



**Project Title: OrphanCare – A Digital Platform for Child Adoption and Orphan Care**

**Supervised By**

Rakhi Moni Saha

Lecturer

Department of Software Engineering  
Daffodil International University

**Submitted By**

Sayad Md Asif

ID: 213-35-763

Department of Software Engineering  
Daffodil International University

This Project report has been submitted in fulfillment of the requirements for the Degree of Bachelor of Science in Software Engineering.

**© All right Reserved by Daffodil International University**

## APPROVAL

This project titled on “OrphanCare – A Digital Platform for Child Adoption and Orphan Care”, submitted by Sayad Md Asif (ID: 213-35-763) to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Software Engineering and approval as to its style and contents.

### BOARD OF EXAMINERS



**Dr. Imran Mahmud**  
**Professor & Head**

Department of Software Engineering  
Faculty of Science and Information Technology  
Daffodil International University

**Chairman**



**Md Shohel Arman**  
**Assistant Professor**

Department of Software Engineering  
Faculty of Science and Information Technology  
Daffodil International University

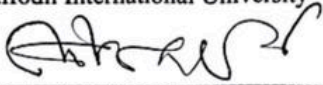
**Internal Examiner 1**



**Md. Rajib Mia**  
**Lecturer (Senior Scale)**

Department of Software Engineering  
Faculty of Science and Information Technology  
Daffodil International University

**Internal Examiner 2**



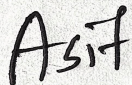
**Md Habibur Rahman**  
**Associate Professor**

Department of Computer Science and Engineering  
Islamic University, Bangladesh

**External Examiner**

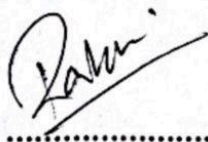
## Declaration

I hereby declare that this project has been completed under the supervision of Rakhi Moni Saha, Lecturer, Department of Software Engineering, Daffodil International University. I also affirm that this project is my original work, submitted for the degree of B.Sc. in Software Engineering, and neither the entire work nor any portion has been previously submitted for another degree at this or any other university.



.....  
Sayad Md Asif  
ID : 213-35-763  
Department of Software Engineering  
Daffodil International University

Certified By:



.....  
Rakhi Moni Saha  
Lecturer  
Department of Software Engineering  
Daffodil International University

## Acknowledgement

I owe my deepest thanks to **Ms. Rakhi Moni Saha** for steady supervision and timely, practical advice. Her questions sharpened the scope of the project and her comments helped turn early ideas into a working system. I am also grateful to the staff of **Hope Haven Orphanage** for opening their doors and explaining how records are maintained, where delays occur, and what information administrators truly need. Those conversations grounded the design in daily realities rather than assumptions.

My classmates and friends deserve recognition for trying out unfinished builds, spotting edge cases, and suggesting clearer messages and simpler flows. Their feedback made the application easier to use for both staff and adopters.

Finally, I am thankful to my parents for constant encouragement and patience during long evenings of coding and testing. Their confidence helped me deal with setbacks and continue refining the solution until it met the objectives of this final-year project.

## Abstract

**OrphanCare** is a browser-based application that digitises routine tasks in orphanages: maintaining child records, handling adoption requests, and sharing announcements. The platform supports two roles. **Administrators** create and update children's profiles, review and decide on adoption requests, post news, and keep the system in good order. **Adopters/Users** can browse children, submit an adoption request, follow the status of their submission, and manage a simple personal profile.

The implementation uses **PHP 8** for server logic and **MySQL 8** as the data store, deployed on a low-cost **LAMP** host. Choosing familiar, open-source tools reduces expense and makes the system easy to maintain on shared hosting commonly available in Bangladesh.

By replacing scattered paper files with a single secure database, the application lowers administrative effort and speeds up decision-making. The main modules **Child Registry**, **Adoption Request Management**, **News Publishing**, **Profile Management**, and light **Reporting** - were designed with short forms, clear validation, and responsive pages to suit both desktops and mobile devices.

While the current release focuses on core needs, the architecture allows surface-level extensions such as SMS/email notifications, a small analytics dashboard, and multilingual content. In summary, OrphanCare offers a practical, scalable step toward transparent and efficient orphanage administration, improving how information flows among staff, prospective parents, and the wider community.

## Table of Content

<b>Acknowledgement.....</b>	<b>iii</b>
<b>Abstract.....</b>	<b>iv</b>
<b>Chapter 1: Introduction.....</b>	<b>10</b>
1.1 Project Overview and Purpose.....	10
1.2 Project Planning and Initiation.....	10
Phase 1: Preliminary Analysis & Project Scope.....	11
1. Problem Statement.....	11
2. Proposed Solution.....	11
3. Project Scope.....	11
In-Scope Features.....	11
Out-of-Scope Features.....	11
Phase 2 : Market Feasibility.....	11
Phase 3 : Technical Feasibility.....	12
Phase 4 : Financial Feasibility.....	13
1.3 Target User Profile and Tentative Elicitation.....	13
1.4 System Requirements.....	14
1.5 Project Scheduling.....	15
a. Time Frame / Gantt Chart (30 Weeks).....	15
b. Risk Management.....	16
<b>Chapter 2: Design and Implementation.....</b>	<b>17</b>
2.1 .1 Functional Requirements.....	17
2.3 System Design.....	20
a. Use case Diagram.....	20
b : Case Description.....	22
Case Description-01: User Registration.....	22
Case Description-02: Login.....	23
Case Description-03: Add Child Details.....	24
Case Description-04: Edit Child Details.....	24
Case Description-05: Delete Child Details.....	25
Case Description-06 : View Child Details.....	26
Case Description-07: Send Adoption Request.....	27
Case Description-08: View All Adoption Requests.....	28
Case Description-09: Accept Adoption Request.....	28
Case Description-10: Reject Adoption Request.....	29
Case Description-11: View News.....	30
Case Description-12: Add News.....	30
Case Description-13 : Delete News.....	31

