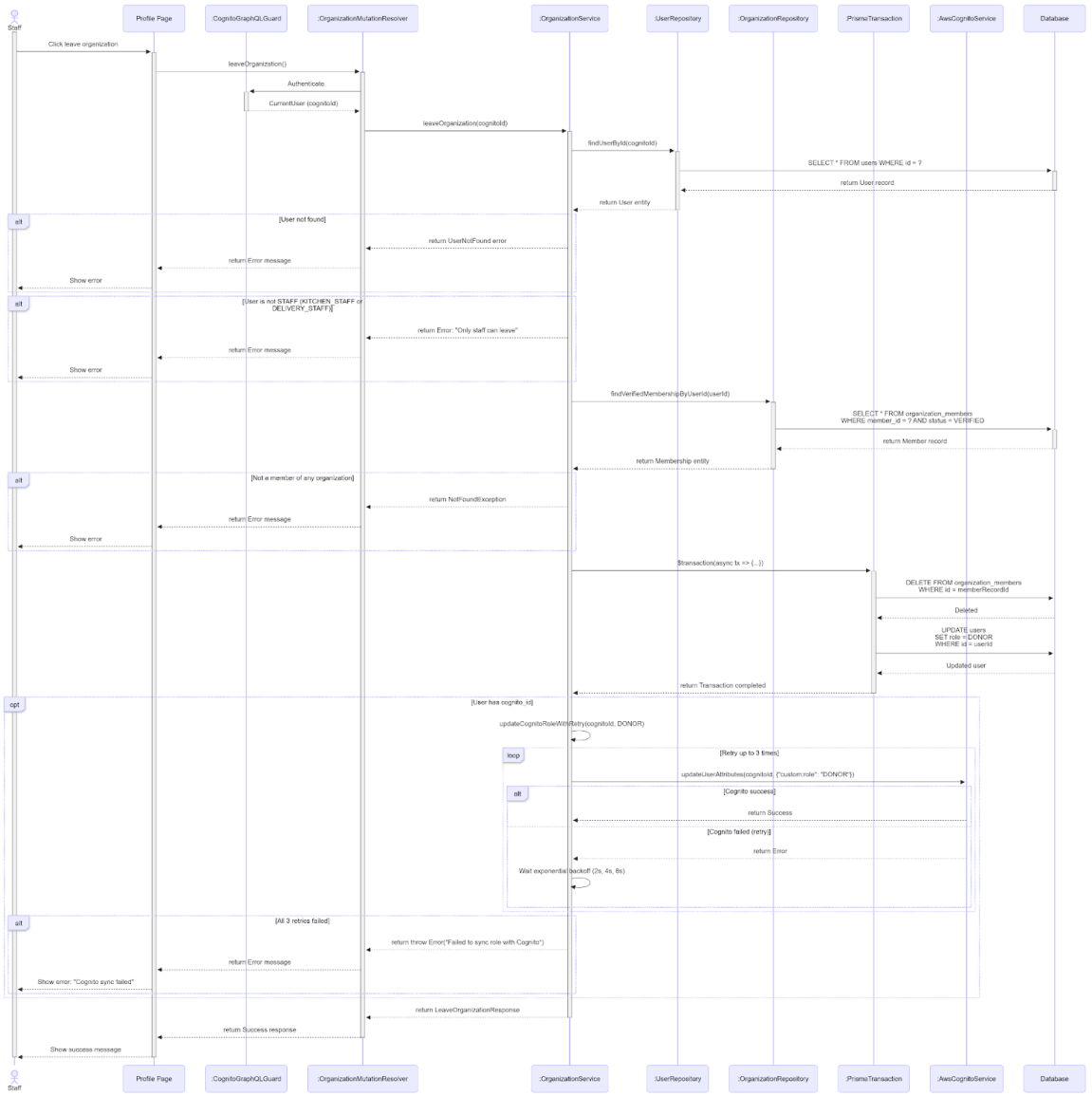


Figure 37. Leave Organization Sequence Diagram

3.4.7 Organization State Diagram

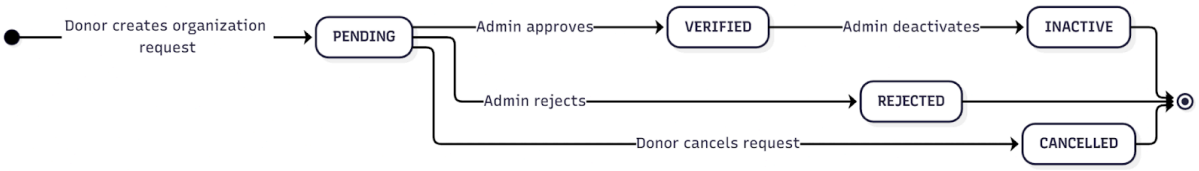


Figure 38. Organization State Diagram

3.5 Wallet Management

3.5.1 Class Diagram

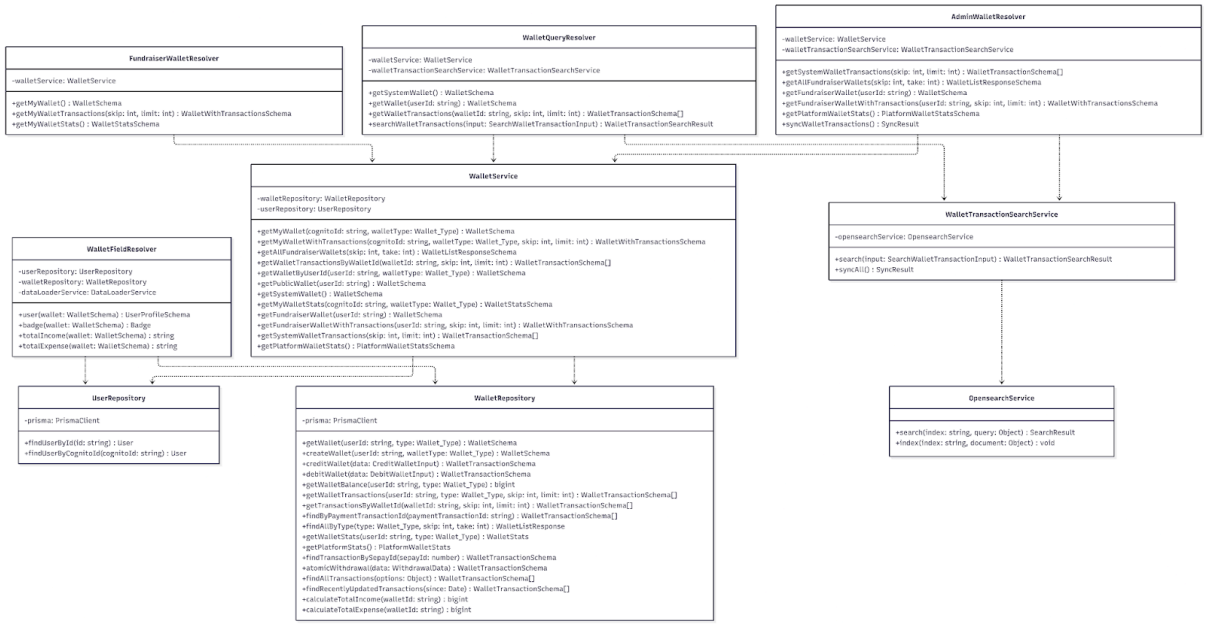


Figure 39. Wallet Management Class Diagram

3.5.2 Class Specification

WalletQueryResolver

No	Method	Description
1	**getSystemWallet()**	Get a system wallet.
2	**getWallet(userId)**	Get wallet transactions by user ID.
3	**searchWalletTransactions(input)**	Search wallet transactions.

Table 39. WalletQueryResolver Class Specification

FundraiserWalletResolver

No	Method	Description
1	**getMyWallet()**	Get a fundraiser wallet.
2	**getMyWalletTransactions(skip, limit)**	Get a fundraiser wallet with transactions.
3	**getMyWalletStats()**	Get fundraiser wallet statistics.

Table 40. FundraiserWalletResolver Class Specification

AdminWalletResolver

No	Method	Description
1	**getSystemWalletTransactions(skip, limit)**	Get system wallet transactions.
2	**getAllFundraiserWallets(skip, take)**	Get all fundraiser wallets.
3	**getFundraiserWallet(userId)**	Get a fundraiser wallet by ID.
4	**getFundraiserWalletWithTransactions(userId, skip, limit)**	Get a fundraiser wallet with transactions by ID.
5	**getPlatformWalletStats()**	Get platform-wide wallet statistics.
6	**syncWalletTransactions()**	Sync all wallet transactions to OpenSearch.

Table 41. AdminWalletResolver Class Specification

3.5.3 Search Wallet Transactions Sequence Diagram

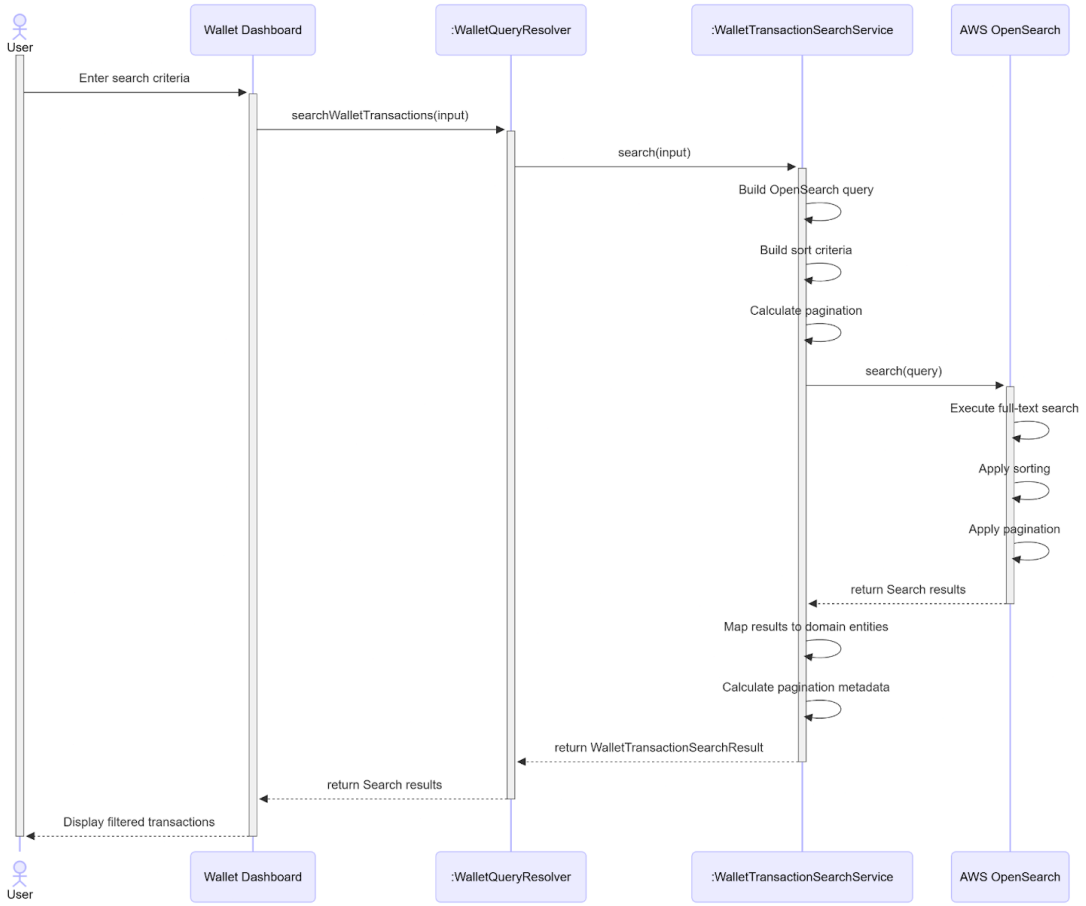


Figure 40. Search Wallet Transactions Sequence Diagram

3.5.4 Get System Wallet Sequence Diagram

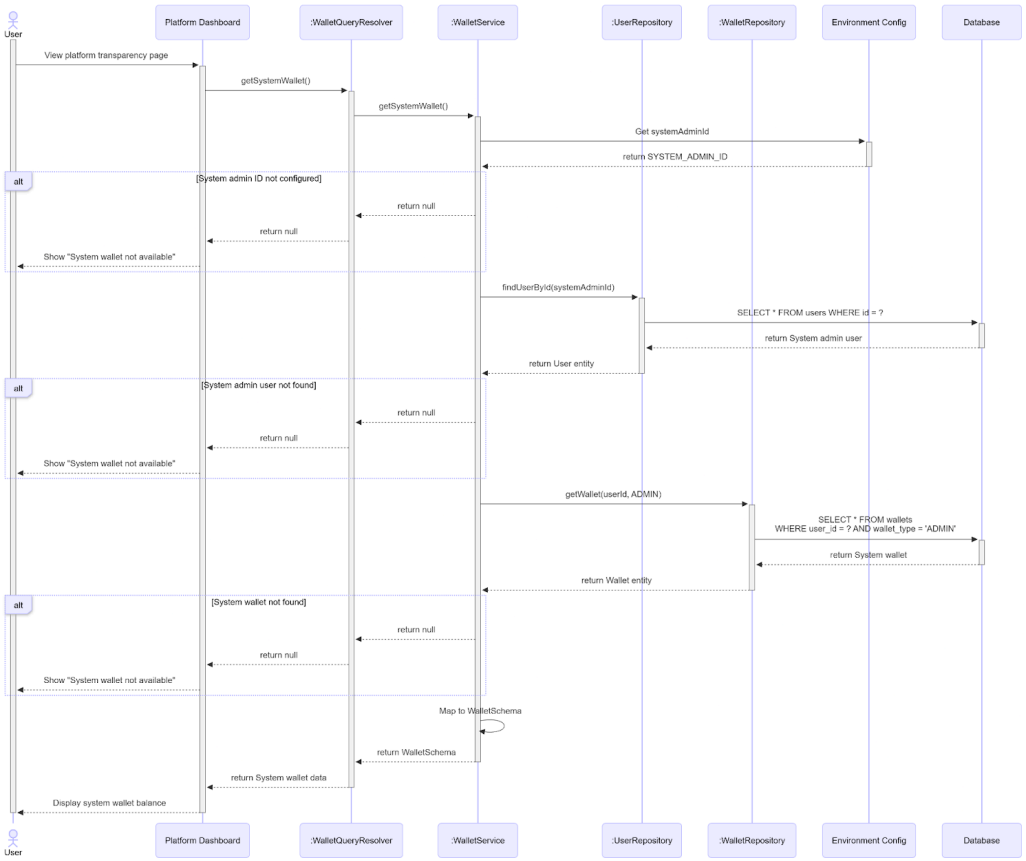


Figure 41. Get System Wallet Sequence Diagram

3.5.5 Get Fundraiser Wallet Sequence Diagram

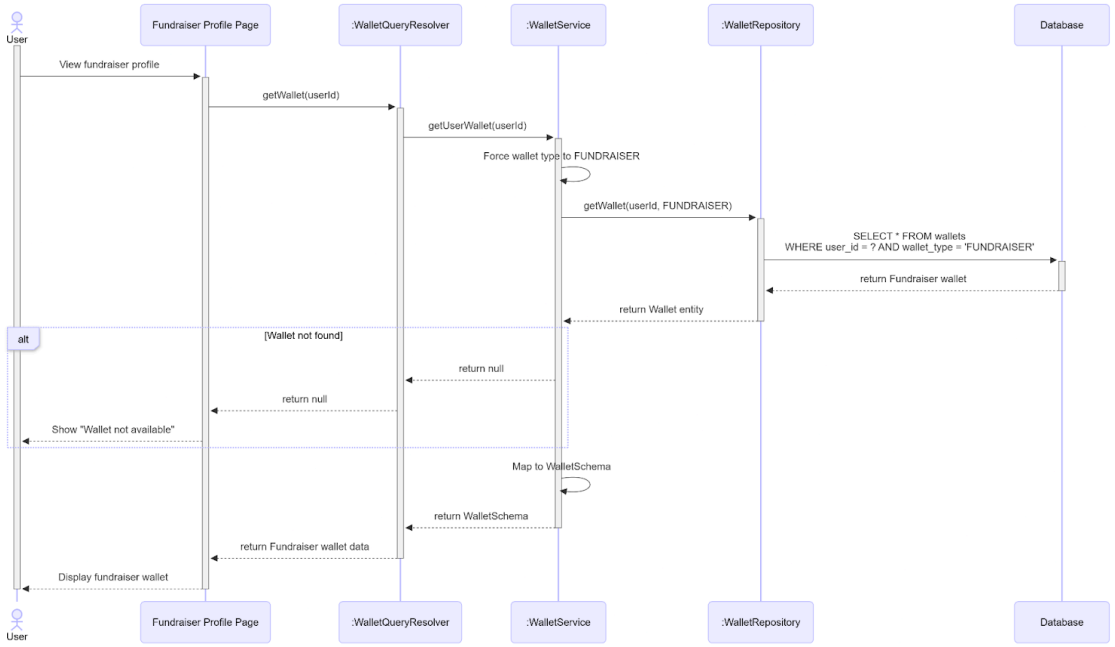


Figure 42. Get Fundraiser Wallet Sequence Diagram

3.6 Badge Management

3.6.1 Class Diagram

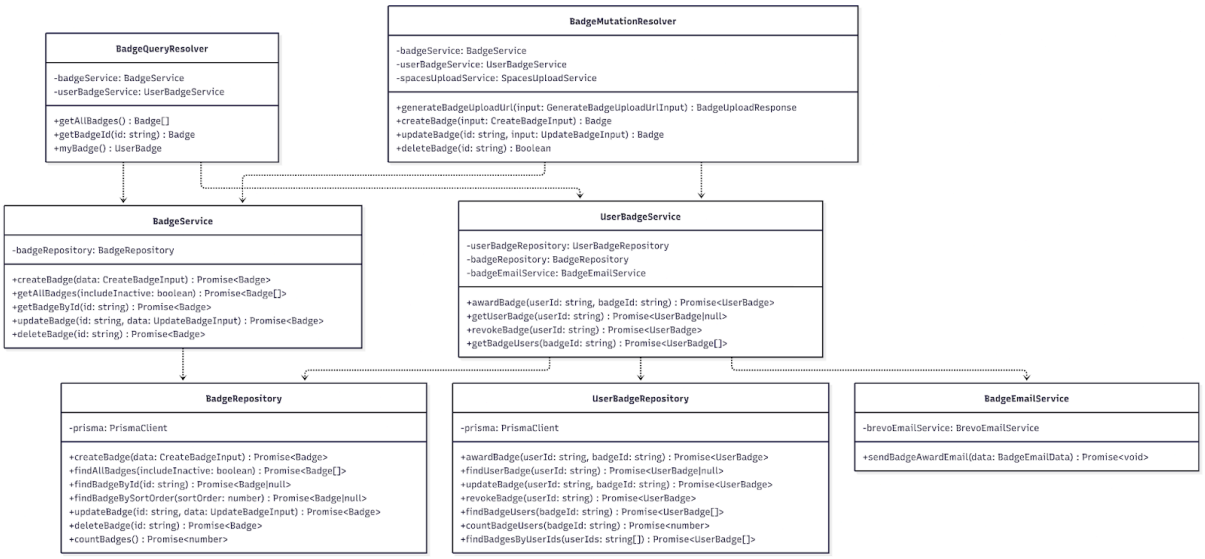


Figure 43. Badge Management Class Diagram

3.6.2 Class Specification

BadgeQueryResolver

No	Method	Description
1	**getAllBadges()**	Get all active badges available in the system.
2	**getBadgeId(id)**	Get details of a specific badge by ID.
3	**myBadge()**	Get the badge currently assigned to the authenticated user

Table 42. BadgeQueryResolver Class Specification

BadgeMutationResolver

No	Method	Description
1	**generateBadgeUploadUrl(input)**	Generate a presigned URL to upload a badge icon.
2	**createBadge(input)**	Create a new badge.
3	**updateBadge(id, input)**	Update an existing badge's details.
4	**deleteBadge(id)**	Delete a badge.

Table 43. BadgeMutationResolver Class Specification

3.6.3 View Profile Badge Sequence Diagram

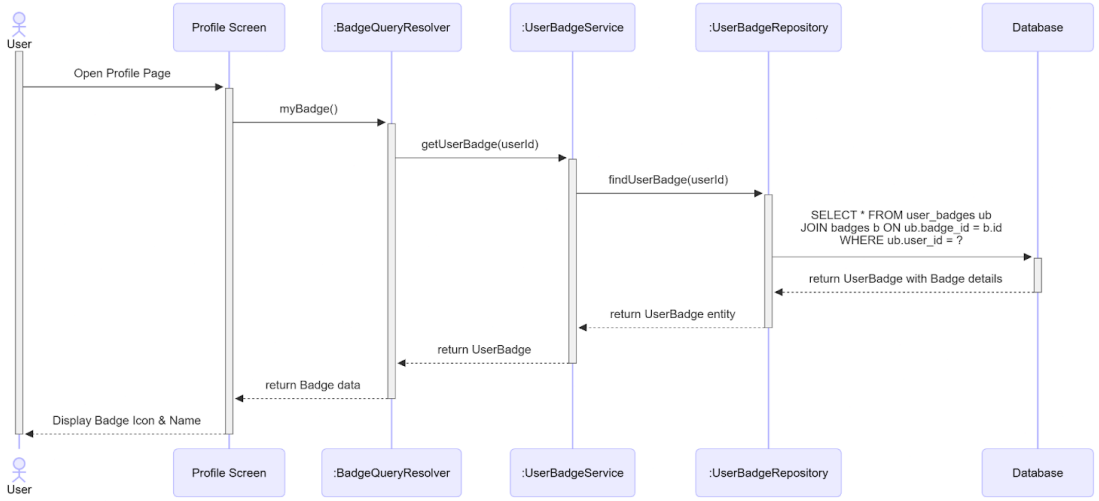


Figure 44. View Profile Badge Sequence Diagram

3.7 Donation Management

3.7.1 Class Diagram

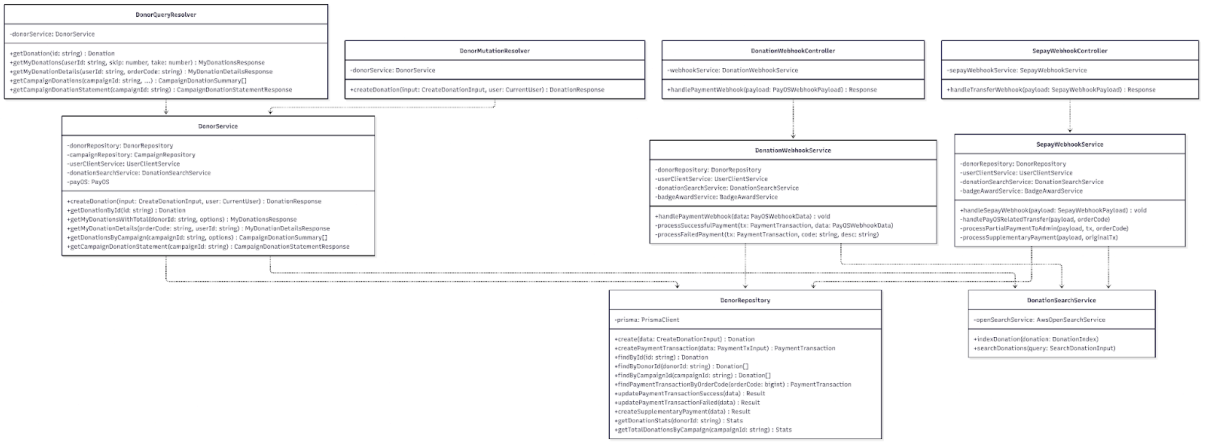


Figure 45. Donation Management Class Diagram

3.7.2 Class Specification

DonorQueryResolver

No	Method	Description
1	**getDonation(id)**	Retrieve a single donation by ID.
2	**getMyDonations(skip, take)**	Get a paginated list of donations made by the current authenticated user, including total stats.
3	**getMyDonationDetails(orderCode)**	Get detailed information about a specific donation using its order code, including payment status.
4	**getCampaignDonations(campaignId, skip, take, sortBy, sortOrder, searchDonorName)**	Get a public list of donations for a specific campaign.

No	Method	Description
5	**getCampaignDonationStatement(campaignId)**	Generate a detailed statement of all successful donations for a campaign.

Table 44. DonorQueryResolver Class Specification

DonorMutationResolver

No	Method	Description
1	**createDonation(input)**	Creates a PENDING donation record and generates a QR Code

Table 45. DonorMutationResolver Class Specification

DonationWebhookController

No	Method	Description
1	**handlePaymentWebhook(payload)**	Receives payment status updates from PayOS. Processes successful or failed payments, updates transaction status, and triggers post-payment actions (badges, emails).

Table 46. DonationWebhookController

SepayWebhookController

No	Method	Description
1	**handleTransferWebhook(payload)**	Receives bank transfer notifications from Sepay. Handles matching transfers to donations (via content parsing), partial payments, and supplementary payments.

Table 47. SepayWebhookController

3.7.3 Create Donation Sequence Diagram

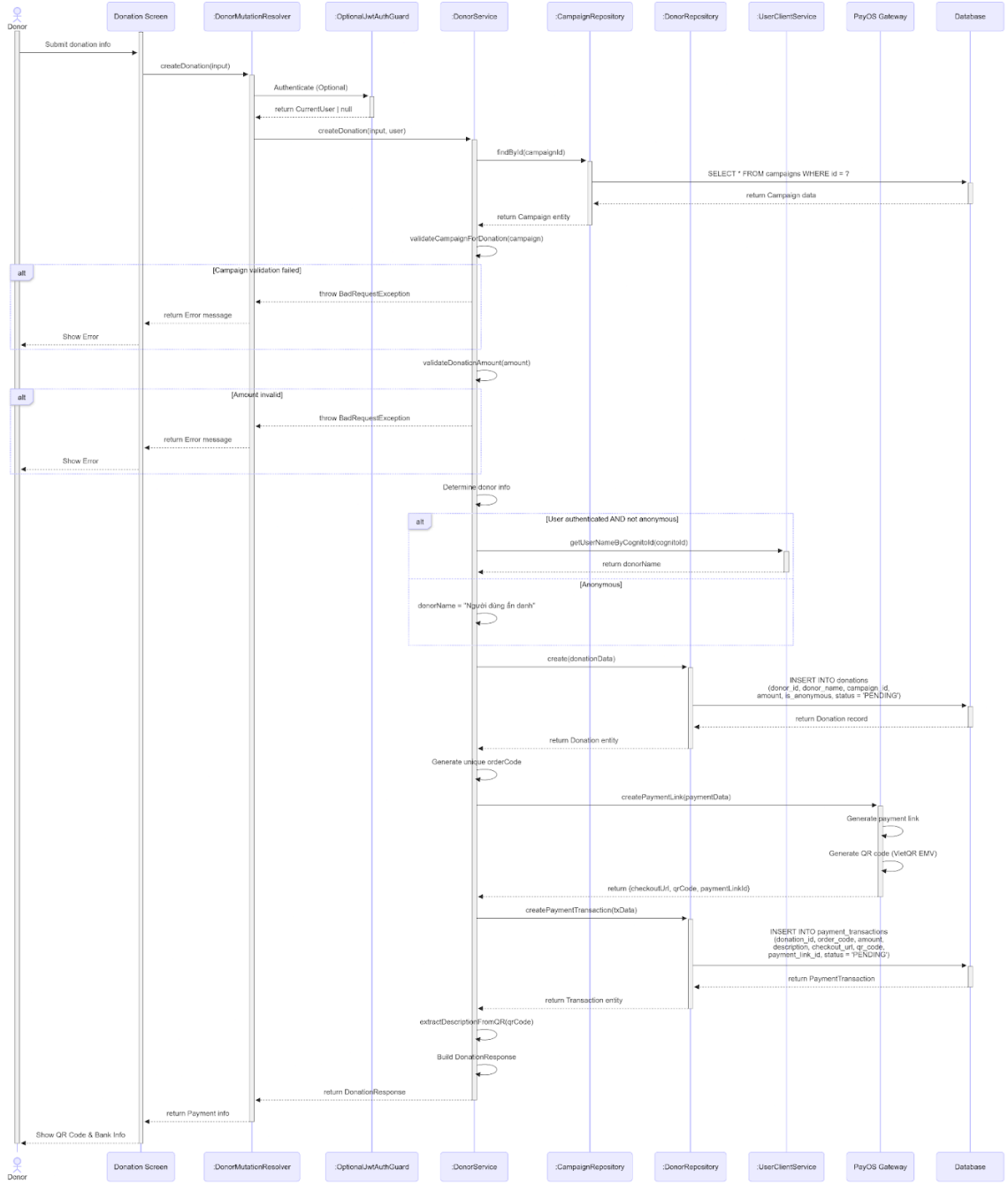


Figure 46. Create Donation Sequence Diagram

3.7.4 PayOS Webhook Donation Payment Process Sequence Diagram

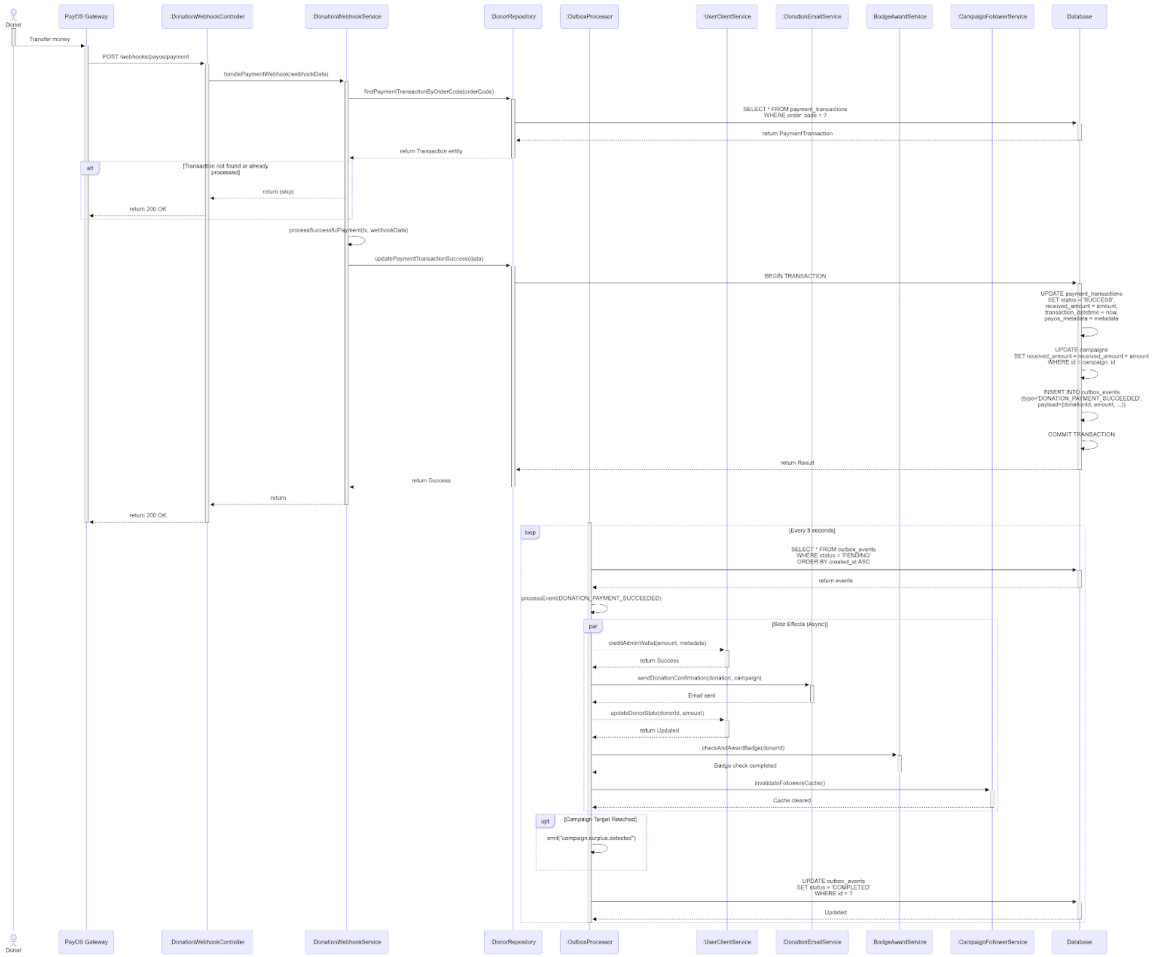


Figure 47. PayOS Webhook Donation Payment Process Sequence Diagram

3.7.5 Sepay Webhook Bank Transfer Process Sequence Diagram

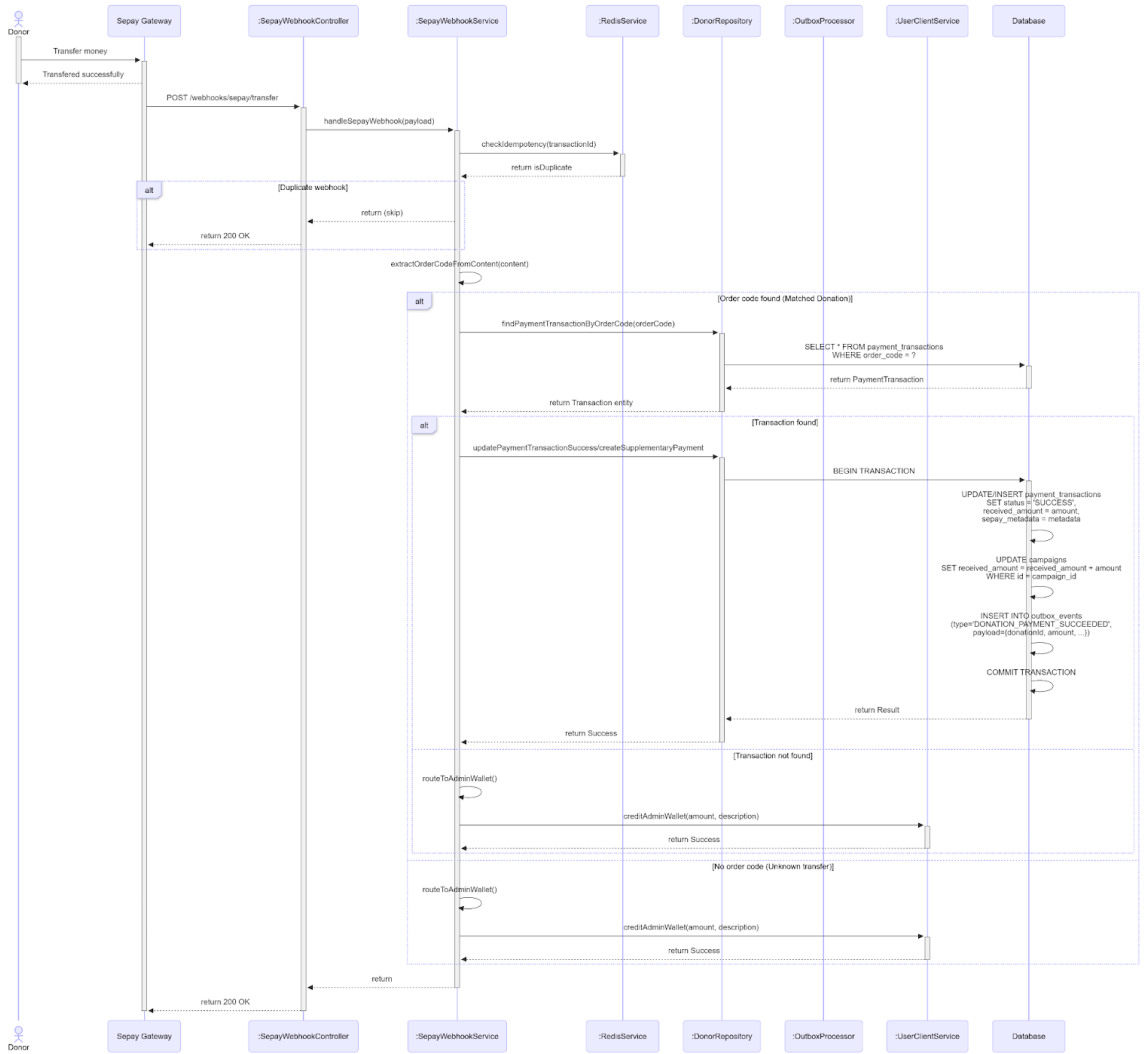


Figure 48. Sepay Webhook Bank Transfer Process Sequence Diagram

3.7.6 Donation State Diagram

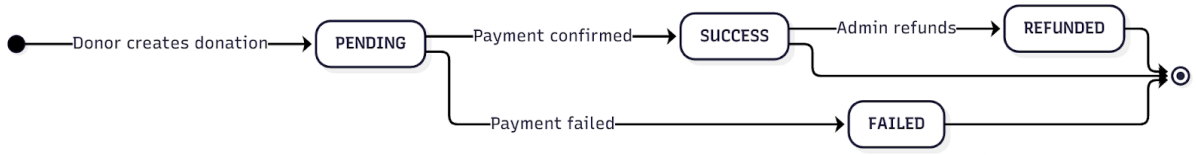


Figure 49. Donation State Diagram

3.8 Disbursement Management

3.8.1 Class Diagram

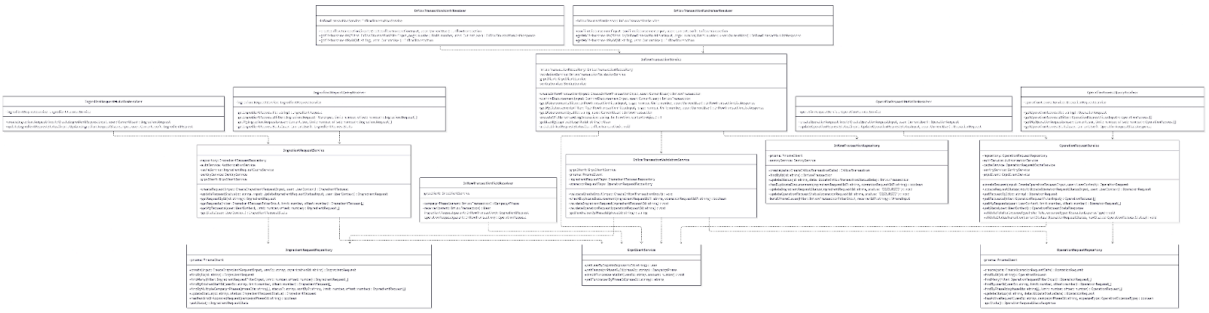


Figure 50. Disbursement Management Class Diagram

3.8.2 Class Specification

InflowTransactionAdminResolver

No	Method	Description
1	**createInflowTransaction(input)**	Creates disbursement for approved request
2	**getListInflowTransaction(filter, page, limit)**	Get all disbursements with filters
3	**getInflowTransactionDetails(id)**	Get single disbursement by ID

Table 48. InflowTransactionAdminResolver Class Specification

IngredientRequestMutationResolver

No	Method	Description
1	**createIngredientRequest(input)**	Creates ingredient request
2	**updateIngredientRequestStatus(id, input)**	Approves/Rejects by changing status ingredient request

Table 49. IngredientRequestMutationResolver Class Specification

OperationRequestMutationResolver

No	Method	Description
1	**createOperationRequest(input)**	Creates operation request
2	**updateOperationRequestStatus(input)**	Approves/Rejects by changing status operation request

Table 50. OperationRequestMutationResolver Class Specification

InflowTransactionFundraiserResolver

No	Method	Description
1	**confirmDisbursement(input)**	Confirms received disbursement from admin
2	**getMyDisbursements(filter, page, limit)**	Get own disbursements with filters
3	**getMyDisbursementById(id)**	Get single own disbursement by ID

Table 51. InflowTransactionFundraiserResolver Class Specification

IngredientRequestQueryResolver

No	Method	Description
1	**getIngredientRequest(id)**	Get single ingredient request by ID
2	**getIngredientRequests(filter, limit, offset)**	Get all ingredient requests with filters
3	**getMyIngredientRequests(limit, offset)**	Get own ingredient requests
4	**getIngredientRequestStats()**	Get statistics for own requests

Table 52. IngredientRequestQueryResolver Class Specification

OperationRequestQueryResolver

No	Method	Description
1	**getOperationRequest(id)**	Get single operation request by ID
2	**getOperationRequests(filter)**	Get all operation requests with filters
3	**getMyOperationRequests(limit, offset)**	Get own operation requests
4	**getOperationRequestStats()**	Get statistics for own requests