Case Description-14: View Profile.....	32
Case Description-15: Update Profile.....	33
Case Description- 16: Forgot Password.....	33
Case Description- 17: Log Out.....	34
c. Activity Diagram.....	35
Activity Diagram-01: User Registration.....	35
Activity Diagram-02: Login.....	36
Activity Diagram-03: Add Child Details.....	37
Activity Diagram-04: Edit Child Details.....	37
Activity Diagram-05: Delete Child Details.....	38
Activity Diagram-06 : View Child Details.....	39
Activity Diagram-07: Send Adoption Request.....	40
Activity Diagram-08: View All Adoption Requests.....	41
Activity Diagram-09: Accept Adoption Request.....	41
Activity Diagram-10: Reject Adoption Request.....	42
Activity Diagram-11: View News.....	43
Activity Diagram-12: Add News.....	43
Activity Diagram-13 : Delete News.....	44
Activity Diagram-14: View Profile.....	44
Activity Diagram-15: Update Profile.....	45
Activity Diagram- 16: Forgot Password.....	46
Activity Diagram- 17: Log Out.....	46
d. Sequence Diagram.....	47
Sequence Diagram-01: User Registration.....	47
Sequence Diagram-02: Login.....	48
Sequence Diagram-03: Add Child Details.....	49
Sequence Diagram-04: Edit Child Details.....	50
Sequence Diagram-05: Delete Child Details.....	51
Sequence Diagram-06 : View Child Details.....	52
Sequence Diagram-07: Send Adoption Request.....	52
Sequence Diagram-08: View All Adoption Requests.....	53
Sequence Diagram-09: Accept Adoption Request.....	54
Sequence Diagram-10: Reject Adoption Request.....	54
Sequence Diagram-11: View News.....	55
Sequence Diagram-12: Add News.....	55
Sequence Diagram-13 : Delete News.....	56
Sequence Diagram-14: View Profile.....	56
Sequence Diagram-15: Update Profile.....	57

Sequence Diagram- 16: Forgot Password.....	58
Sequence Diagram- 17: Log Out.....	58
e . UML Diagram.....	59
f. ERD.....	60
<b>Chapter 3: Software Testing.....</b>	<b>61</b>
3.2 Testing Strategies.....	61
<b>Chapter 4: Deployment and Maintenance.....</b>	<b>62</b>
4.1 Software Release Life Cycle.....	62
<b>Chapter 5: User Manual.....</b>	<b>63</b>
For Adopter.....	63
5.1 User Registration.....	63
5.2 Log in.....	64
5.3 View Child Details.....	64
5.4 Send Adoption Request.....	65
5.5 View News :.....	66
For Admin.....	67
5.6 Dashboard.....	67
5.7 Add Child Details.....	67
5.8 Edit Child Details.....	68
5.9 Delete Child Details.....	69
5.10 View all Adoption Request.....	69
5.11 Update Profile.....	70
5.12 Accept Adoption Request.....	70
5.13 Rejection Adoption request.....	71
5.14 Add News.....	71
5.15 Update news.....	72
5.16 Delete News.....	72
<b>Chapter 6: Project Summary.....</b>	<b>73</b>
6.1 Overview.....	73
6.2 Summary.....	73
6.3 Future Work.....	73
6.4 Conclusion.....	73
<b>References.....</b>	<b>74</b>
<b>Appendix A: Source Code.....</b>	<b>74</b>

## List of Tables

<b>Point</b>	<b>Name of table</b>	<b>Page no.</b>
Table 1.2.1	Feasibility Study	10
Table 1.2.2	Market Feasibility Assessment	12
Table 1.2.3	Technical Feasibility	12
Table 1.2.4	Financial Feasibility	13
Table 1.3.1	Target User Profile: Admin/Staff	14
Table 1.3.2	Target User Profile: Adopter/Prospective Parent	14
Table 1.4.1	System Requirements	15
Table 1.4.2	User Environment Requirements	15
Table 1.5.1	Risk Management	16
Table 2.1.1	Functional Requirements catalogue	17 - 19
Table 2.2.1	Non-Functional Requirements	20
Table 3.1.1	Use-Case “CaseDescription” tables (C-01 ... C-017)	22 - 35
Table 3.3.1	System Testing	61-62

## List of Figures

<b>Point</b>	<b>Name of figure</b>	<b>Page no.</b>
Figure 1.6.1	Gantt Chart (30-week plan)	16
Figure 2.3.1	Use-Case Diagram	21
Figures 2.3.2.a–q	Activity Diagrams 01–17	35-46
Figures 2.3.3.a–q	Sequence Diagrams 01–17	47-58
Figure 2.3.4	Class Diagram	59
Figure 2.3.5	ER Diagram	60
Figure 4.1.1	SDLC (Software Development Life Cycle)	62

# Chapter 1: Introduction

## 1.1 Project Overview and Purpose

Many orphanages in Bangladesh still depend on handwritten registers and phone calls to run daily activities. This slows the adoption pipeline, makes follow-up difficult, and exposes records to loss or inconsistency.

**OrphanCare** addresses these issues with a small, web-based system that assigns tasks by role and stores everything in one place.

- **Admins** register and update child profiles, review adoption requests, and publish news or notices.
- **Adopters/Users** browse the list of children, submit a request with required details, and track the status online.

By replacing manual registers and scattered spreadsheets with an integrated solution, OMS aims to:

- Speed up adoption processing.
- Reduce clerical workload.
- Improve transparency and trust between orphanages and prospective parents.
- 

## 1.2 Project Planning and Initiation

### 1.2.1 Feasibility Study

Table 1.2.1: Feasibility snapshot for OrphanCare (lens–verdict–highlights)

Lens	Verdict	Highlights
Technical	High	PHP 8, MySQL 8, Bootstrap 5; skills available locally.
Economic	High	Year-1 hosting $\approx$ \$35; most tools open-source.
Operational	Moderate	Staff need 2-hour onboarding; otherwise low disruption.
Market High	High	No comparable Bengali-language orphanage CMS.

## **Phase 1: Preliminary Analysis & Project Scope**

The project began with a structured planning phase that included feasibility studies, scope definition, and requirement gathering from both staff and adopters. The goal was to design a system that is practical, affordable, and easy to maintain within the resource constraints of a typical Bangladeshi orphanage.

### **1. Problem Statement**

Manual registers slow the adoption pipeline, duplicate effort, and make audit trails nearly impossible. Adopters wait weeks for status updates; staff re-enter the same data in multiple notebooks .

### **2. Proposed Solution**

**OMS introduces:**

1. **Role-based login** (Admin, Adopter).
2. **Child module** – add, edit, deactivate child records with photos.
3. **Adoption module** – online request form, admin accept / reject workflow.
4. **News module** – publish orphanage updates for the community.
5. **Reports** – filter requests by date or status for quick oversight.

### **3. Project Scope**

#### *In-Scope Features*

- Secure authentication and password reset
- CRUD operations for children, news, and adoption requests
- Soft-delete policy to keep historical data
- Dashboard metrics (total children, pending requests, etc.)
- Mobile-responsive interface

#### *Out-of-Scope Features*

- Multi-orphanage tenancy
- SMS or WhatsApp notifications
- Online donation or payment gateway
- AI-based matching or recommendation engine
- Real-time chat or video-call integration

## Phase 2 : Market Feasibility

The market study determines whether an orphanage-focused web application can gain real traction in Bangladesh. The analysis looks at the size of the opportunity, unmet needs, existing alternatives, regulatory climate, and practical adoption drivers such as cost and mobile reach. The outcome shows a clear gap that OMS can fill at very low entry cost.

Table 1.2.2 : Market Feasibility (opportunity size, alternatives, and adoption drivers)

Factor	Evidence (Bangladesh context)	Assessment
<b>Target segment</b>	≈ 1,200 registered child-care organisations; majority still paper-based.	<i>Large, under-served</i>
<b>Unmet need</b>	Faster adoption processing and public transparency requested by NGO Bureau reports.	<i>High</i>
<b>Competitive landscape</b>	No Bengali-language orphanage CMS; foreign SaaS (e.g. ChildTrak) priced >\$600/yr.	<i>Favourable</i>
<b>Regulatory fit</b>	Aligns with the Department of Social Services electronic-record initiative; no licence fee required.	<i>Compliant</i>
<b>Adoption drivers</b>	Low hosting cost (< ₳4,000/yr) and mobile-friendly UI encourage quick uptake.	<i>Strong</i>
<b>Overall viability</b>	Clear gap + low barrier to entry = <b>HIGH</b> market feasibility.	

## Phase 3 : Technical Feasibility

This phase answers the question “*Can we build and run OMS with the tools and skills already available?*”

All chosen components are mature, open-source, and have large support communities, keeping risk low and maintainability high.

Table 1.2.3: Technical Feasibility (Stack & Operations):

Layer / Module	Selected Technology	Feasibility & Rationale
<b>Front-end</b>	HTML 5, Bootstrap 5, vanilla JS	Lightweight; runs on any browser; easy for future interns to maintain.

<b>Back-end</b>	PHP 8 with PDO	Widely supported on low-cost shared hosting; developer skill set already available.
<b>Database</b>	MySQL 8	ACID compliance, simple backups, free; fits ERD.
<b>Image storage</b>	/uploads/ + filename hashing	No external CDN cost; ≤ 1 GB fits typical orphanage.
<b>Hosting</b>	Apache (LAMP)	Bundled with cPanel plans; one-click deploy.
<b>Security</b>	bcrypt password hashing, PDO prepared statements, CSRF tokens	Proven techniques; no specialised library needed.
<b>Integration</b>	PHPMailer for email notice	SMTP only; avoids SMS gateway cost.
<b>Verdict</b>	All components are mature, free, and have local community support → <b>HIGH</b> technical feasibility.	

**Phase 4 : Financial Feasibility**

A quick cost-benefit check shows the solution is affordable for a single orphanage and scales inexpensively as usage grows.

Table 1.2.4 : Financial Feasibility (Cost & Viability):

Cost Item	Qty / Term	Unit Cost (USD)	Sub-Tota
Shared LAMP hosting	1 yr	24	24
.org domain	1 yr	10	10
Optional backup plug-in	1 yr	05	5
Contingency (10 %)	—	—	4
<b>Total Year-1 Budget</b>			<b>\$43</b>

### 1.3 Target User Profile and Tentative Elicitation

Table 1.3.1 : Target User 1: Admin/ Staff

Aspect	Details – Target User 1 (Admin / Staff)
<b>Profile</b>	<ul style="list-style-type: none"> <li>• 3 semi-structured interviews with staff</li> <li>• 2-hour job-shadowing session during intake day</li> </ul>
<b>Goals</b>	Duplicate data entry, misplaced files, slow status enquiries from adopters.
<b>Pain Points</b>	<ul style="list-style-type: none"> <li>• Register children quickly and accurately</li> <li>• Track every adoption request in one place</li> <li>• Publish news / reports without calling a developer</li> </ul>
<b>Elicitation</b>	Social worker or office clerk responsible for child records and adoption paperwork at the orphanage.

Table 1.3.2: Target User 2: Adopter/ Prospective Parent

Aspect	Details – Target User 2 (Adopter / Prospective Parent)
<b>Profile</b>	<ul style="list-style-type: none"> <li>• Online survey (15 questions) posted in adoption forums</li> <li>• Two 30-minute follow-up phone interviews for in-depth insights</li> </ul>
<b>Goals</b>	Repeated trips or phone calls to orphanage, slow manual feedback, uncertainty about document status.
<b>Pain Points</b>	<ul style="list-style-type: none"> <li>• Browse available children 24/7</li> <li>• Submit an adoption request online</li> <li>• Receive timely updates on application status</li> </ul>
<b>Elicitation</b>	Middle-income couple (28 – 45 yrs) with smartphone and reliable internet, interested in adopting a child.

### 1.4 System Requirements

The Orphanage Management System (OMS) is intentionally lightweight so it can run on a typical low-cost shared host and be accessed from any modern browser, even on older PCs used by many orphanages.

Table 1.4.1 : Functional & Non-Functional Summary:

Category	Minimum Specification
Server OS	Linux or Windows Server (shared LAMP/CPanel acceptable)
Web Server	Apache 2.4+ (mod_php)
PHP	Version 8.0 or later with PDO extensions
Database	MySQL 8.x community edition
Memory	512 MB RAM (1 GB recommended)
Storage	1 GB free disk ( $\approx$ 200 child photos + backups)
Client Browser	Chrome 90+, Firefox 80+, Edge 90+, mobile Chrome/Firefox (latest)
Network	1 Mbps uplink minimum; system works well on 3 G mobile data

Table 1.4.2 User Environment Requirements ( Client-Side Setup table):

Category	Requirements
Hardware	Any modern desktop / laptop or Android 8+ smartphone with $\geq$ 1 GB RAM
Software	Modern web browser (see System Requirements)
Network	Stable broadband or 3 G/4 G data connection; average page weight $\approx$ 200 kB

## 1.5 Project Scheduling

### a. Time Frame / Gantt Chart (30 Weeks)

The project is planned over 30 weeks in five phases. Each phase ends with a clear milestone and deliverable.