Table 53. OperationRequestQueryResolver Class Specification

3.8.3 Create Ingredient Request Sequence Diagram

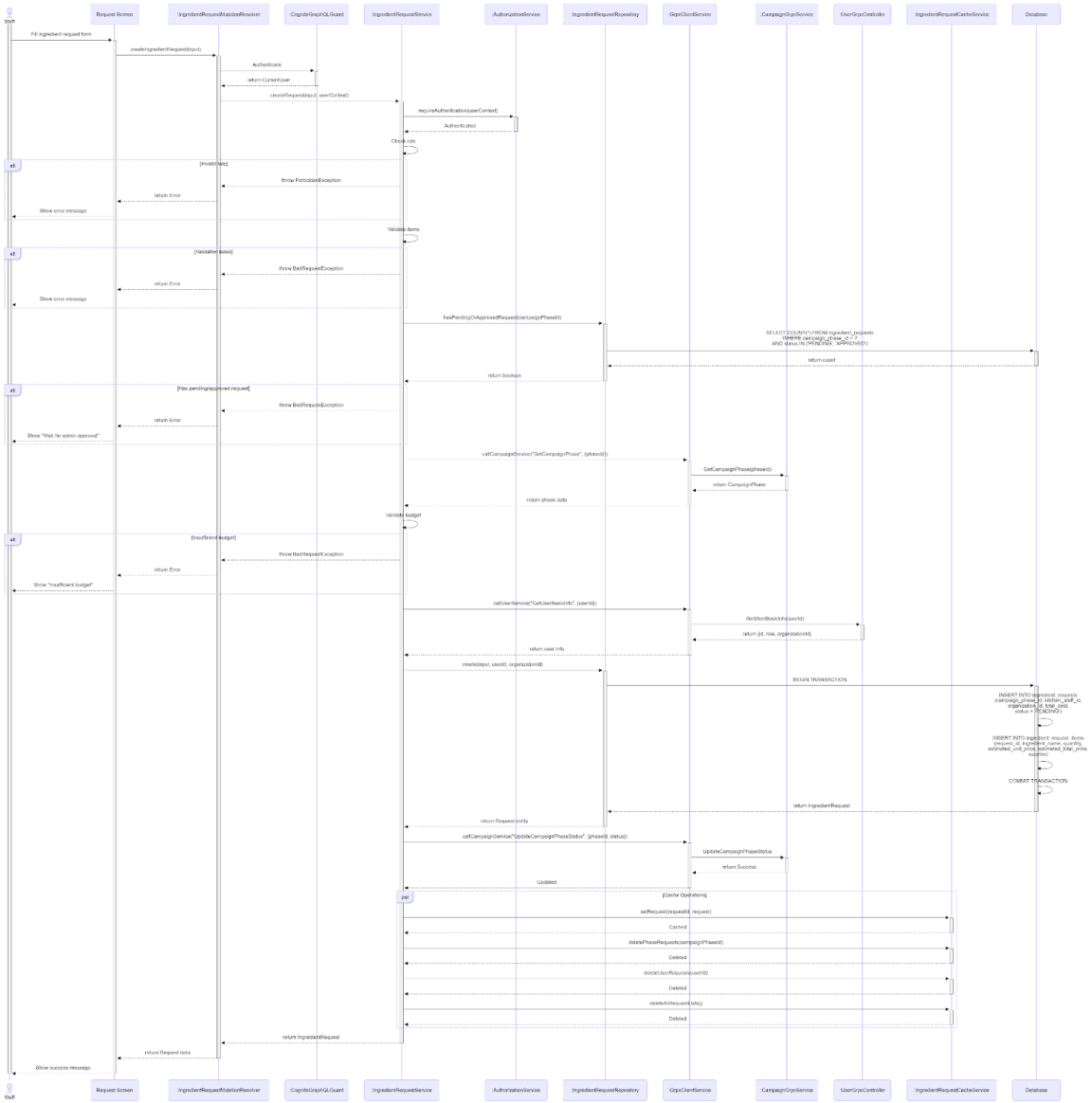


Figure 51. Create Ingredient Request Sequence Diagram

3.8.4 Approve Ingredient Request Sequence Diagram

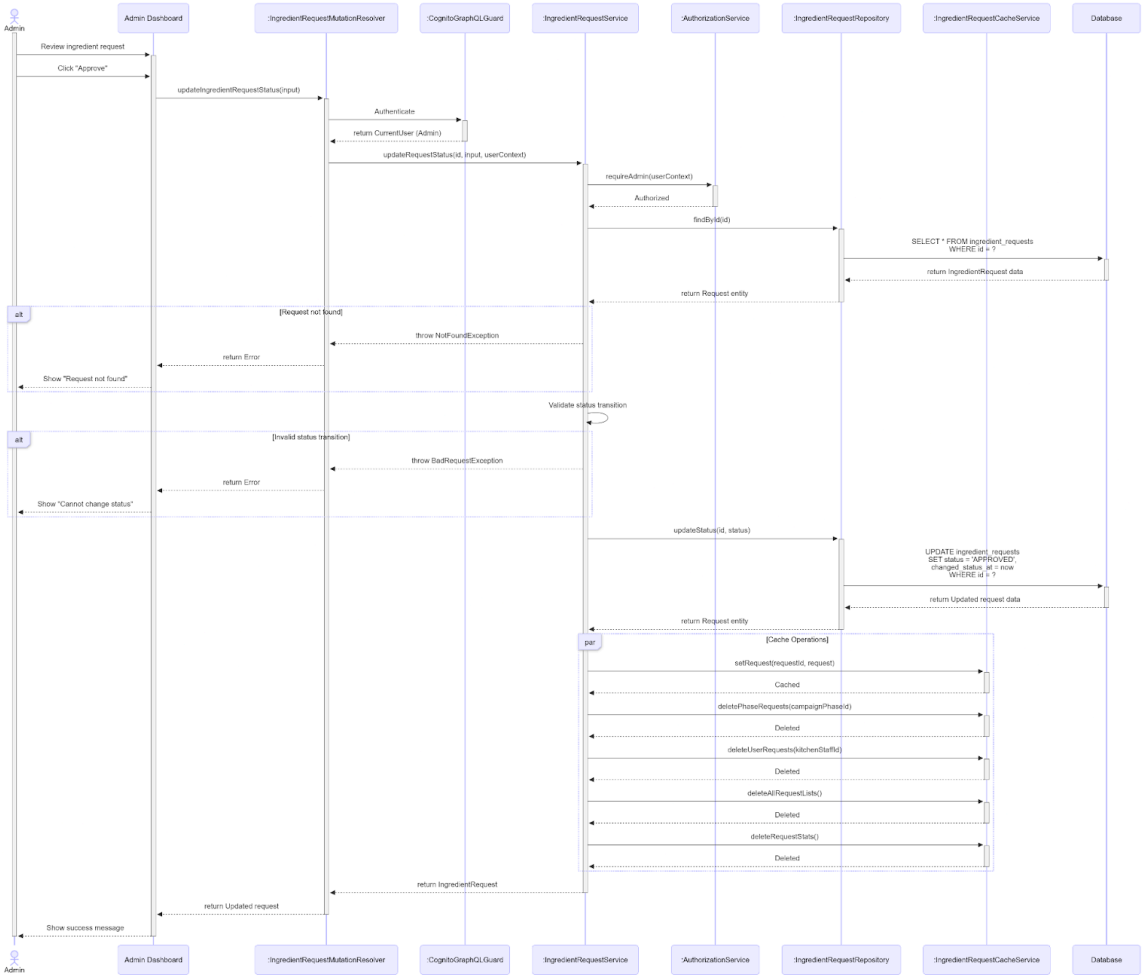


Figure 52. Approve Ingredient Request Sequence Diagram

3.8.5 Create Disbursement Sequence Diagram

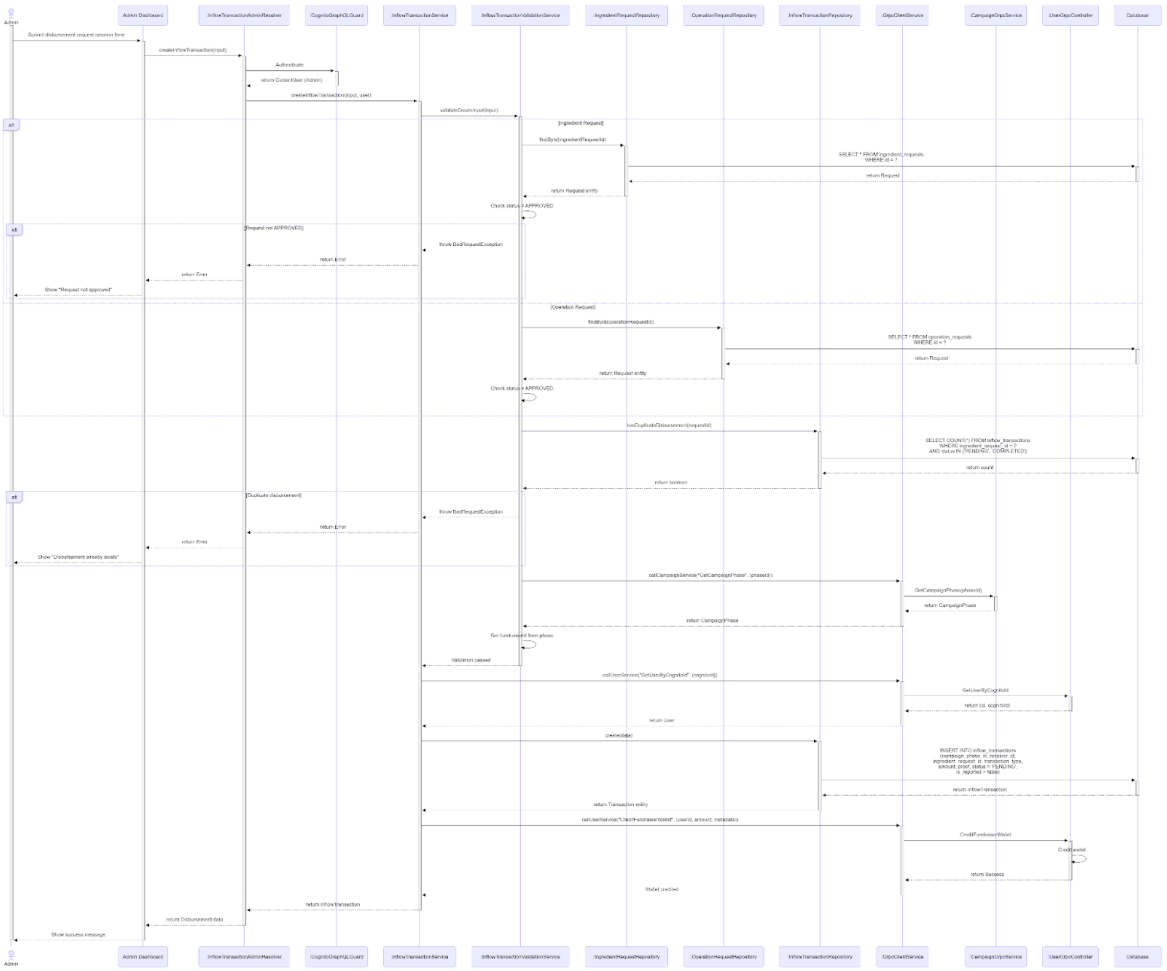


Figure 53. Create Disbursement Sequence Diagram

3.8.6 Confirm Disbursement Sequence Diagram

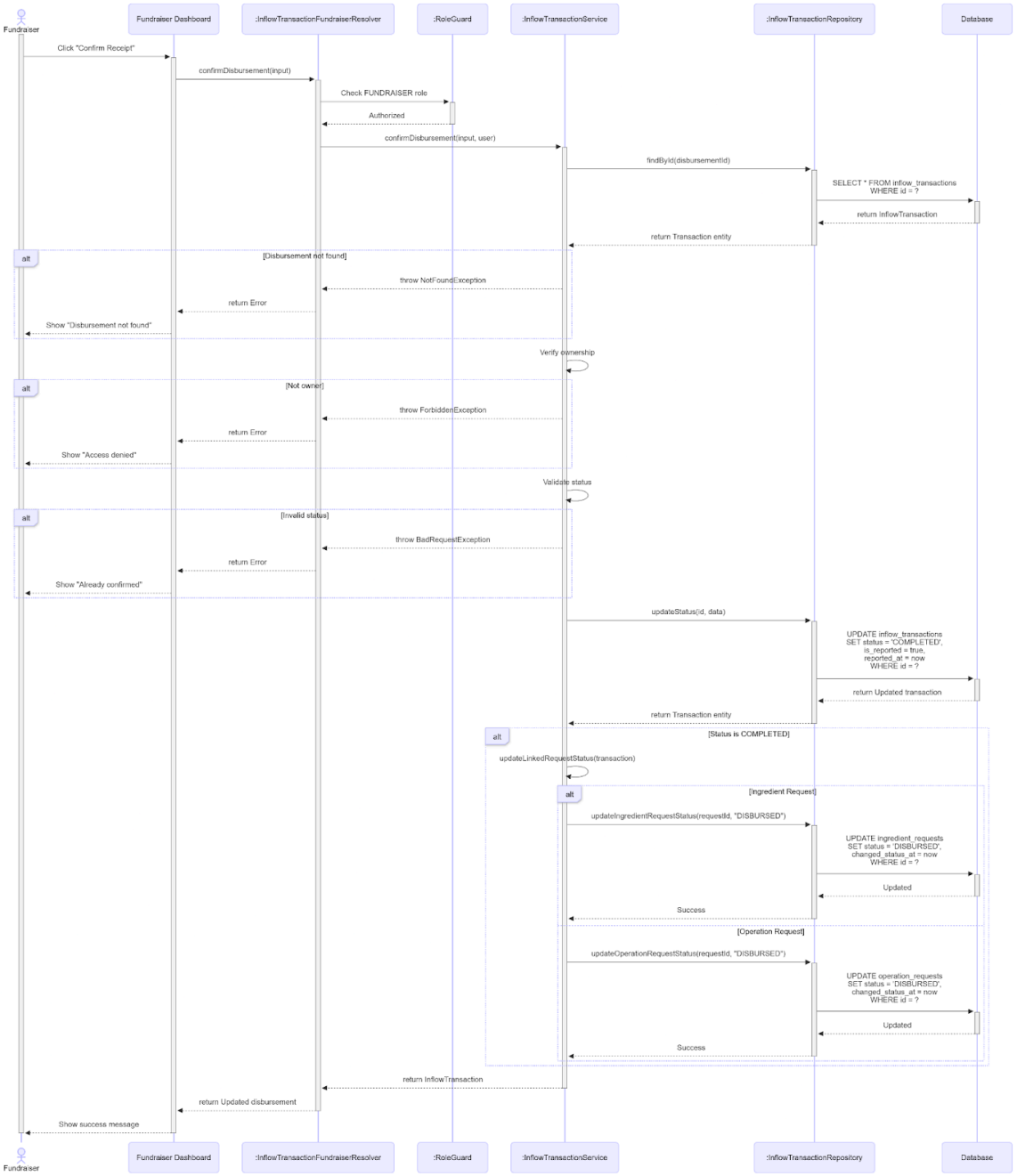


Figure 54. Confirm Disbursement Sequence Diagram

3.9 Campaign Management

3.9.1 Class Diagram

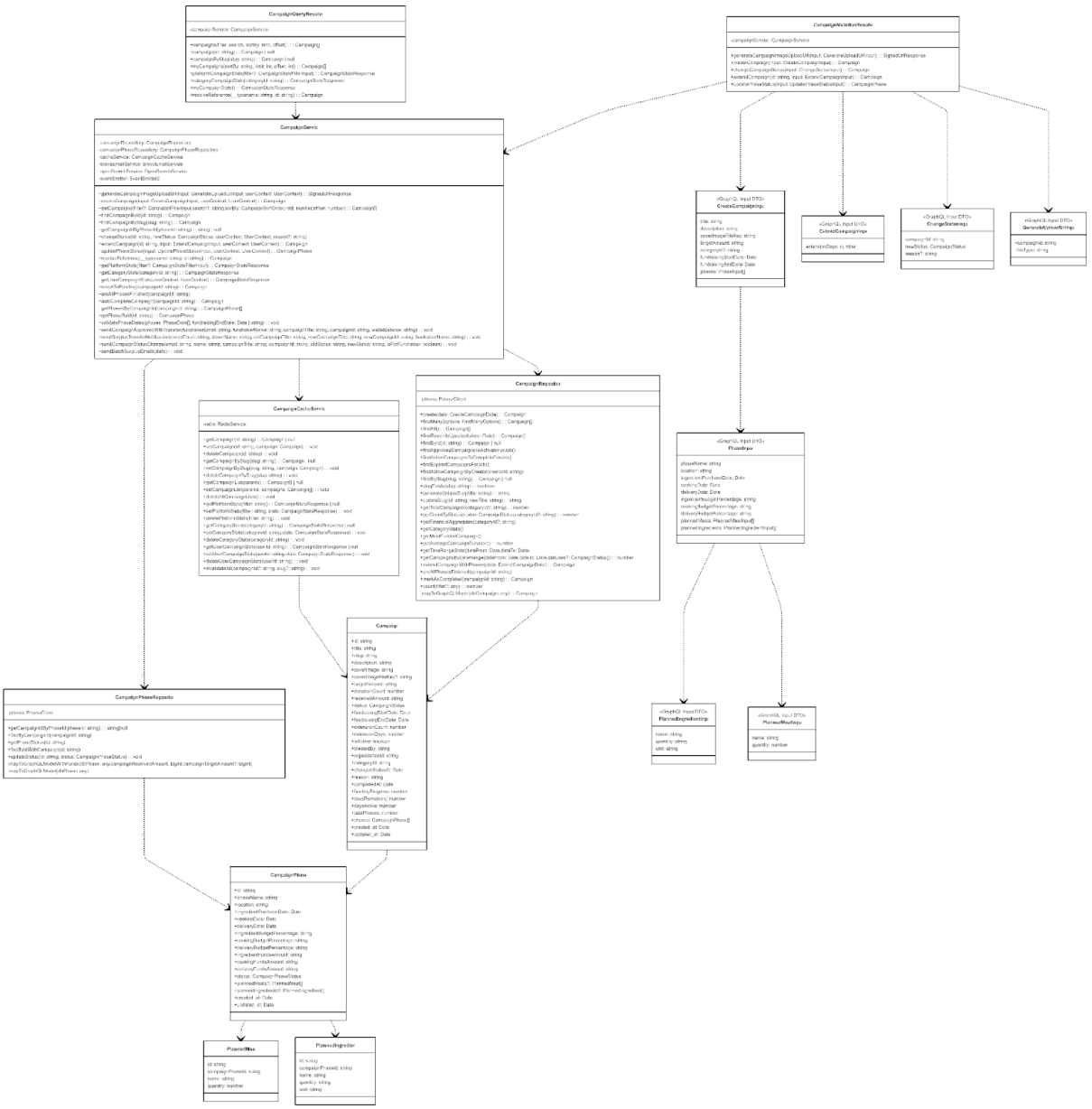


Figure 55. Campaign Management Class Diagram

3.9.2 Class Specifications

CampaignQueryResolver

No	Method	Description
1	**campaigns(filter, search, sortBy, limit, offset)**	Retrieves a paginated list of campaigns based on optional filters (status, category), search terms, and sorting preferences.
2	**campaign(id)**	Retrieves the details of a specific campaign by its unique identifier. Returns null if not found.
3	**campaignBySlug(slug)**	Retrieves a campaign's details using its unique URL-friendly slug.
4	**myCampaigns(sortBy, limit, offset)**	Fetches a list of campaigns created by the currently authenticated user (Fundraiser).
5	**platformCampaignStats(filter)**	Retrieves aggregated statistics for the entire platform (or filtered scope), including total funds raised, success rates, and campaign counts.
6	**categoryCampaignStats(categoryId)**	Provides statistical performance data specific to a campaign category.
7	**myCampaignStats()**	Provides statistical data specific to the campaigns owned by the authenticated user.
8	**resolveReference(__typename, id)**	Resolves the campaign entity for GraphQL Federation when referenced by other subgraphs.

Table 54. CampaignQueryResolver Class Specification

CampaignMutationResolver

No	Method	Description
1	**createCampaign(input)**	Orchestrates the creation of a new campaign, including its basic information and initial phases.
2	**changeCampaignStatus(input)**	Transitions a campaign through its lifecycle.
3	**extendCampaign(id, input)**	Extends the fundraising end date of a campaign.
4	**generateCampaignImageUploadUrl(input)**	Generates a pre-signed URL for secure image uploading to object storage (S3/Spaces). Enforces ownership checks if updating an existing campaign.
5	**updatePhaseStatus(input)**	Transitions a campaign phase through its lifecycle.

Table 55. CampaignMutationResolver Class Specification

3.9.3 Create Campaign Sequence Diagram

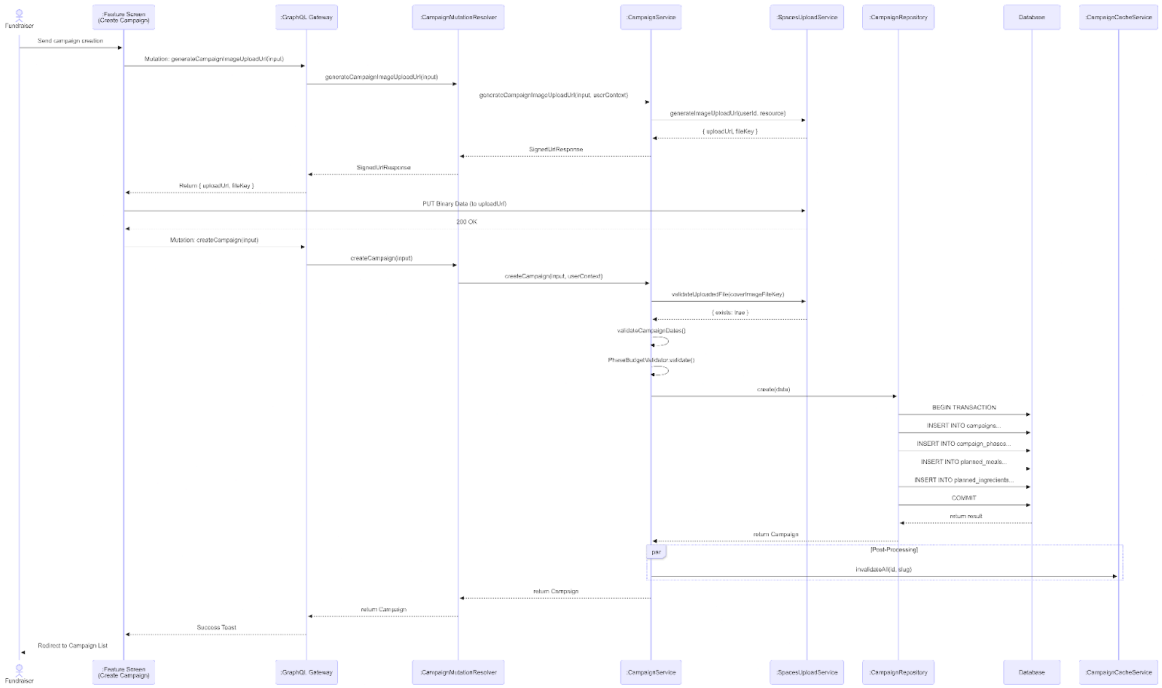


Figure 56. Create Campaign Sequence Diagram

3.9.4 Search Campaigns Sequence Diagram

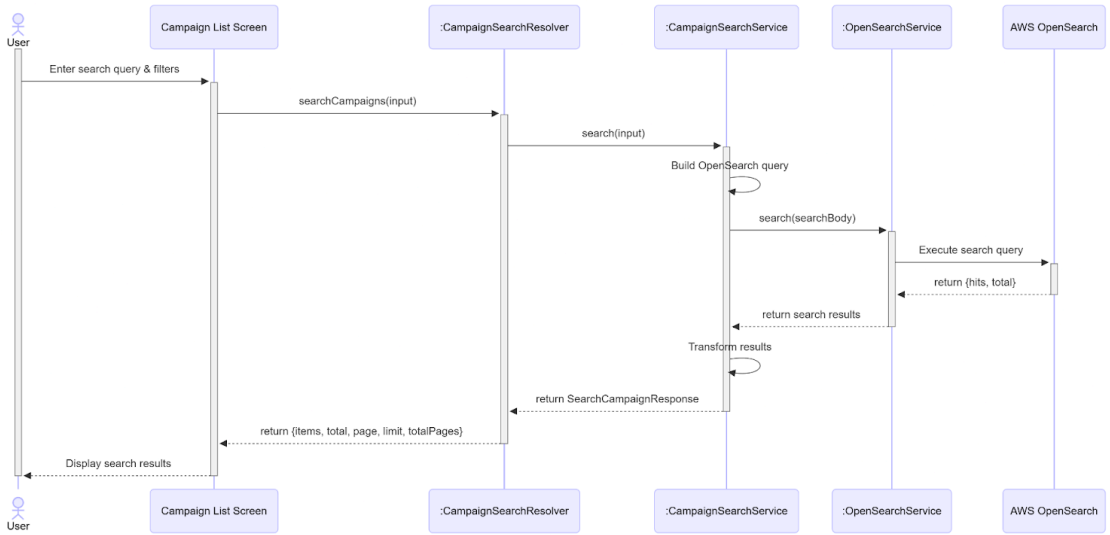


Figure 57. Search Campaigns Sequence Diagram

3.9.5 View Campaign Detail Sequence Diagram

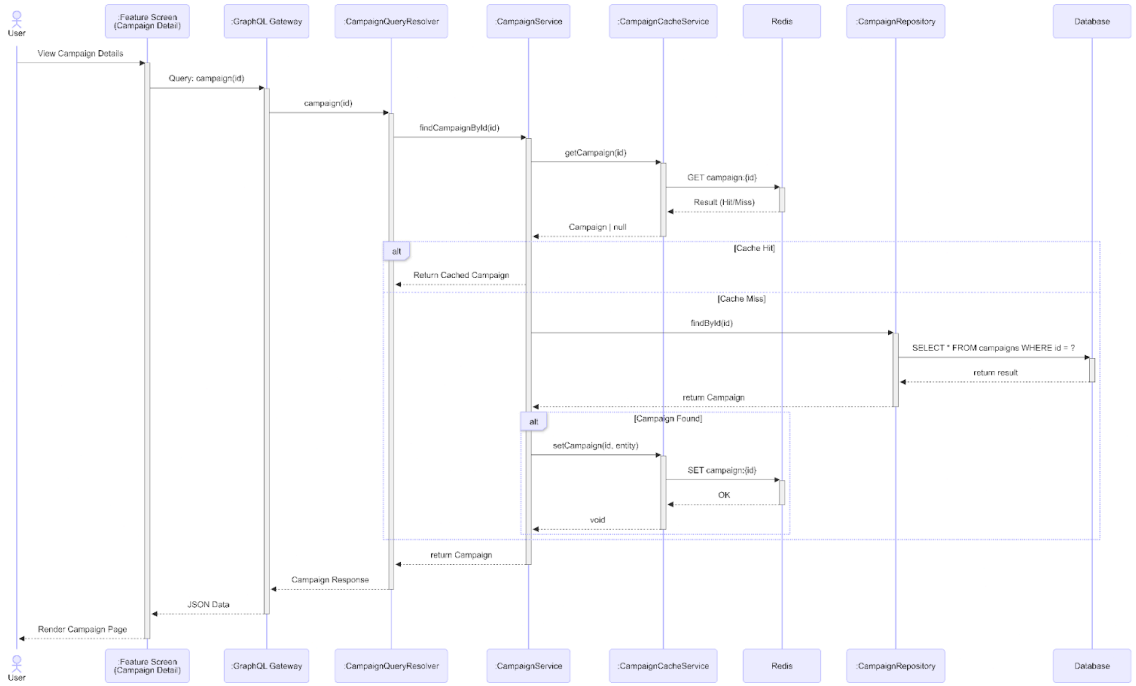


Figure 58. View Campaign Detail Sequence Diagram

3.9.6 View My Campaigns Sequence Diagram

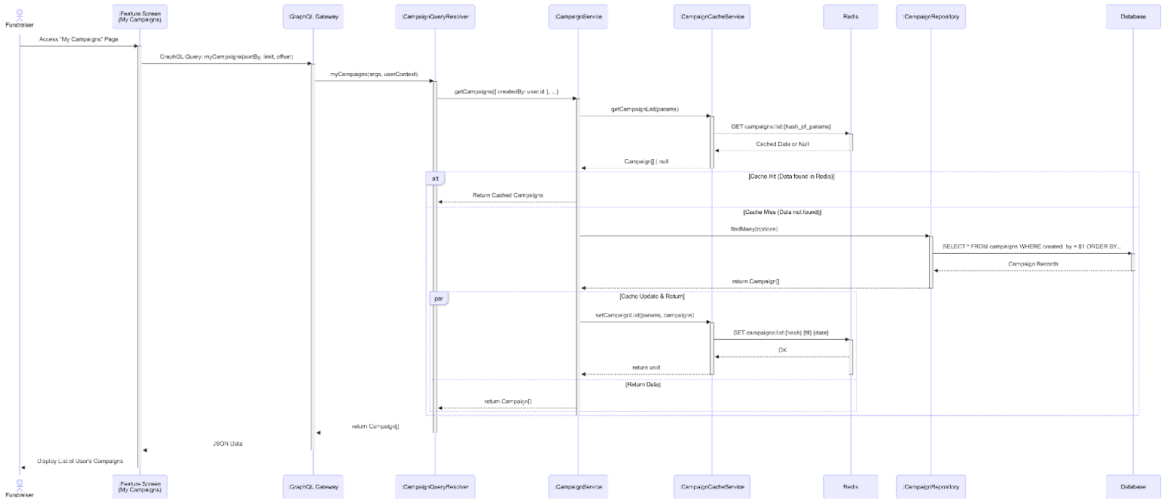


Figure 59. View My Campaigns Sequence Diagram

3.9.7 View Campaign Statistics Sequence Diagram

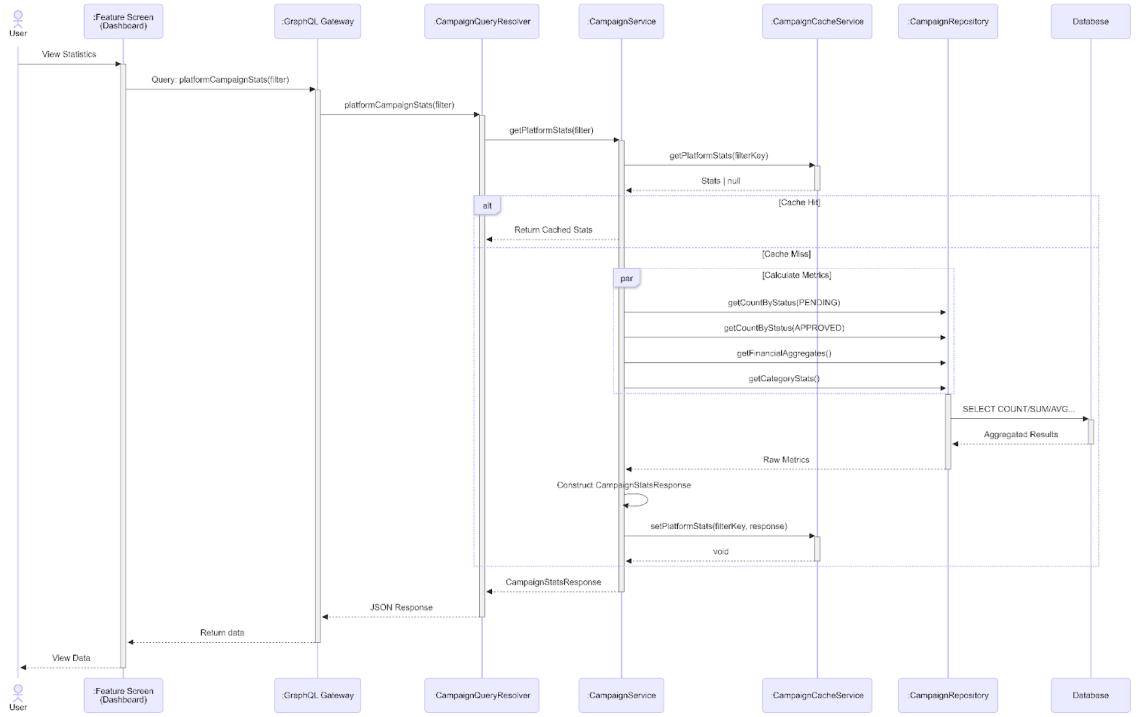


Figure 60. View Campaign Statistics Sequence Diagram

3.9.8 Change Campaign Status Sequence Diagram

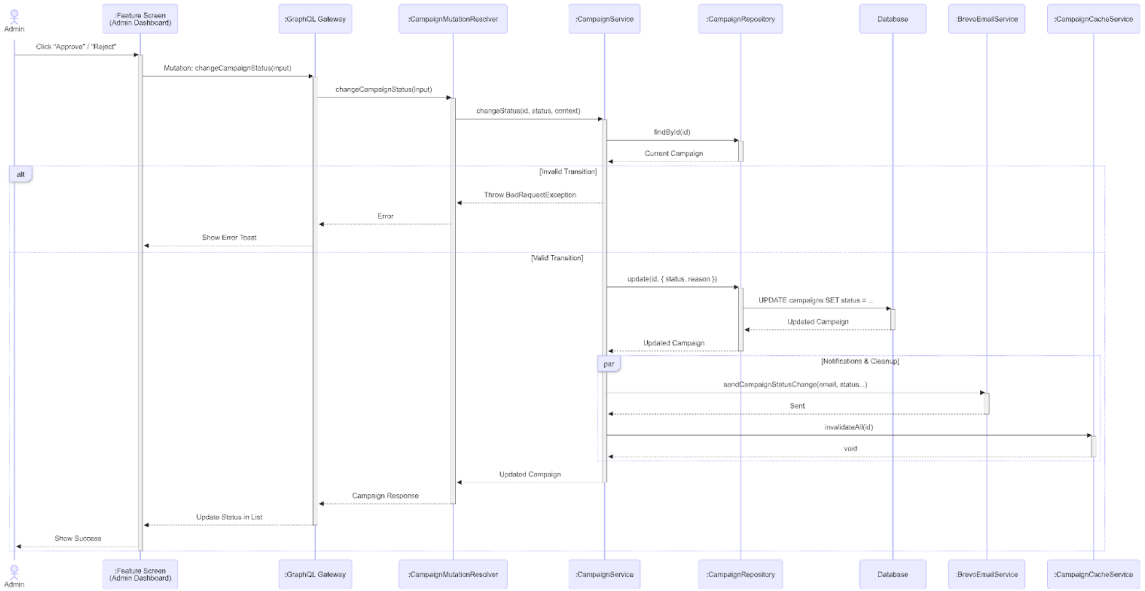


Figure 61. Change Campaign Status Sequence Diagram

3.9.9 Extend Campaign Sequence Diagram

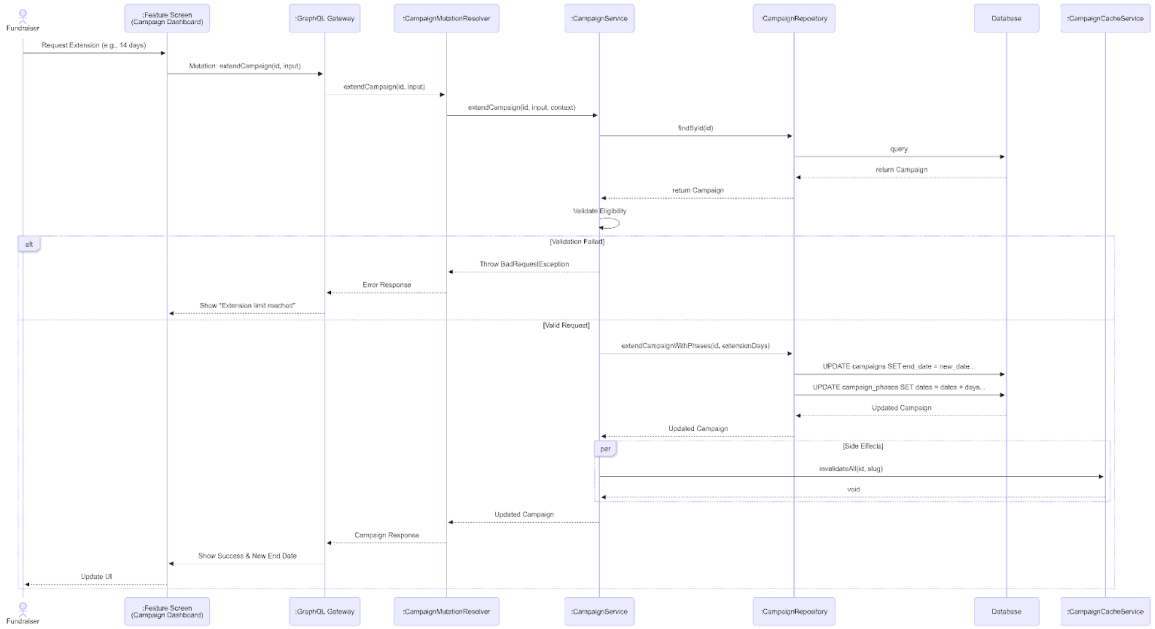


Figure 62. Extend Campaign Sequence Diagram

3.9.10 Update Phase Status Sequence Diagram

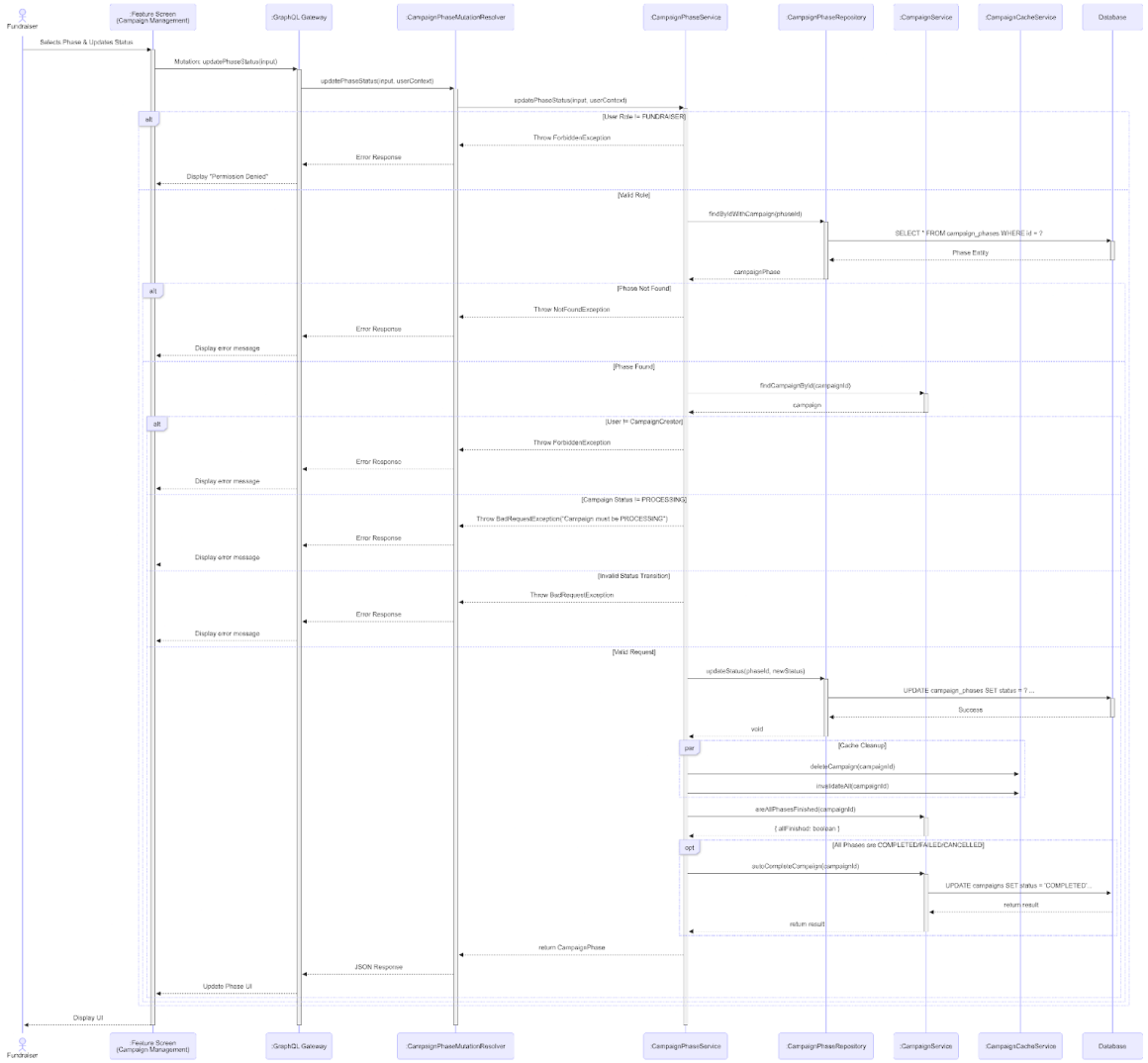


Figure 63. Update Phase Status Sequence Diagram

3.9.11 Campaign State Diagram

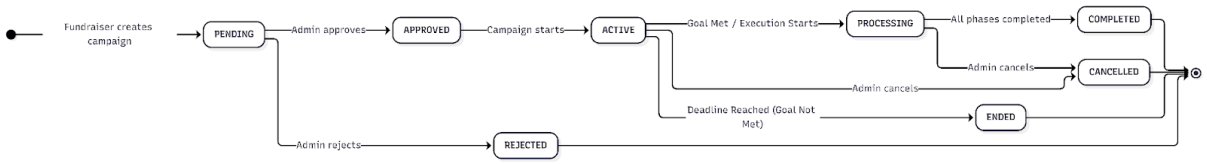


Figure 64: Campaign State Diagram

3.9.12 Campaign Phase State Diagram

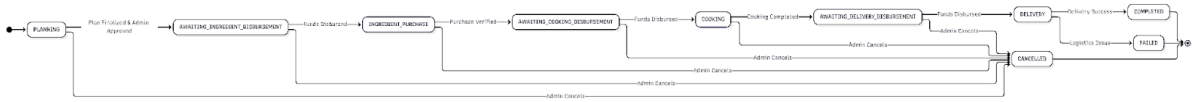


Figure 65: Campaign Phase State Diagram

3.10 Campaign Reassignment Management

3.10.1 Class Diagram

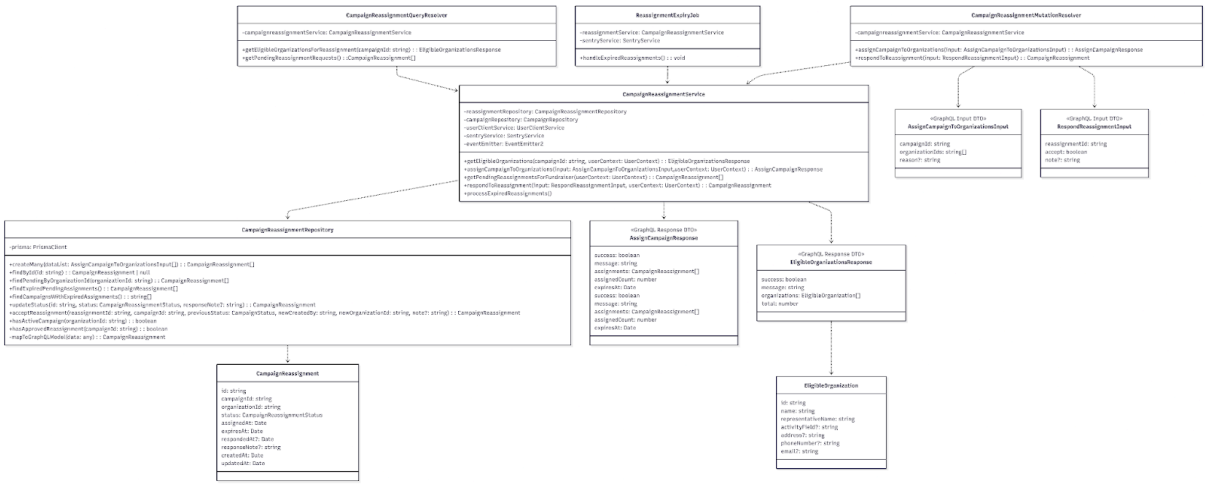


Figure 66. Campaign Reassignment Management Class Diagram

3.10.2 Class Specifications

CampaignReassignmentQueryResolver

No	Method	Description
1	**getEligibleOrganizationsForReassignment(campaignId)**	Retrieval endpoint used by Administrators. It delegates to the service layer to fetch a list of verified organizations that are currently eligible to take over the specified cancelled campaign (i.e., they have no currently active campaigns).
2	**getPendingReassignmentRequests()**	Retrieval endpoint used by Fundraisers. It fetches all incoming reassignment proposals targeting the authenticated user's organization that are currently in a PENDING state.

Table 56. CampaignReassignmentQueryResolver Class Specification

CampaignReassignmentMutationResolver

No	Method	Description
1	**assignCampaignToOrganizations(input)**	Administrative mutation that initiates the reassignment workflow. It accepts a campaign ID and a list of target organization IDs, instructing the service to create formal reassignment requests for them.
2	**respondToReassignment(input)**	Fundraiser mutation used to accept or reject a reassignment proposal. It processes the decision and provides an optional response note.

Table 57. CampaignReassignmentMutationResolver Class Specification

3.10.3 Assign Campaign To Organizations Sequence Diagram

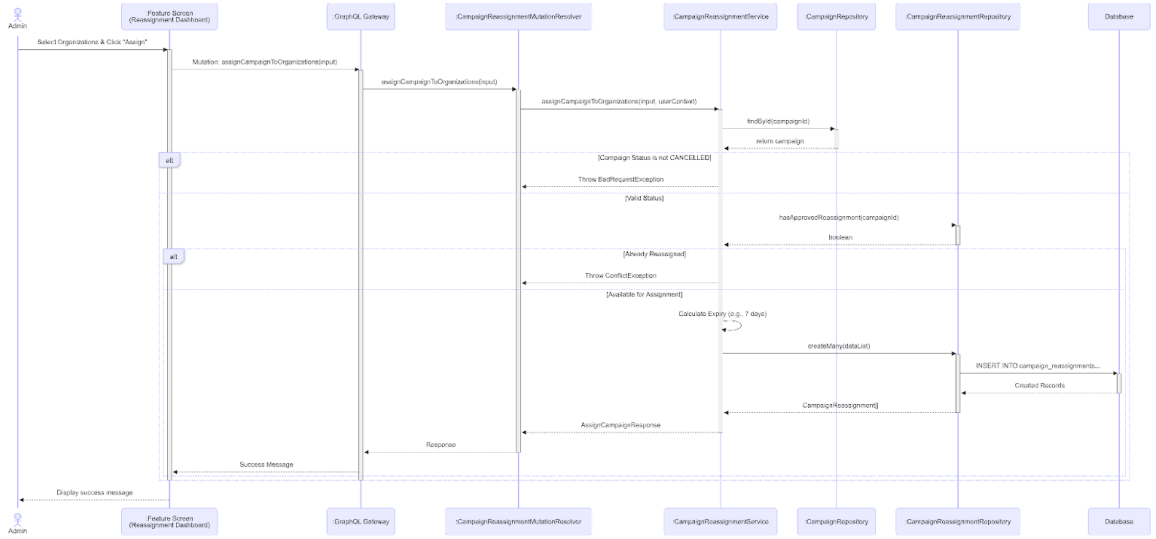


Figure 67. Assign Campaign To Organizations Sequence Diagram3.10.4 Get Pending Reassignment Requests Sequence Diagram

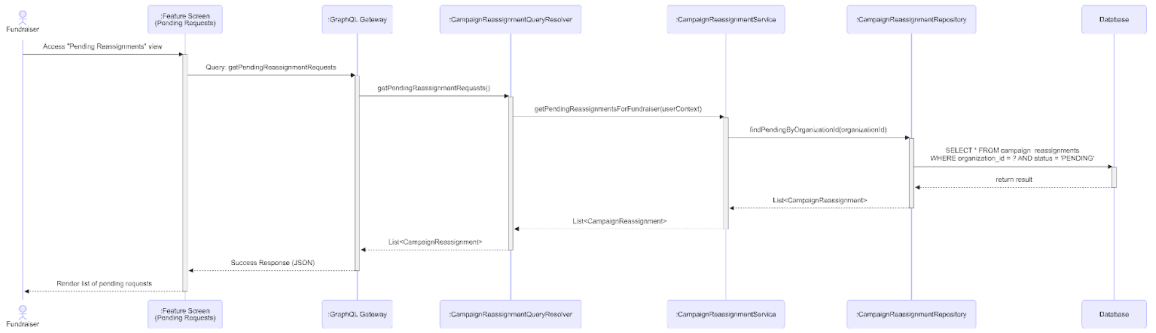


Figure 68. Get Pending Reassignment Requests Sequence Diagram

3.10.5 Respond Campaign Assignment Sequence Diagram

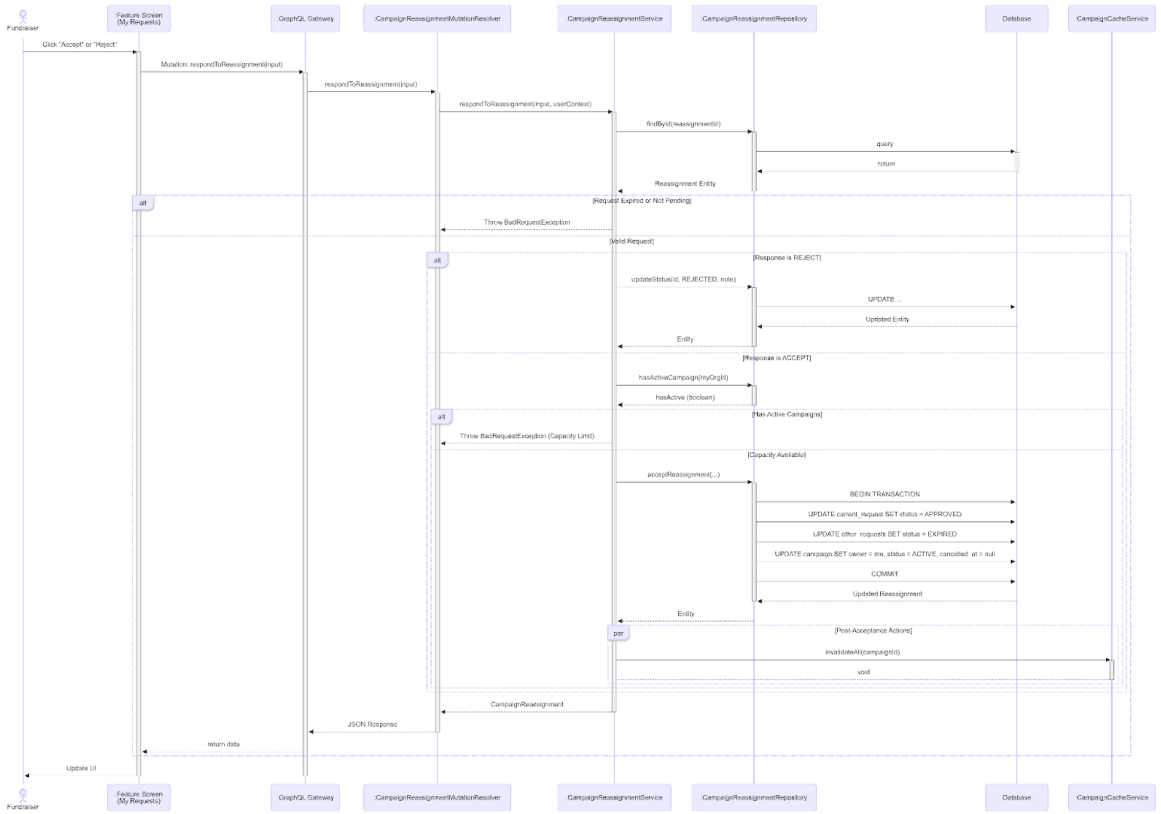


Figure 69. Respond Campaign Assignment Sequence Diagram

3.10.6 Campaign Reassignment State Diagram

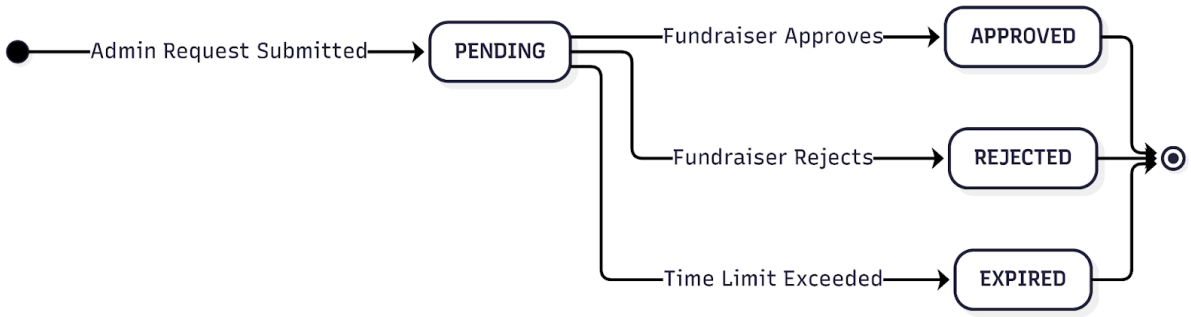


Figure 70: Campaign Reassignment State Diagram

3.11 Post Management

3.11.1 Class Diagram

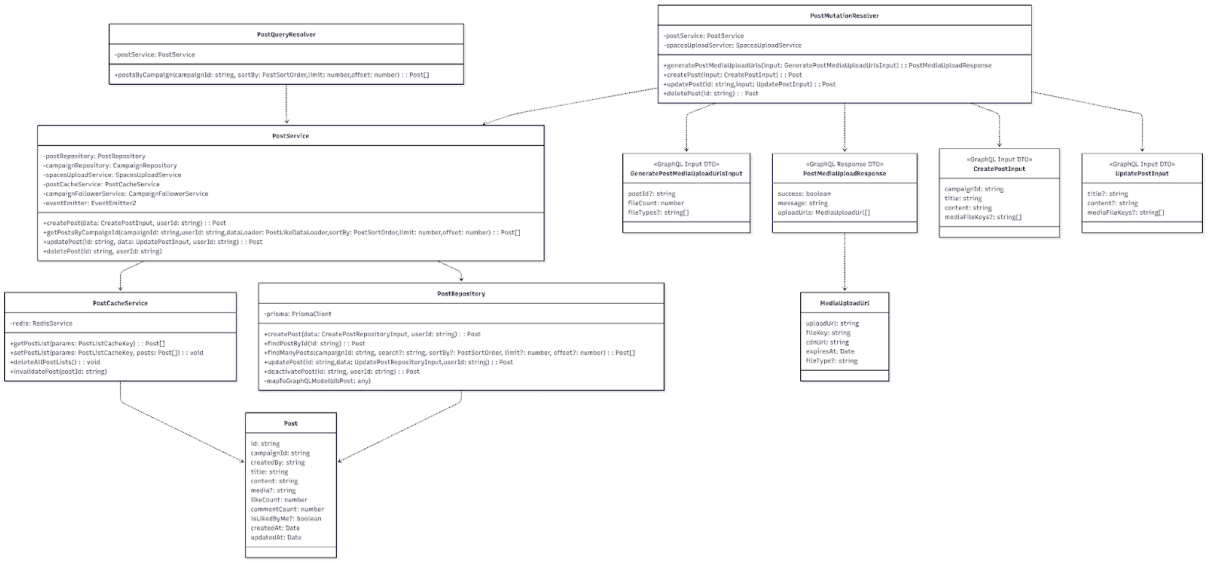


Figure 71. Post Management Class Diagram

3.11.2 Class Specifications

PostQueryResolver

No	Method	Description
1	**postsByCampaign(campaignId: string, sortBy: PostSortOrder, limit: number, offset: number): Post[]**	Retrieves a paginated list of posts belonging to a specific campaign, applying sorting rules defined by PostSortOrder. Delegates business logic to PostService.