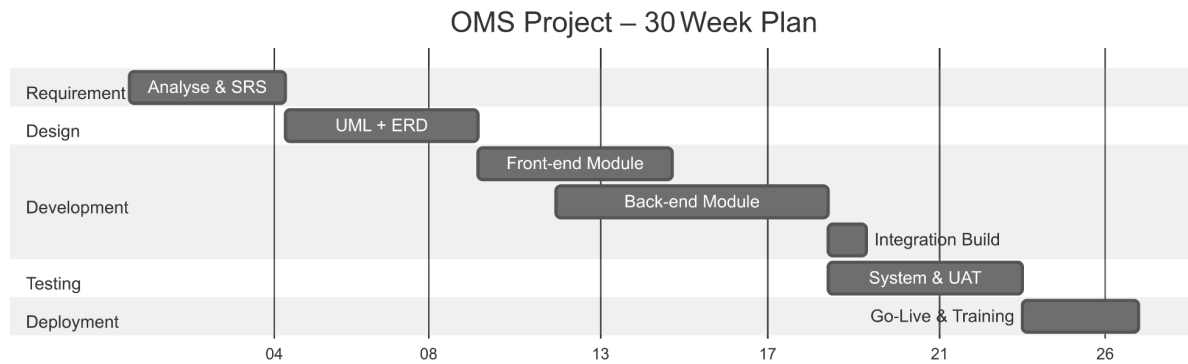


Figure 1.5.1 Gantt Chart (30-week plan)

### b. Risk Management

Main risks include hosting outages, security vulnerabilities, scope creep, admin turnover, and staff resistance. Mitigation includes regular backups, secure coding practices, requirement freeze after SRS, backup admin accounts, and simple training.

Table 1.5.1 : Risks, Mitigations, Owners & Status table:

ID	Risk Description	Mitigation Strategy
R1	Hosting provider outage or data loss.	Schedule nightly automated backups; keep off-site copy on Google Drive; enable provider uptime alerts.
R2	SQL-Injection or XSS vulnerability exploited.	Use PDO prepared statements, output-escape all user input, run quarterly security scan (OWASP ZAP).
R3	Scope creep adds features beyond 30-week timeline.	Freeze requirements after SRS sign-off; log extra ideas in a “v2” backlog for later semesters.
R4	Single admin leaves organisation.	Create procedure to assign a second admin account; document reset steps.
R5	Staff resistance to new system	Create procedure to assign a second admin account; document reset steps.

## Chapter 2: Design and Implementation

### 2.1 .1 Functional Requirements

<b>FR01</b>	<b>User Registration</b>
<b>Description</b>	Provide a sign-up form so visitors can create an adopter account.
<b>Stakeholders</b>	Adopter

<b>FR02</b>	<b>Login</b>
<b>Description</b>	Authenticate users or admins with username + password and open their respective dashboards.
<b>Stakeholders</b>	Adopter, Admin

<b>FR03</b>	<b>Add Child Details</b>
<b>Description</b>	Allow an admin to register a new child profile including photo and basic data.
<b>Stakeholders</b>	Admin

<b>FR04</b>	<b>Edit Child Details</b>
<b>Description</b>	Enable an admin to update existing child information or replace the photo.
<b>Stakeholders</b>	Admin

<b>FR05</b>	<b>Delete Child Details</b>
<b>Description</b>	Let an admin remove (soft-delete) a child record that has not yet been adopted.
<b>Stakeholders</b>	Admin

<b>FR06</b>	<b>View Child Details</b>
-------------	---------------------------

<b>Description</b>	Display a public details page for each active child; show an <b>Adopt</b> button to logged-in users only.
<b>Stakeholders</b>	Adopter

<b>FR07</b>	<b>Send Adoption Request</b>
<b>Description</b>	Permit a logged-in adopter to submit an online adoption request for a selected child.
<b>Stakeholders</b>	Adopter

<b>FR08</b>	<b>View All Adoption Requests</b>
<b>Description</b>	Show the admin a paginated table of all adoption requests with search and sorting.
<b>Stakeholders</b>	Admin

<b>FR09</b>	<b>Accept Adoption Request</b>
<b>Description</b>	Allow the admin to accept a pending adoption request and mark the child as adopted.
<b>Stakeholders</b>	Admin

<b>FR10</b>	<b>Reject Adoption Request</b>
<b>Description</b>	Allow the admin to reject a pending adoption request and record an optional reason.
<b>Stakeholders</b>	Admin

<b>FR11</b>	<b>View News</b>
<b>Description</b>	List news cards on the home page and open the full article on demand.
<b>Stakeholders</b>	Adopter

<b>FR12</b>	<b>Add News</b>
-------------	-----------------

<b>Description</b>	Enable the admin to publish a new news article with optional image.
<b>Stakeholders</b>	Admin

<b>FR13</b>	<b>Delete News</b>
<b>Description</b>	Let the admin remove outdated news items from public view.
<b>Stakeholders</b>	Admin

<b>FR14</b>	<b>View Profile</b>
<b>Description</b>	Show the logged-in user (adopter or admin) their personal profile details.
<b>Stakeholders</b>	Admin, Adopter

<b>FR15</b>	<b>Update Profile</b>
<b>Description</b>	Allow the logged-in user to update profile fields (name, email, phone, etc.).
<b>Stakeholders</b>	Admin, Adopter

<b>FR16</b>	<b>Forgot Password</b>
<b>Description</b>	Terminate the current session and redirect the user to the public home page.
<b>Stakeholders</b>	Adopter, Admin,

<b>FR17</b>	<b>Login</b>
<b>Description</b>	Provide a self-service form for users to reset a forgotten password after verifying
<b>Stakeholders</b>	Adopter, Admin



Table 2.2.1 Non-Functional Requirements

<b>ID</b>	<b>Category</b>	<b>Description</b>
<b>NFR1</b>	Performance	Public pages must load in $\leq 2$ s over 1 Mbps link; adoption-form save $\leq 1$ s.
<b>NFR2</b>	Security	Passwords hashed with bcrypt; all traffic forced over HTTPS; CSRF tokens on every form.
<b>NFR3</b>	Usability	Interface usable on 360 px mobile screen; no action requires more than three clicks from Home.
<b>NFR4</b>	Scalability	System must support 100 concurrent adopters without visible slowdown on a 1 CPU shared host.
<b>NFR5</b>	Reliability	Application uptime $\geq 99$ % per month; failures logged and emailed to admin within 5 min.
<b>NFR</b>	Maintainability	Codebase commented, PSR-12 compliant; new developer can set up dev environment in $< 30$ min.

## 2.3 System Design

### a. Use case Diagram

Primary Actors : Adaptor

Secondary Actors : Admin

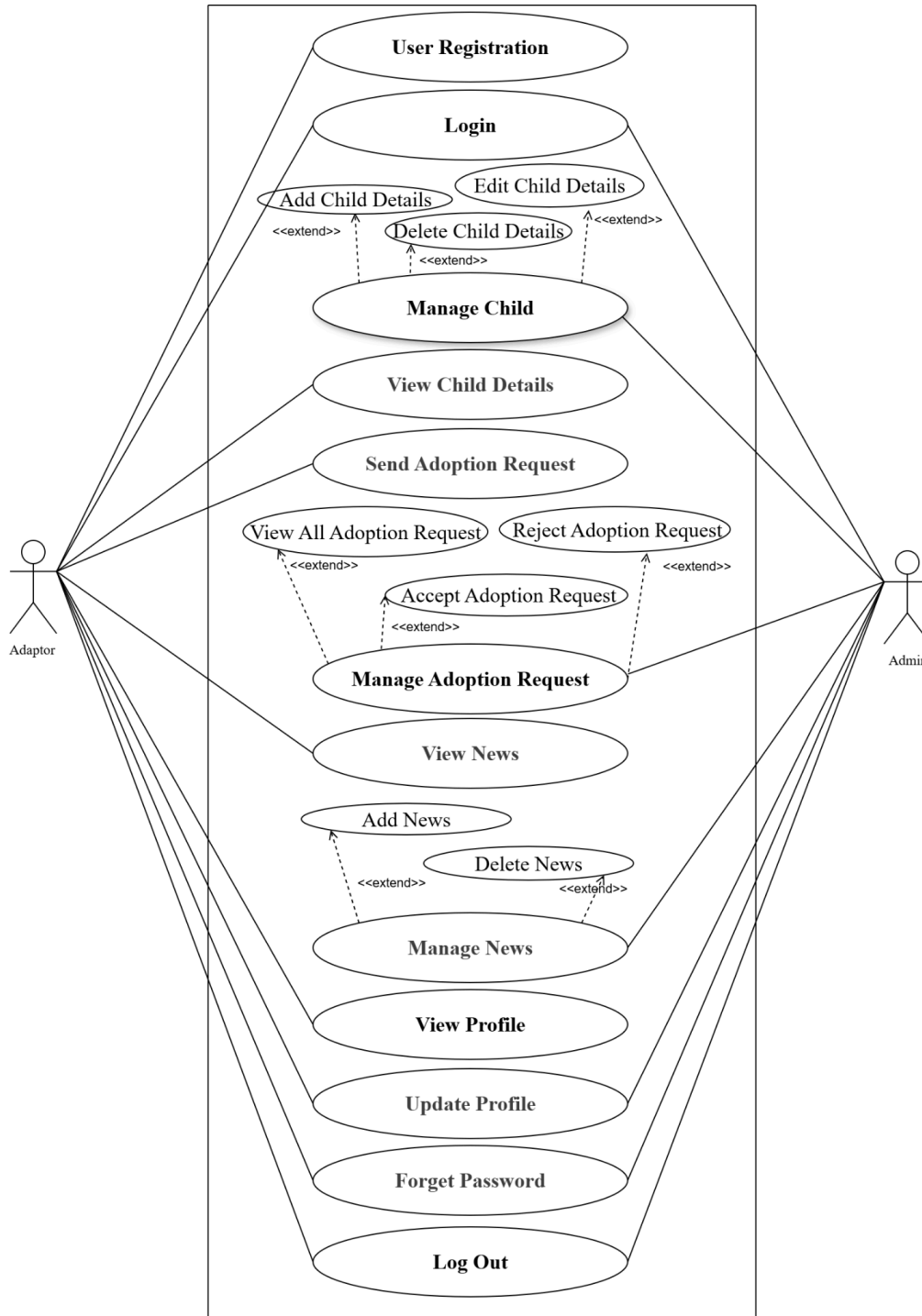


Figure 2.3.1 : Use Case Diagram

## b : Case Description

### Case Description-01: User Registration

<b>Use Case</b>	User Registration
<b>Goal</b>	Let a new visitor open an account so they can log in and request adoptions.
<b>Precondition</b>	The visitor is not signed in.
<b>Success Condition</b> <b>End</b>	Account is stored successfully
<b>Failed End Condition</b>	Data rejected; form shows specific errors.
<b>Primary Actors:</b>	Adoptor, Non Registered User
<b>Secondary Actors:</b>	N/A
<b>Description / Main Success Scenario</b>	<ol style="list-style-type: none"><li>1. Visitor clicks <b>Sign Up</b>.</li><li>2. Fills name, username, email, phone, gender, password.</li><li>3. Website checks format and that username/email are unique.</li><li>4. Data saved; password hashed.</li><li>5. Site redirects to <b>Login</b> page.</li></ol>
<b>Alternative Flows</b>	<b>A1:</b> Email or username already in use → highlight field, stay on form. <b>A2:</b> Weak password (< 6 chars or no digit) → show rule list.
<b>Quality Requirements</b>	<ol style="list-style-type: none"><li>1. Server reply <math>\leq 2</math> s.</li></ol>

## Case Description-02: Login

<b>Use Case</b>	Login
<b>Goal</b>	Authenticate user or admin and open the right dashboard.
<b>Precondition</b>	Account status is <b>Active</b> .
<b>Success End Condition</b>	Secure session starts; role-based menu appears
<b>Failed End Condition</b>	User stays on Login with error notice.
<b>Primary Actors:</b>	Adopter, Admin
<b>Secondary Actors:</b>	N/A
<b>Description / Main Success Scenario</b>	<ol style="list-style-type: none"> <li>1. Adopter enter username &amp; password.</li> <li>2. Click <b>Send</b>. (Sign in)</li> <li>3. Site verify credentials.</li> <li>4. a For normal user → go to Home or Profile.</li> <li>4. b For admin → go to Admin Dashboard.</li> </ol>
<b>Alternative Flows</b>	A1: Password Incorrect : message :”Invalid Details”
<b>Quality Requirements</b>	<ol style="list-style-type: none"> <li>1. 3 wrong tries → 5-minute lockout</li> <li>2. Idle auto-logout after 20 min.</li> <li>3. All traffic over HTTPS.</li> </ol>

### Case Description-03: Add Child Details

<b>Use Case</b>	Add Child Details
<b>Goal</b>	Register a new child profile with photo and basic info.
<b>Precondition</b>	Admin is signed in.
<b>Success Condition</b> <b>End</b>	New child appears in <b>Manage Child</b> list
<b>Failed End Condition</b>	Record not saved; admin sees validation error.
<b>Primary Actors:</b>	Admin
<b>Secondary Actors:</b>	N/A
<b>Description / Main Success Scenario</b>	<ol style="list-style-type: none"> <li>1. Admin selects <b>Child to Add</b>.</li> <li>2. Fills name, gender, arrival date, age, blood group, allergies; uploads JPEG/PNG.</li> <li>3. Website checks required fields and file size (<math>\leq 200</math> KB).</li> <li>4. Image saved to server; DB row created.</li> <li>5. Confirmation banner "<b>Child added.</b>"</li> </ol>
<b>Alternative Flows</b>	<b>A1:</b> Missing mandatory field → highlight field. <b>A2:</b> Photo > 2 MB or wrong type → reject upload.
<b>Quality Requirements</b>	<ol style="list-style-type: none"> <li>1. Save operation <math>\leq 1</math> s</li> <li>2. image auto-resized.</li> </ol>