Table 58. PostQueryResolver Class Specification

PostMutationResolver

No	Method	Description
1	**generatePostMediaUploadUrls(input: GeneratePostMediaUploadUrlsInput): PostMediaUploadResponse**	Produces pre-signed upload URLs for post media files using the SpacesUploadService.
2	**createPost(input: CreatePostInput):**	Creates a new post using the provided input payload. Delegates validation and persistence logic to PostService.

No	Method	Description
	Post	
3	**updatePost(id: string, input: UpdatePostInput): Post**	Updates an existing post with new values provided by the client.
4	**deletePost(id: string): Post**	Soft-deletes an existing post through the service layer.

Table 59. PostMutationResolver Class Specification

3.11.3 Create Post Sequence Diagram

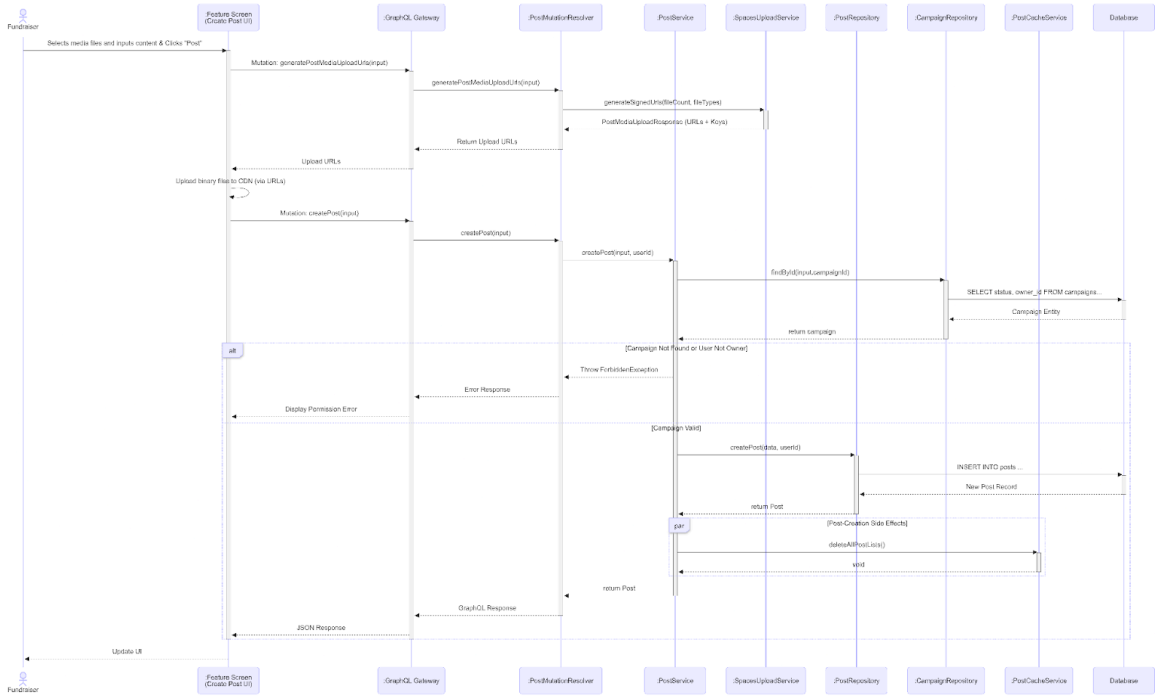


Figure 72. Create Post Sequence Diagram

3.11.4 Get Posts by Campaign Sequence Diagram

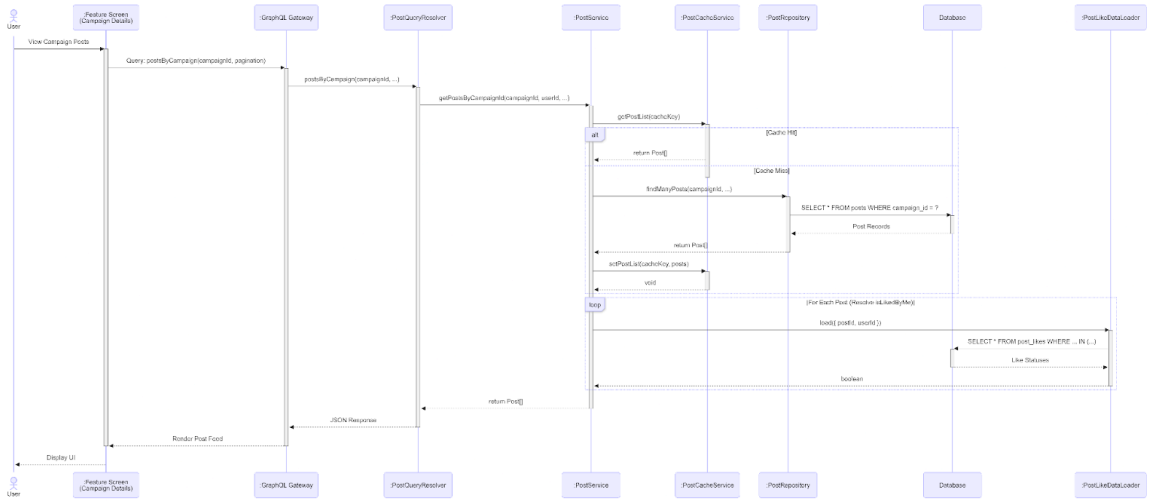


Figure 73. Get Posts by Campaign Sequence Diagram

3.11.5 Update Post Sequence Diagram

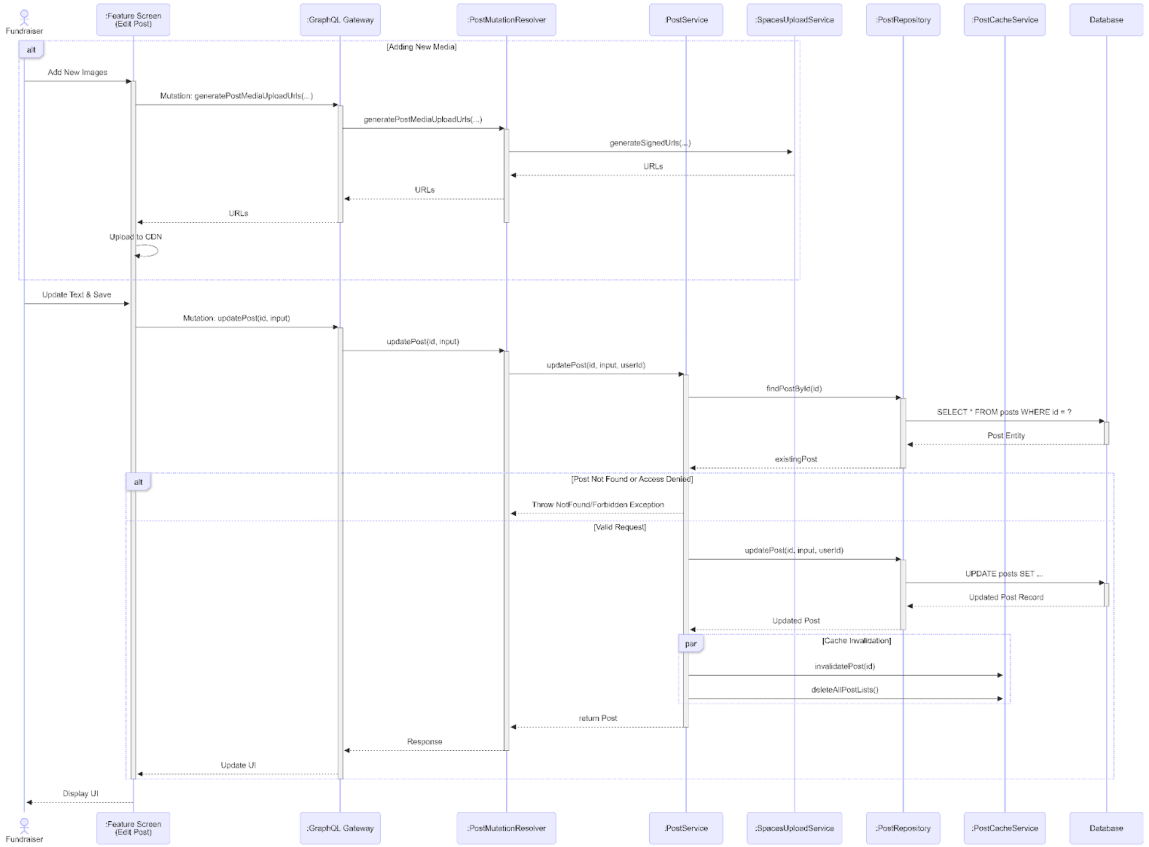


Figure 74. Update Post Sequence Diagram

3.11.6 Delete Post Sequence Diagram

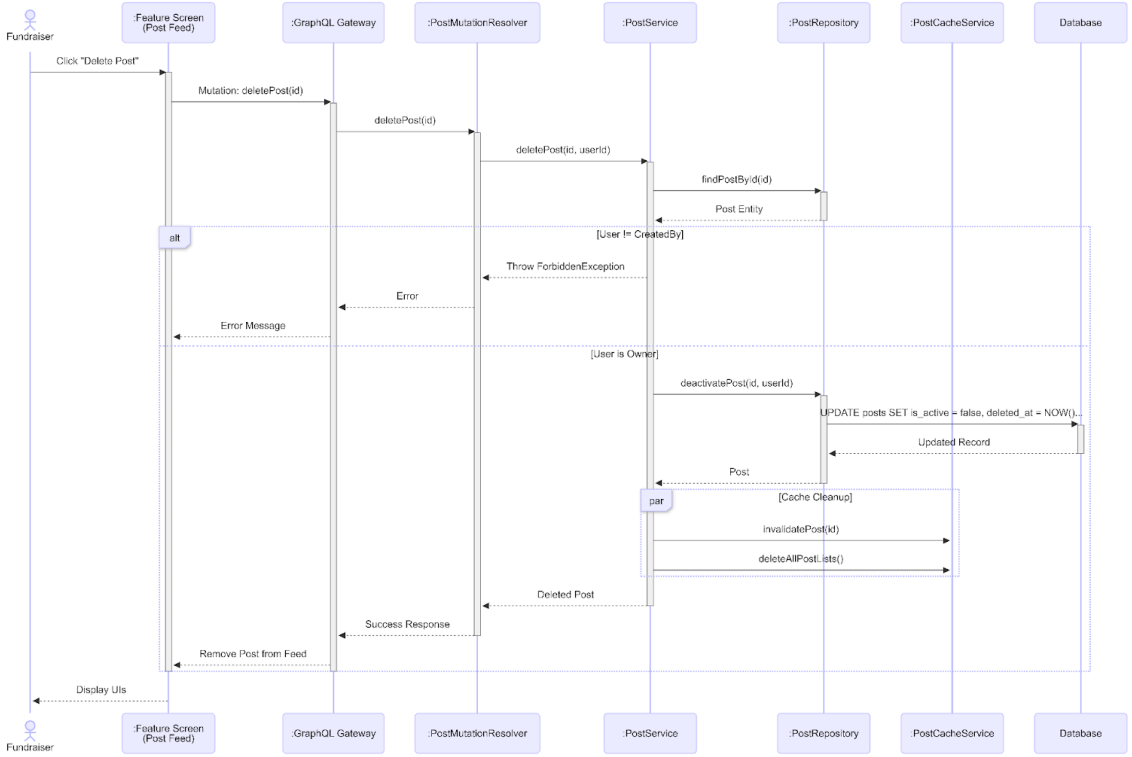


Figure 75. Delete Post Sequence Diagram

3.12 Post Like Management

3.12.1 Class Diagram

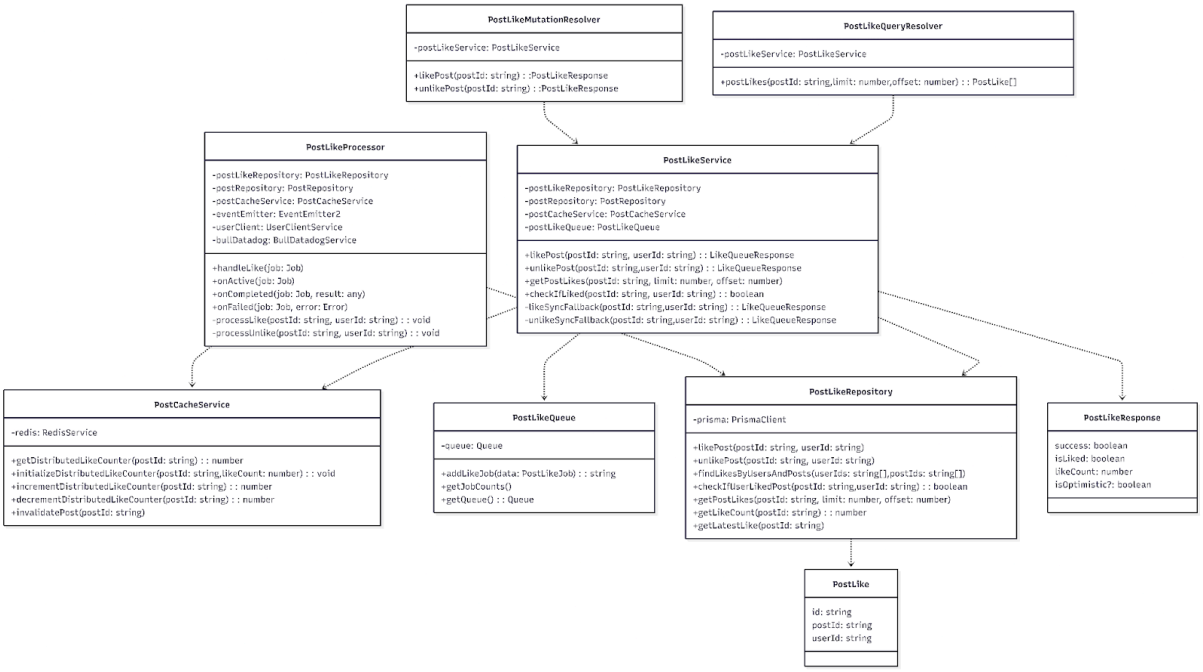


Figure 76. Post Like Management Class Diagram

3.12.2 Class Specifications

PostLikeQueryResolver

No	Method	Description
1	**postLikes(postId, limit, offset)**	Retrieves a paginated list of PostLike objects for a specific post. It delegates the data retrieval to the PostLikeService.

Table 60. PostLikeQueryResolver Class Specification

PostLikeMutationResolver

No	Method	Description
1	**likePost(postId)**	Initiates the workflow for a user to like a specific post. It calls the service layer to handle the operation asynchronously or synchronously and returns a PostLikeResponse.
2	**unlikePost(postId)**	Initiates the workflow for a user to remove a like from a specific post. It delegates to the service layer and returns the updated state in a PostLikeResponse.

Table 61. PostLikeMutationResolver Class Specification

3.12.3 Like Post Sequence Diagram

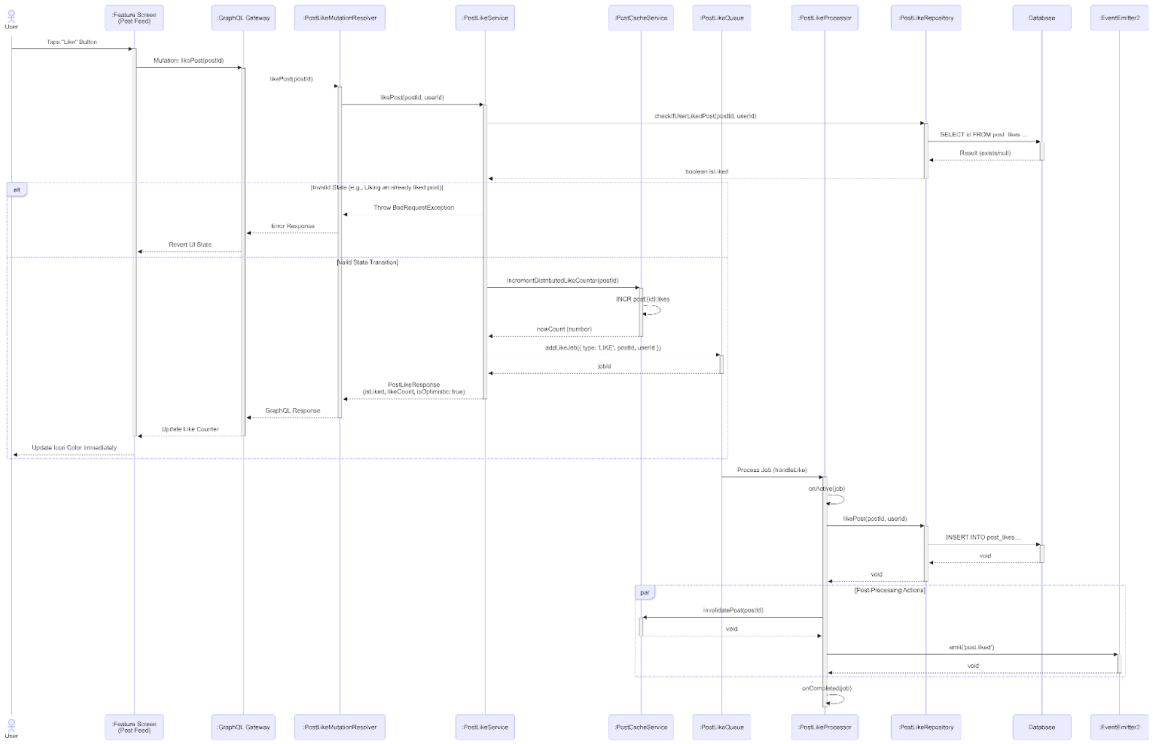


Figure 77. Like Post Sequence Diagram

3.12.4 Unlike Post Sequence Diagram

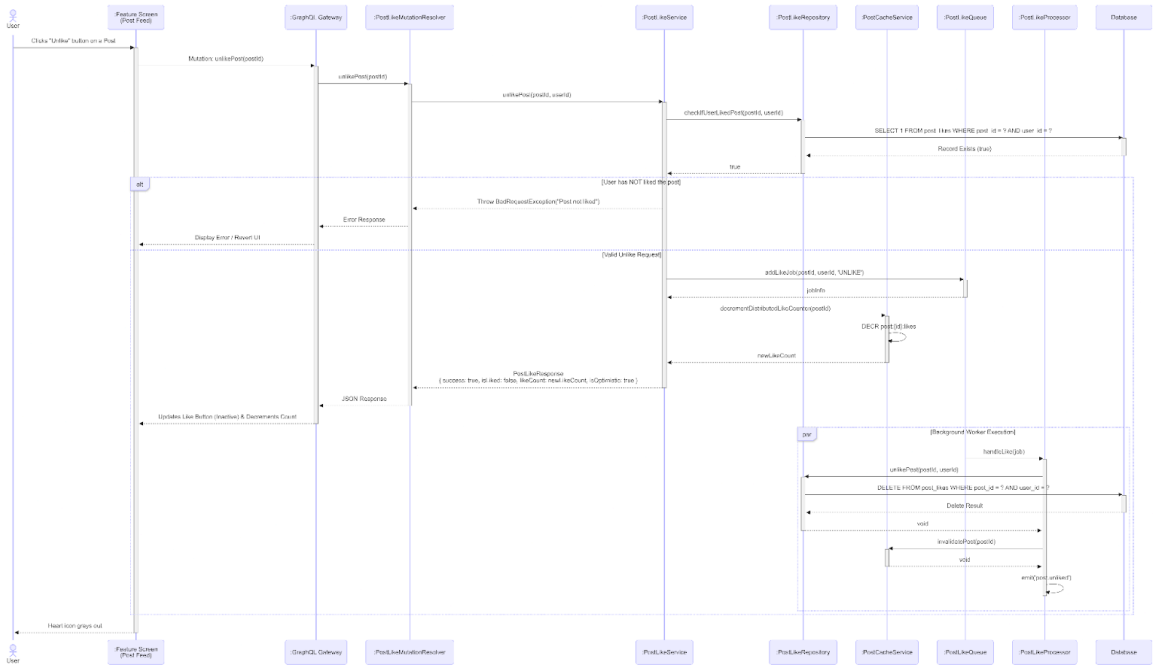


Figure 78. Unlike Post Sequence Diagram

3.12.5 View Post Likers Sequence Diagram

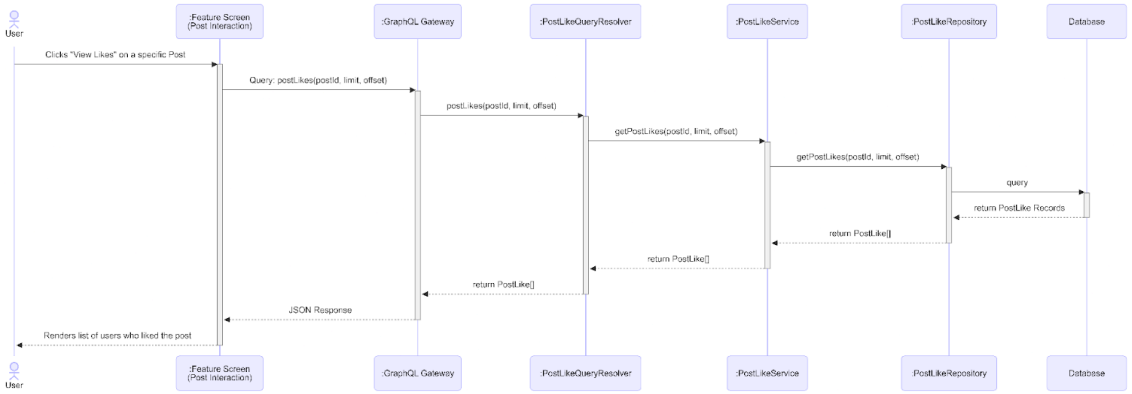


Figure 79. View Post Likers Sequence Diagram

3.13 Post Comment Management

3.13.1 Class Diagram

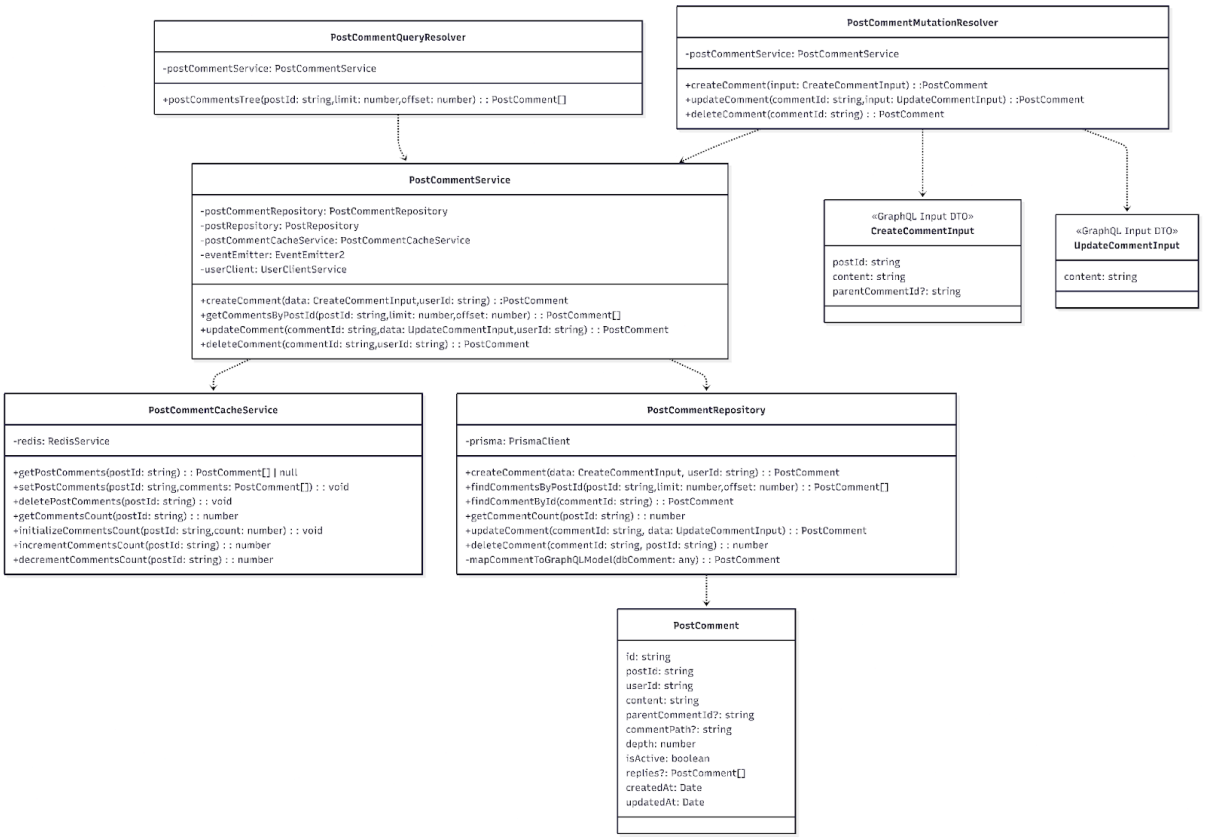


Figure 80. Post Comment Management Class Diagram

3.13.2 Class Specifications

PostCommentQueryResolver

No	Method	Description
1	**postCommentsTree(postId, limit, offset)**	Retrieves a paginated list of comments associated with a specific post. It delegates the execution to the service layer to reconstruct the hierarchical structure (threading) of comments and replies based on the provided pagination parameters.

Table 62. PostCommentQueryResolver Class Specification

PostCommentMutationResolver

No	Method	Description
1	**createComment(input)**	Facilitates the creation of a new comment or a reply to an existing comment. It accepts a CreateCommentInput object containing the post ID, content, and optional parent comment ID, then delegates the creation logic to the PostCommentService.
2	**updateComment(commentId, input)**	Handles requests to modify the content of an existing comment. It accepts the unique identifier of the comment and an UpdateCommentInput object, delegating the update operation to the service layer for validation and persistence.

No	Method	Description
3	**deleteComment(commentId)**	Processes a request to remove a specific comment from the system. It delegates the deletion logic (typically a soft-delete or deactivation) to the PostCommentService using the provided comment identifier.

Table 63. PostCommentMutationResolver Class Specification

3.13.3 Create Comment Sequence Diagram

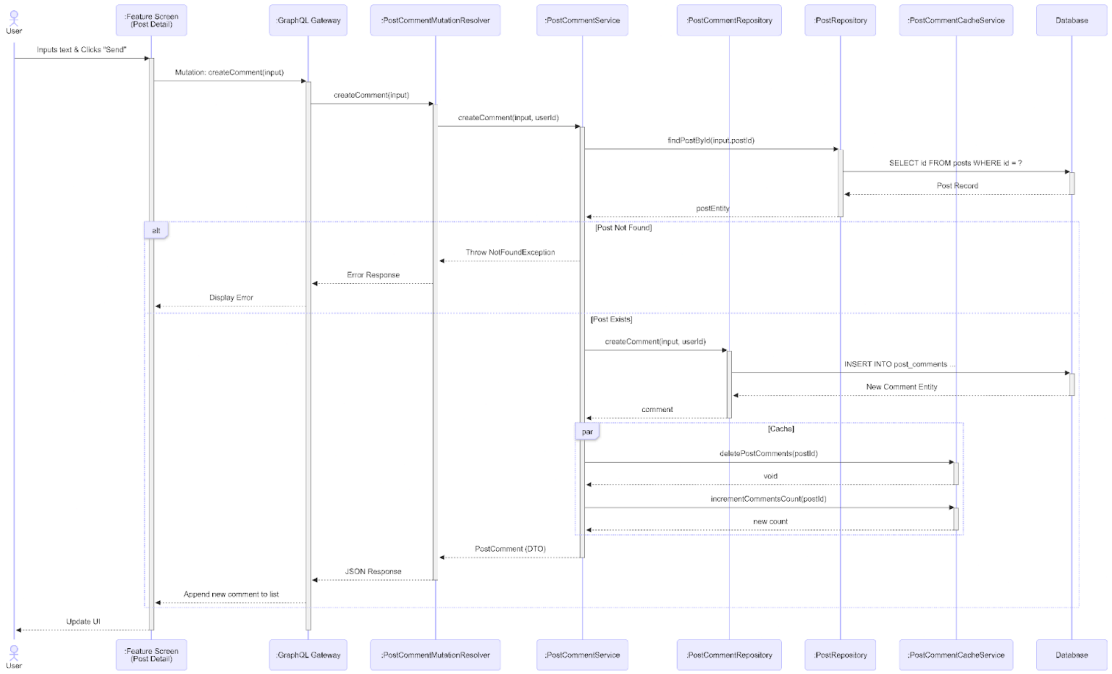


Figure 81. Create Comment Sequence Diagram

3.13.4 Get Post Comments Sequence Diagram

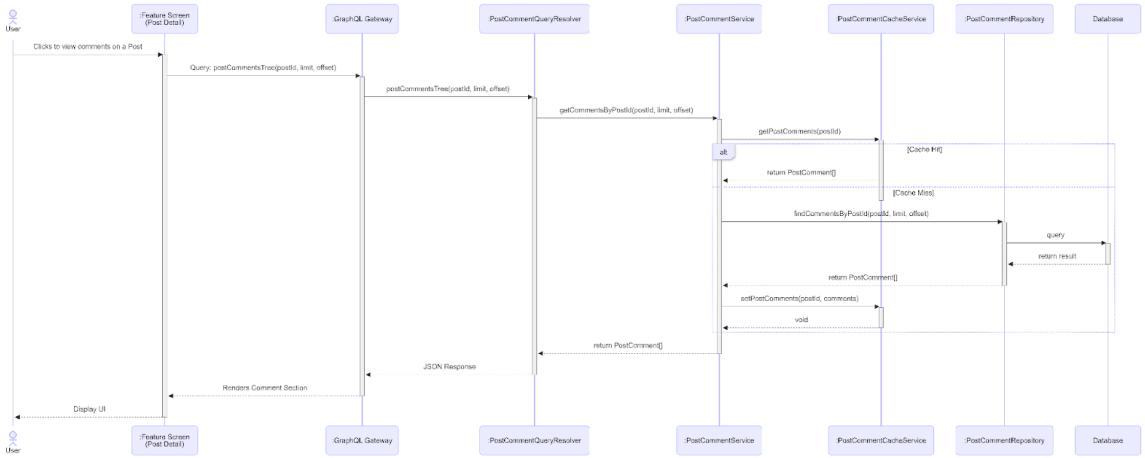


Figure 82. Get Post Comments Sequence Diagram

3.13.5 Update Comment Sequence Diagram

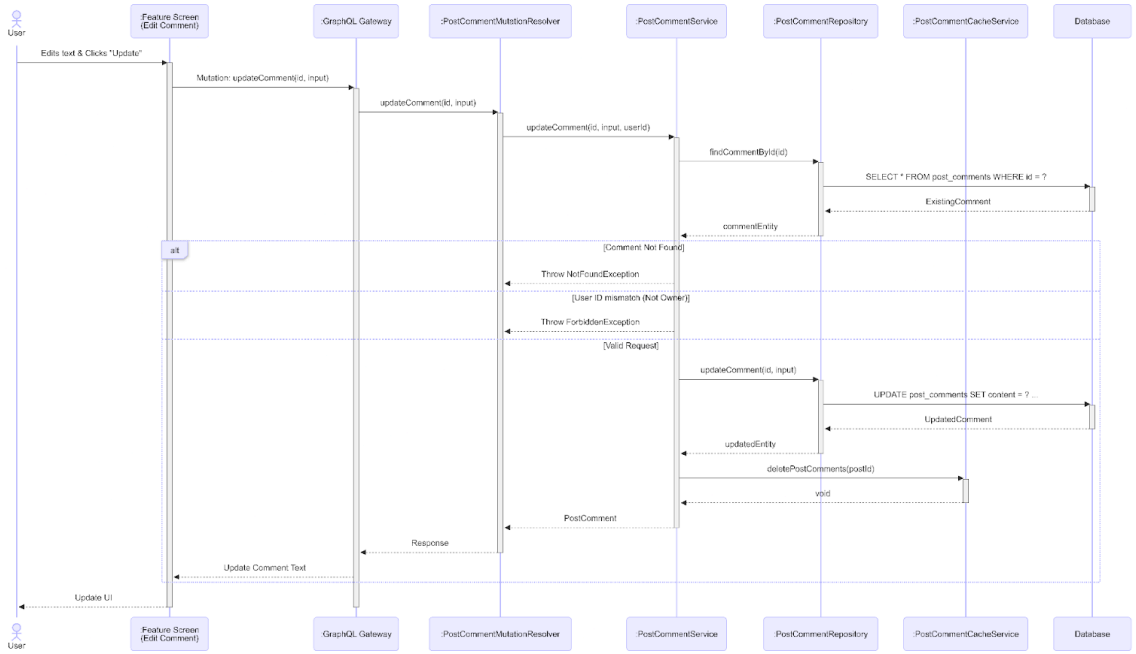


Figure 83. Update Comment Sequence Diagram

3.13.6 Delete Comment Sequence Diagram

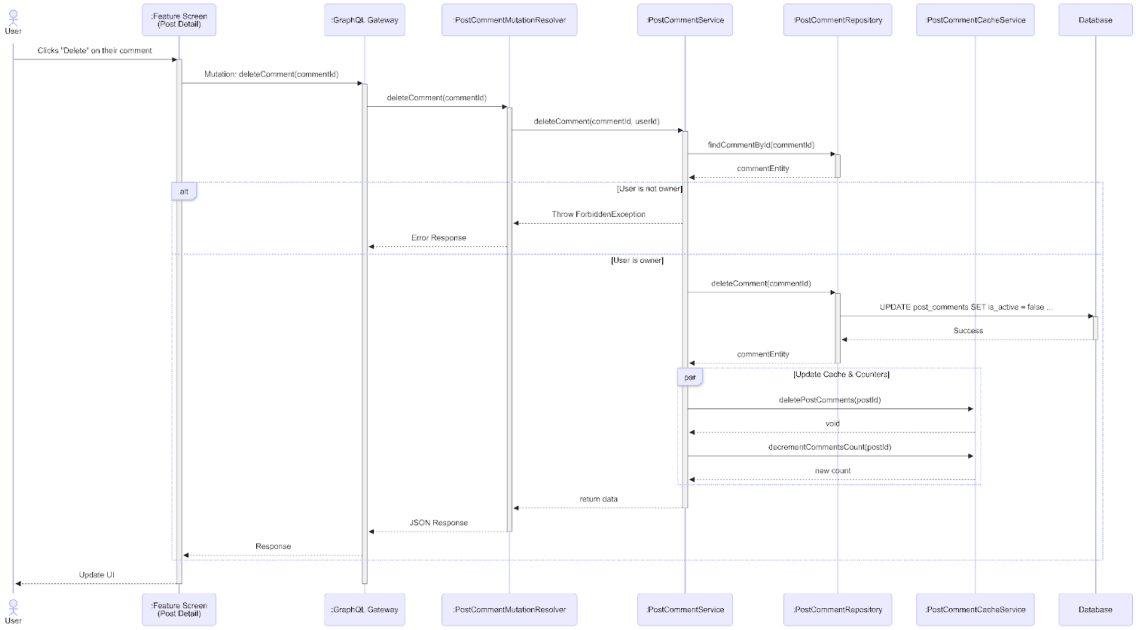


Figure 84. Delete Comment Sequence Diagram

3.14 Expense Proof Management

3.14.1 Class Diagram

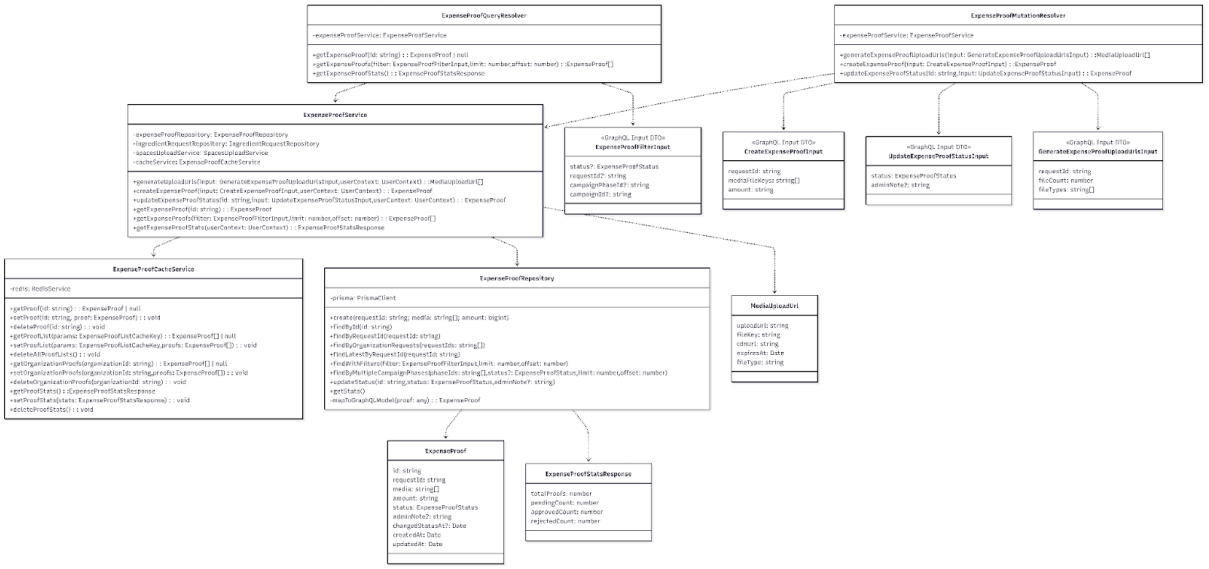


Figure 85. Expense Proof Management Class Diagram

3.14.2 Class Specifications

ExpenseProofQueryResolver

No	Method	Description
1	**getExpenseProof(id)**	Retrieves the details of a specific expense proof entity identified by its unique identifier. This method delegates the retrieval logic to the service layer to ensure the record exists and is accessible.
2	**getExpenseProofs(filter, limit, offset)**	Retrieves a paginated collection of expense proof entities based on specified filtering criteria, such as status, request ID, or associated campaign phase. It supports pagination via limit and offset parameters to manage large datasets efficiently.
3	**getExpenseProofStats()**	Aggregates and retrieves statistical data regarding expense proofs within the system or context. It provides a summary of total, pending, approved, and rejected counts, typically utilized for administrative dashboards.

Table 64. ExpenseProofQueryResolver Class Specification

ExpenseProofMutationResolver

No	Method	Description
1	**generateExpenseProofUploadUrls(input)**	Generates pre-signed upload URLs for secure object storage interaction. This operation delegates to the service layer to provision temporary access credentials, allowing clients to upload evidence files (images/documents) directly to the CDN/Storage provider before creating the record.

No	Method	Description
2	**createExpenseProof(input)**	Persists a new expense proof record in the system. It associates the previously uploaded media file keys and the declared monetary amount with a specific operation request, formally initiating the auditing process.
3	**updateExpenseProofStatus(id, input)**	Modifies the administrative status of an existing expense proof (e.g., transitioning from PENDING to APPROVED or REJECTED). This mutation allows administrators to adjudicate the validity of the submitted proof and attach optional review notes.

Table 65. ExpenseProofMutationResolver Class Specification

3.14.3 Create Expense Proof Sequence Diagram

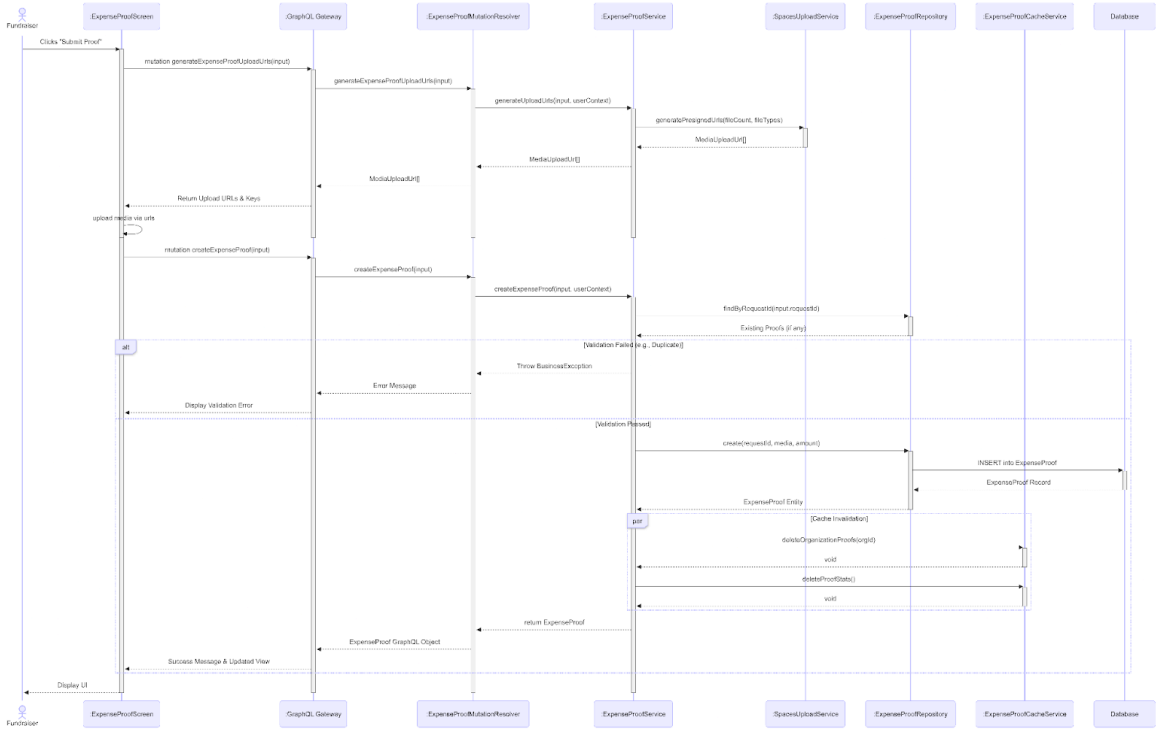


Figure 86. Create Expense Proof Sequence Diagram

3.14.4 Update Expense Proof Status Sequence Diagram

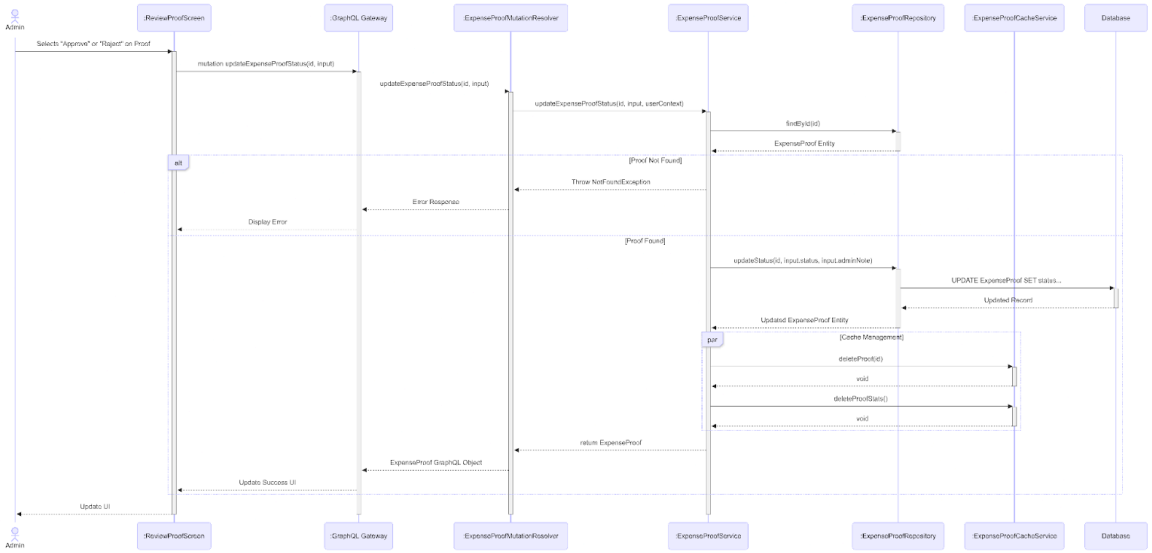


Figure 87. Update Expense Proof Status Sequence Diagram

3.14.5 View Expense Proof List Sequence Diagram

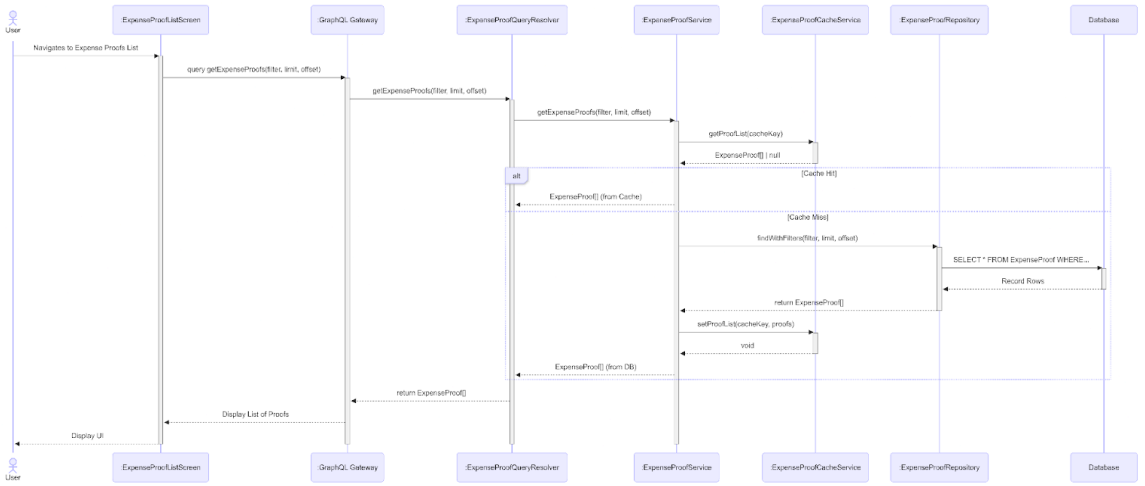


Figure 88. View Expense Proof List Sequence Diagram

3.14.6 View Detail Expense Proof Sequence Diagram

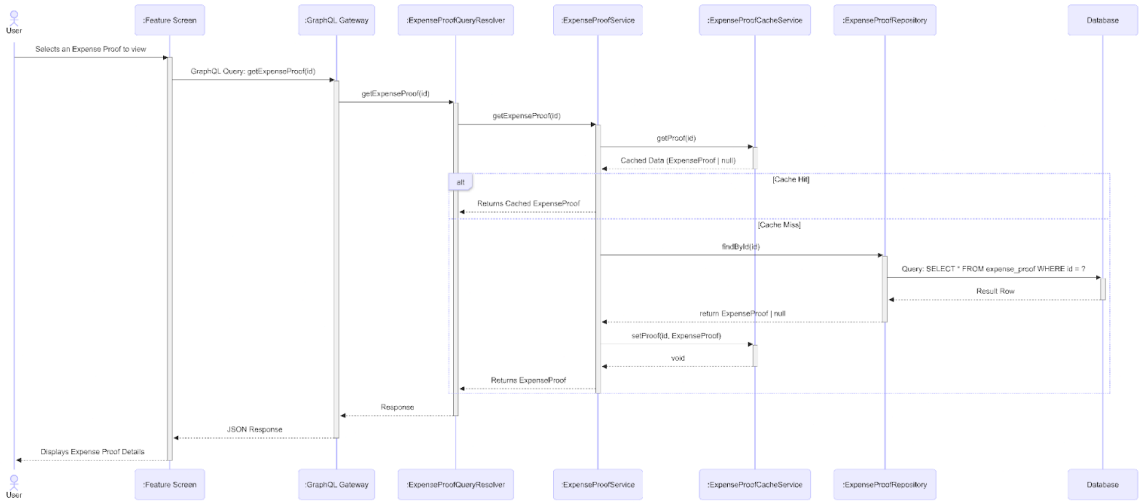


Figure 89. View Detail Expense Proof Sequence Diagram

3.14.7 View Expense Proof Statistics Sequence Diagram

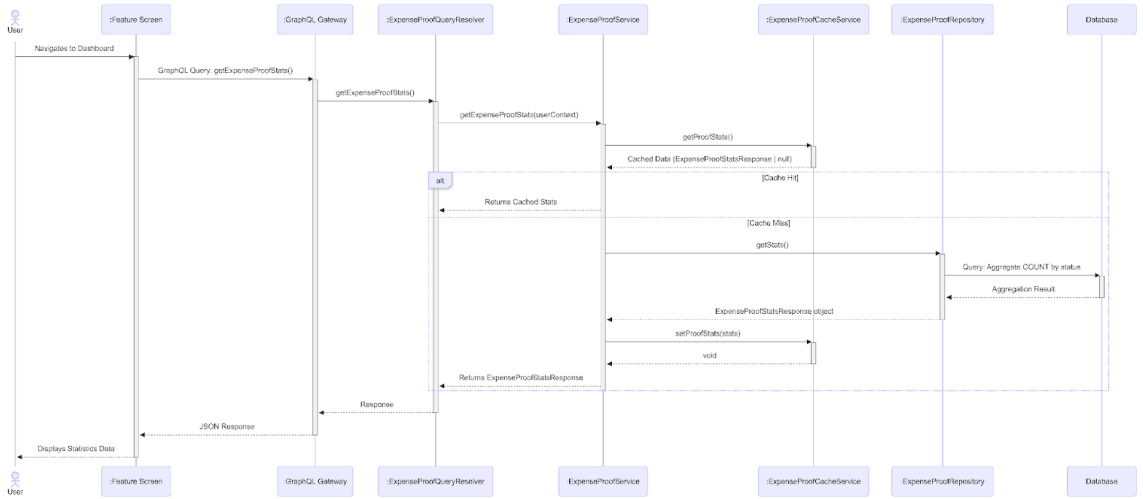


Figure 90. View Expense Proof Statistics Sequence Diagram

3.15 Meal Batch Management

3.15.1 Class Diagram

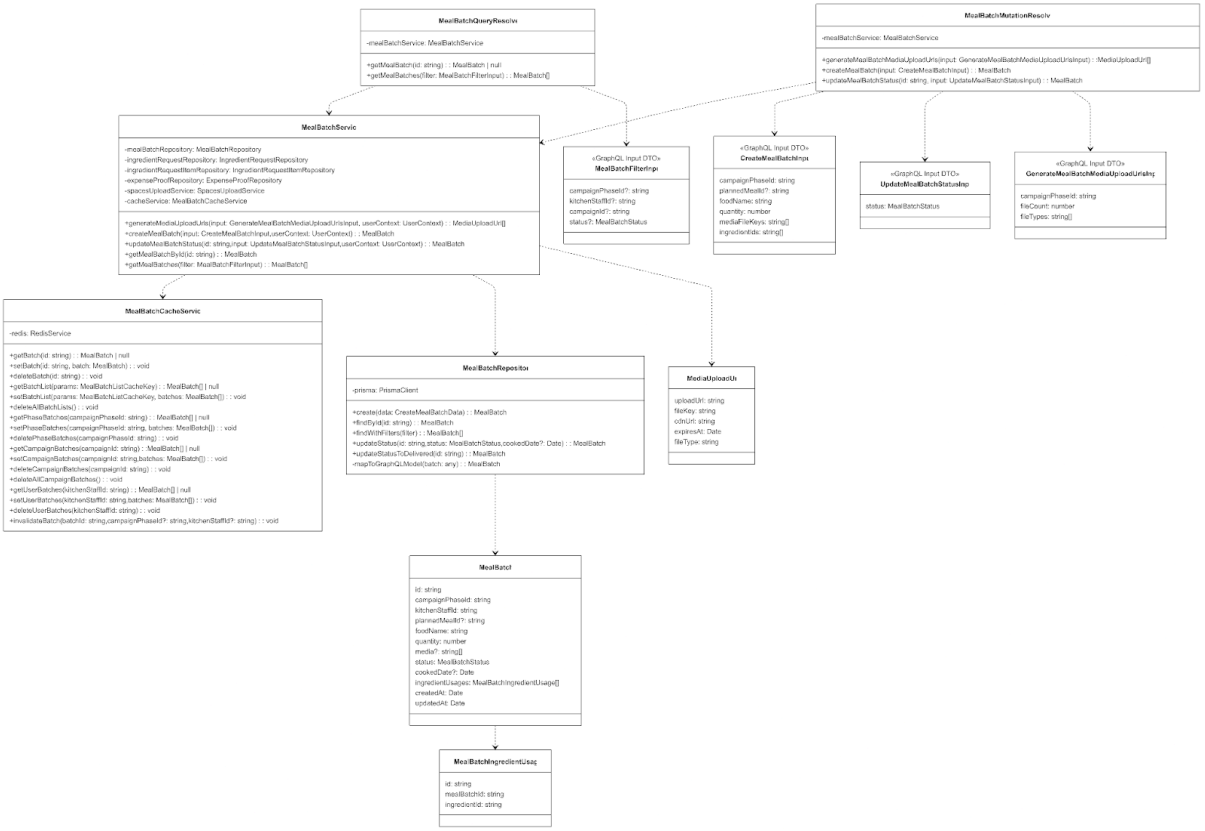


Figure 91. Meal Batch Class Diagram

3.15.2 Class Specifications

MealBatchQueryResolver

No	Method	Description
1	**getMealBatch(id: string)**	Retrieves the detailed information of a specific meal batch identified by its unique identifier. This method delegates the retrieval logic to the service layer and returns null if the record does not exist.
2	**getMealBatches(filter: MealBatchFilterInput)**	Retrieves a list of meal batches that match the criteria specified in the filter object. Filtering options may include the campaign phase, assigned kitchen staff, campaign ID, or the current status of the batch.

Table 66. MealBatchQueryResolver Class Specification

MealBatchMutationResolver

No	Method	Description
1	**generateMealBatchMediaUploadUrls(input)**	Generates and returns a list of pre-signed URLs to facilitate the secure upload of media files (e.g., images or videos) associated with a meal batch. This is typically invoked prior to creating a batch record to ensure media assets are staged in the storage service.
2	**createMealBatch(input)**	Initiates the creation of a new meal batch record within the system. This method accepts input parameters defining the food name, quantity, associated ingredients, and media keys, then delegates the creation process to the service layer.
3	**updateMealBatchStatus(id, input)**	Updates the status of an existing meal batch identified by id. This method allows for lifecycle transitions of the meal batch (e.g., marking it as cooked or delivered) based on the provided input status.

Table 67. MealBatchMutationResolver Class Specification

3.15.3 Create Meal Batch Sequence Diagram

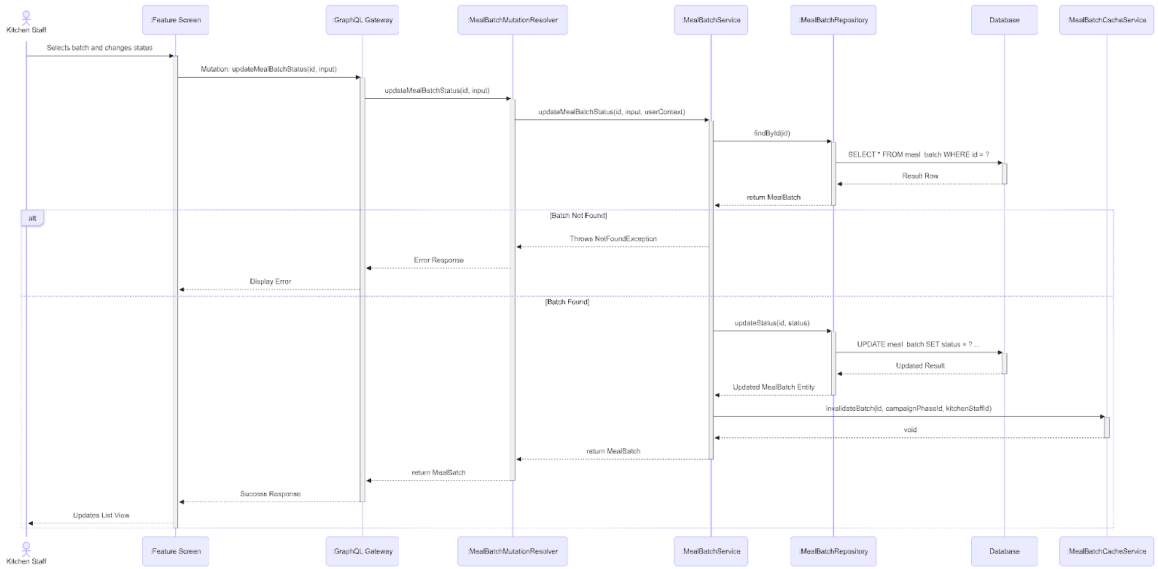


Figure 92. Create Meal Batch Sequence Diagram

3.15.4 Update Meal Batch Status Sequence Diagram

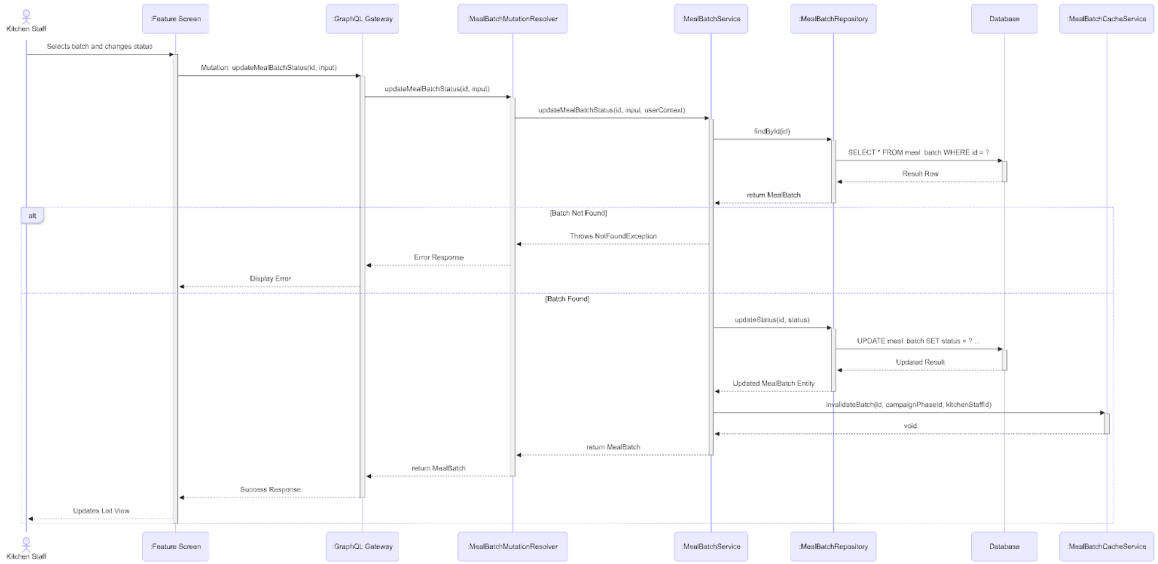


Figure 93. Update Meal Batch Status Sequence Diagram

3.15.5 View Meal Batches Sequence Diagram

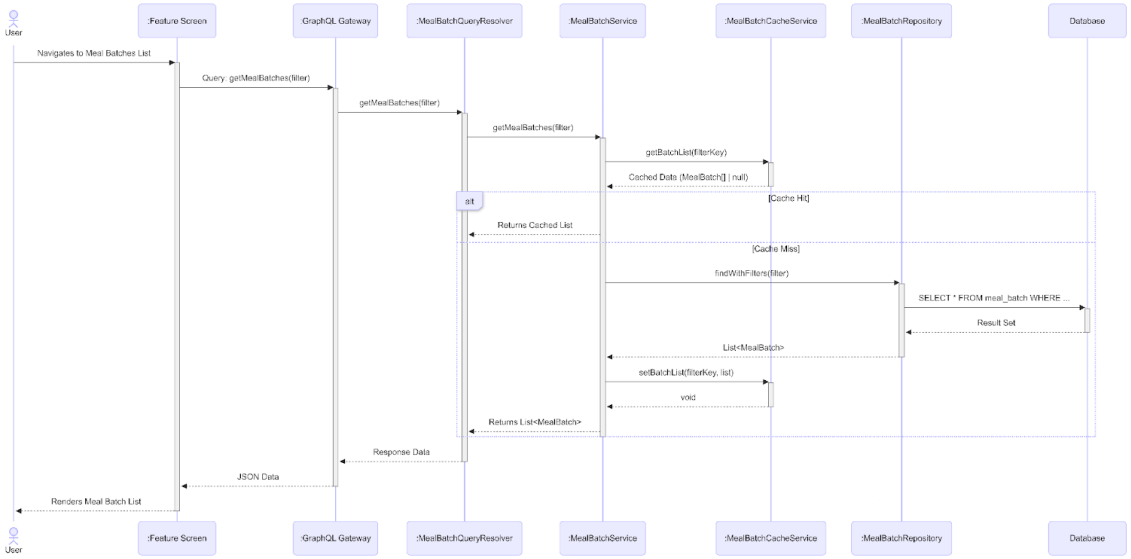


Figure 94. View Meal Batches Sequence Diagram

3.15.6 View Meal Batch Detail Sequence Diagram

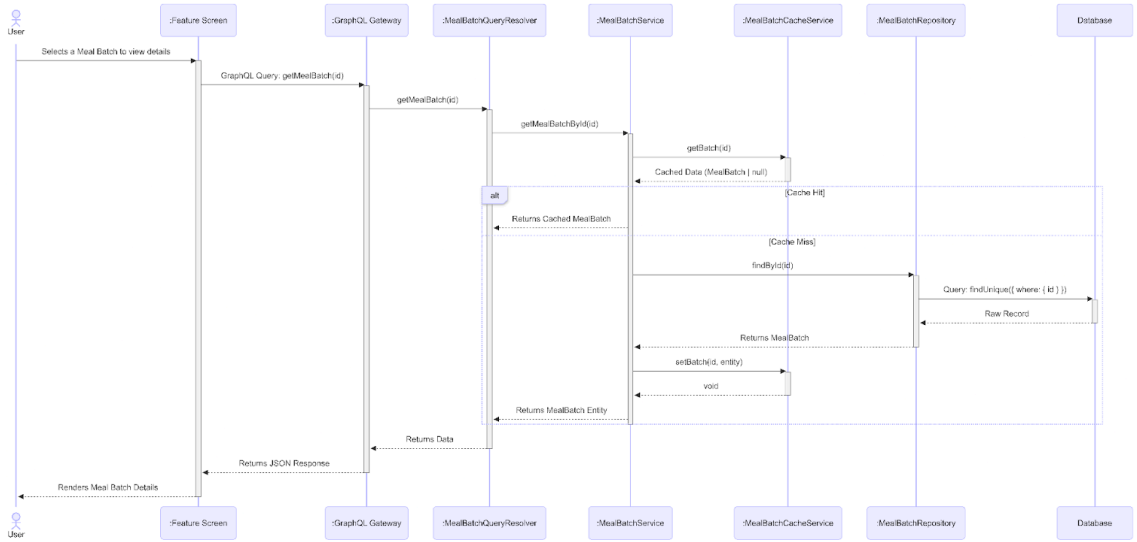


Figure 95. View Meal Batch Detail Sequence Diagram

3.15.7 Meal Batch State Diagram

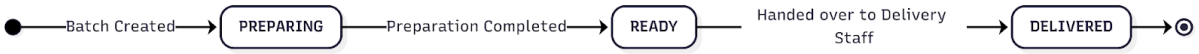


Figure 96: Meal Batch State Diagram

3.16 Delivery Task Management

3.16.1 Class Diagram

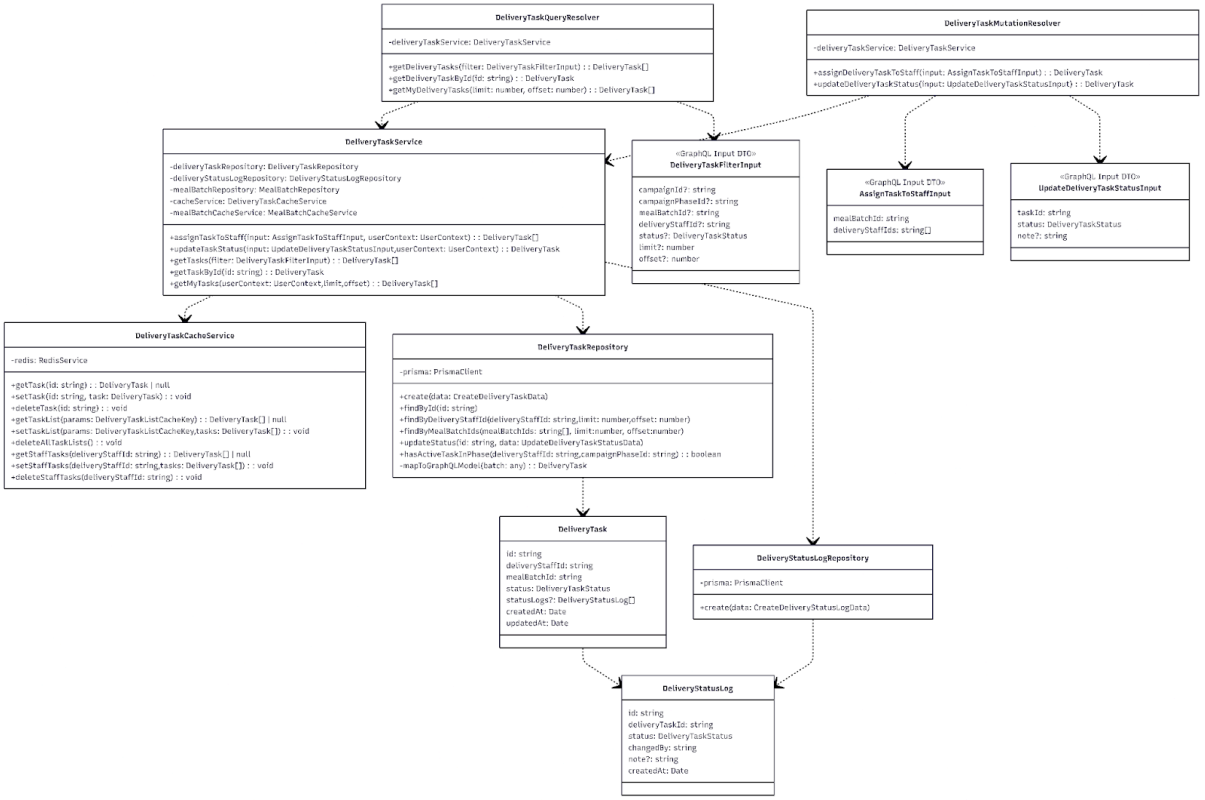


Figure 97. Delivery Task Management

3.16.2 Class Specifications

DeliveryTaskQueryResolver

No	Method	Description
1	**getDeliveryTasks(filter: DeliveryTaskFilterInput): DeliveryTask[]**	Retrieves a collection of delivery tasks that align with the criteria specified in the filter object. Filtering capabilities include parameters such as campaign ID, campaign phase ID, specific meal batch ID, delivery staff ID, and the current status of the task.
2	**getDeliveryTaskById(id: string): DeliveryTask**	Fetches the detailed entity of a specific delivery task identified by its unique identifier. This method invokes the service layer to locate the record and returns the DeliveryTask object if found.
3	**getMyDeliveryTasks(limit: number, offset: number): DeliveryTask[]**	Retrieves a paginated list of delivery tasks assigned to the currently authenticated user. This method utilizes the user context to filter tasks specific to the requester, applying the provided limit and offset for pagination.

Table 68. DeliveryTaskQueryResolver Class Specification

DeliveryTaskMutationResolver

No	Method	Description
1	**assignDeliveryTaskToStaff(input: AssignTaskToStaffInput): DeliveryTask**	Orchestrates the assignment of a meal batch to specific delivery staff members. It accepts an input object containing the target mealBatchId and a list of deliveryStaffIds. Upon successful execution by the service, it returns the resulting DeliveryTask.
2	**updateDeliveryTaskStatus(input: UpdateDeliveryTaskStatusInput): DeliveryTask**	Modifies the status of an existing delivery task based on the provided input. This method facilitates the lifecycle transitions of a task (e.g., marking as picked up or delivered) and allows for the inclusion of optional notes regarding the status change.

Table 69. DeliveryTaskMutationResolver Class Specification

3.16.3 Get Delivery Tasks Sequence Diagram

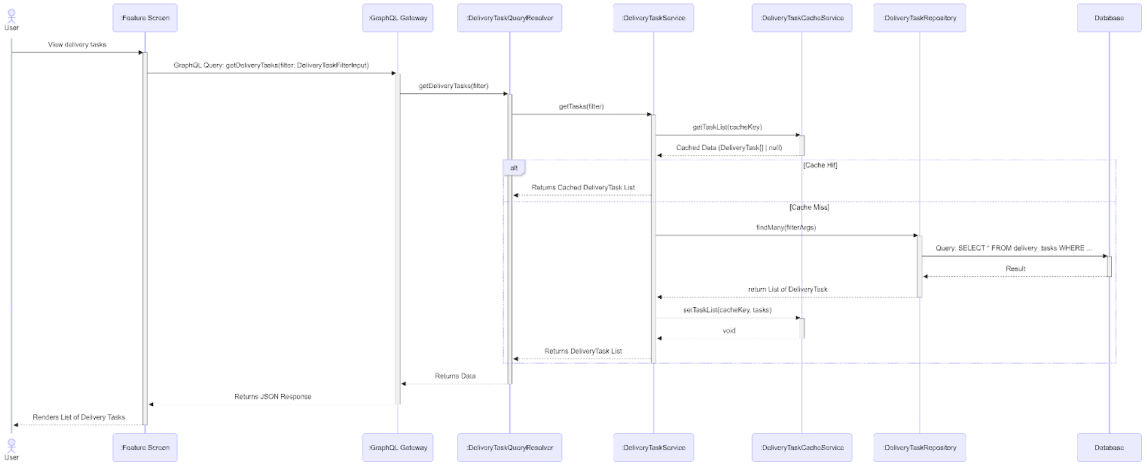


Figure 98. Get Delivery Tasks Sequence Diagram

3.16.4 Get Delivery Task Detail Sequence Diagram

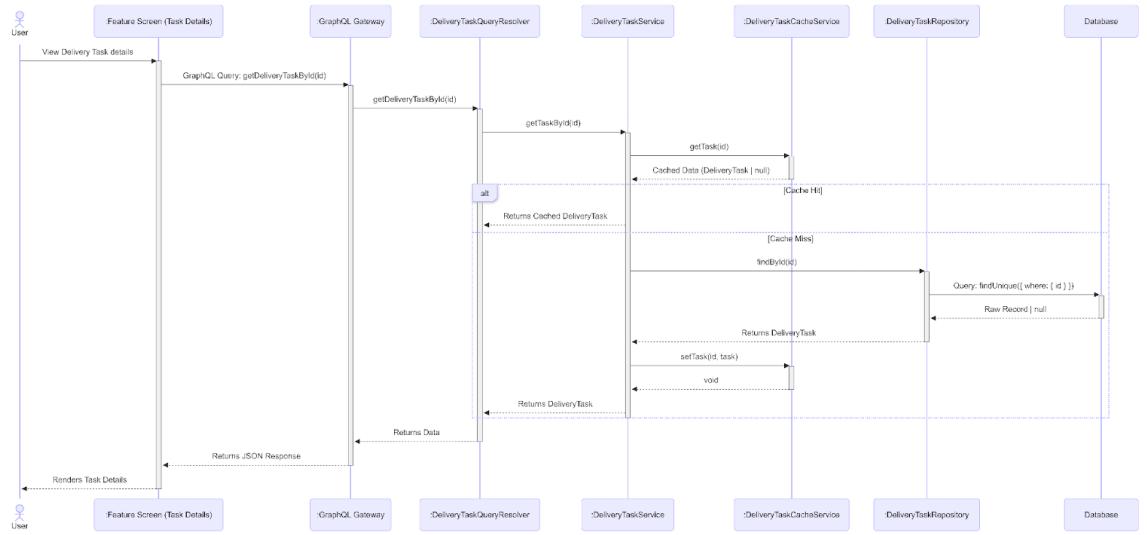


Figure 99. Get Delivery Task Detail Sequence Diagram

3.16.5 Assign Delivery Task to Staff Sequence Diagram

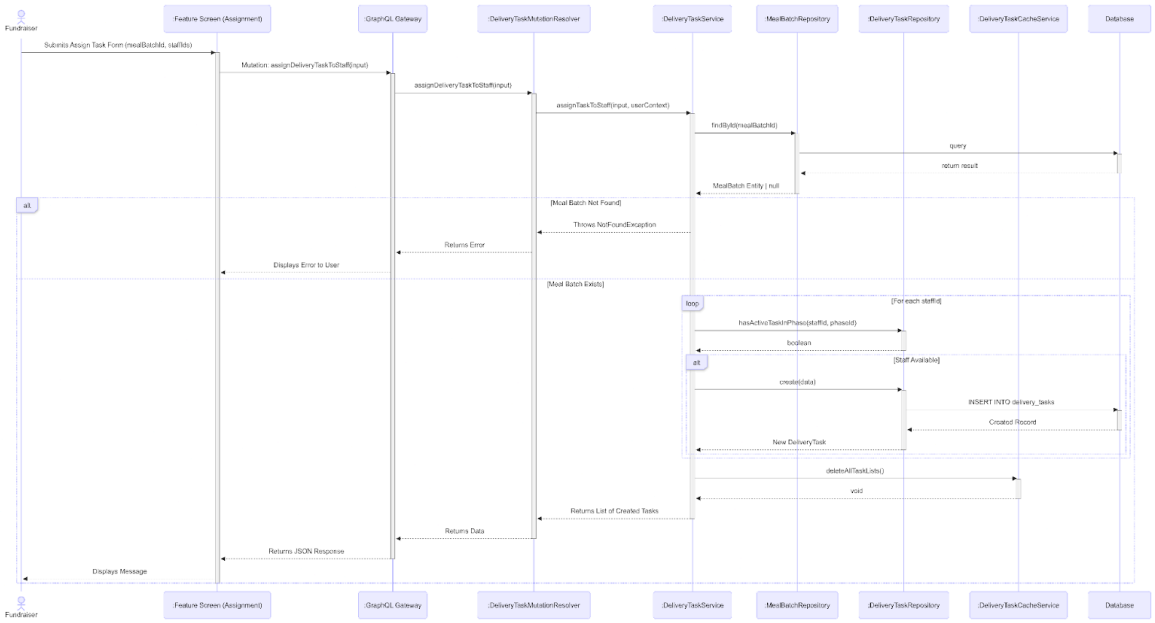


Figure 100. Assign Delivery Task to Staff Sequence Diagram

3.16.6 Update Delivery Task Status Sequence Diagram

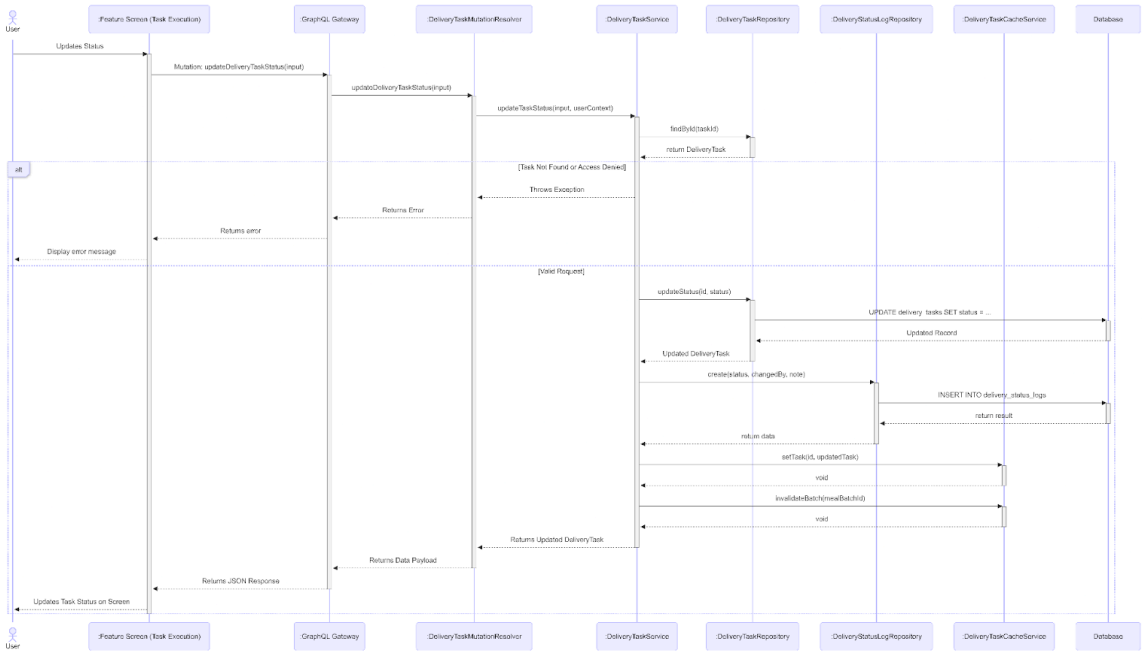


Figure 101. Update Delivery Task Status Sequence Diagram

3.16.7 Get My Delivery Tasks Sequence Diagram

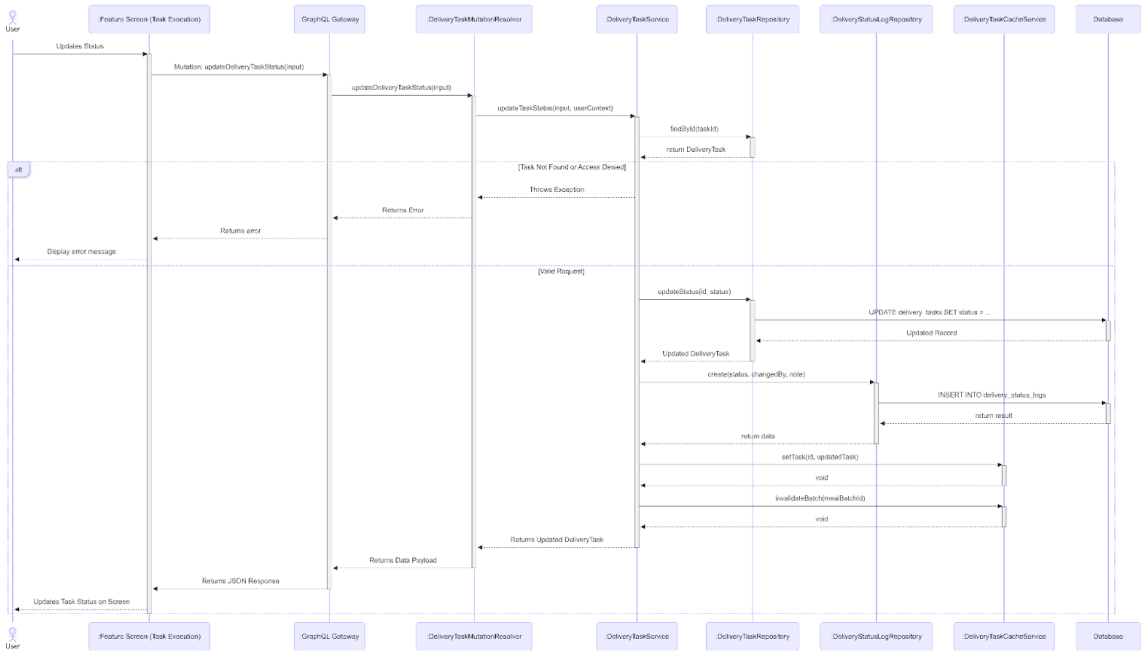


Figure 102. Get My Delivery Tasks Sequence Diagram

3.16.8 Delivery Task State Diagram

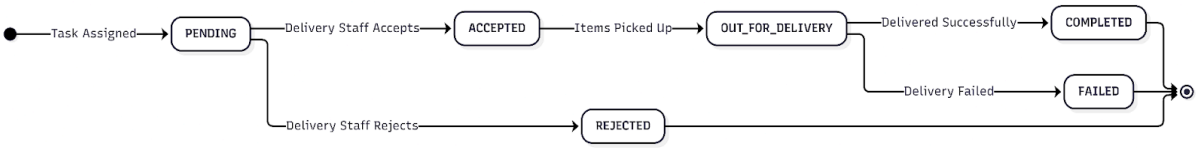


Figure 103: Delivery Task State Diagram

V. Software Testing Documentation

1. Scope of Testing

The testing will cover all critical features, functions, and non-functional requirements of FoodFund - A crowdfunding system for food preparation and distribution to support communities in need. This includes the essential features like organization management, campaign management, donation, disbursement management, among others.

Feature and Function to be tested:

- User Registration and Sign In.
- Organization Management.
- Donation and Payment Processing.
- Wallet & Transaction Management.
- Ingredient & Operation Request Management.
- Campaign Management

Feature and Function not to be tested:

- Extensive cross-browser or mobile responsiveness testing.
- Performance testing under extreme load.
- Manual accessibility testing.

2. Test Strategy

2.1 Testing Types

- **Functional Testing**
  - Objective: Verify that each function of the platform operates in conformance with the requirement specification.
  - Technique: Black-box testing will be used to evaluate the system from the end-user perspective.
  - Completion Criteria: All functional test cases must pass without any major defects.
- **User Acceptance Testing**
  - Objective: Ensure that the platform meets the business needs and is ready for production.
  - Technique: All team members will perform tasks in real-world scenarios.
  - Completion Criteria: The system must satisfy all user stories and acceptance criteria defined in the UAT plan.

2.2 Test Levels

Type of Tests	Test Level			
	Unit	Integration	System	Acceptance
Functionality	X	X	X	X
User Interface			X	X
Data Validation	X	X	X	X

Table 70. Test Levels

2.3 Supporting Tools

Purpose	Tool	Vendor/In-house	Version
Test API	Postman	Vendor (Postman, Inc.)	v11.19
Test case management	Google Sheet	Google	Cloud version(latest)

Table 71. Supporting Tools

3. Test Plan

3.1 Human Resources

Worker/Doer	Role	Specific Responsibilities/Comments
Trương Minh Phước	Leader	Planning, Unit testing, Integration testing, System testing, and Acceptance testing
Huỳnh Đình Luyện	Member	Unit testing, Integration testing, System testing, and Acceptance testing
Phan Cảnh Bảo Duy	Member	Integration testing, System testing, and Acceptance testing
Huỳnh Lê Nhật Hoàng	Member	Integration testing, System testing, and Acceptance testing

Table 72. Human Resources

3.2 Test Environment

Purpose	Tool	Provider	Version
Test case management	Google Sheet	Google	Cloud version
Version control	Github	Github	Cloud version
Development environment	Visual Studio Code	Microsoft	
Database	PostgreSQL	PostgreSQL	
Operating System	Window 11	Microsoft	
Mobile app development	Expo	Expo	

Table 73. Test Environment

3.3 Test Milestones

Milestone Task	Start Date	End Date
Test planning	13/10/2025	19/10/2025
Test case development	24/10/2025	01/11/2025
Unit testing	02/11/2025	25/11/2025
Integration testing	02/11/2025	25/11/2025
System testing	26/11/2025	01/12/2025
Testing closure and Reporting	02/12/2025	11/12/2025

Table 74. Test Milestones

4. Test Cases

- Unit Test Cases: Refer to **Report5_Test Report**
- Other Test Case (IT, ST, UAT): Refer to **Report5_Unit Test**

5. Test Reports

- **System Testing, Acceptance Testing**

TEST STATISTICS			
Project Name	*FoodFund - A crowdfunding system for food preparation and distribution to support communities in need*	**Creator**	**PhuocTM**
Project Code	*FA25SE52*	**Reviewer**	Nguyễn Minh Sang
Document Code	*FA25SE52_Test Report_v1.1*	**Issue Date**	*02/12/2025*

No	Module code	Passed	Failed	Pending	N/A	Total Test Cases
1	Authentication	23	0	0	0	23
2	Organization	20	0	0	0	20
3	Donation	19	0	0	0	19
4	Wallet	15	0	0	0	15
5	Ingredient Request	30	0	0	0	30
6	Operation Request	30	0	0	0	30
7	Meal Batch Request	27	0	0	0	27
8	Delivery Task Request	25	0	0	0	25
9	Inflow Transaction	13	0	0	0	13
10	Campaign	34	0	0	0	34
11	Notification	13	0	0	0	13
12	System Config	11	0	0	0	11
13	Expense Proof	17	0	0	0	17
14	Profile	12	0	0	0	12

No	Module code	Passed	Failed	Pending	N/A	Total Test Cases
15	Campaign Category	13	0	0	0	13
	Sub total	302	0	0	0	302
	Test coverage					100 %
	Test successful coverage					100 %

Table 75. Test Statistics Report

- **Unit Testing**

UNIT TEST REPORT			
Project Name	*FoodFund - A crowdfunding system for food preparation and distribution to support communities in need*	Creator	**PhuocTM**
Project Code	*FA25SE52*	Reviewer	Nguyễn Minh Sang
Document Code	*FA25SE52_Test Report_v1.1*	Issue Date	*02/12/2025*

No	Function code	Passed	Failed	Untested	N	A	B	Total Test Cases
1	SignUp	7	0	0	1	3	3	7
2	SignIn	8	0	0	1	5	2	8
3	ConfirmSignUp	8	0	0	1	4	3	8
4	RequestCreateOrganization	13	0	0	2	4	7	13
5	CreateCampaign	19	0	0	1	7	11	19
6	CreateDonation	16	0	0	4	4	8	16
7	CreateOperationRequest	13	0	0	2	5	6	13
8	CreateIngredientRequest	10	0	0	1	3	6	10
9	CreateMealBatch	17	0	0	2	6	9	17
10	CreateCampaignCategory	9	0	0	1	5	3	9
	Sub total	120	0	0	16	46	58	120
	Test coverage							100 %
	Test successful coverage							100 %
	Normal case							13.33 %
	Abnormal case							38.33 %
	Boundary case							48.33 %

Table 76. Unit Test Report

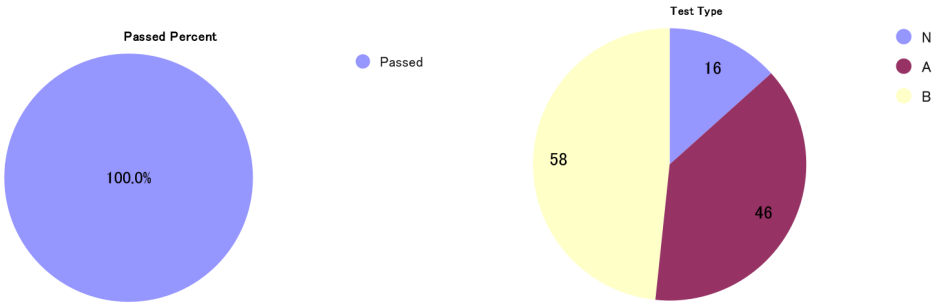


Figure 104. Test Case Report Pie Chart

VI. Release Package & User Guides

1. Deliverable Package

No.	Deliverable Item	Description
1	Project Schedule/Tracking	Report3_Project Tracking.xlsx
2	Project Backlog	Github commit logs
3	Source Codes	Frontend (ReactJS) & Backend (NestJS) repositories
4	Final Report Document	Report7_Final Project Report.docx
5	Test Cases Document	Report5_Test Report.xlsx
6	Defects List	Report3_Project Tracking.xlsx
7	Issues List	Report3_Project Tracking.xlsx
8	Slide	FoodFund_Slide.pptx