### Case Description-04: Edit Child Details

<b>Use Case</b>	Edit Child Details
-----------------	--------------------

<b>Goal</b>	Update existing child information.
<b>Precondition</b>	Child record exists and the admin is logged in.
<b>Success Condition</b> <b>End</b>	Data updated; audit timestamp stored.
<b>Failed End Condition</b>	Old data remains.
<b>Primary Actors:</b>	Admin
<b>Secondary Actors:</b>	N/A
<b>Description / Main Success Scenario</b>	<ol style="list-style-type: none"> <li>1. In <b>Manage Child</b>, the admin clicks the blue edit icon.</li> <li>2. Form loads with current data.</li> <li>3. Admin edits fields or uploads a new photo.</li> <li>4. Clicks <b>Save</b>; website validates and updates DB.</li> <li>5. Site shows <b>“Update successful.”</b></li> </ol>
<b>Alternative Flows</b>	<b>A1:</b> Another admin deleted the record meanwhile → site shows “Record not found,” returns to list.
<b>Quality Requirements</b>	Same field checks as Add; changes saved atomically.

#### Case Description-05: Delete Child Details

<b>Use Case</b>	Delete Child Details
<b>Goal</b>	Remove a child profile from public view.
<b>Precondition</b>	Child is not marked <b>Adopted</b> .
<b>Success Condition</b> <b>End</b>	Status flips to <b>Inactive</b> (soft delete).

<b>Failed End Condition</b>	Record still visible.
<b>Primary Actors:</b>	Admin
<b>Secondary Actors:</b>	N/A
<b>Description / Main Success Scenario</b>	<ol style="list-style-type: none"> <li>1. Admin clicks the red <b>Delete</b> icon.</li> <li>2. The website asks “Are you sure you want to delete?”.</li> <li>3. Admin confirms.</li> <li>4. System sets isActive = No.</li> <li>5. Row disappears from list.</li> </ol>
<b>Alternative Flows</b>	<b>A1:</b> Child already adopted → system blocks, shows “Cannot delete adopted profile.”
<b>Quality Requirements</b>	Confirm dialog mandatory; action logged.

#### Case Description-06 : View Child Details

<b>Use Case</b>	View Child Details
<b>Goal</b>	Show information of one child to any visitor.
<b>Precondition</b>	Child is Active.
<b>Success Condition</b> <b>End</b>	Details page renders with photo and bio.
<b>Failed End Condition</b>	404 page shown.
<b>Primary Actors:</b>	Adopter
<b>Secondary Actors:</b>	N/A

<b>Description / Main Scenario</b>	<ol style="list-style-type: none"> <li>1. User clicks a child card in <b>Children</b> list.</li> <li>2. Website loads details page.</li> <li>3. If the user is logged in, <b>the Adopt</b> button is enabled; if not, button is not work/disable.</li> </ol>
<b>Alternative Flows</b>	<b>A1:</b> Visitor clicks grey (disable) button → site redirects to Login.
<b>Quality Requirements</b>	Page load ≤ 2 s; mobile friendly layout.

### Case Description-07: Send Adoption Request

<b>Use Case</b>	Send Adoption Request
<b>Goal</b>	Submit an online form to adopt a specific child.
<b>Precondition</b>	User logged in, profile complete, child not adopted.
<b>Success End Condition</b>	Request stored with status <b>Pending</b> ; user sees unique request #.
<b>Failed End Condition</b>	No record saved.
<b>Primary Actors:</b>	Adopter
<b>Secondary Actors:</b>	N/A
<b>Description / Main Scenario</b>	<ol style="list-style-type: none"> <li>1. User clicks <b>Adopt</b>.</li> <li>2. Form asks address, occupation, monthly income, reason, marital status, and (if married) spouse info.</li> <li>3. User submits; site validates.</li> <li>4. System generates 8-digit Request No.; saves record.</li> <li>5. The website thanks the user and shows the number.</li> </ol>
<b>Alternative Flows</b>	<b>A1:</b> Same user already has a request for this child → site says “Request already on file.”

<b>Quality Requirements</b>	CSRF token on form; save ≤ 1 s.
-----------------------------	---------------------------------

### Case Description-08: View All Adoption Requests

<b>Use Case</b>	View All Adoption Requests
<b>Goal</b>	Show admin every adoption request in one table.
<b>Precondition</b>	Admin signed in.
<b>Success Condition</b> <b>End</b>	Table loads with search, sort, and link to detail view.
<b>Failed End Condition</b>	
<b>Primary Actors:</b>	Admin
<b>Secondary Actors:</b>	N/A
<b>Description / Main Success Scenario</b>	<ol style="list-style-type: none"> <li>1. Admin chooses <b>Adoption Request</b> → <b>All</b>.</li> <li>2. Website returns paginated list.</li> <li>3. Admin can click the file icon to open request details.</li> </ol>
<b>Alternative Flows</b>	N/A
<b>Quality Requirements</b>	Page server-side pagination; each page within 500 ms.

### Case Description-09: Accept Adoption Request

<b>Use Case</b>	Accept Adoption Request
<b>Goal</b>	Approve a pending request and mark the child as adopted.
<b>Precondition</b>	Request status = Pending.

<b>Success End Condition</b>	Request → <b>Accepted</b> ; child → <b>Adopted</b> ; email notice sent.
<b>Failed End Condition</b>	Nothing changed.
<b>Primary Actors:</b>	Admin
<b>Secondary Actors:</b>	N/A
<b>Description / Main Success Scenario</b>	<ol style="list-style-type: none"> <li>1. Admin views request detail.</li> <li>2. Clicks <b>Accept</b>.</li> <li>3. The system updates both tables in one transaction.</li> <li>4. Status cards on Dashboard refresh.</li> </ol>
<b>Alternative Flows</b>	<b>A1:</b> Another admin already accepted → site shows concurrency error.
<b>Quality Requirements</b>	Atomic update; action logged with admin username + timestamp.

### Case Description-10: Reject Adoption Request

<b>Use Case</b>	Reject Adoption Request
<b>Goal</b>	Deny a pending adoption request.
<b>Precondition</b>	Request status = Pending.
<b>Success End Condition</b>	Status becomes <b>Rejected</b> ; adopter emailed.
<b>Failed End Condition</b>	Status unchanged.
<b>Primary Actors:</b>	Admin
<b>Secondary Actors:</b>	N/A

<b>Description / Main Success Scenario</b>	<ol style="list-style-type: none"> <li>1. In request detail, admin clicks <b>Reject</b>.</li> <li>2. Enter a short reason (optional).</li> <li>3. Site updates status and stores reason.</li> </ol>
<b>Alternative Flows</b>	N/A
<b>Quality Requirements</b>	Reason $\leq$ 255 chars; update $\leq$ 1 s.

### Case Description-11: View News

<b>Use Case</b>	View News
<b>Goal</b>	Allow visitors to read news articles.
<b>Precondition</b>	At least one active news post.
<b>Success Condition</b> <b>End</b>	News cards shown; users can open full article.
<b>Failed End Condition</b>	
<b>Primary Actors:</b>	Adopter
<b>Secondary Actors:</b>	N/A
<b>Description / Main Success Scenario</b>	<ol style="list-style-type: none"> <li>1. User scrolls to <b>News &amp; Updates</b>.</li> <li>2. Cards load (title, image, date).</li> <li>3. User clicks <b>Read More</b> to see full content.</li> </ol>
<b>Alternative Flows</b>	N/A
<b>Quality Requirements</b>	Images lazy-load; articles paginated by 5.

### Case Description-12: Add News

<b>Use Case</b>	Add News
<b>Goal</b>	Publish a new news article.
<b>Precondition</b>	Admin signed in
<b>Success Condition</b> <b>End</b>	Article live in News list.
<b>Failed End Condition</b>	
<b>Primary Actors:</b>	Admin
<b>Secondary Actors:</b>	N/A
<b>Description / Main Success Scenario</b>	<ol style="list-style-type: none"> <li>1. Admin chooses <b>News</b> → <b>Add</b>.</li> <li>2. Fills title + body + optional image.</li> <li>3. Clicks <b>Save</b>; site stores and displays post.</li> </ol>
<b>Alternative Flows</b>	N/A
<b>Quality Requirements</b>	Strip risky HTML; limit image ≤ 200 KB.

### Case Description-13 : Delete News

<b>Use Case</b>	Delete News
<b>Goal</b>	Remove outdated news.
<b>Precondition</b>	News exists.
<b>Success Condition</b> <b>End</b>	Post hidden from site.

<b>Failed End Condition</b>	
<b>Primary Actors:</b>	Admin
<b>Secondary Actors:</b>	N/A
<b>Description / Main Success Scenario</b>	<ol style="list-style-type: none"> <li>1. Admin presses delete icon.</li> <li>2. Confirms prompt.</li> <li>3. Site flags post as <b>Inactive</b>.</li> </ol>
<b>Alternative Flows</b>	N/A
<b>Quality Requirements</b>	Confirm dialog; soft delete preferred.

#### Case Description-14: View Profile

<b>Use Case</b>	View Profile
<b>Goal</b>	Show personal data to the logged-in user.
<b>Precondition</b>	The user signed in.
<b>Success End Condition</b>	Profile displayed; username read-only.
<b>Failed End Condition</b>	
<b>Primary Actors:</b>	Adopter, Admin
<b>Secondary Actors:</b>	N/A
<b>Description / Main Success Scenario</b>	<ol style="list-style-type: none"> <li>1. User opens <b>My Account</b> → <b>Profile</b>.</li> <li>2. Website fetches and shows data.</li> </ol>
<b>Alternative Flows</b>	N/A

<b>Quality Requirements</b>	User sees only own record (IDOR safe).
-----------------------------	--

### Case Description-15: Update Profile

<b>Use Case</b>	Update Profile
<b>Goal</b>	Edit personal information.
<b>Precondition</b>	Profile form open.
<b>Success End Condition</b>	Data saved; green “Updated” alert.
<b>Failed End Condition</b>	
<b>Primary Actors:</b>	Admin, Adopter
<b>Secondary Actors:</b>	N/A
<b>Description / Main Success Scenario</b>	<ol style="list-style-type: none"> <li>1. User edits fields.</li> <li>2. Clicks <b>Update</b>.</li> <li>3. Site validates and saves.</li> <li>4. Reloads page with new data.</li> </ol>
<b>Alternative Flows</b>	<b>A1:</b> Duplicate email → error banner.
<b>Quality Requirements</b>	Email RFC 5322; phone numeric 10–15 digits.

### Case Description- 16: Forgot Password

<b>Use Case</b>	Forgot Password
-----------------	-----------------

<b>Goal</b>	Reset password when forgotten.
<b>Precondition</b>	User knows registered email & username.
<b>Success Condition</b> <b>End</b>	Password changed; user redirected to Login.
<b>Failed End Condition</b>	
<b>Primary Actors:</b>	Admin, Adopter
<b>Secondary Actors:</b>	N/A
<b>Description / Main Success Scenario</b>	<ol style="list-style-type: none"> <li>1. User clicks <b>Forgot Password</b>.</li> <li>2. Enters email, username, new pass, confirm.</li> <li>3. Site verifies match, updates pass hash.</li> <li>4. Shows “Password reset—please sign in.”</li> </ol>
<b>Alternative Flows</b>	N/A
<b>Quality Requirements</b>	New pass $\geq$ 6 chars with digit + letter.

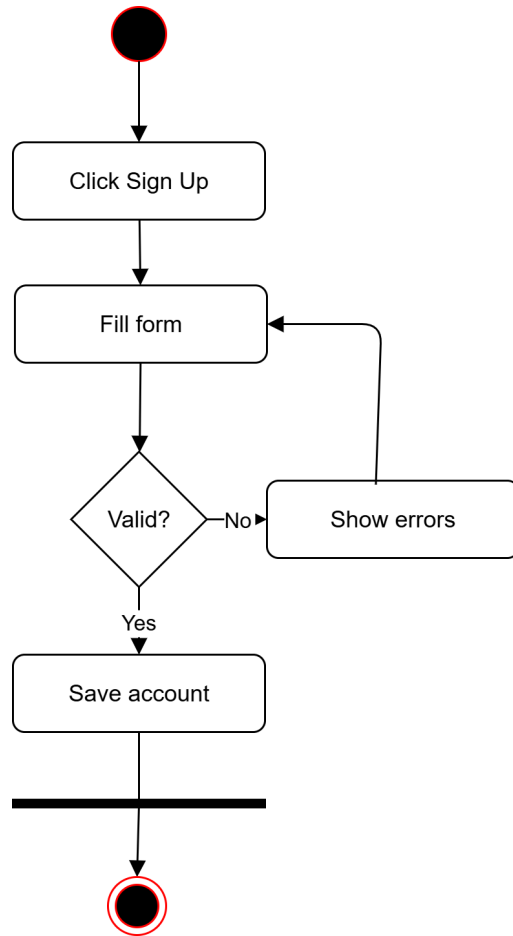
### Case Description- 17: Log Out

<b>Use Case</b>	Log Out
<b>Goal</b>	End the current session.
<b>Precondition</b>	User logged in.
<b>Success Condition</b> <b>End</b>	Session destroyed; user back to public home.
<b>Failed End Condition</b>	