Table 77. Deliverable Package

2. Installation Guides

2.1 System Requirements

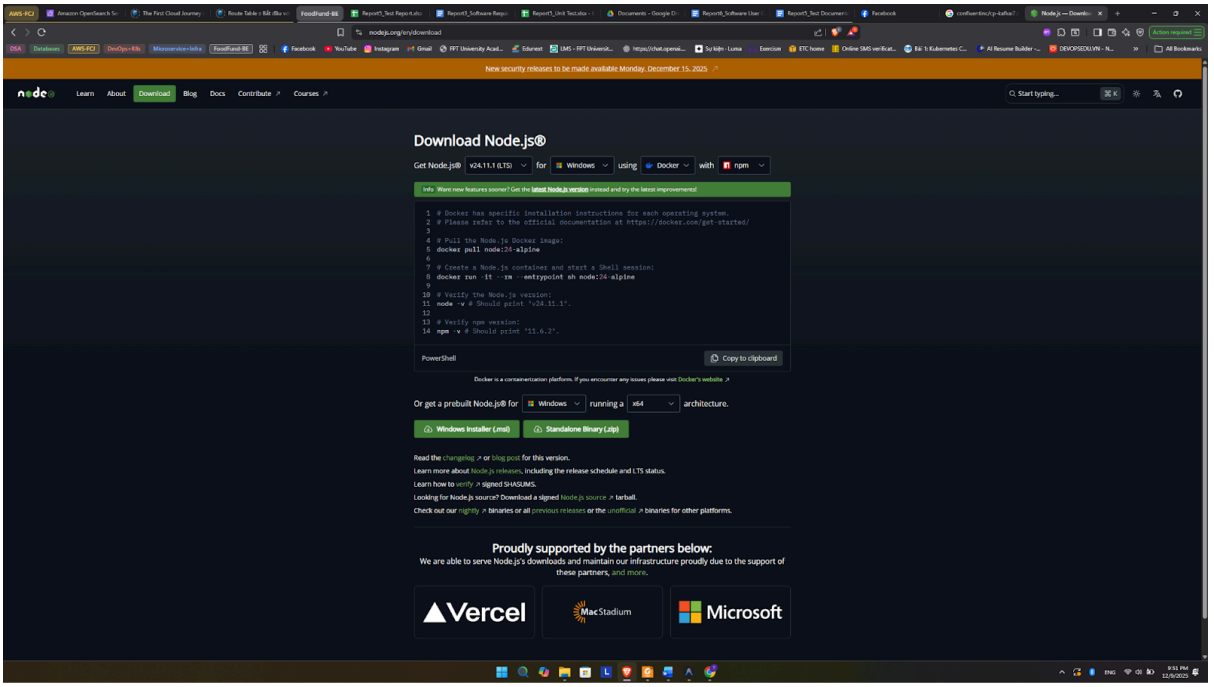
- **Operating System:** Windows 10+ or any modern Linux distribution
- **Frontend:**
  - Node.js 20.10.0+ & npm 10.2.3+
  - Visual Studio Code 2022
- **Backend:**
  - NestJS 11.1.6+
  - Visual Studio Code 2022
- **Database:**
  - PostgreSQL 17
  - OpenSearch 3.1
  - Valkey 8
- **Cloud:**
  - AWS and Digital Ocean account is configured
- **Payment:**
  - Sepay webhook defined
  - PayOS credentials
- **Browser:**
  - Chrome, Firefox, Edge (latest versions)

2.2 Installation Instruction

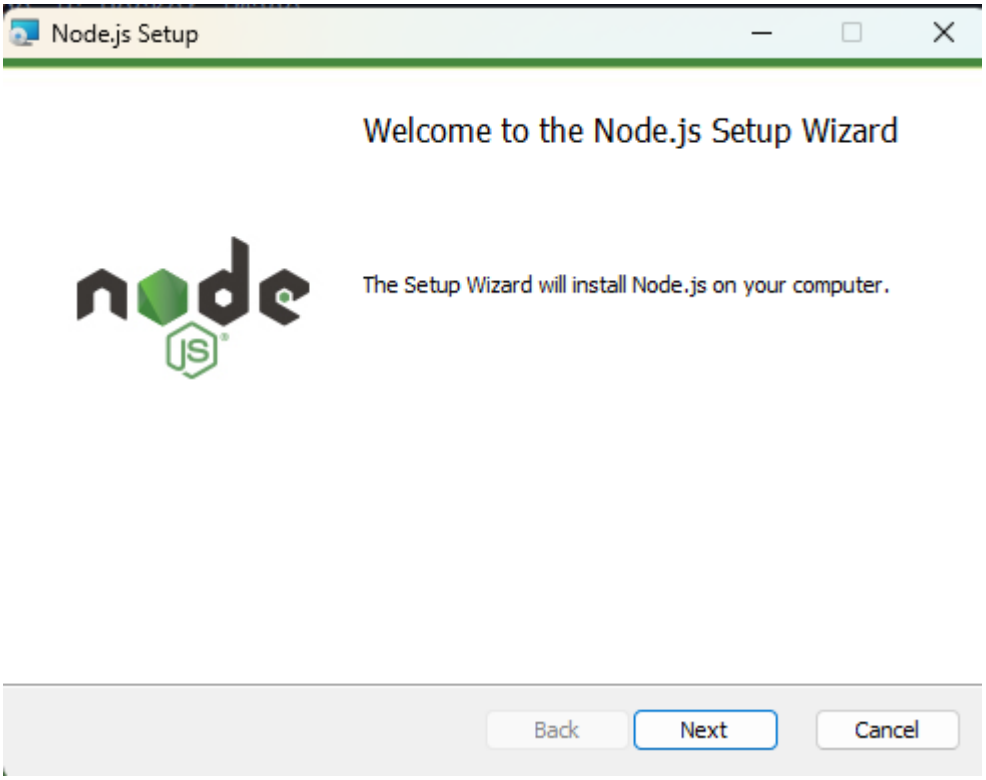
2.2.1 Environment Installation

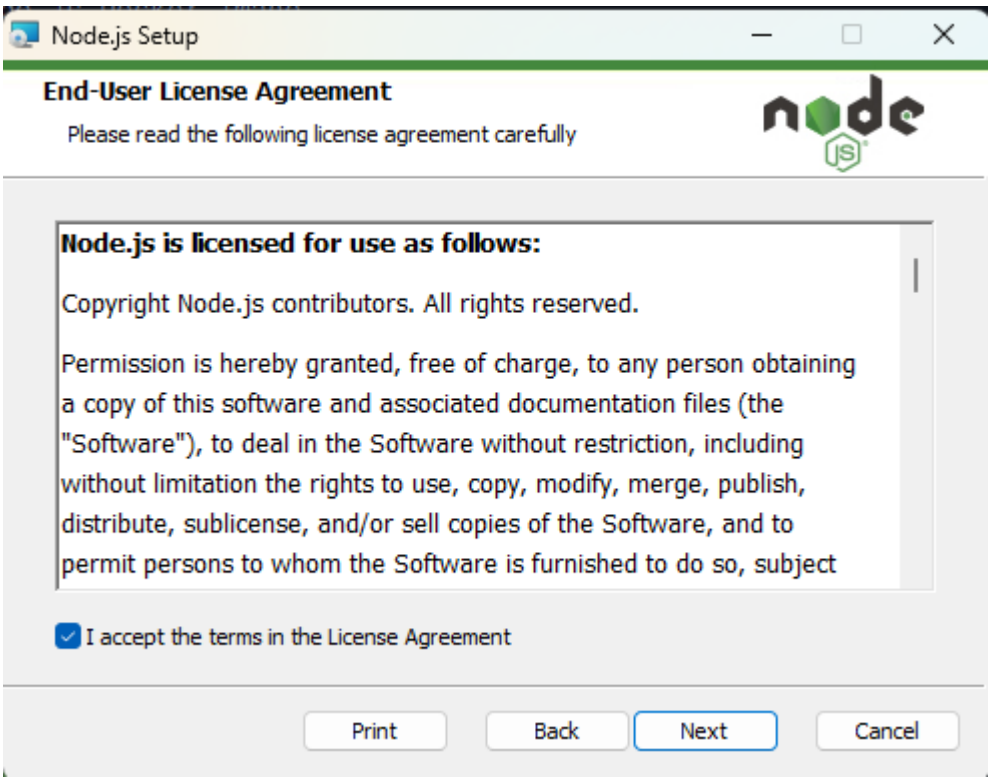
Node.js:

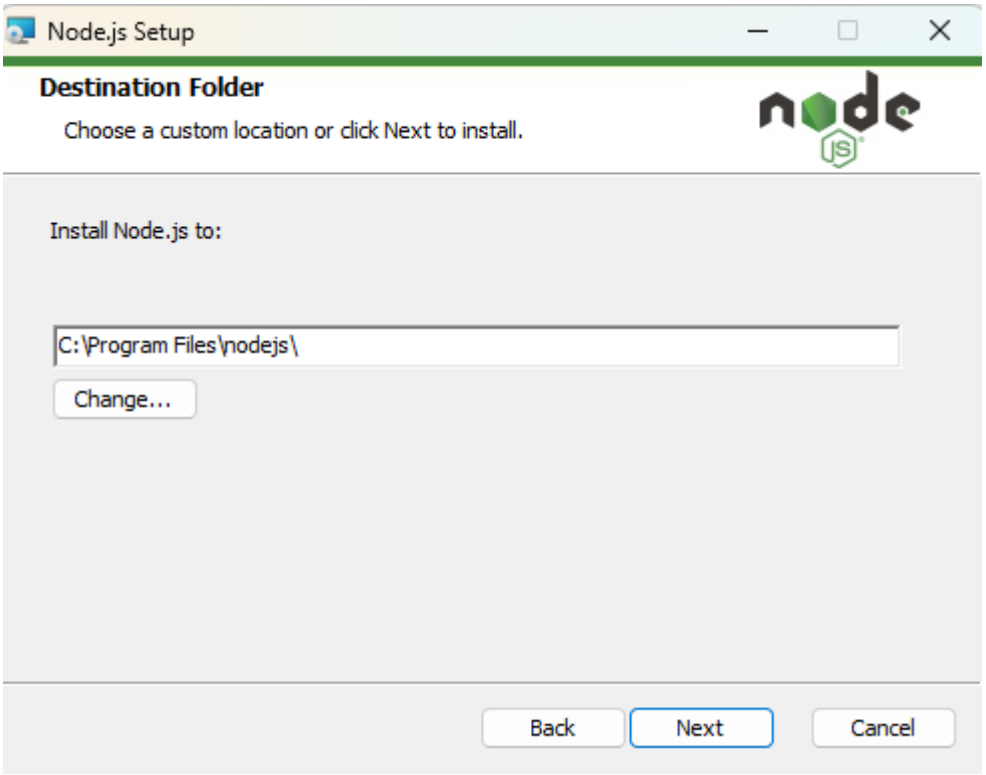
- **Step 1: Download [Node.js]**

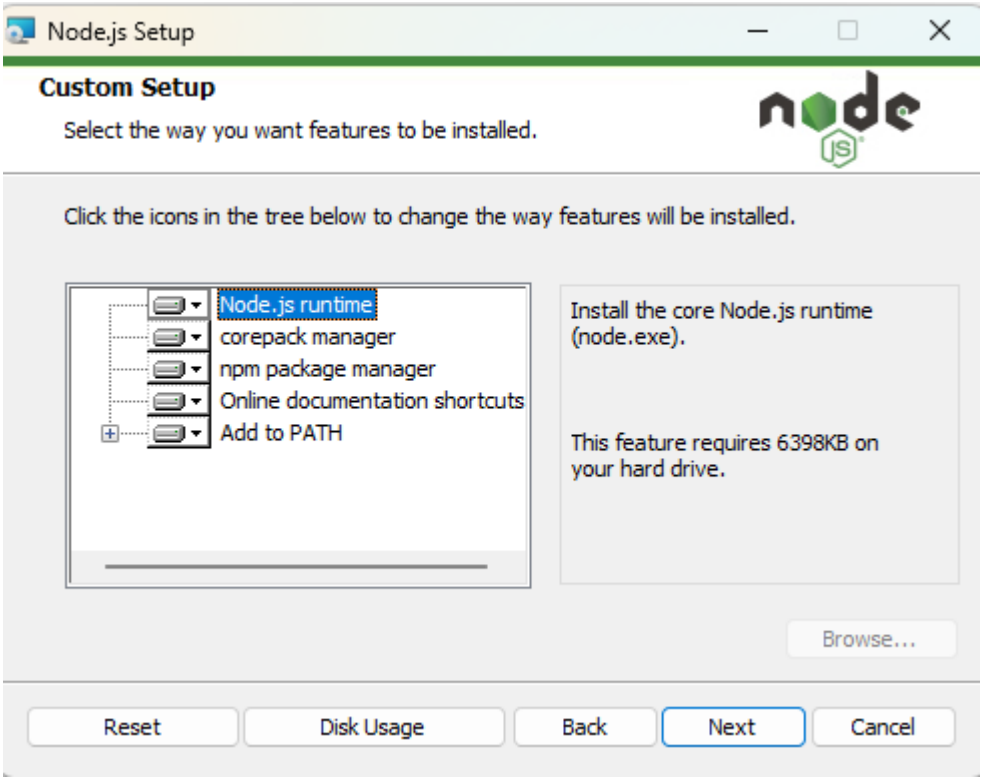


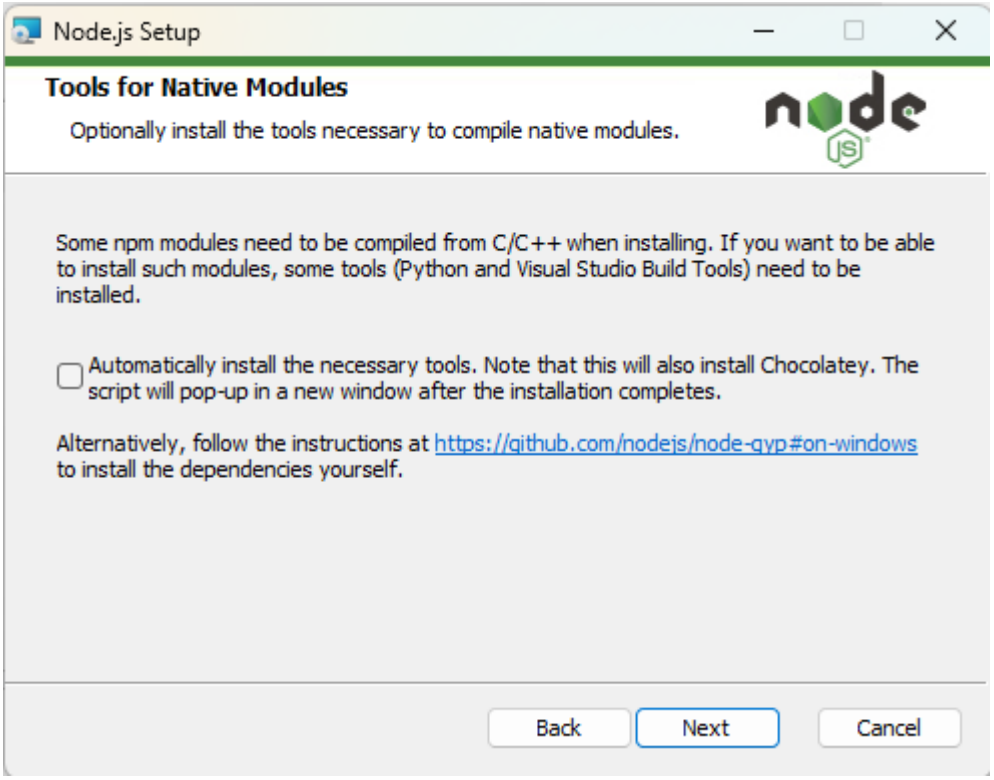
- **Step 2: Install Node.js**

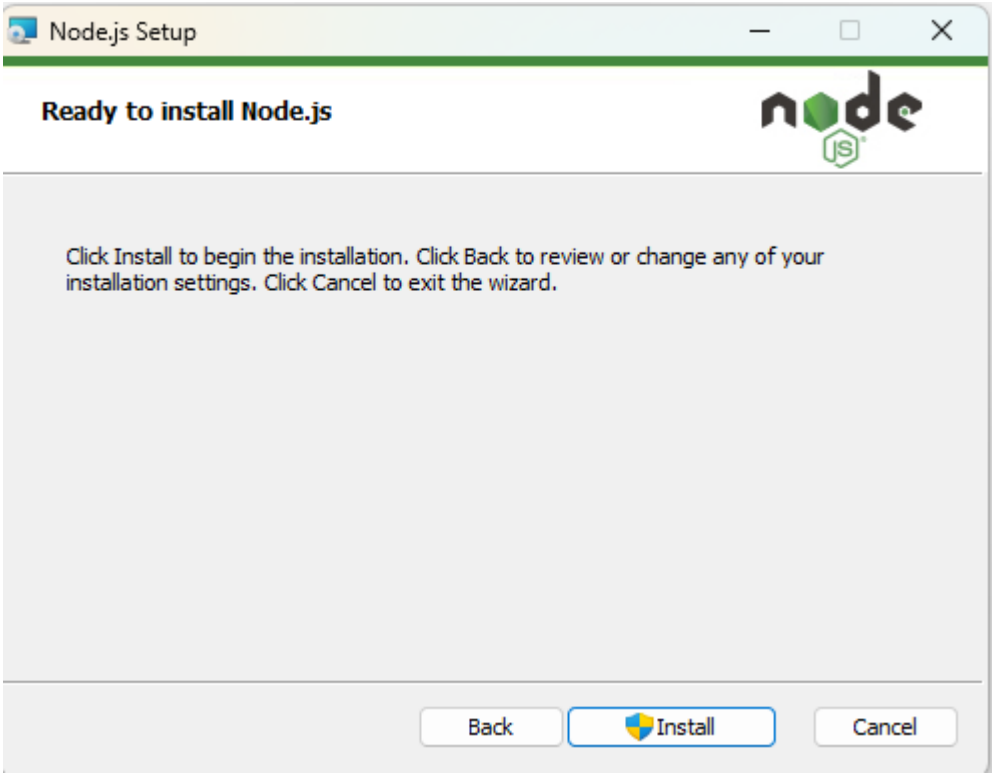



Node.js Setup
End-User License Agreement
Please read the following license agreement carefully
Node.js is licensed for use as follows:
Copyright Node.js contributors. All rights reserved.
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject
I accept the terms in the License Agreement
Print
Back
Next
Cancel


Node.js Setup
Destination Folder
Choose a custom location or click Next to install.
Install Node.js to:
C:\Program Files\nodejs\
Change...
Back
Next
Cancel

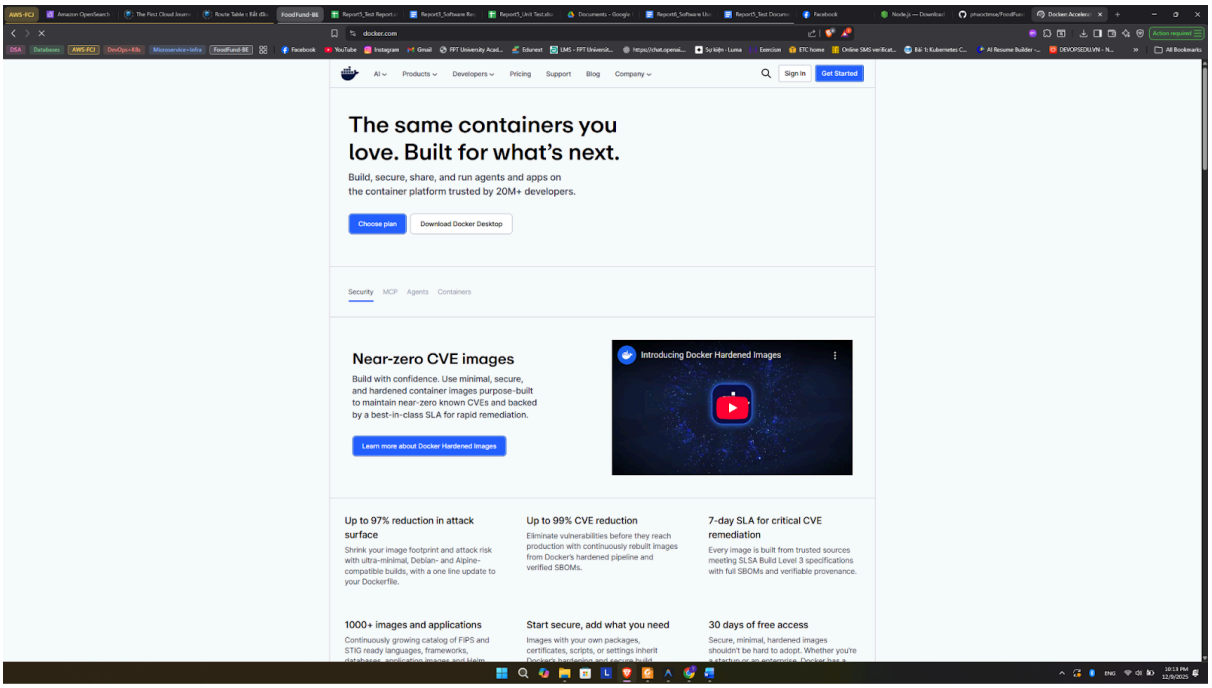

Node.js Setup
Custom Setup
Select the way you want features to be installed.
Click the icons in the tree below to change the way features will be installed.
Node.js runtime
corepack manager
npm package manager
Online documentation shortcuts
Add to PATH
Install the core Node.js runtime (node.exe).
This feature requires 6398KB on your hard drive.
Browse...
Reset
Disk Usage
Back
Next
Cancel


Node.js Setup
Tools for Native Modules
Optionally install the tools necessary to compile native modules.
Some npm modules need to be compiled from C/C++ when installing. If you want to be able to install such modules, some tools (Python and Visual Studio Build Tools) need to be installed.
Automatically install the necessary tools. Note that this will also install Chocolatey. The script will pop-up in a new window after the installation completes.
Alternatively, follow the instructions at https://github.com/nodejs/node-gyp#on-windows to install the dependencies yourself.
Back
Next
Cancel

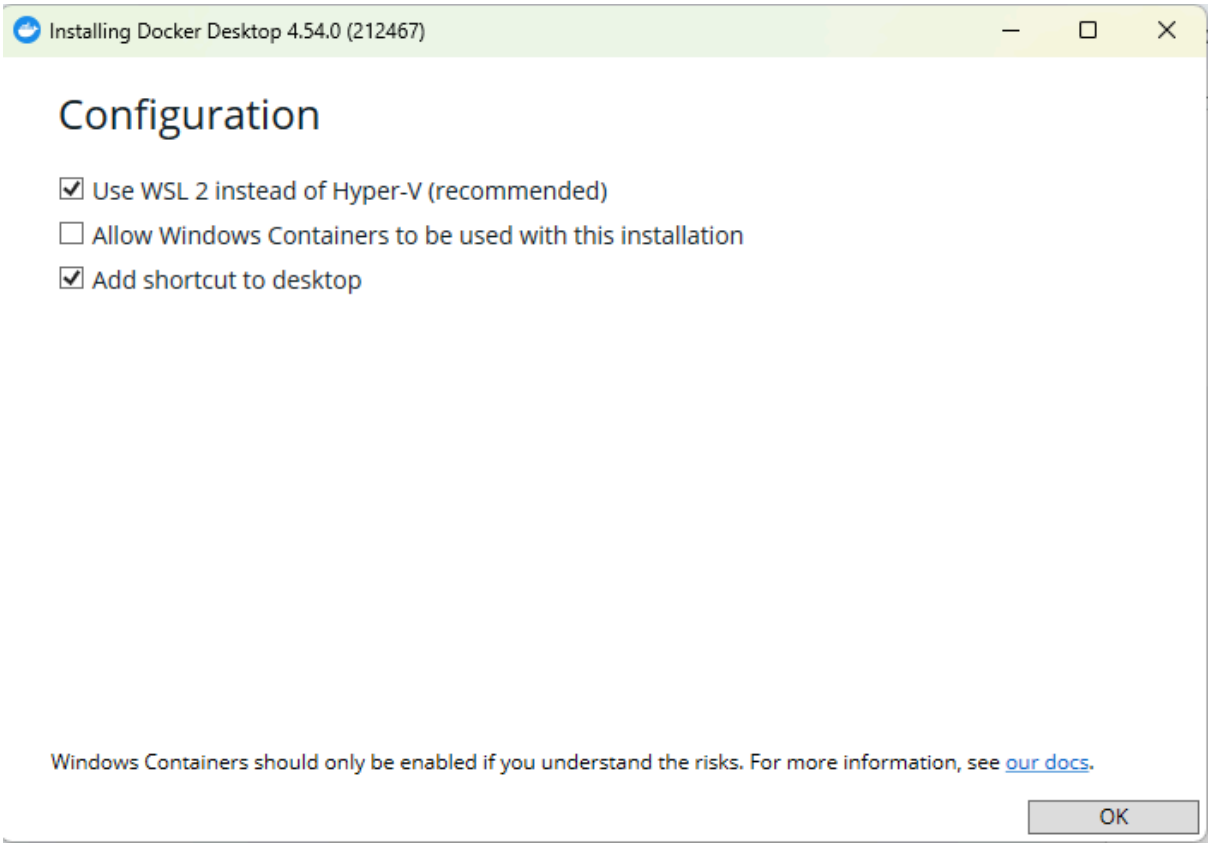


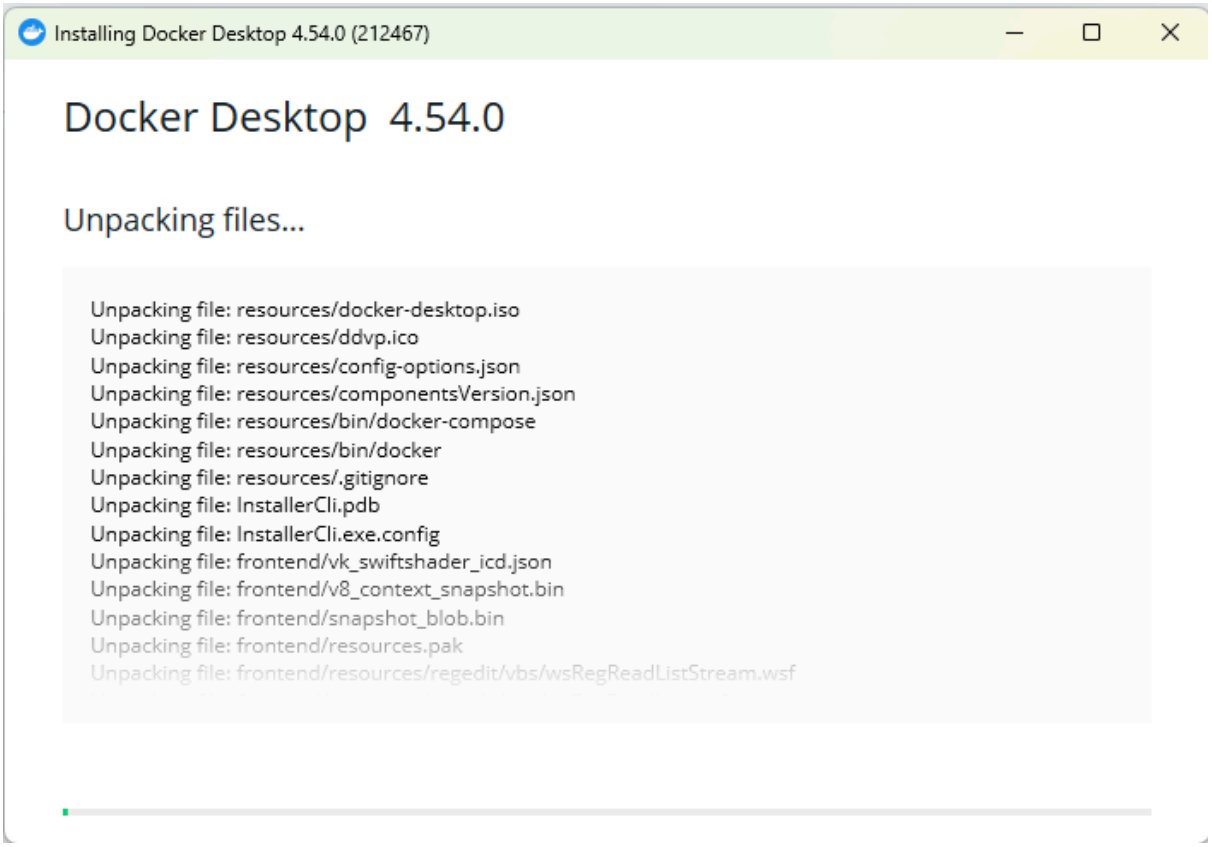
Docker Desktop:

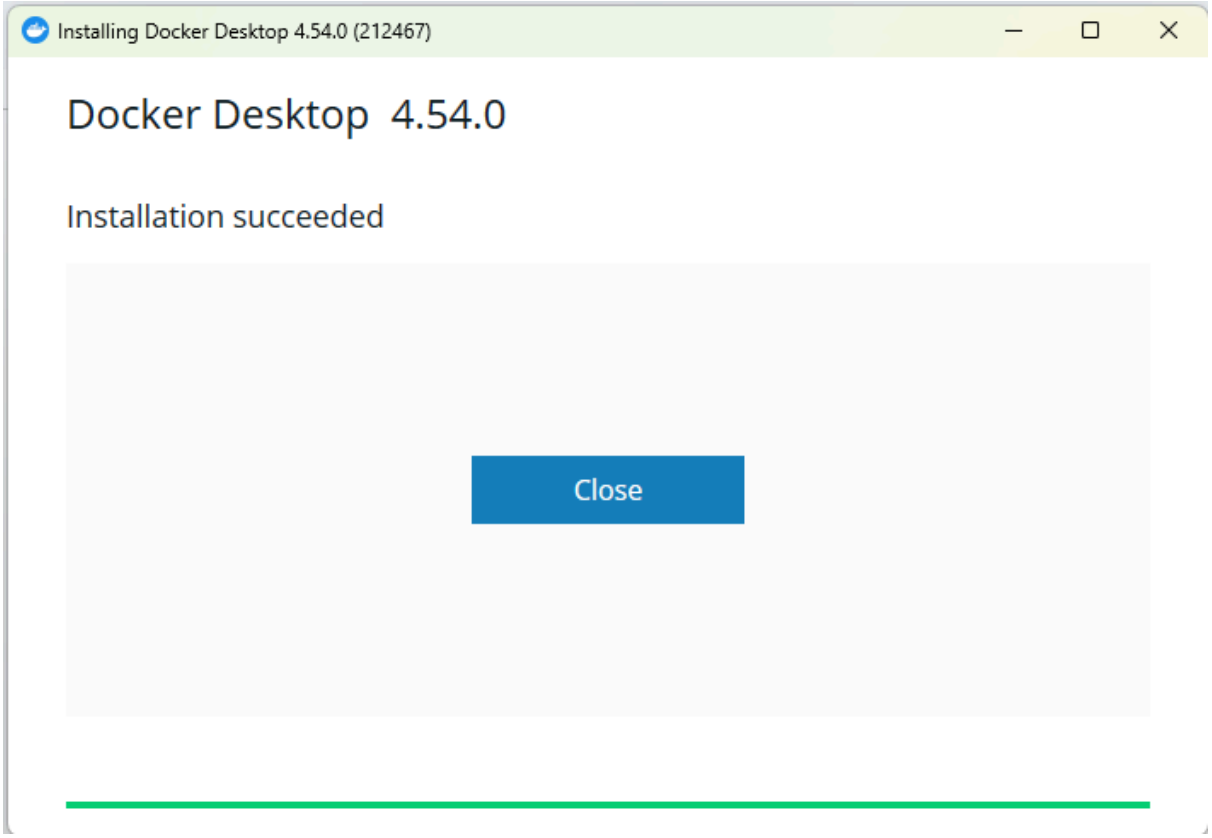
- **Step 1: Download [Docker Desktop]**



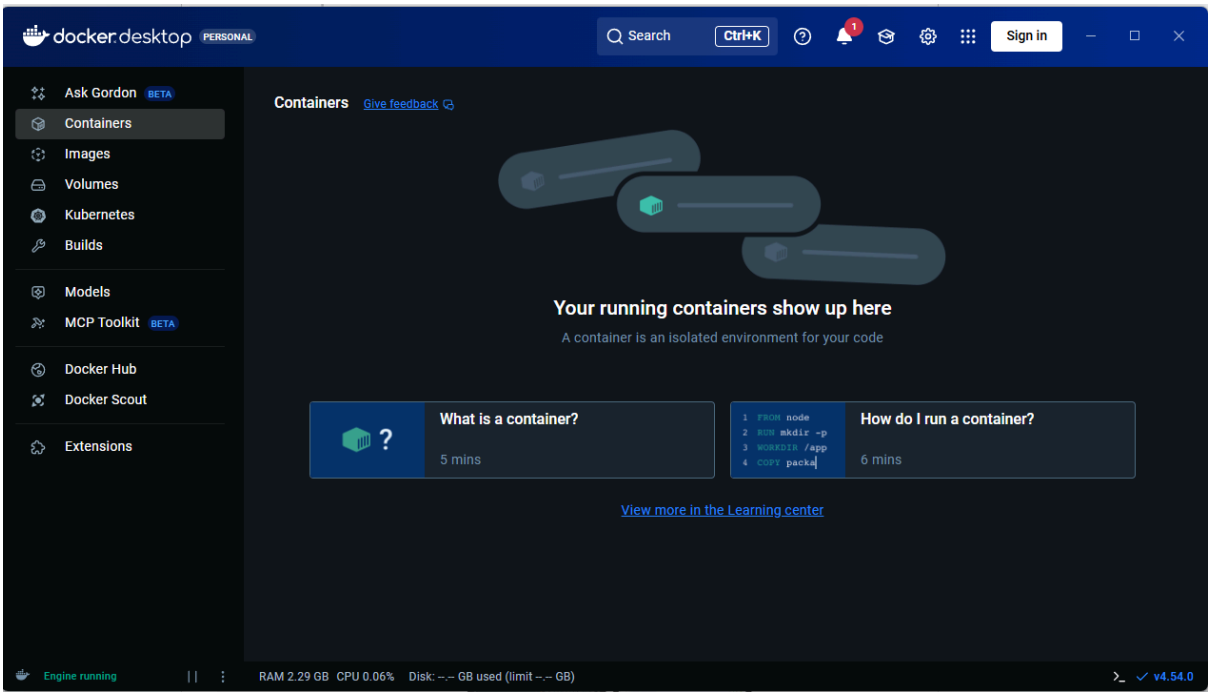
- **Step 2: Install Docker Desktop**


Installing Docker Desktop 4.54.0 (212467)
Configuration
Use WSL 2 instead of Hyper-V (recommended)
Allow Windows Containers to be used with this installation
Add shortcut to desktop
Windows Containers should only be enabled if you understand the risks. For more information, see our docs.
OK


Installing Docker Desktop 4.54.0 (212467)
Docker Desktop 4.54.0
Unpacking files...
Unpacking file: resources/docker-desktop.iso
Unpacking file: resources/ddvp.ico
Unpacking file: resources/config-options.json
Unpacking file: resources/componentsVersion.json
Unpacking file: resources/bin/docker-compose
Unpacking file: resources/bin/docker
Unpacking file: resources/.gitignore
Unpacking file: InstallerCli.pdb
Unpacking file: InstallerCli.exe.config
Unpacking file: frontend/vk_swiftshader_icd.json
Unpacking file: frontend/v8_context_snapshot.bin
Unpacking file: frontend/snapshot_blob.bin
Unpacking file: frontend/resources.pak
Unpacking file: frontend/resources/regedit/vbs/wsRegReadListStream.wsf

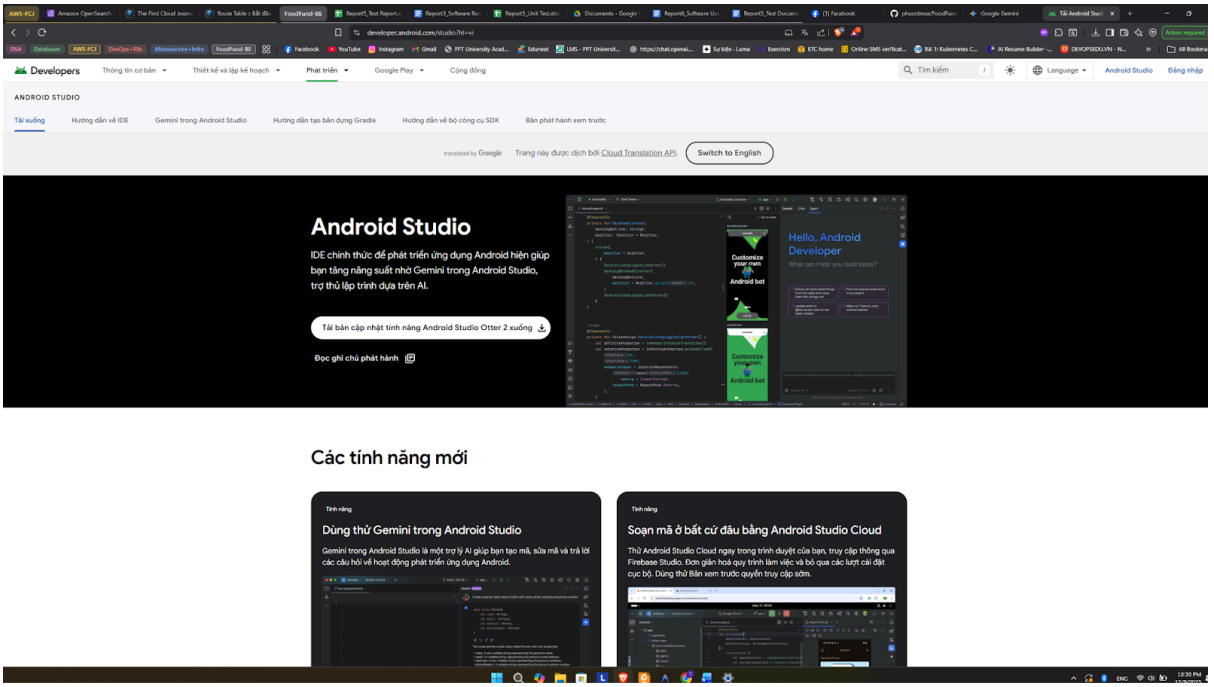


- **Run Docker Desktop**

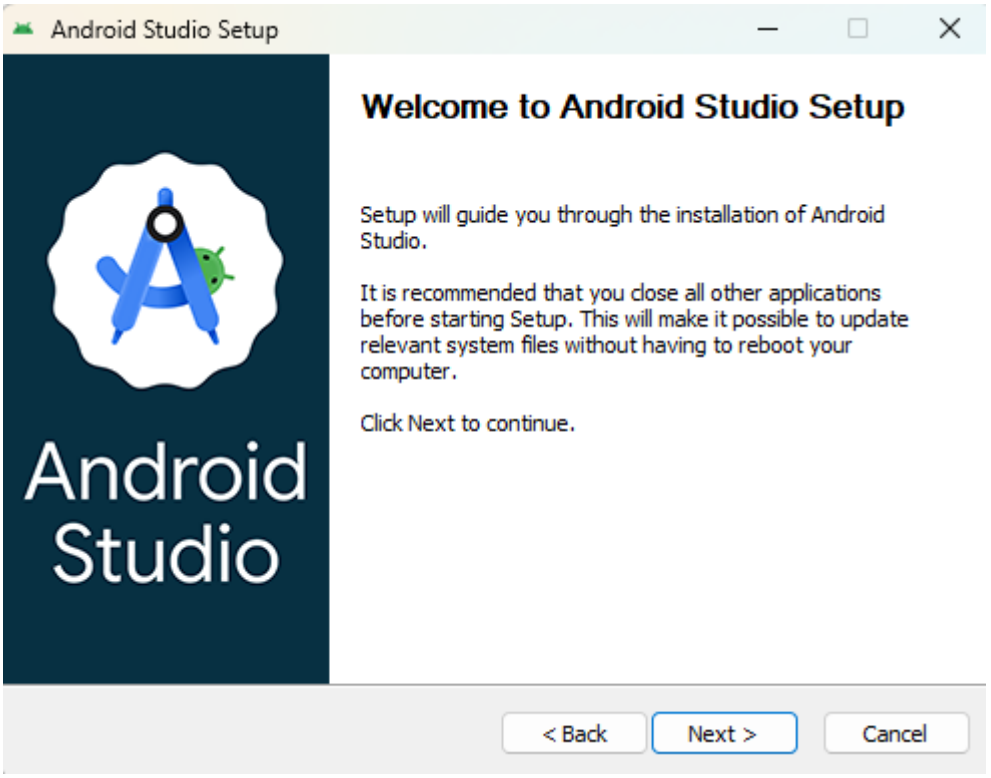


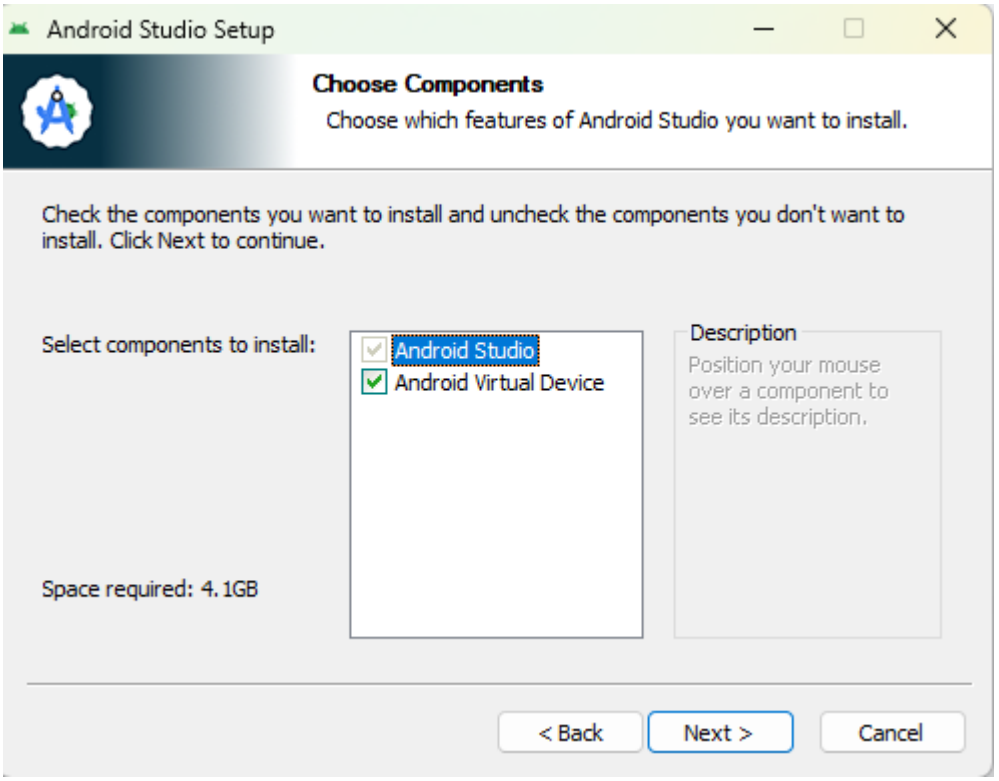
Android Studio:

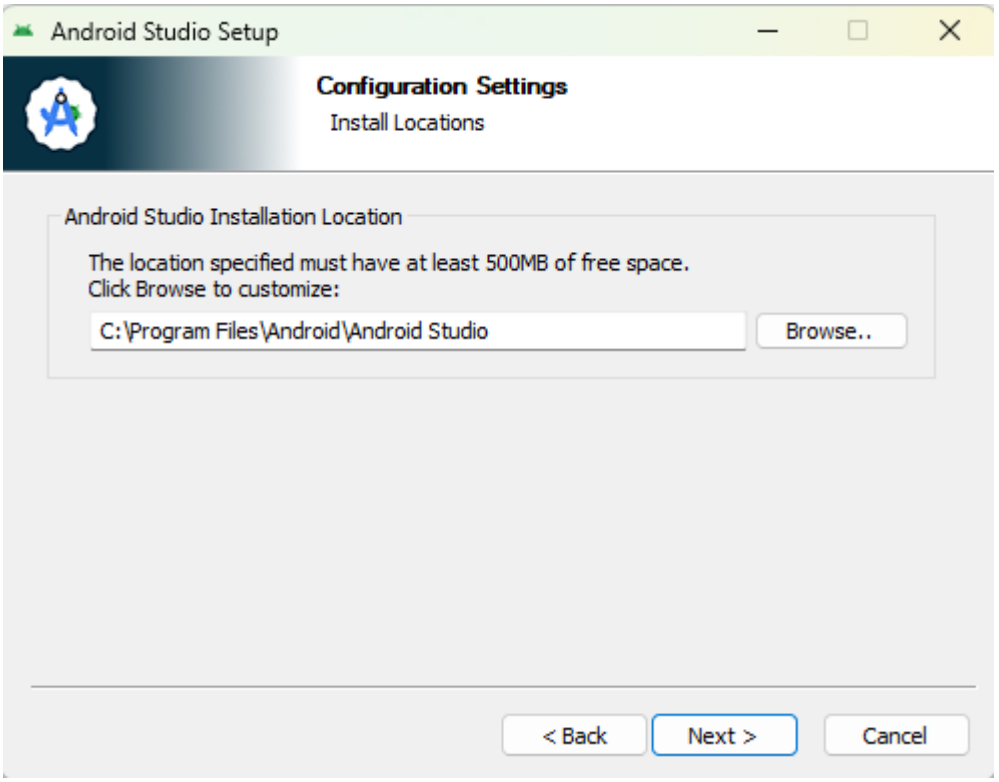
- **Step 1: Download [Android Studio](#):**

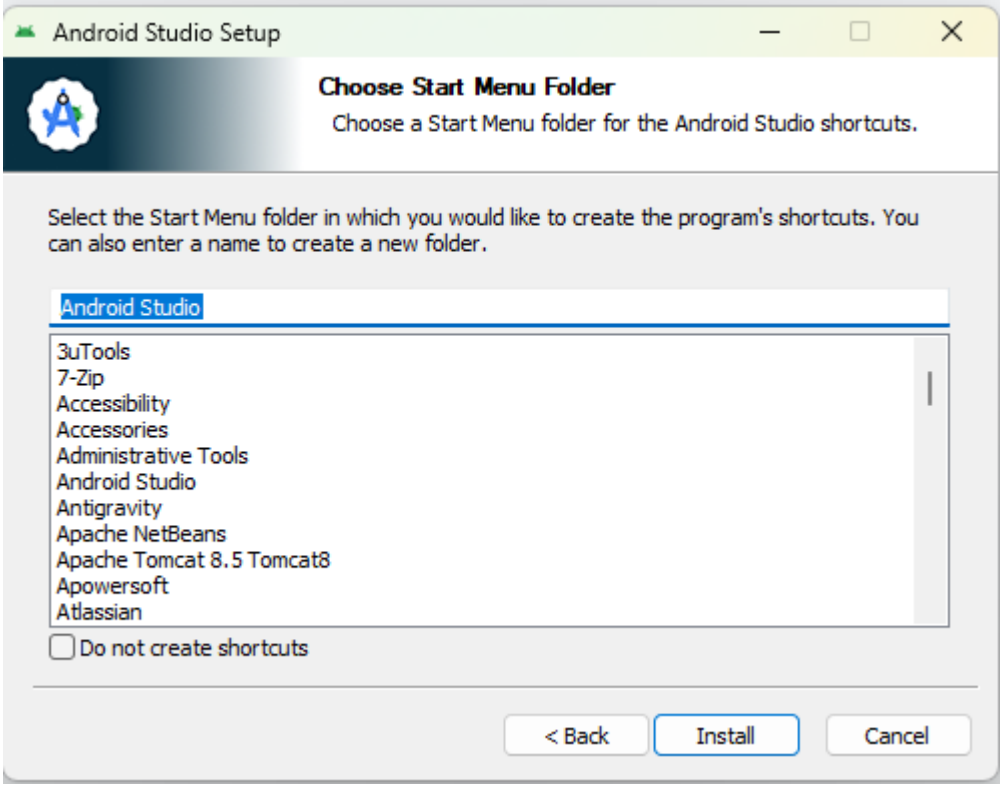


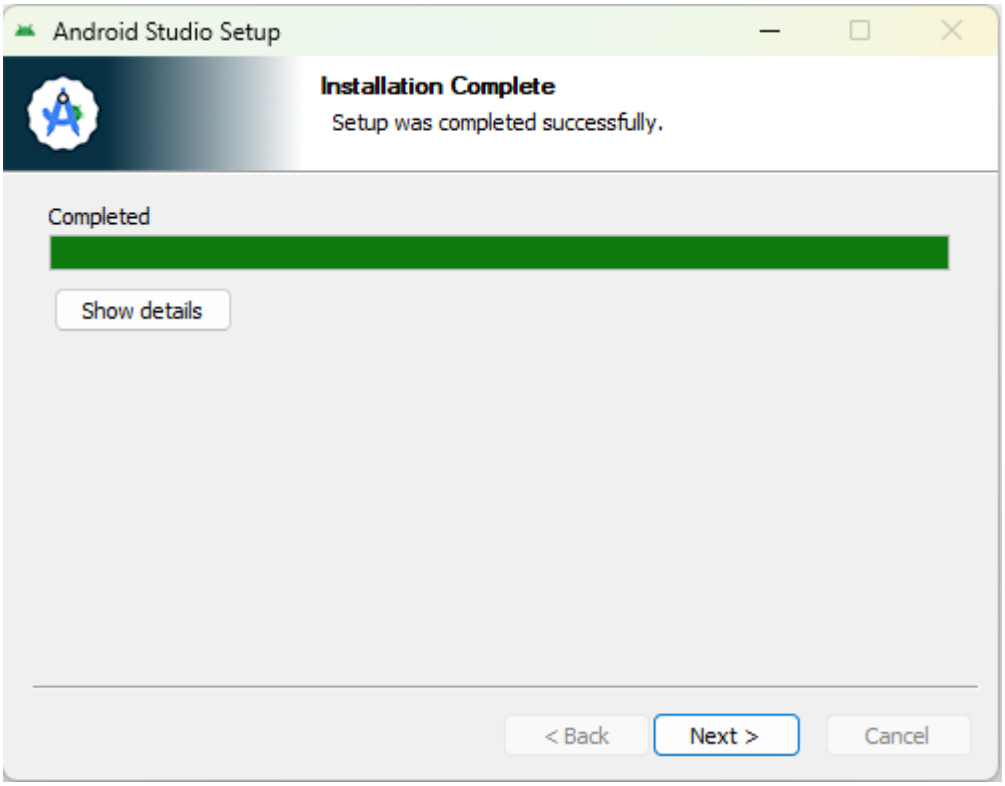
- **Step 2: Install Android Studio:**

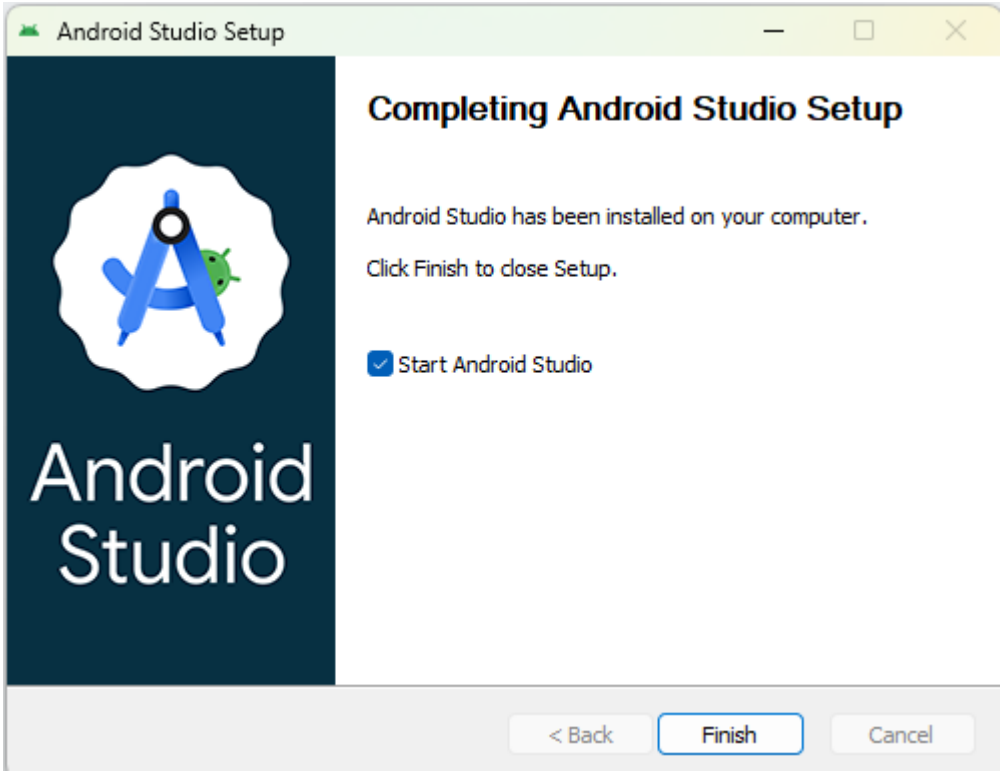



Android Studio Setup
Choose Components
Choose which features of Android Studio you want to install.
Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.
Select components to install:
Android Studio
Android Virtual Device
Description
Position your mouse over a component to see its description.
Space required: 4.1GB
< Back
Next >
Cancel

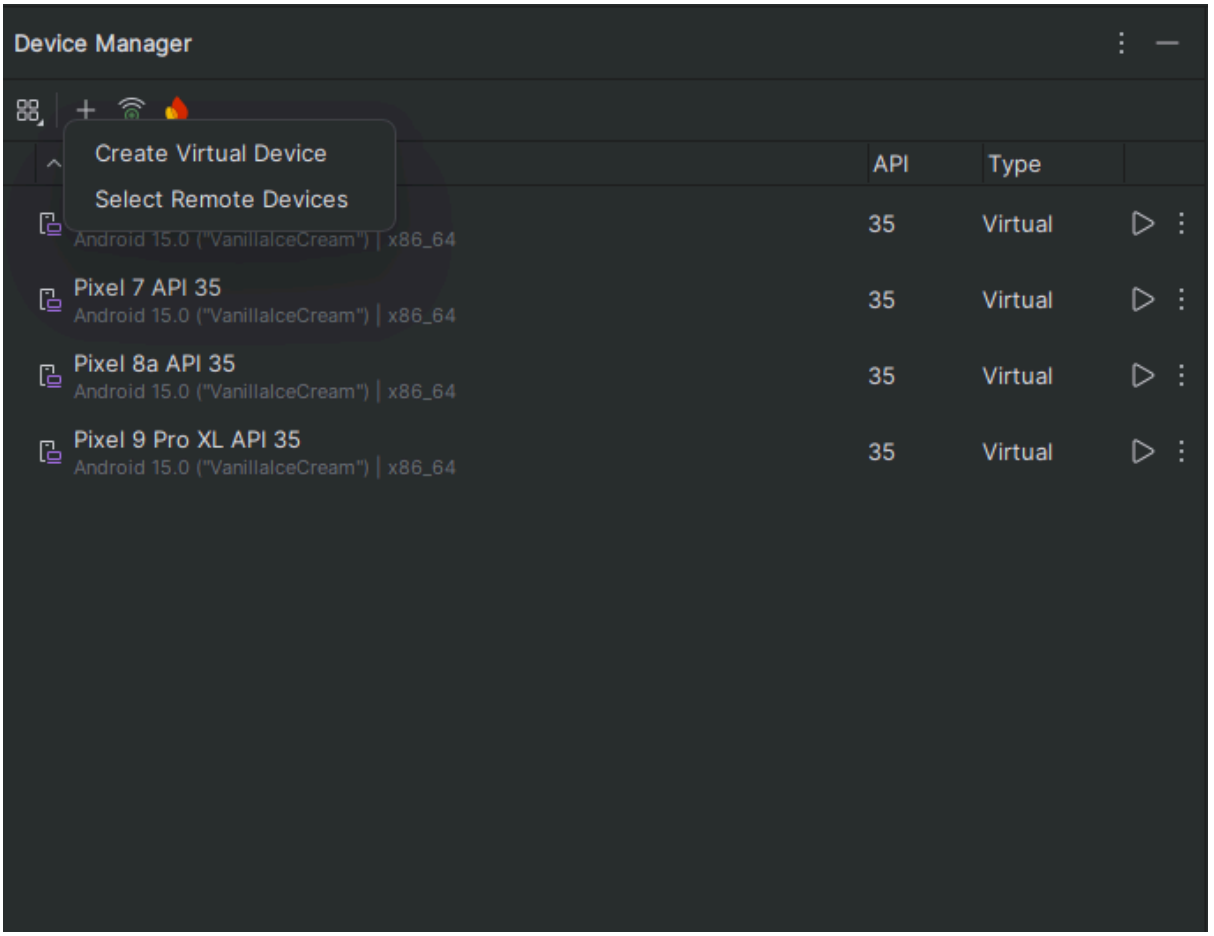

Android Studio Setup
Configuration Settings
Install Locations
Android Studio Installation Location
The location specified must have at least 500MB of free space.
Click Browse to customize:
C:\Program Files\Android\Android Studio
Browse..
< Back
Next >
Cancel


Android Studio Setup
Choose Start Menu Folder
Choose a Start Menu folder for the Android Studio shortcuts.
Select the Start Menu folder in which you would like to create the program's shortcuts. You can also enter a name to create a new folder.
Android Studio
3uTools
7-Zip
Accessibility
Accessories
Administrative Tools
Android Studio
Antigravity
Apache NetBeans
Apache Tomcat 8.5 Tomcat8
Apowersoft
Atlassian
Do not create shortcuts
< Back
Install
Cancel


Android Studio Setup
Installation Complete
Setup was completed successfully.
Completed
Show details
< Back
Next >
Cancel



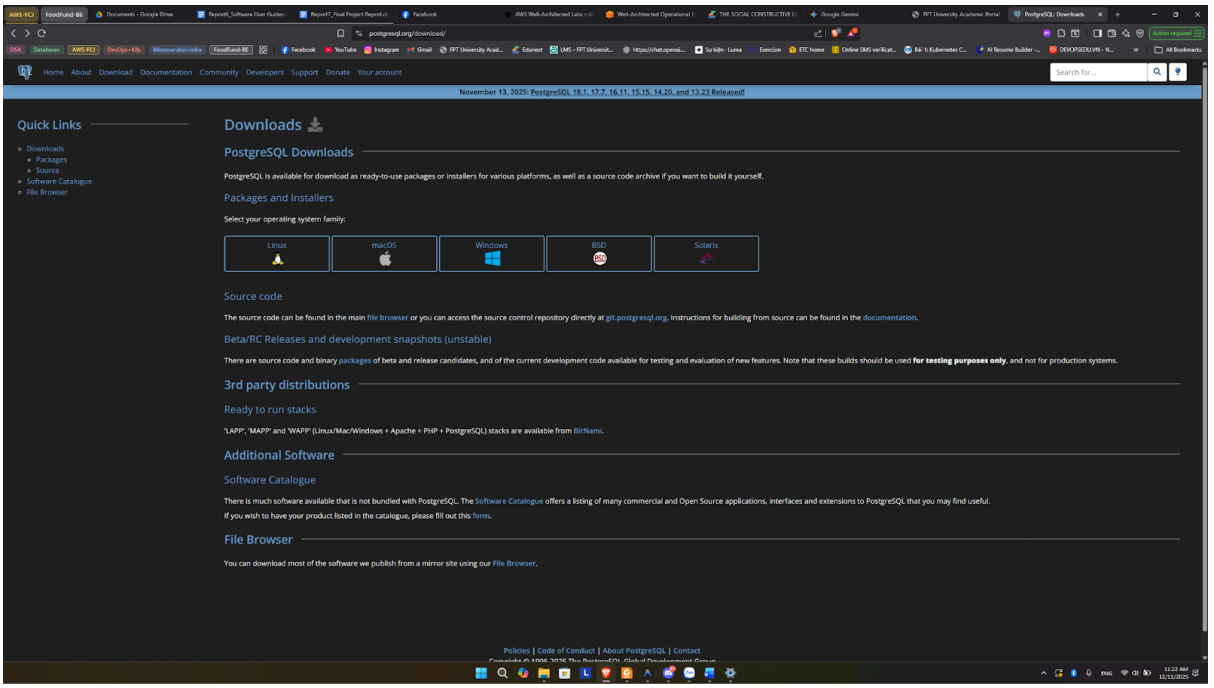
- **Step 3: Open Android Studio and open device**

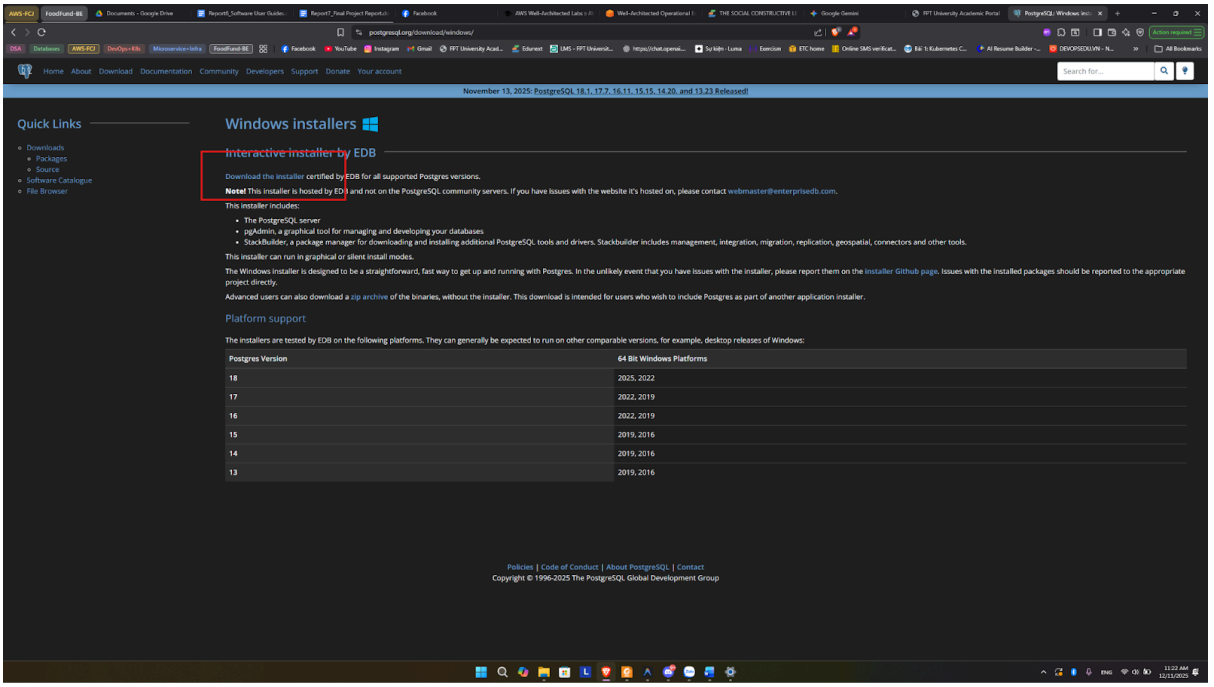


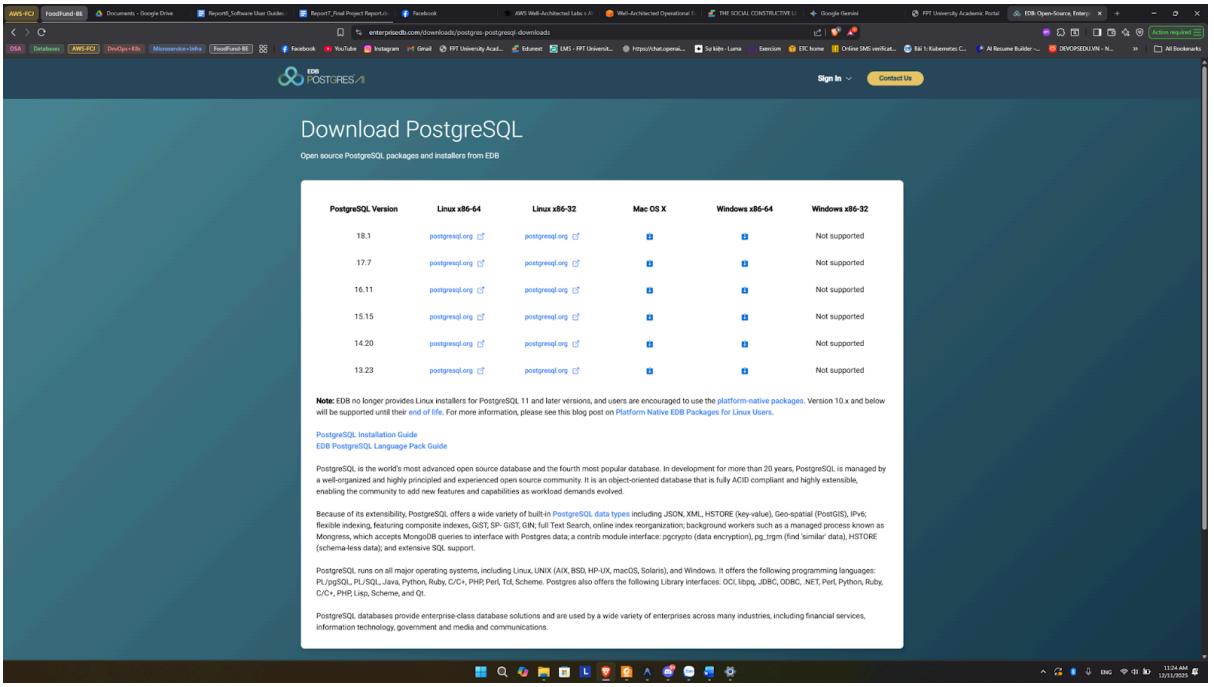
- **Step 4: Turn on the emulator**

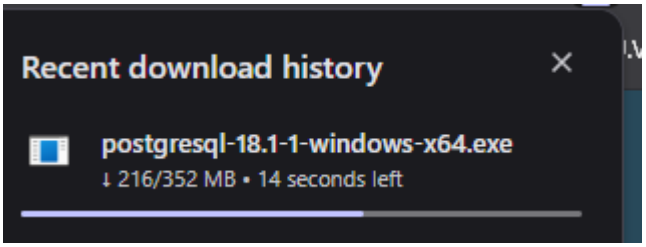
PostgreSQL:

- **Step 1: Download [PostgreSQL]**

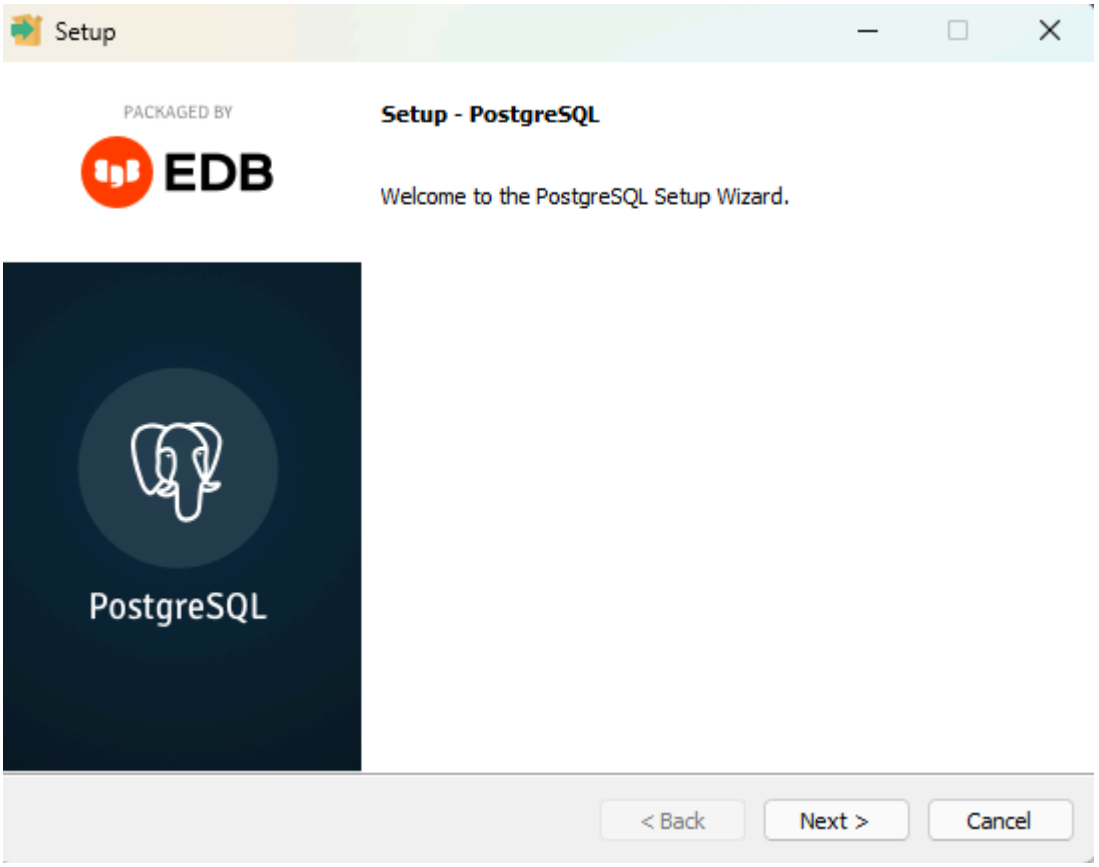


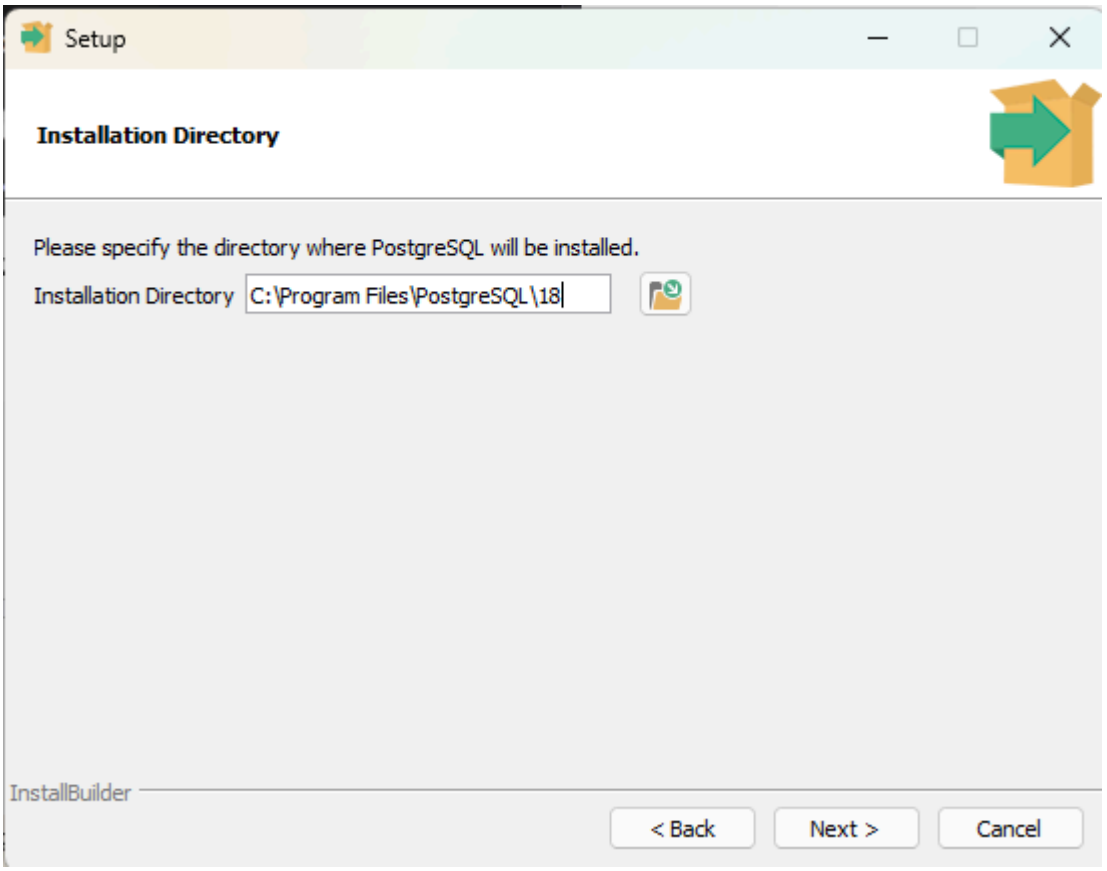


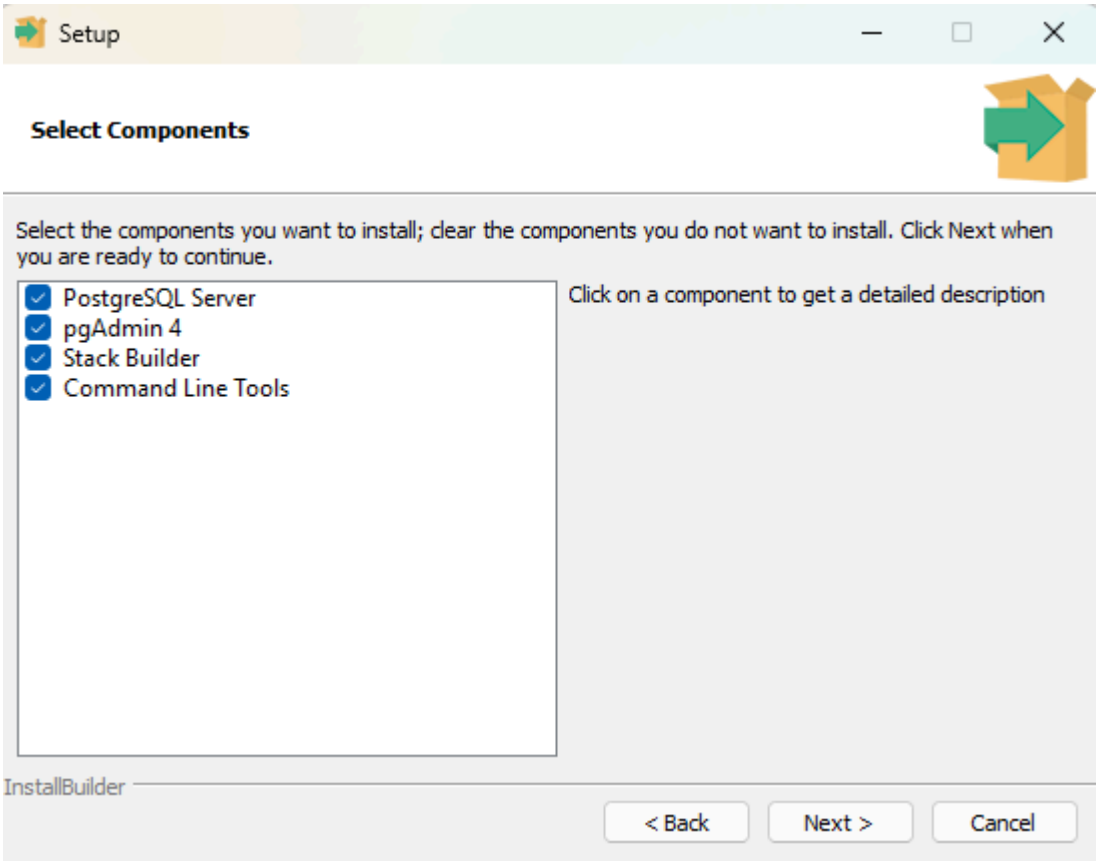


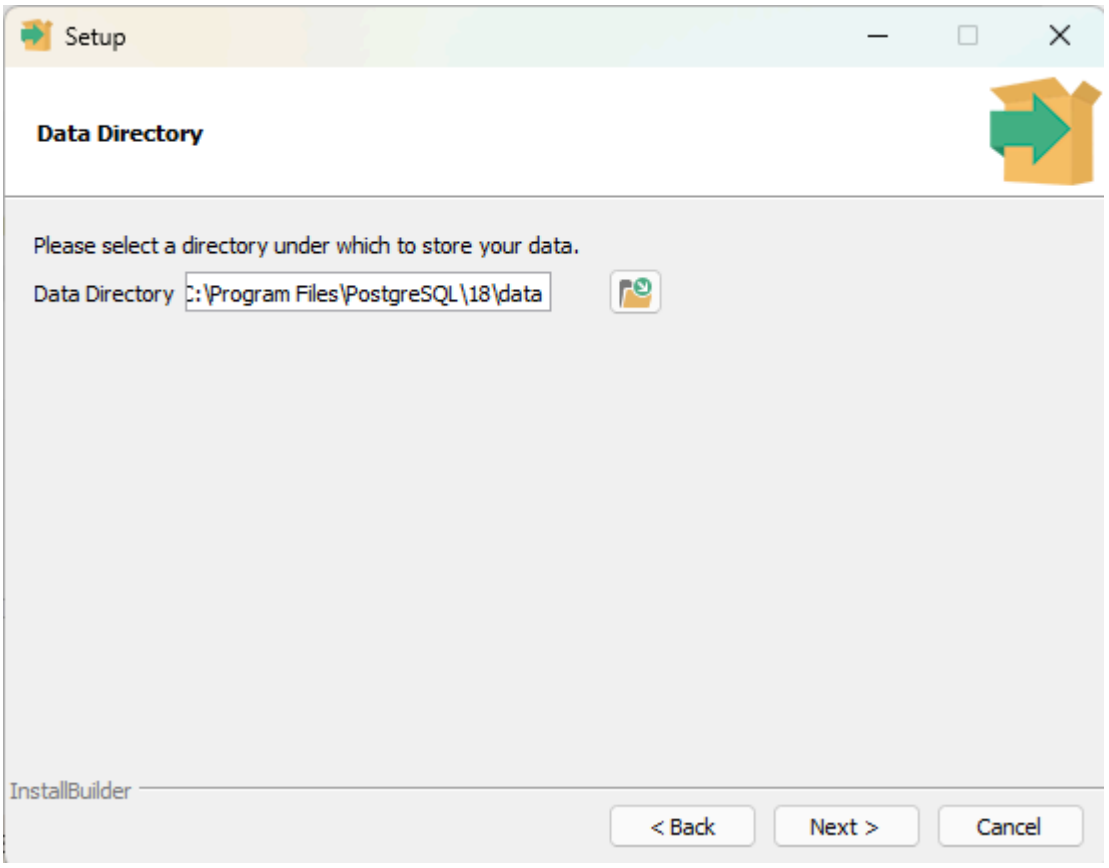


- **Step 2 Install PostgreSQL and pgAdmin:**

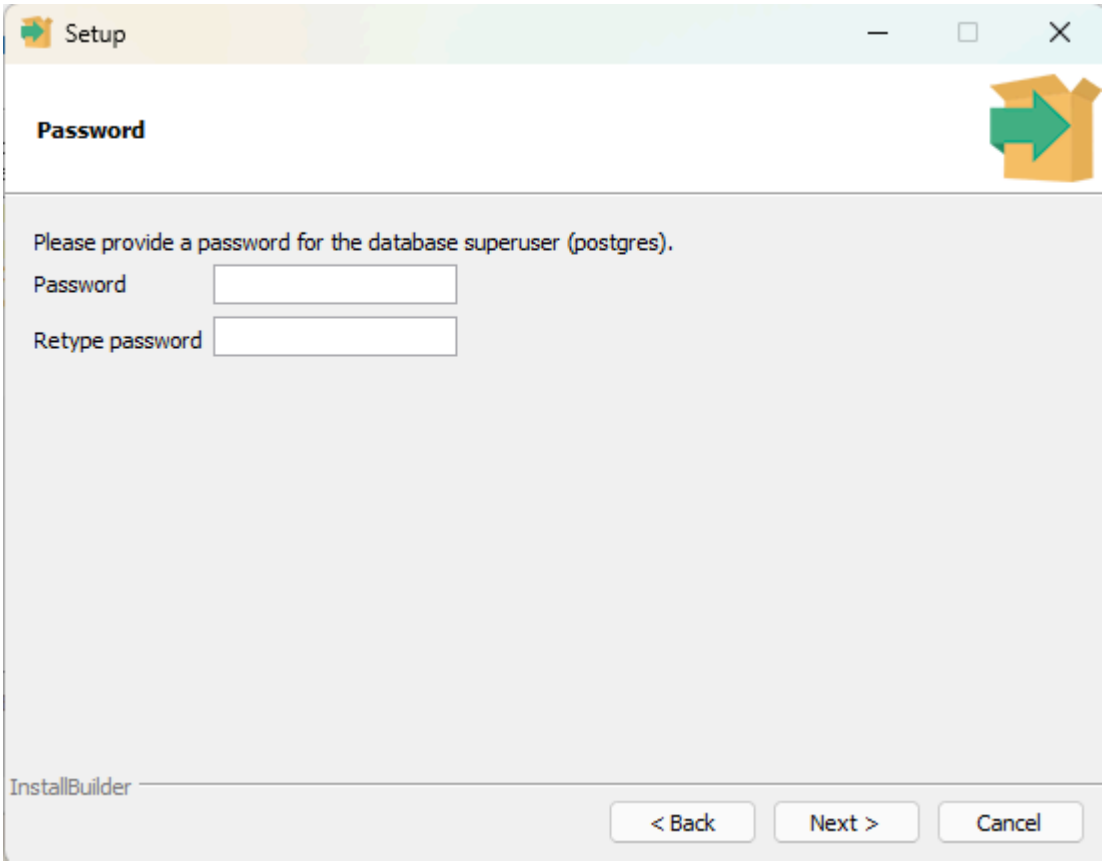

Setup
PACKAGED BY
EDB
Setup - PostgreSQL
Welcome to the PostgreSQL Setup Wizard.
PostgreSQL
< Back
Next >
Cancel

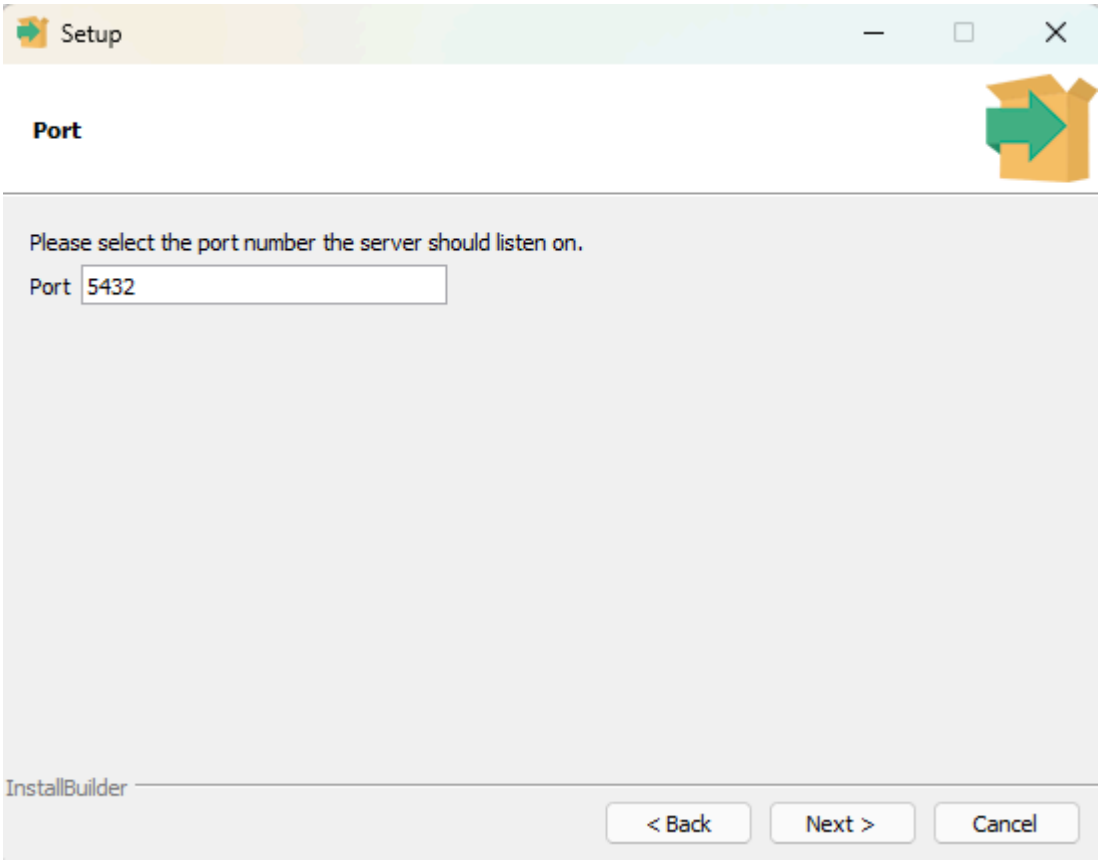

Setup
Installation Directory
Please specify the directory where PostgreSQL will be installed.
Installation Directory
C:\Program Files\PostgreSQL\18
InstallBuilder
< Back
Next >
Cancel

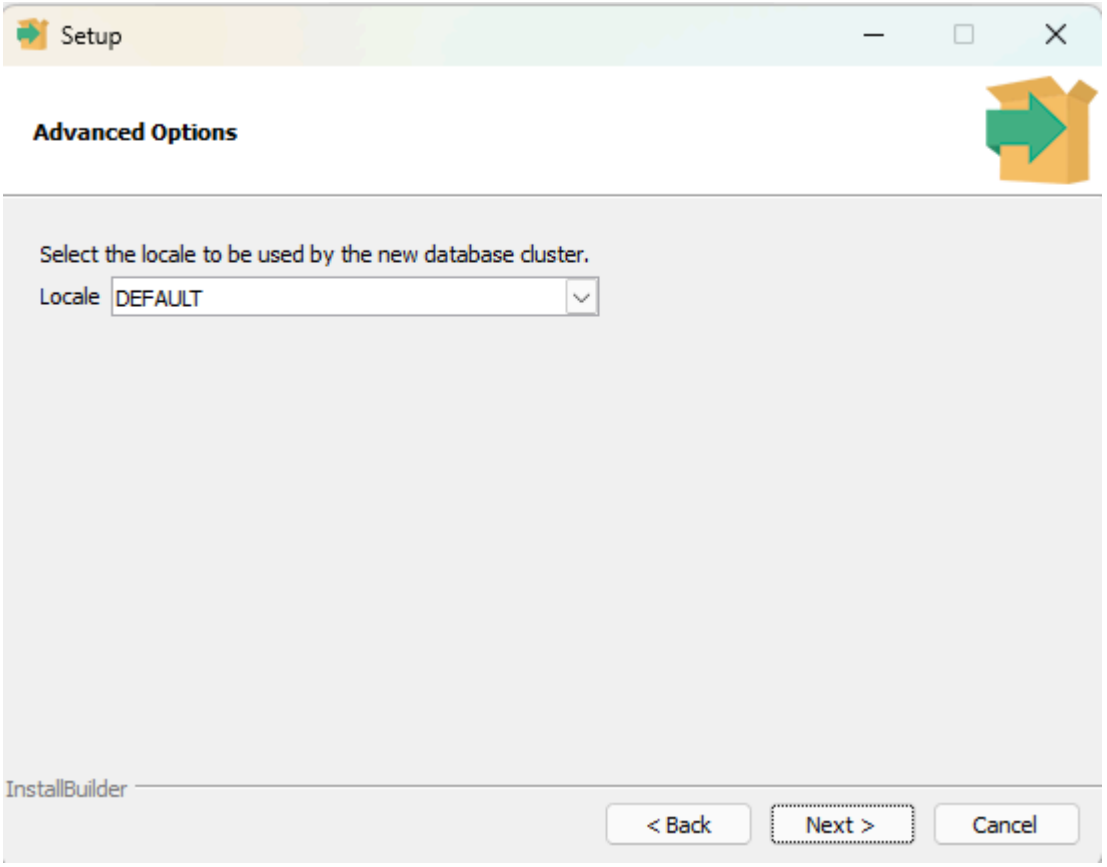


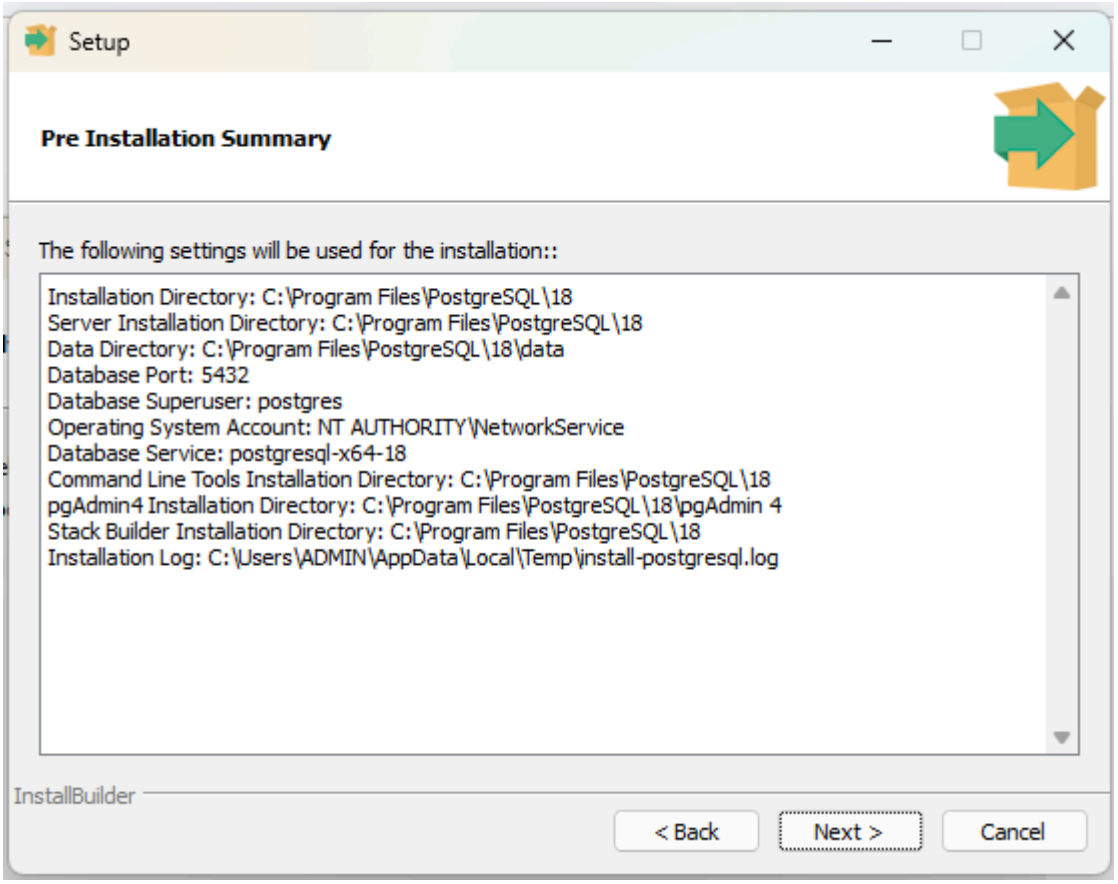


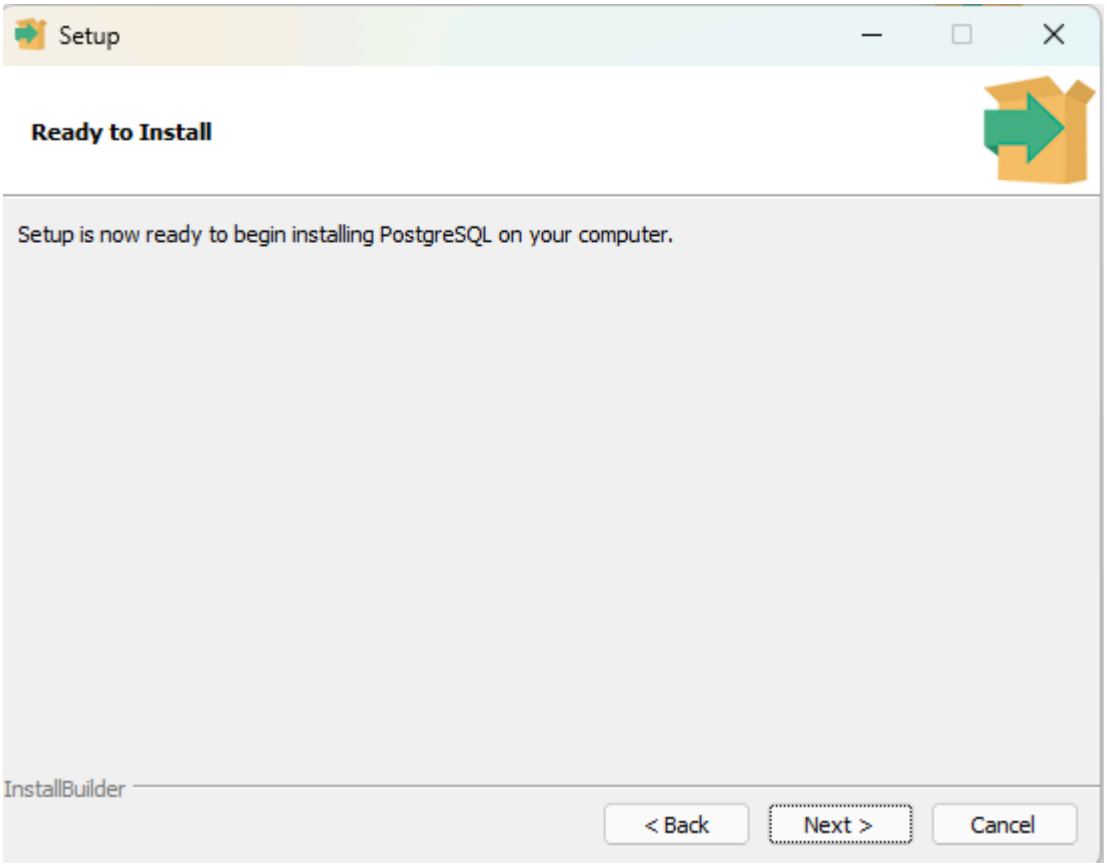
If first time it opens, pgAdmin will ask for postgres user password


Setup
Password
Please provide a password for the database superuser (postgres).
Password
Retype password
InstallBuilder
< Back
Next >
Cancel

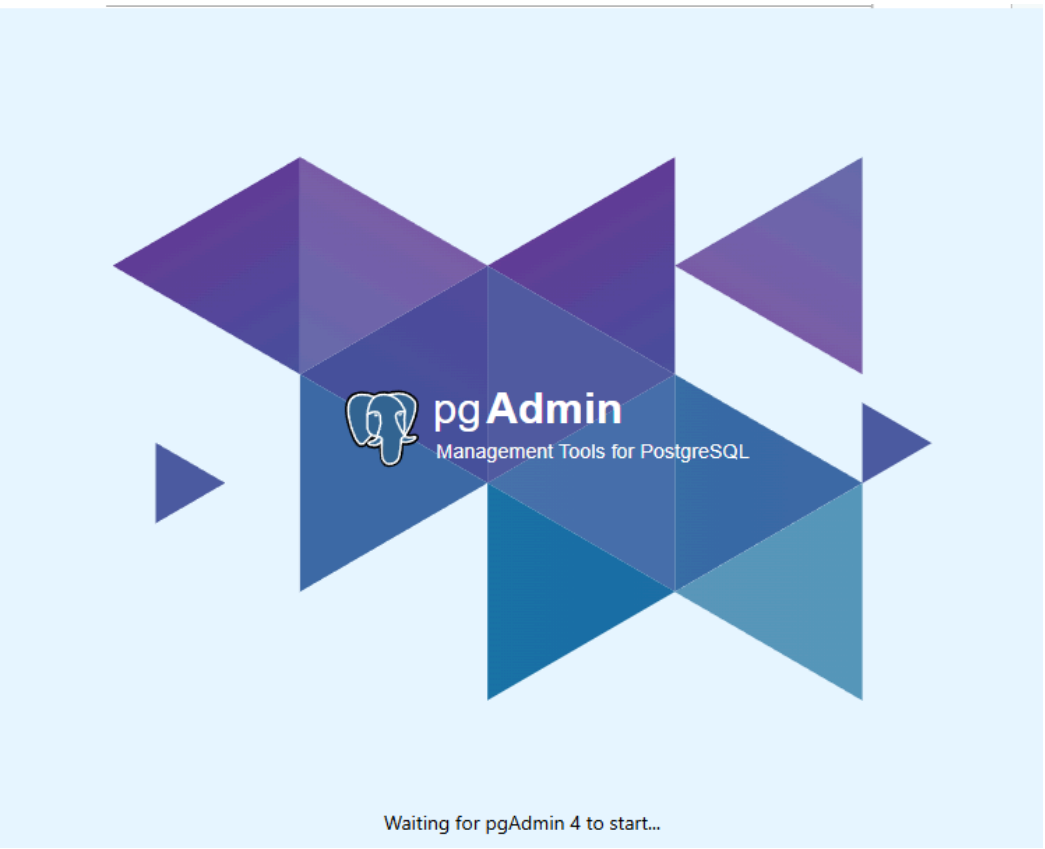

Setup
Port
Please select the port number the server should listen on.
Port
5432
InstallBuilder
< Back
Next >
Cancel

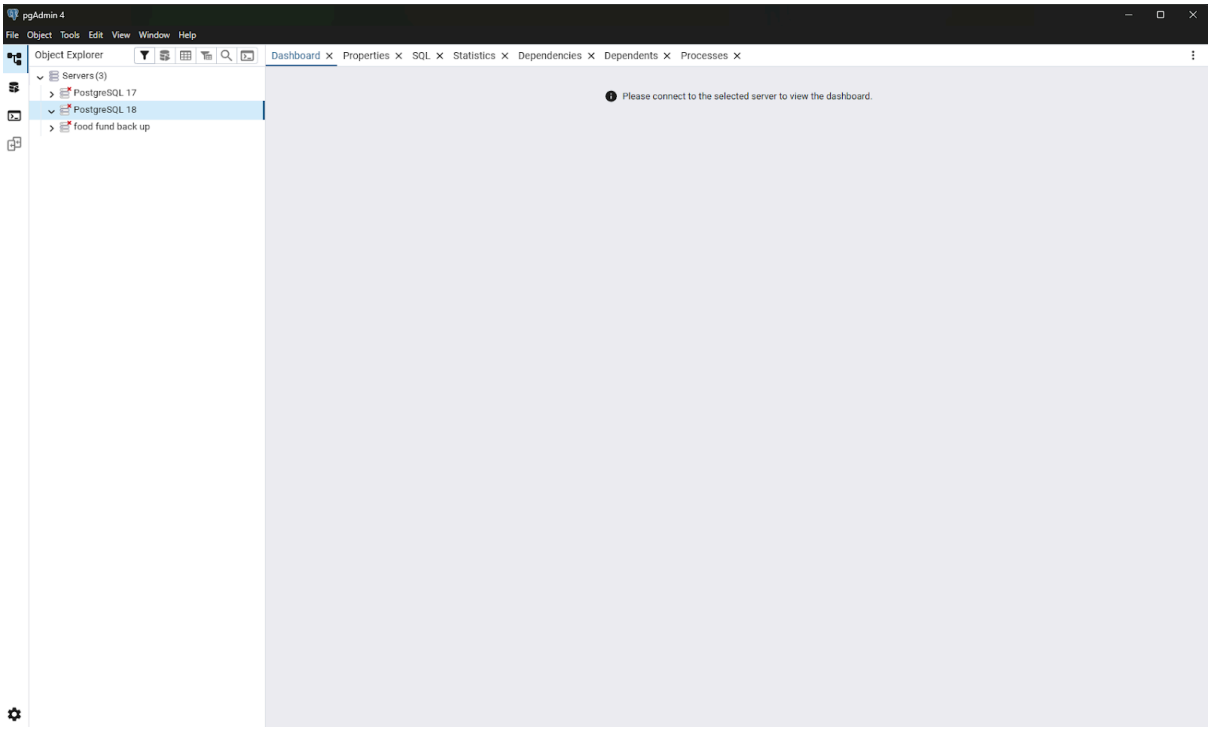

Setup
Advanced Options
Select the locale to be used by the new database cluster.
Locale
DEFAULT
InstallBuilder
< Back
Next >
Cancel


Setup
Pre Installation Summary
The following settings will be used for the installation::
Installation Directory: C:\Program Files\PostgreSQL\18
Server Installation Directory: C:\Program Files\PostgreSQL\18
Data Directory: C:\Program Files\PostgreSQL\18\data
Database Port: 5432
Database Superuser: postgres
Operating System Account: NT AUTHORITY\NetworkService
Database Service: postgresql-x64-18
Command Line Tools Installation Directory: C:\Program Files\PostgreSQL\18
pgAdmin4 Installation Directory: C:\Program Files\PostgreSQL\18\pgAdmin 4
Stack Builder Installation Directory: C:\Program Files\PostgreSQL\18
Installation Log: C:\Users\ADMIN\AppData\Local\Temp\install-postgresql.log
InstallBuilder
< Back
Next >
Cancel

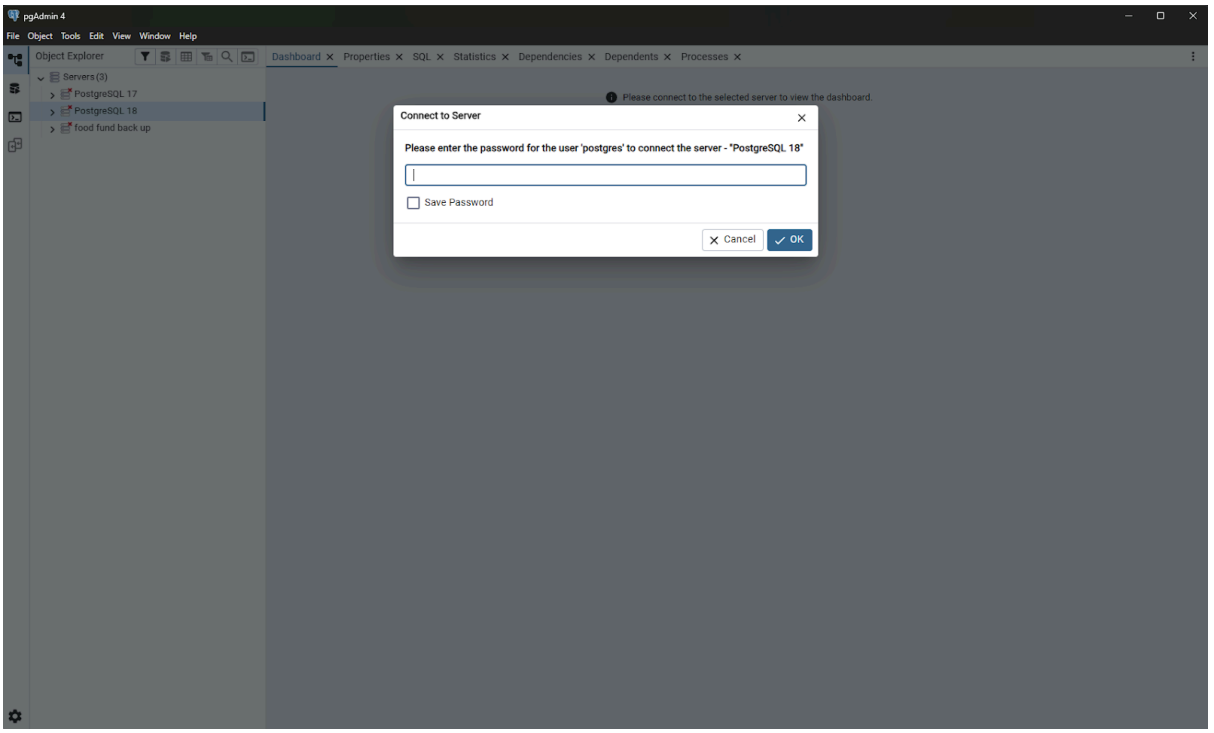


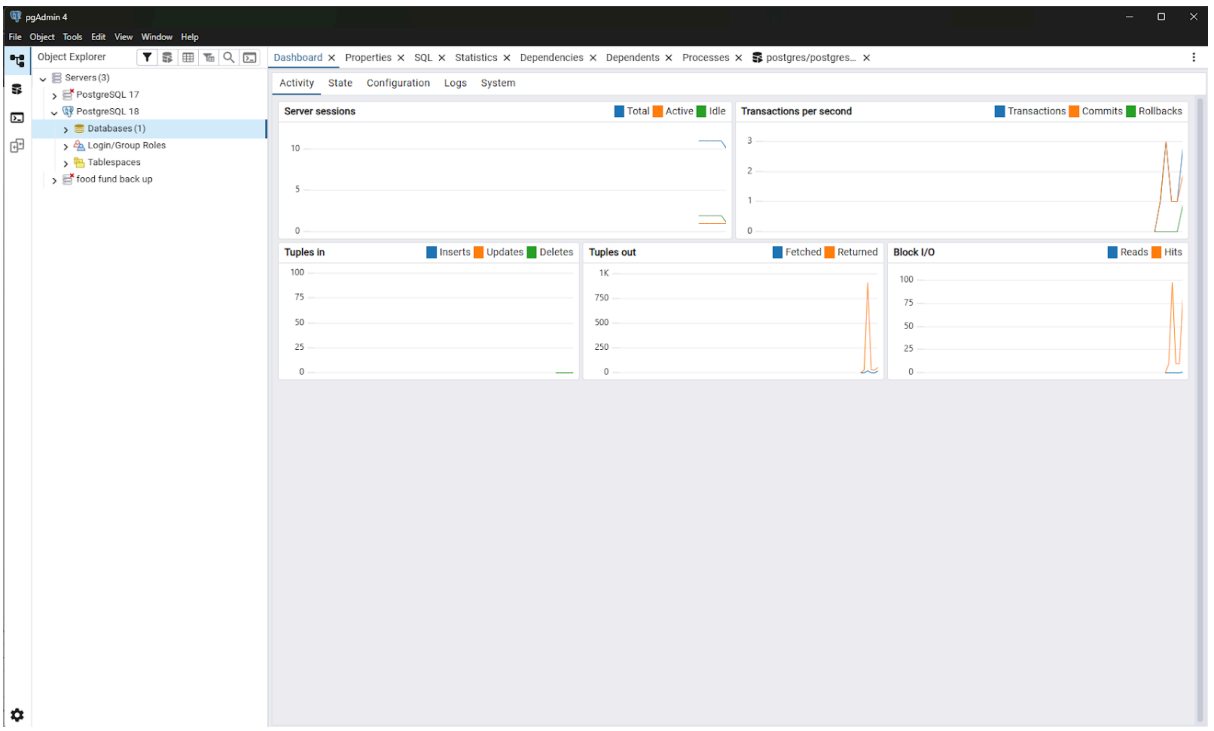
- **Step 3 Once logged in, you'll see Server, Databases, Login/Group Roles, Tables, etc:**





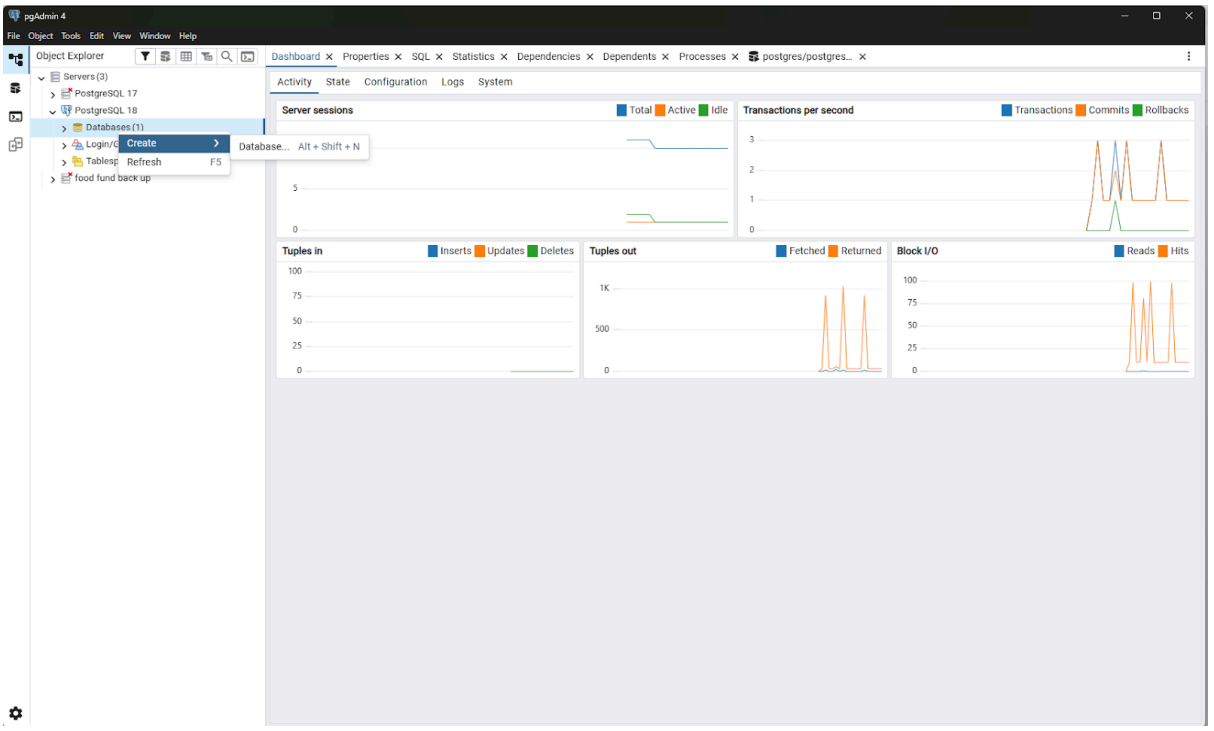
Choose Database Server and enter password



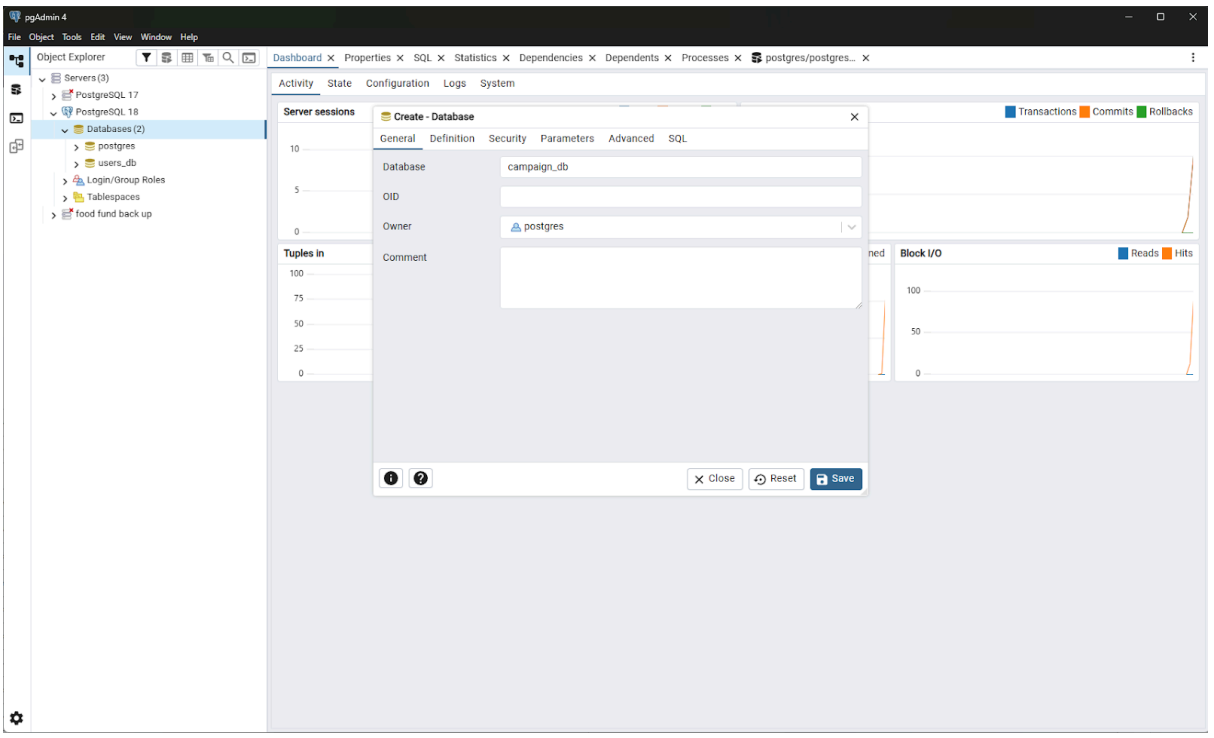


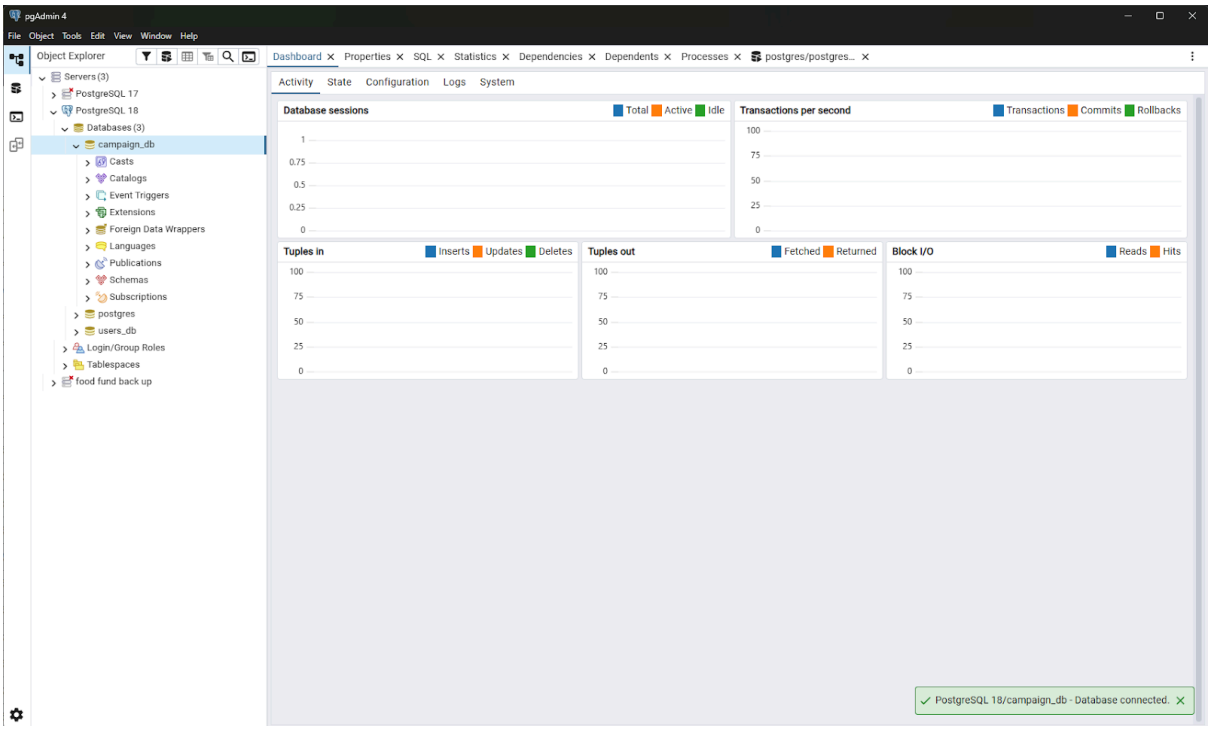
- **Step 4 Create a new databases:**

Right click on Databases -> Create -> Database



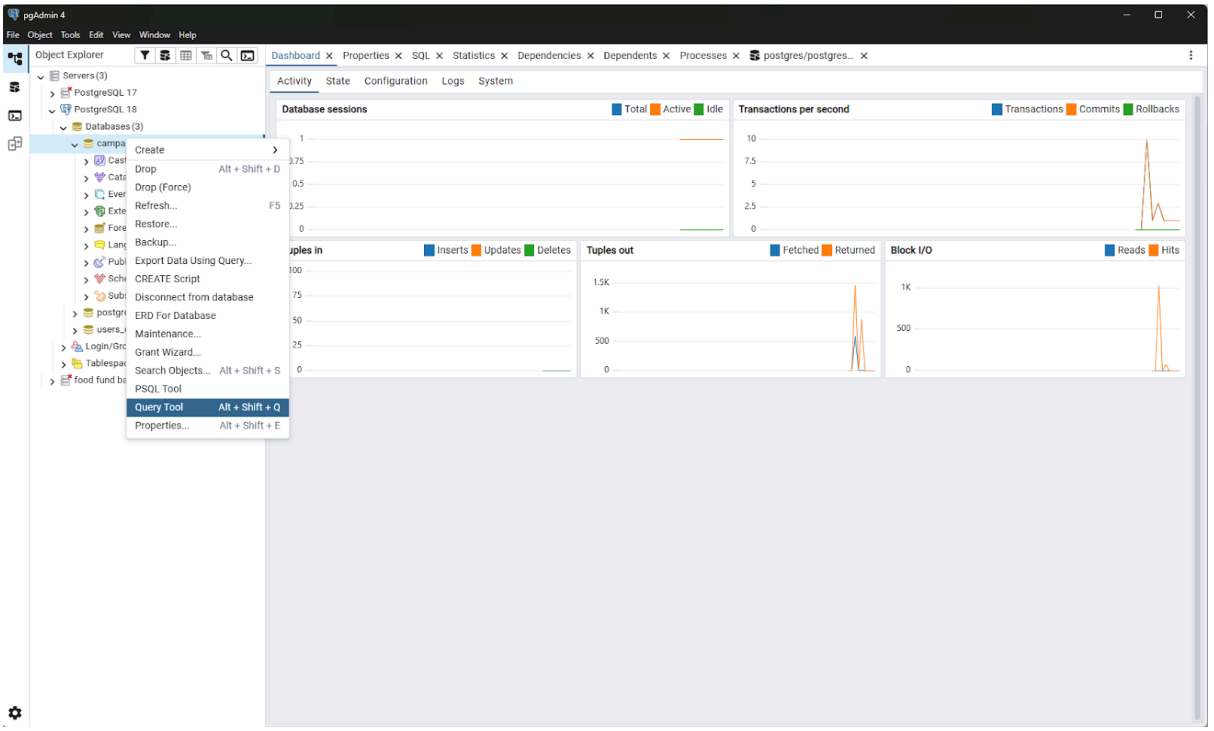
Enter databases name, e.g: campaign_db -> then click Save



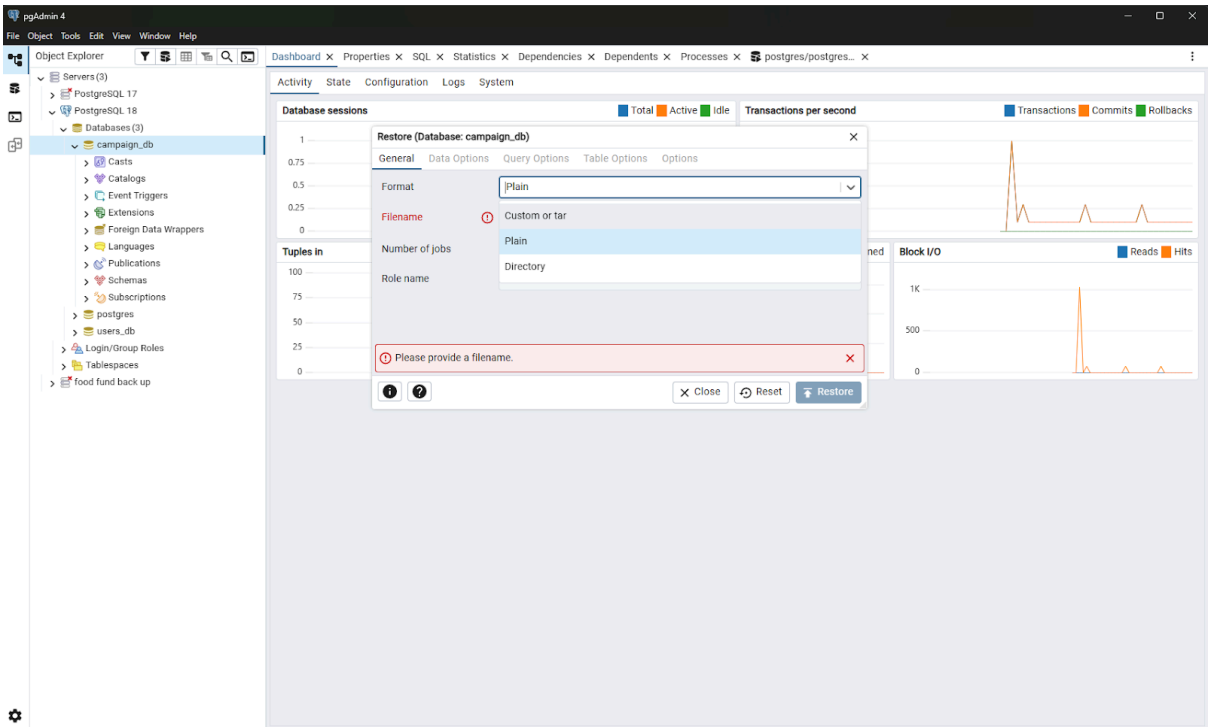


- **Step 5 Apply seed data:**

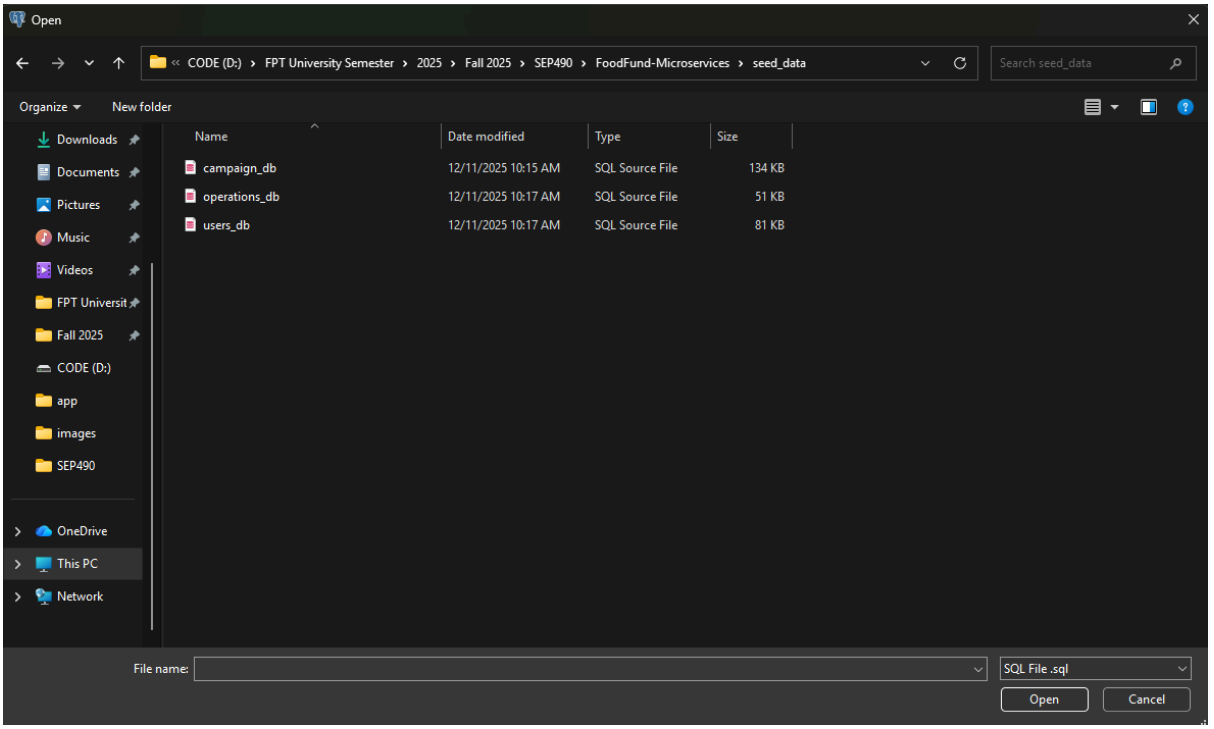
  Right click on database name, then click Restore…

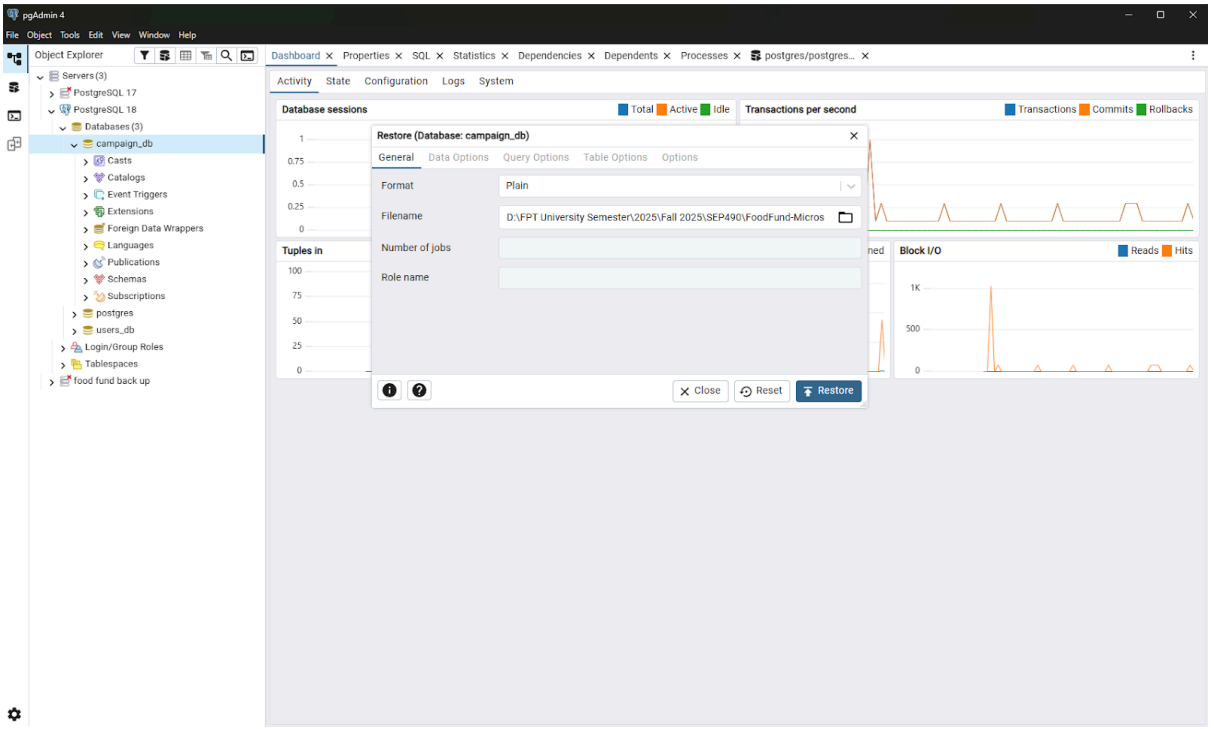


Choose format PLAIN

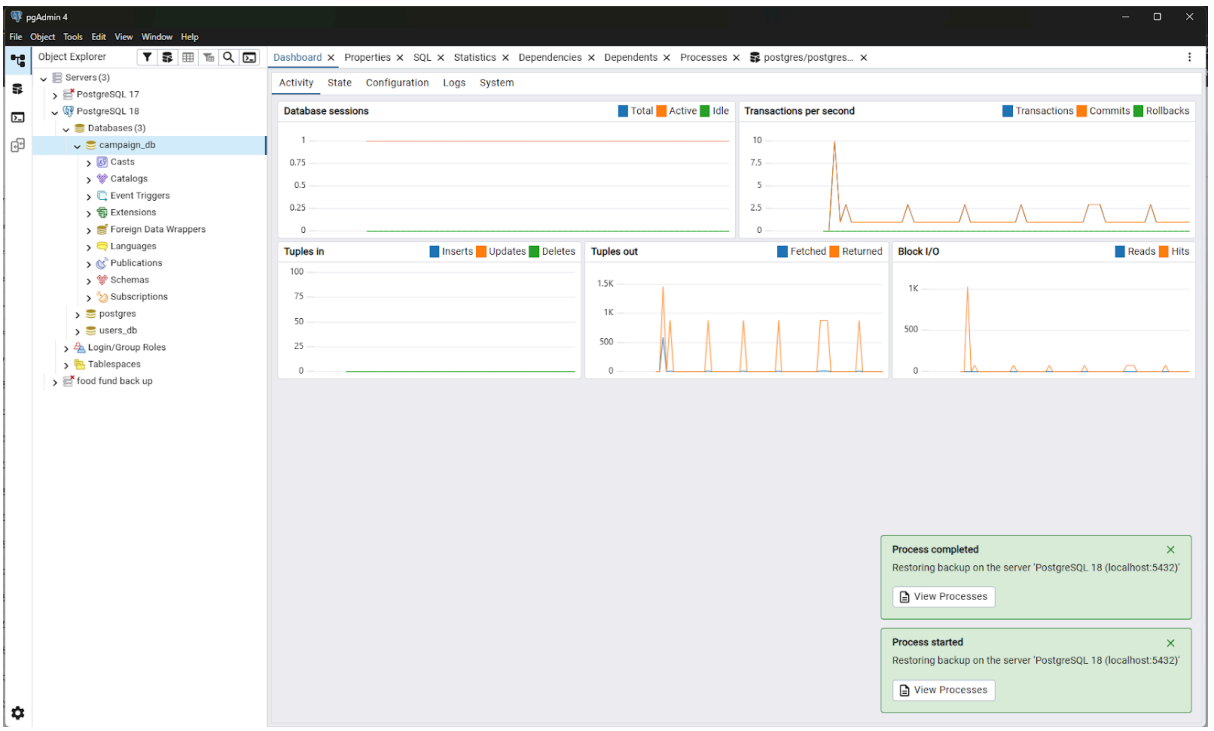


Browse Filename and choose one of file database sql at YourDirectory\FoodFund-Microservices\seed_data\database_name.sql



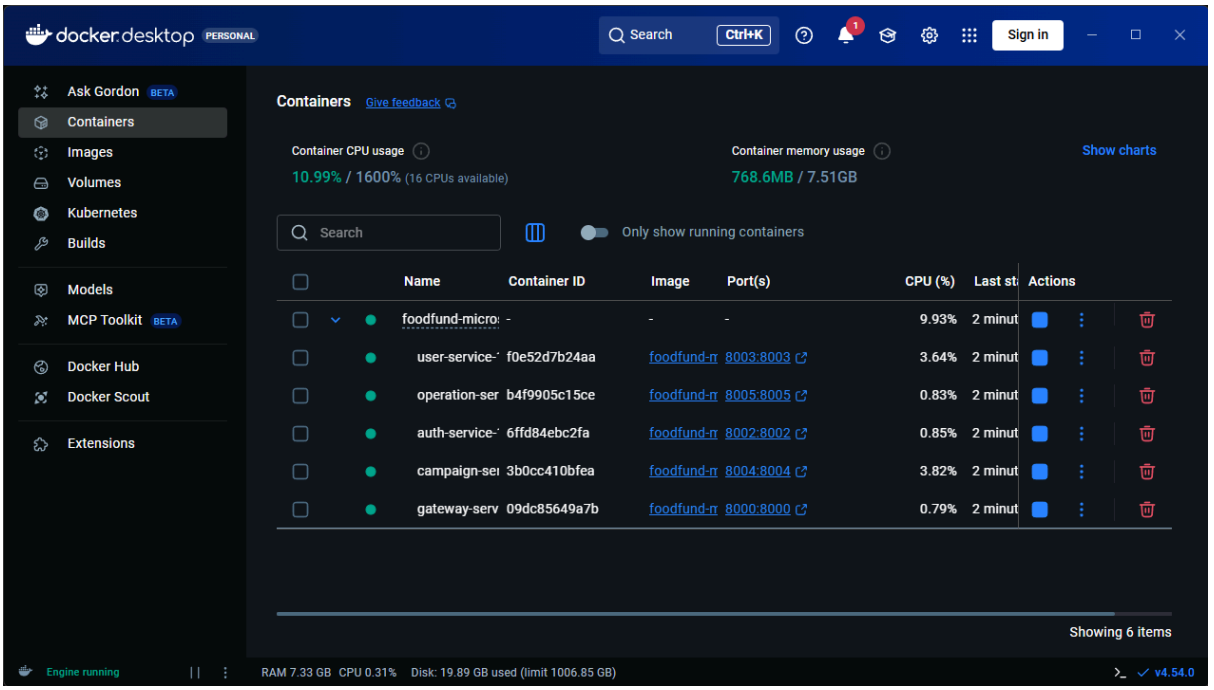


Then click Restore

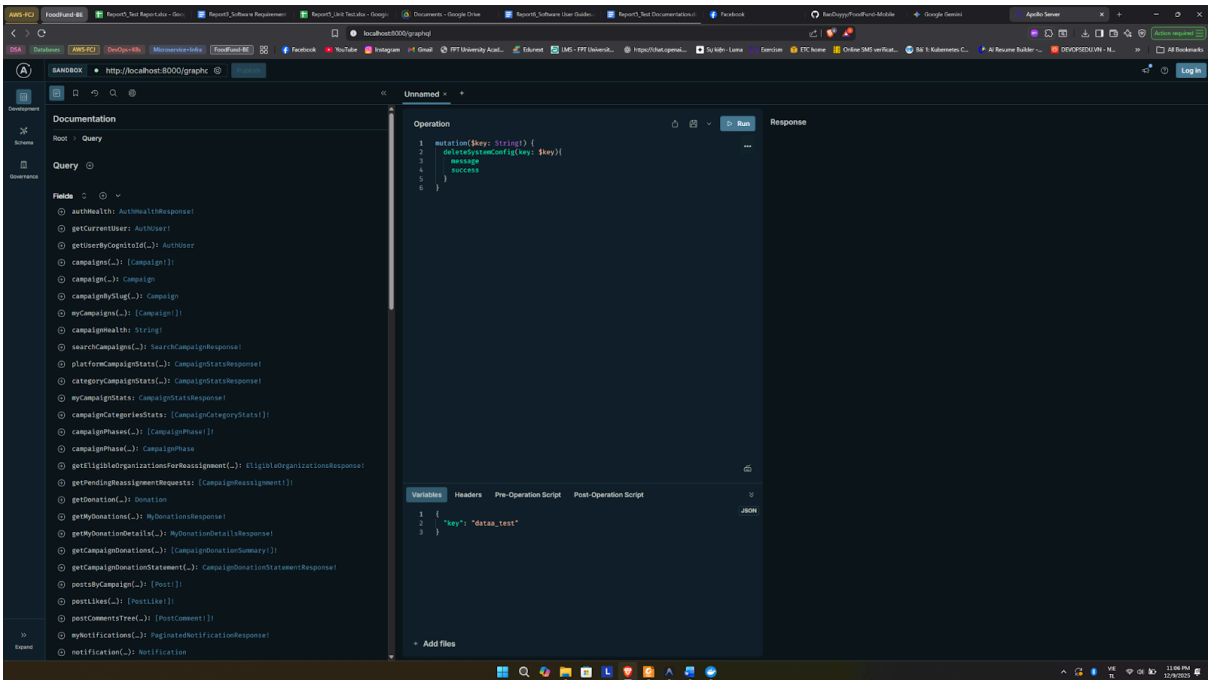


2.2.2 Backend Installation

- Download the source code => run command “git clone https://github.com/phuoctmse/FoodFund-Microservices.git”
- Move to downloaded source code folder => run command “cd FoodFund-Microservices”
- Open source code => run command “code .”
- Create file named “.env.docker” and copy the environment variable from “.env.example” and fill in the values
- Open Docker Desktop application
- Open terminal in IDE or press (control + `) , then run command ‘’docker compose up -d’’

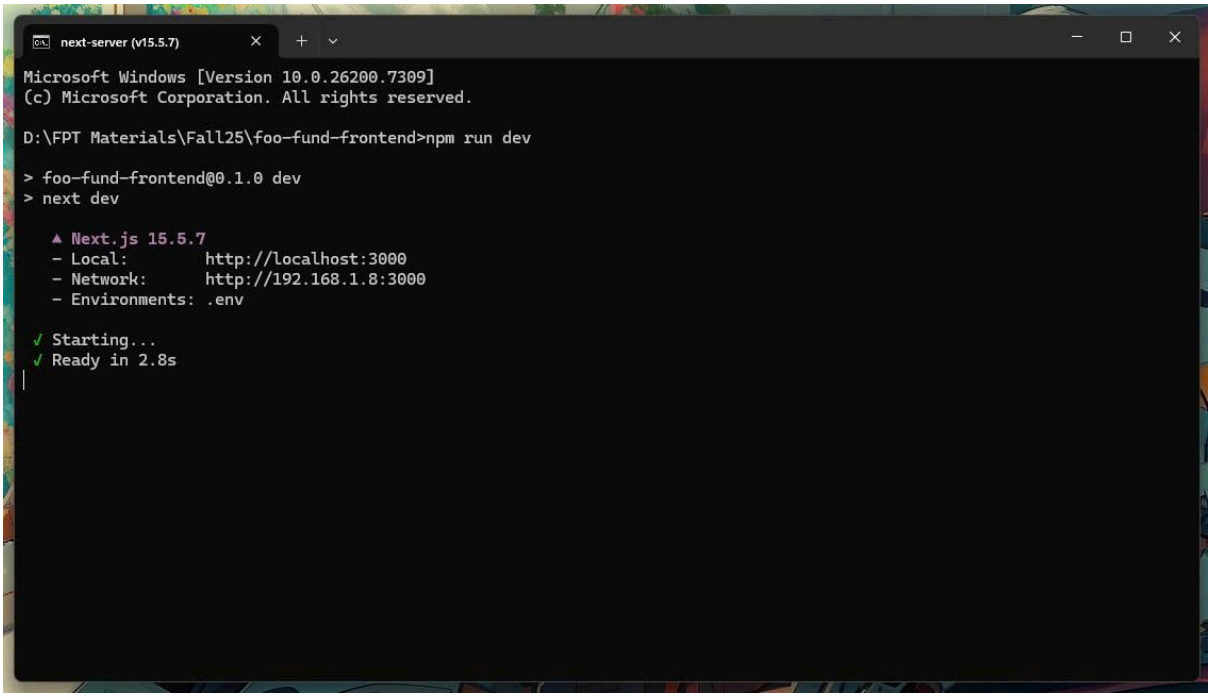


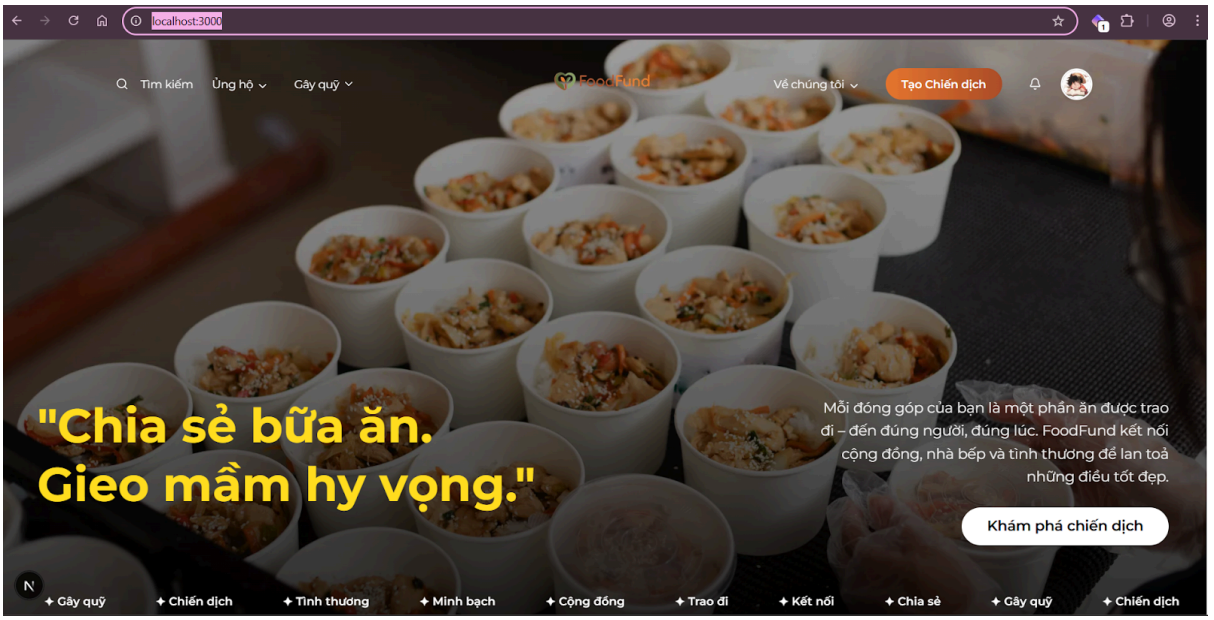
- Run project in development environment, then access http://localhost:8000/graphql to see the api documentation



2.2.3 Frontend Installation (Website)

- Download the source code => run command “git clone https://github.com/bodicode/Food-Fund-Frontend.git”
- Move to downloaded source code folder => run command “cd Food-Fund-Frontend”
- Open source code => run command “code .”
- Create .env to add environment variable
- Install packages for project => run command “npm install’’
- Run project in development environment => run command “npm run dev”, when the command dialog is running like this, click to http://localhost:3000 to open website



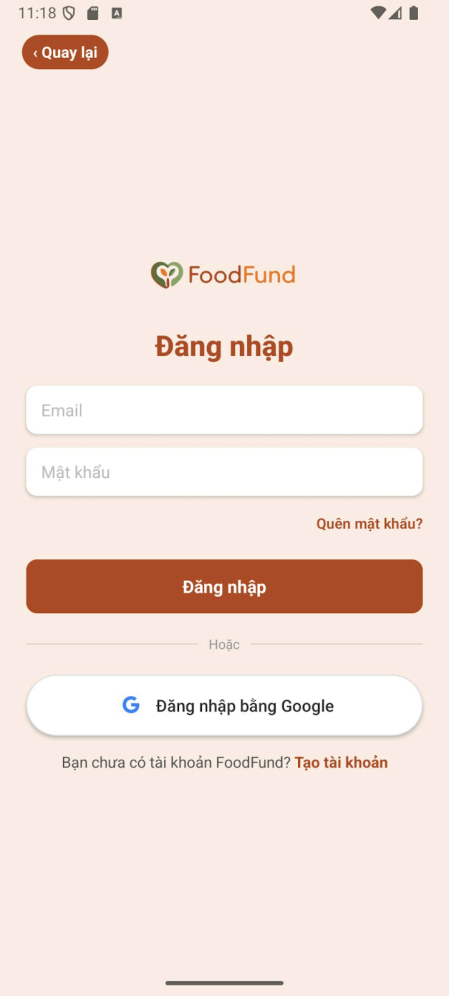


2.2.4 Frontend Installation (Mobile)

- Download the source code => run command “git clone https://github.com/BaoDuyyy/FoodFund-Mobile.git”
- Move to downloaded source code folder => run command “cd FoodFund-Mobile”
- Open source code => run command “code .”
- Create .env to add environment variable
- Install packages for project => run command “npm install’’
- Run project in development environment => run command “npm run start”

```
C:\WINDOWS\system32\cmd.

PS D:\Sem 9\FoodFund_mobile> npm start

> foodfund_mobile@1.0.0 start
> cross-env EXPO_NO_DOTENV_LOAD_LOGS=1 expo start

env: load .env
env: export USERS_DATABASE_URL CAMPAIGN_DATABASE_URL OPERATIONS_DATABASE_URL JWT_SECRET JWT_ACCESS_TOKEN_EXPIRATION JWT_REFRESH_TOKEN_EXPIRATION GRAPHQL_GAT
EWAY_HOST GRAPHQL_GATEWAY_PORT GRAPHQL_GATEWAY_HEALTH_CHECK_URL GRAPHQL_GATEWAY_HEALTH_CHECK_PORT AUTH_HOST AUTH_PORT AUTH_HEALTH_CHECK_URL AUTH_HEALTH_CHEC
K_PORT USERS_SUBGRAPH_HOST USERS_SUBGRAPH_PORT USERS_SUBGRAPH_HEALTH_CHECK_URL USERS_SUBGRAPH_HEALTH_CHECK_PORT CAMPAIGNS_SUBGRAPH_HOST CAMPAIGNS_SUBGRAPH_P
ORT CAMPAIGN_SUBGRAPH_HEALTH_CHECK_URL CAMPAIGN_SUBGRAPH_HEALTH_CHECK_PORT OPERATION_SUBGRAPH_HOST OPERATION_SUBGRAPH_PORT OPERATION_SUBGRAPH_HEALTH_CHECK_U
RL OPERATION_SUBGRAPH_HEALTH_CHECK_PORT AWS_REGION AWS_COGNITO_USER_POOL_ID AWS_COGNITO_CLIENT_ID AWS_ACCESS_KEY_ID AWS_SECRET_ACCESS_KEY AWS_COGNITO_CLIENT
_SECRET AUTH_GRPC_PORT AUTH_GRPC_URL USER_GRPC_PORT USER_GRPC_URL CAMPAIGN_GRPC_PORT CAMPAIGN_GRPC_URL OPERATION_GRPC_PORT OPERATION_GRPC_URL SPACES_KEY SPA
CES_SECRET SPACES_ORIGIN_ENDPOINT SPACES_CDN_ENDPOINT SPACES_BUCKET_NAME SPACES_REGION CORS_ORIGIN AWS_OPENSEARCH_ENDPOINT OPENSEARCH_USERNAME OPENSEARCH_PA
SSWORD AWS_SQS_QUEUE_URL REDIS_HOST REDIS_PORT REDIS_PASSWORD REDIS_USERNAME PAYOS_CLIENT_ID PAYOS_CHECKSUM_KEY PAYOS_API_KEY PAYOS_BANK_NUMBER PAYOS_BANK_N
AME PAYOS_BANK_ACCOUNT_NAME PAYOS_BANK_FULLNAME PAYOS_BANK_LOGO DD_AGENT_HOST DD_AGENT_PORT DD_TRACE_AGENT_PORT DD_ENV DD_VERSION DD_LOGS_INJECTION DD_TRACE
_ENABLED DD_SITE DD_TRACE_SAMPLE_RATE SYSTEM_ADMIN_ID BREVO_API_KEY BREVO_SENDER_EMAIL BREVO_NAME BADGE_FIRST_DONATION_ID BADGE_BRONZE_ID BADGE_SILVER_ID BA
DGE_GOLD_ID BADGE_PLATINUM_ID BADGE_DIAMOND_ID FRONTEND_URL KAFKA_URL
Starting project at D:\Sem 9\FoodFund_mobile
React Compiler enabled
Starting Metro Bundler

› Metro waiting on exp://192.168.1.122:8081
› Scan the QR code above with Expo Go (Android) or the Camera app (iOS)

› Web is waiting on http://localhost:8081

› Using Expo Go
```

- Press “a” to open android emulator or scan qr code to open app in your mobile phone



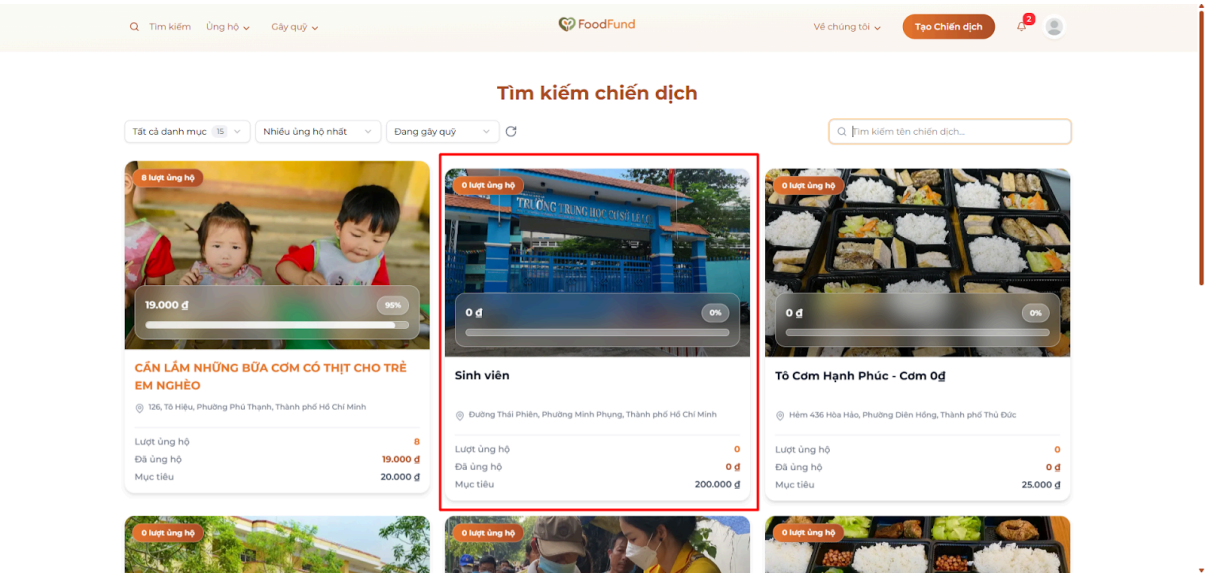
3. User Manual

3.1 Overview

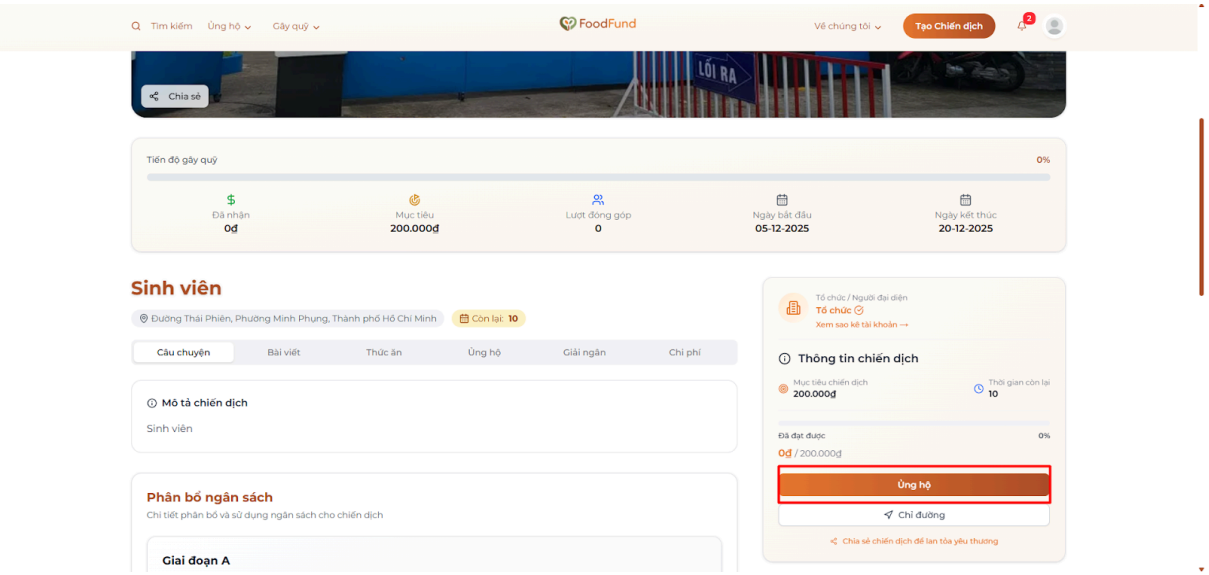
Main flow	Steps
Donation	1. The user clicks on the campaign 2. The user clicks the "Ủng hộ" button 3. The user enters the amount they wish to donate and clicks the "Ủng hộ" button 4. The user scans the QR code to make a donation
User registration	1. User registers an account 2. User checks the email to retrieve the confirmation link 3. User logins with registered account
Create organization	1. User clicks on “Tạo tổ chức” button 2. User uploads a photo of their national ID card, enters organization information and bank account details 3. User submits the organization creation request 4. System reviews organization creation request
Create campaign	1. User clicks on "Tạo chiến dịch" button 2. User enters campaign information 3. User submits campaign creation request 4. System reviews campaign, campaign is published
Operation execution	1. The user creates a request for disbursement of ingredients and operation costs. 2. The system approves and disburses the funds to the organization's bank account. 3. The user confirms receipt of the disbursed funds. 4. Kitchen Staff start on buying and update meal batch. 5. Delivery Staff confirm receiving meal and start delivery to the destination.

3.2 Donation

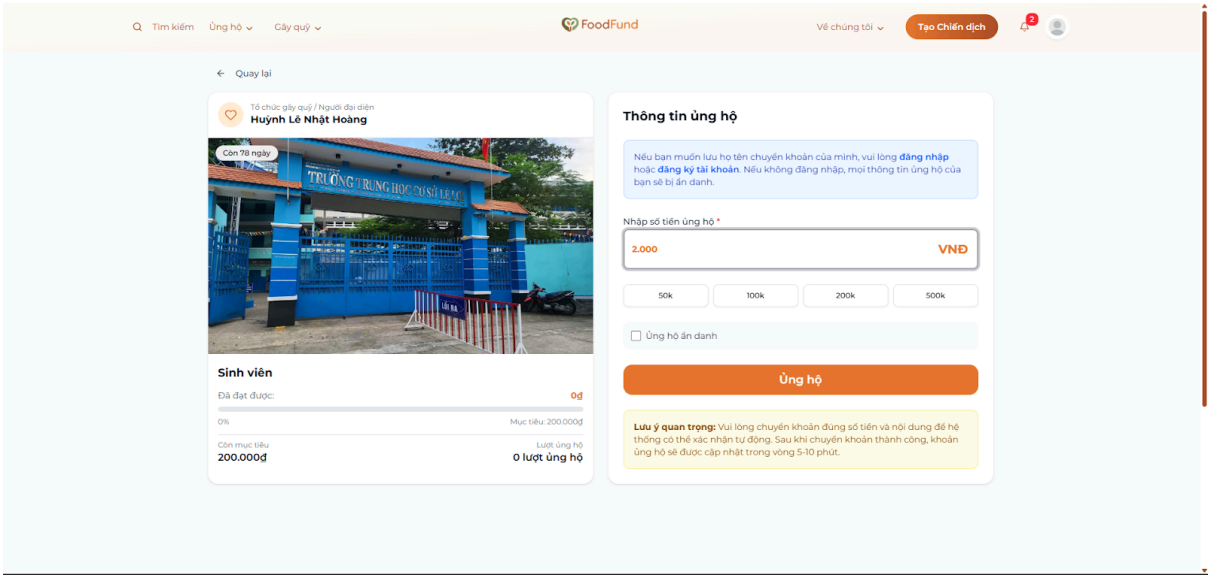
- Step 1: The user clicks on the campaign



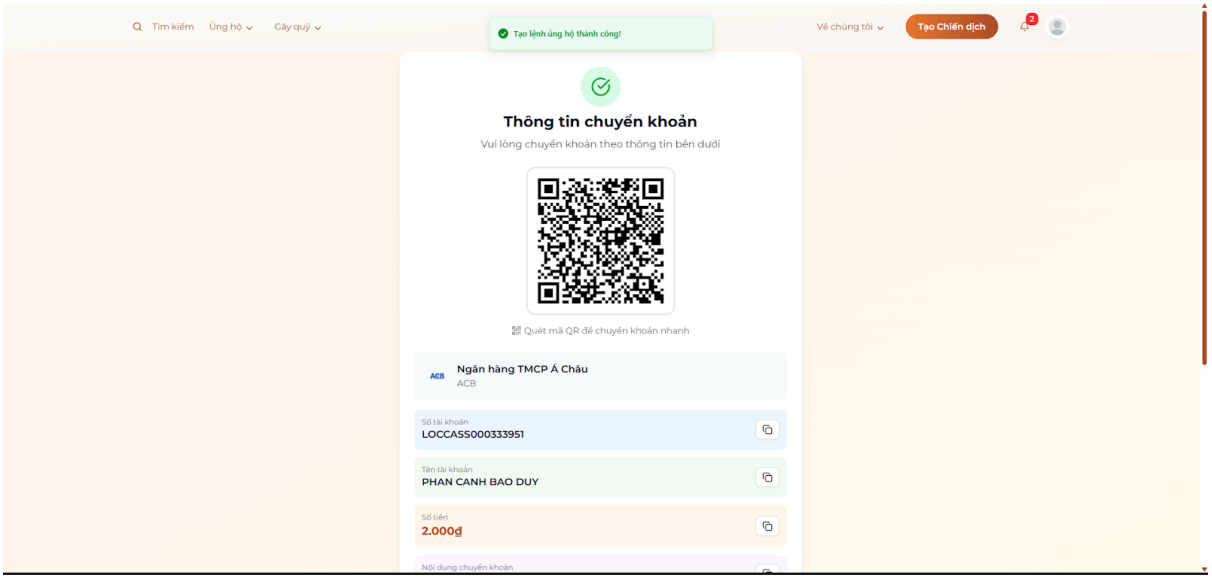
- Step 2: The user clicks the "Ủng hộ" button



- Step 3: The user enters the amount they wish to donate and clicks the "Ủng hộ" button

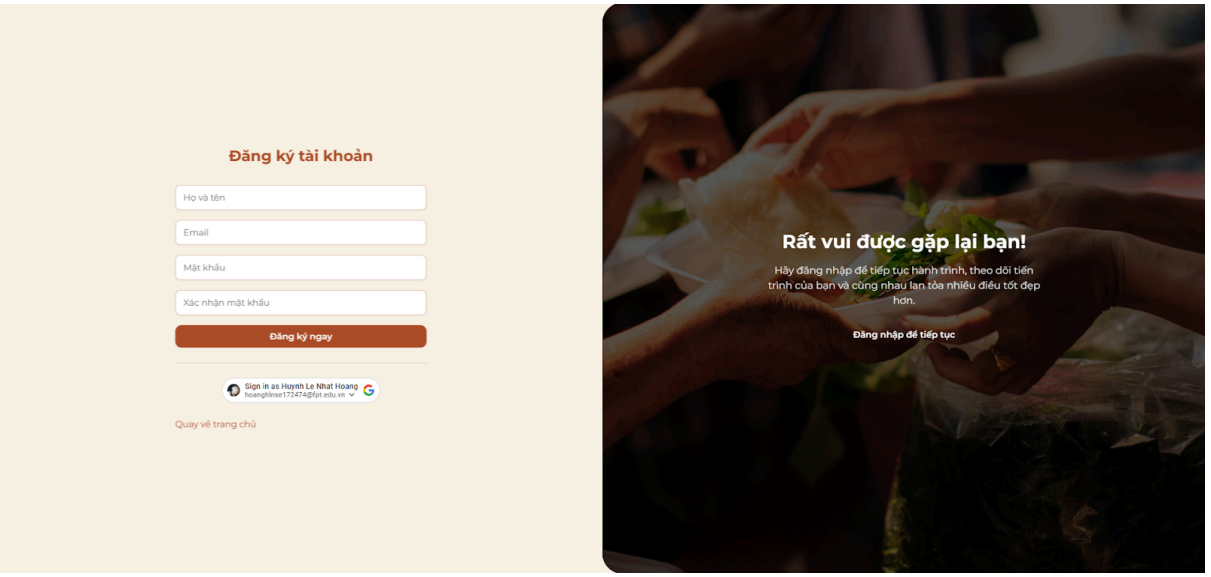


- Step 4: The user scans the QR code to make a donation

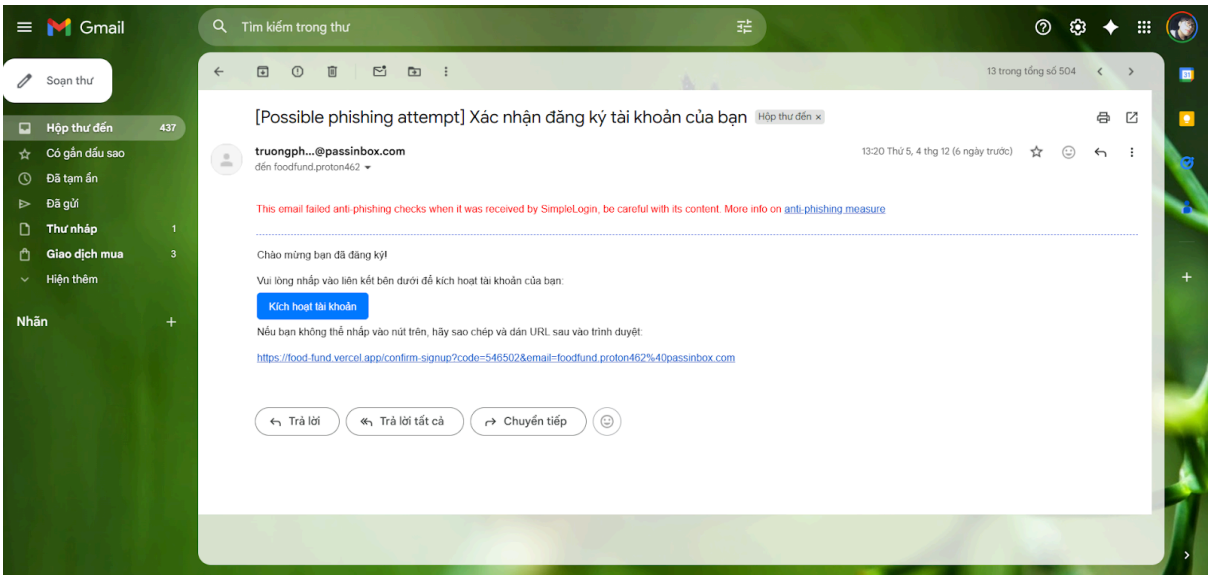


3.3 User registration

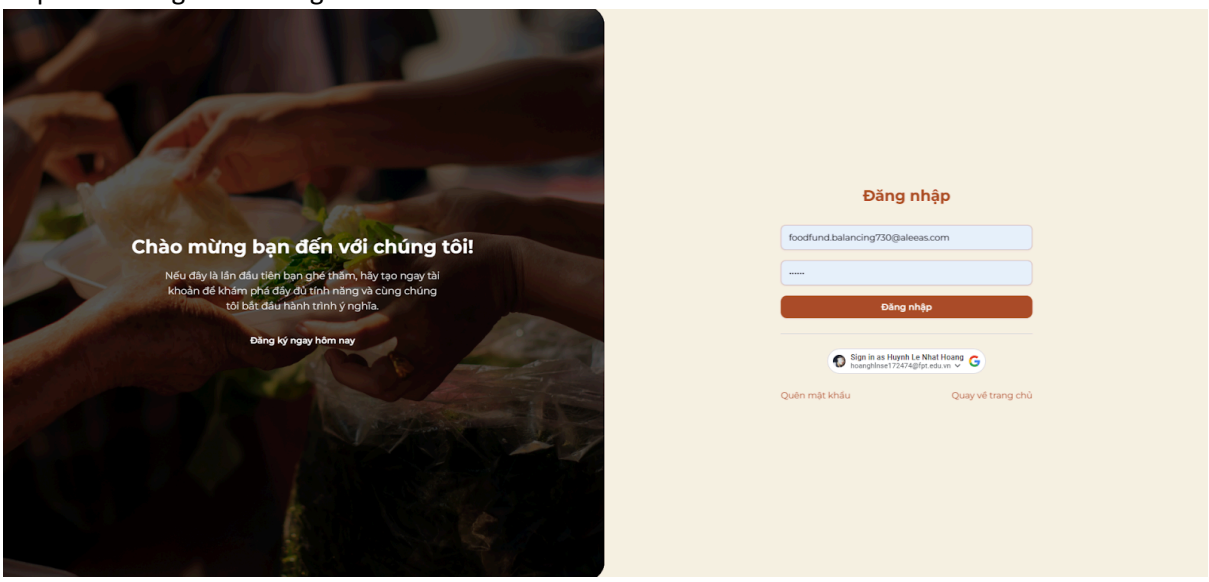
- Step 1: User registers an account



- Step 2: User checks the email to retrieve the confirmation link

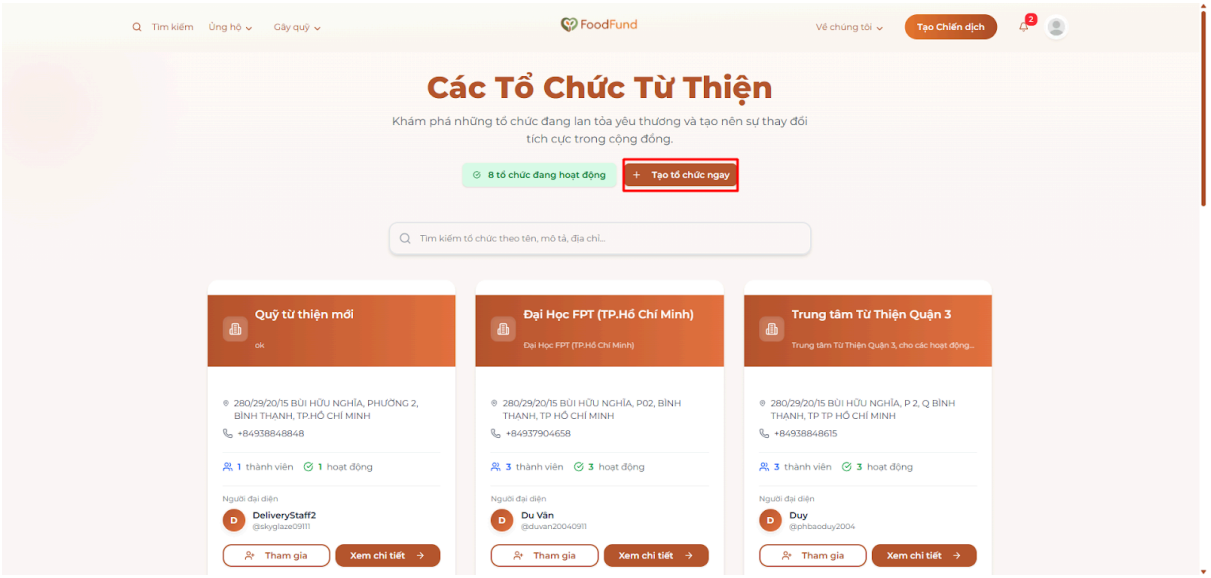


- Step 3: User logins with registered account

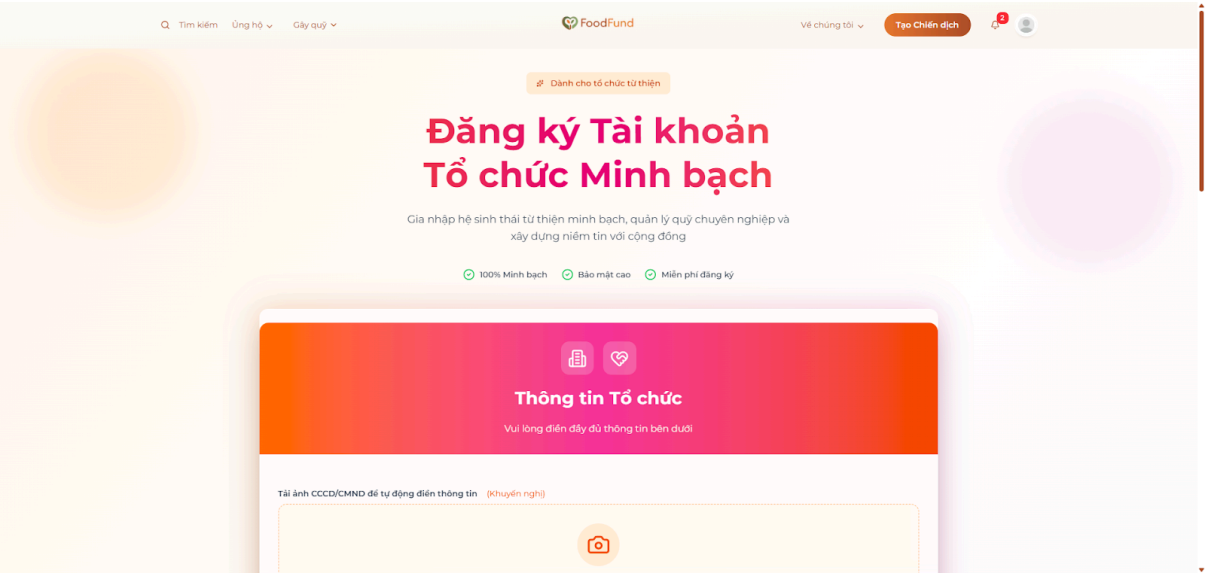


3.4 Create organization

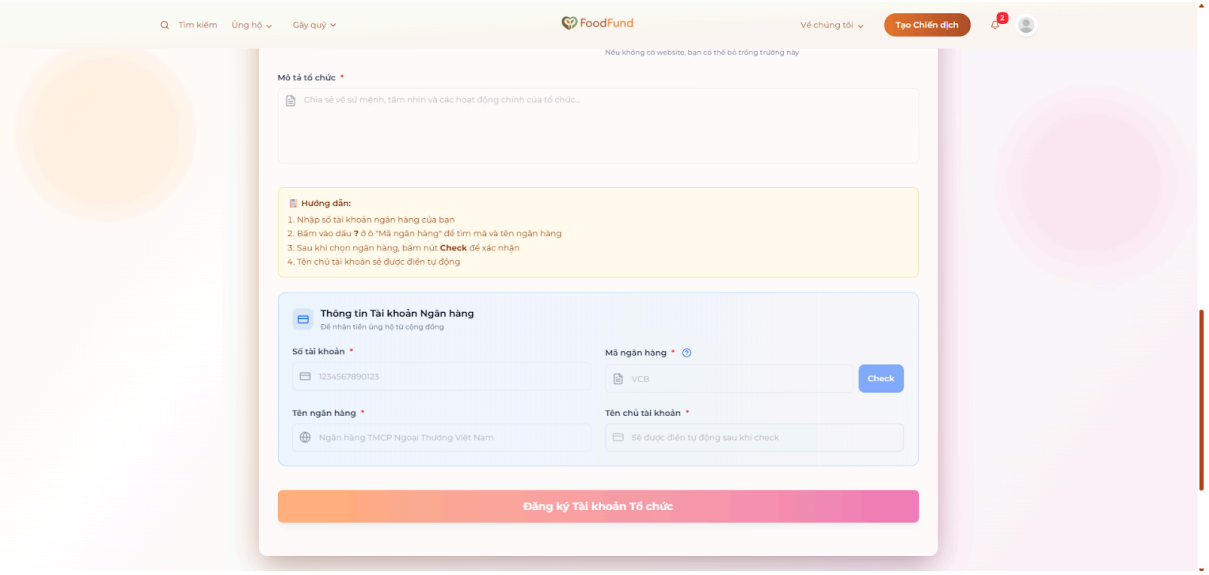
- Step 1: User clicks on “Tạo tổ chức” button



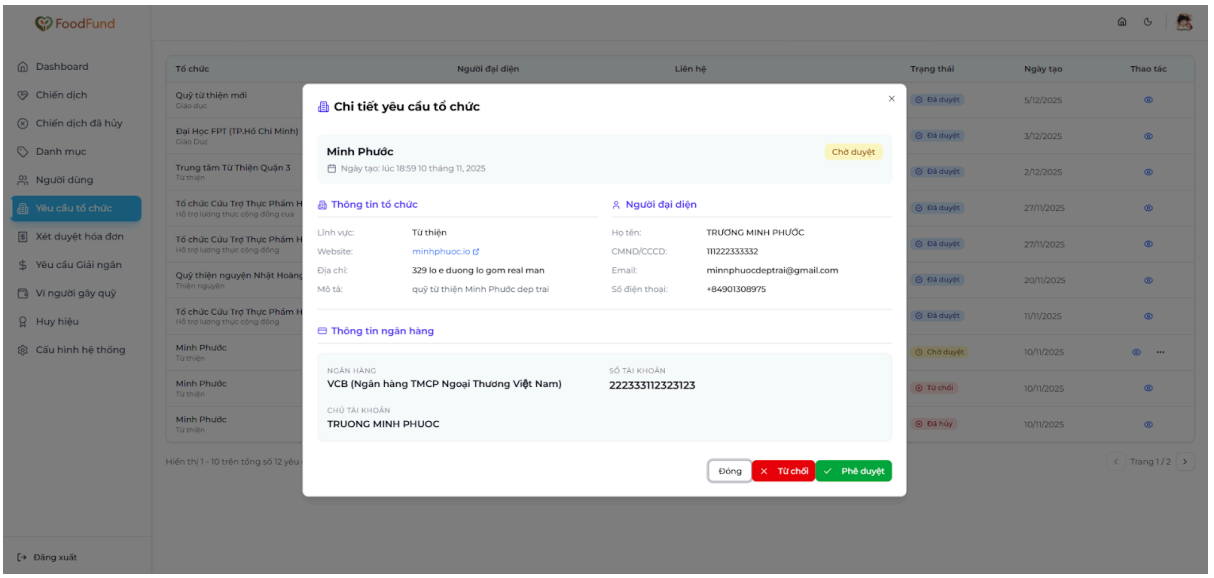
- Step 2: User uploads a photo of their national ID card, enters organization information and bank account details



- Step 3: User submits the organization creation request



- Step 4: System reviews organization creation request

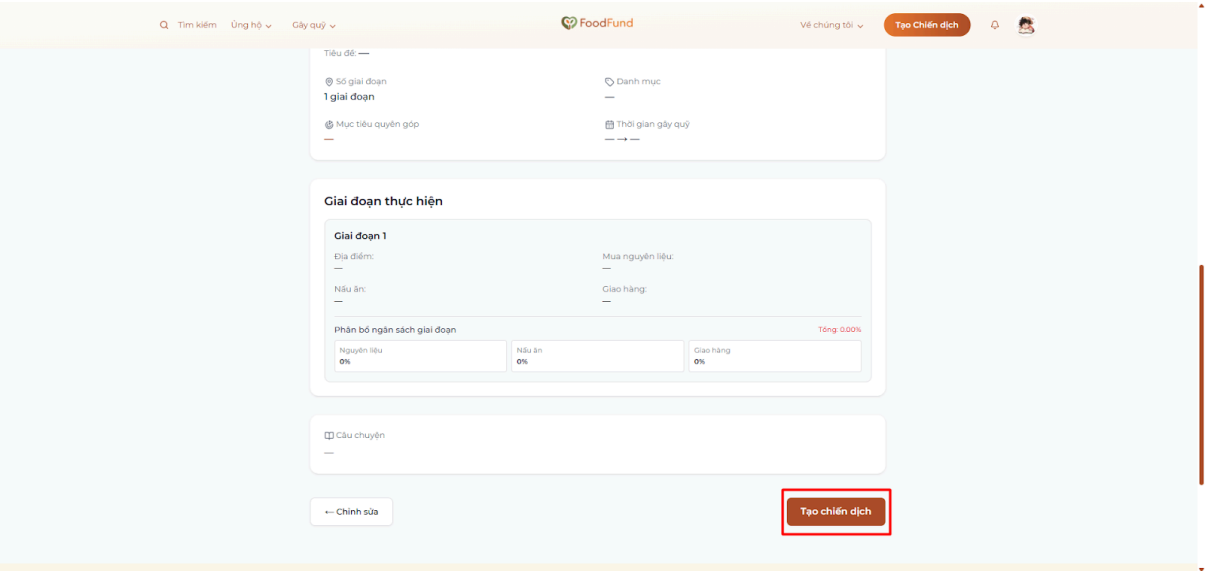


3.5 Create campaign

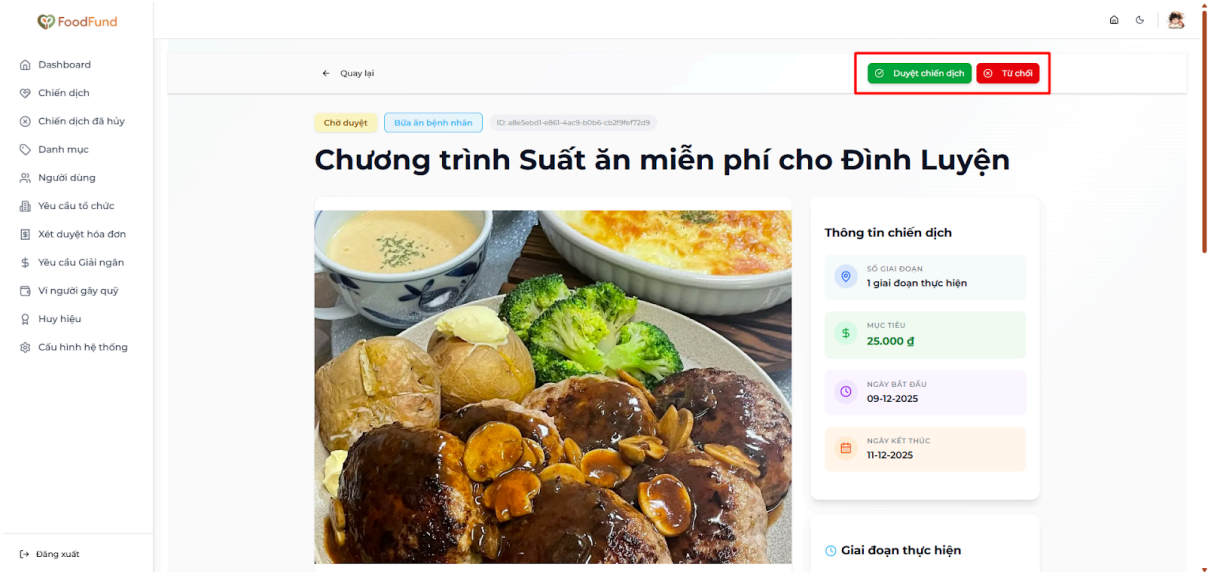
- Step 1: User clicks on "Tạo chiến dịch"

- Step 2: User enters campaign information

- Step 3: User submits campaign creation request

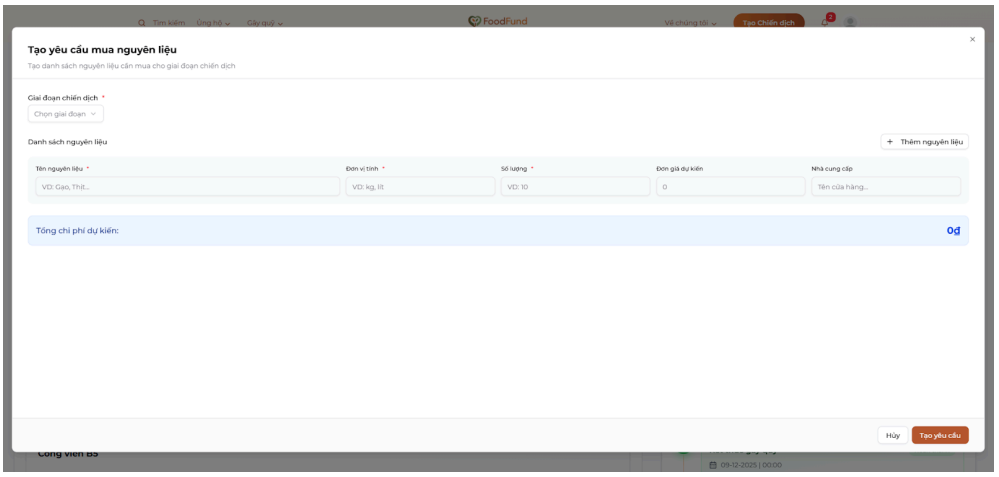


- Step 4: System reviews campaign, campaign is published

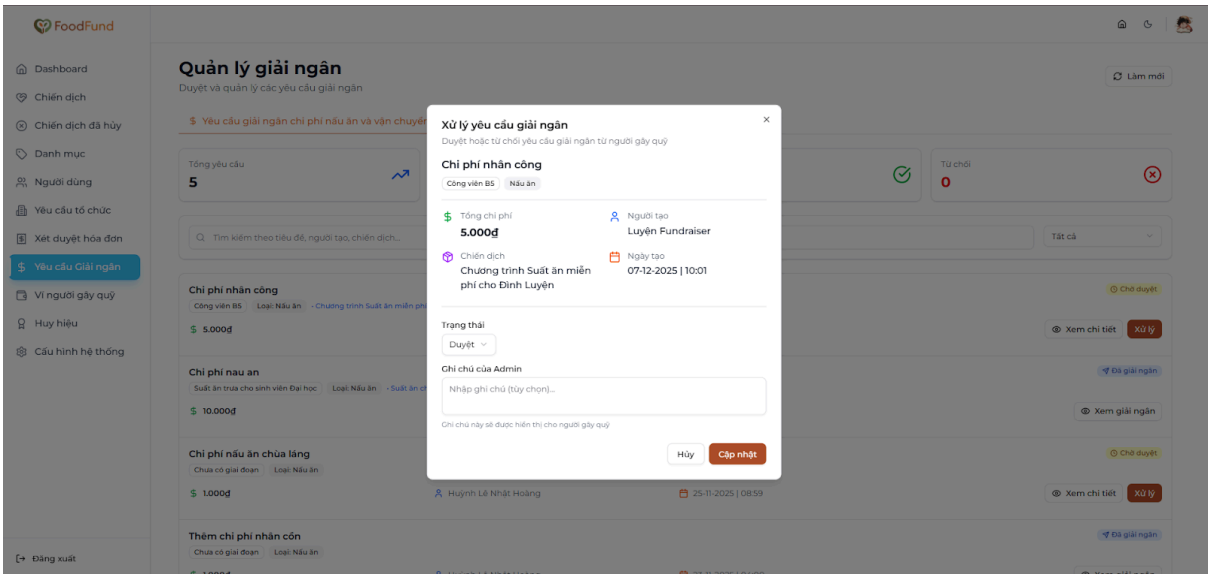


3.6 Operation execution

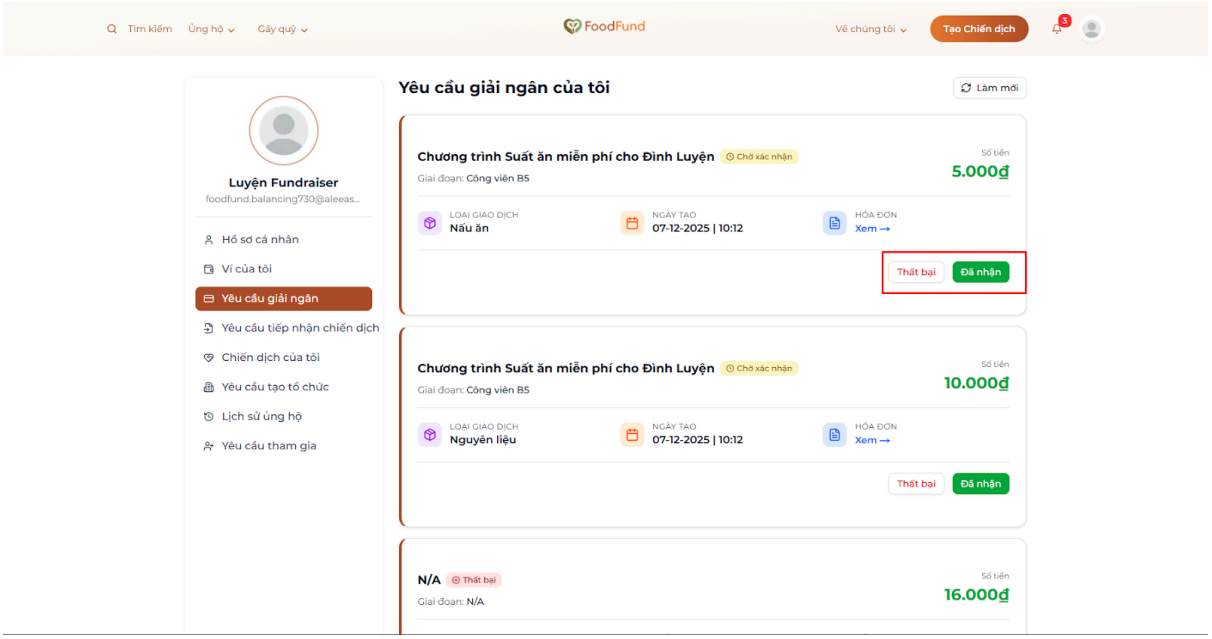
- Step 1: The user creates a request for disbursement of ingredients and operation costs.



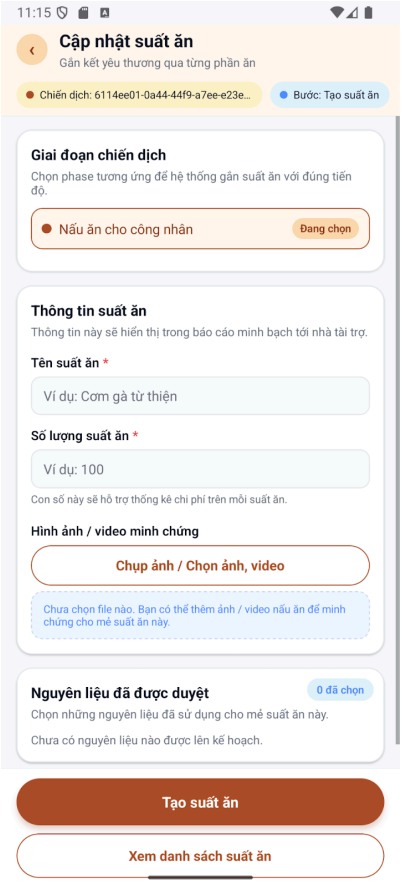
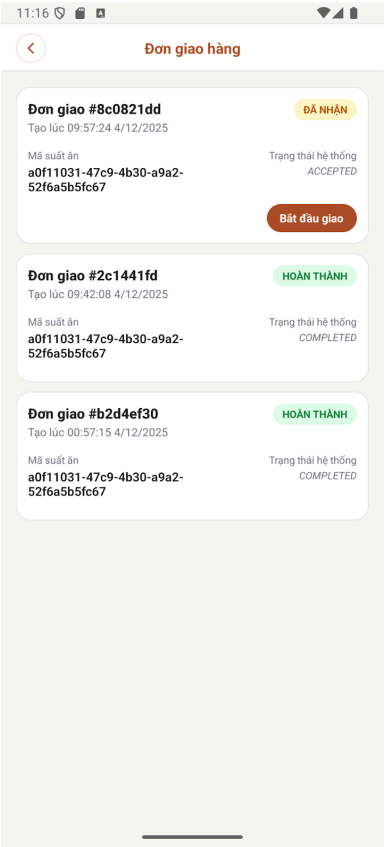
- Step 2: The system approves and disburses the funds to the organization's bank account



- Step 3: The user confirms receipt of the disbursed funds



- Step 4: Kitchen Staff start on buying and updating the meal batch.



- Step 5: Delivery Staff confirm receiving meal and start delivery to the destination.