<b>Primary Actors:</b>	Adopter, Admin
<b>Secondary Actors:</b>	N/A
<b>Description / Main Success Scenario</b>	<ol style="list-style-type: none"> <li>1. User clicks <b>Logout</b>.</li> <li>2. Site clears session cookies.</li> <li>3. Home shows <b>Login</b> links again.</li> </ol>
<b>Alternative Flows</b>	N/A
<b>Quality Requirements</b>	All tokens revoked; no back-button access to private pages.

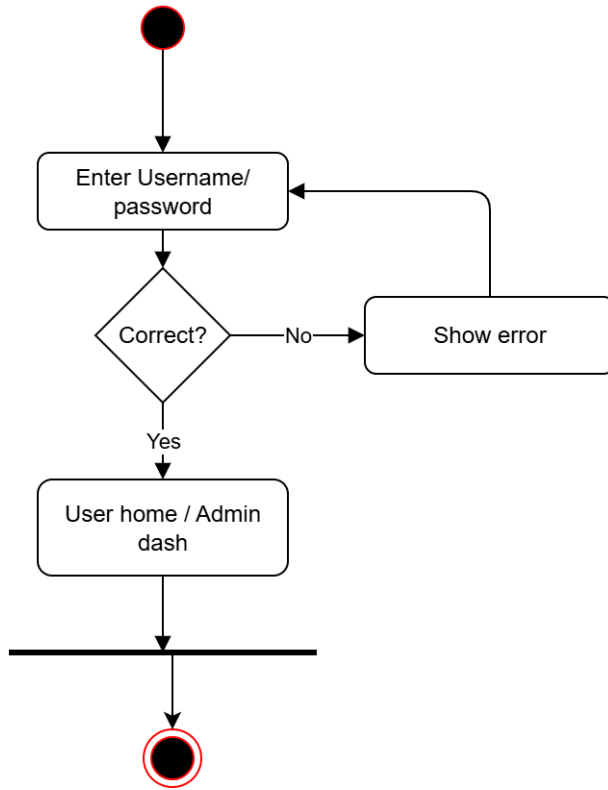
### c. Activity Diagram

#### Activity Diagram-01: User Registration



*Figure 2.3.2.01 : User Registration*

## Activity Diagram-02: Login



*Figure 2.3.2.02 : Login*

### Activity Diagram-03: Add Child Details

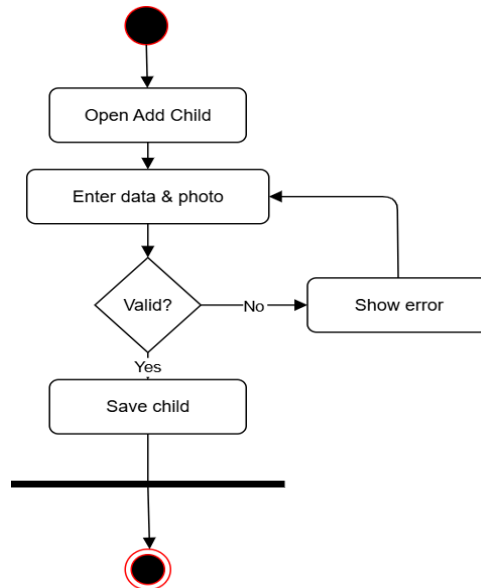


Figure 2.3.2.03 : Add Child Details

### Activity Diagram-04: Edit Child Details

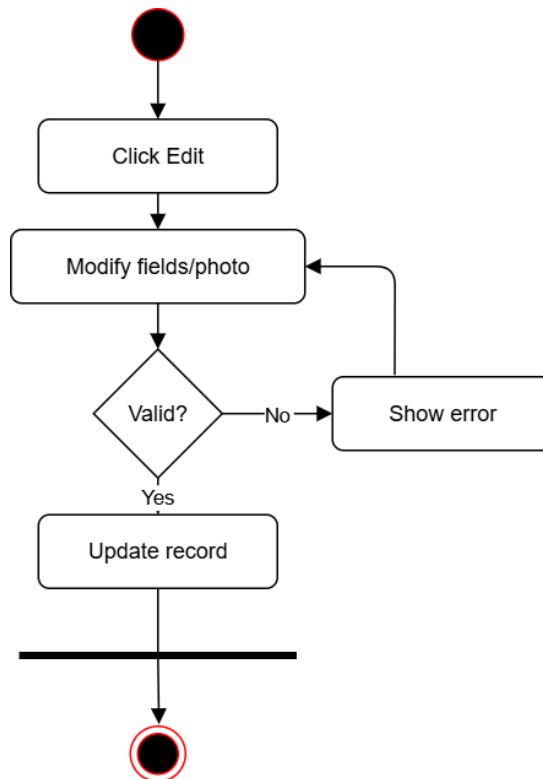
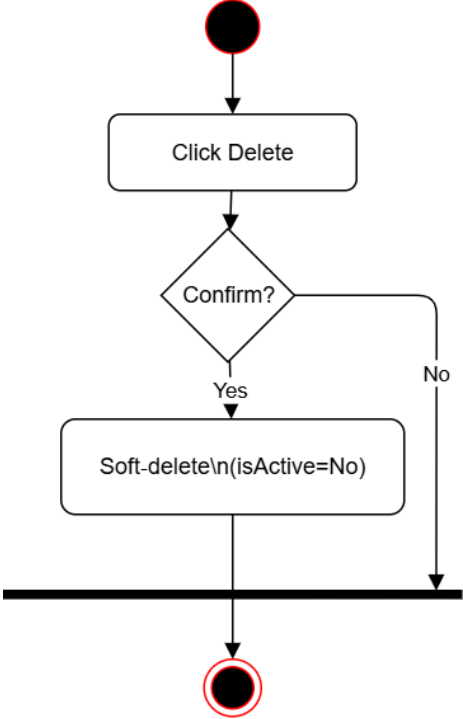


Figure 2.3.2.04 : Edit Child Details

**Activity Diagram-05: Delete Child Details**



*Figure 2.3.2.05: Delete Child Details*

### Activity Diagram-06 : View Child Details

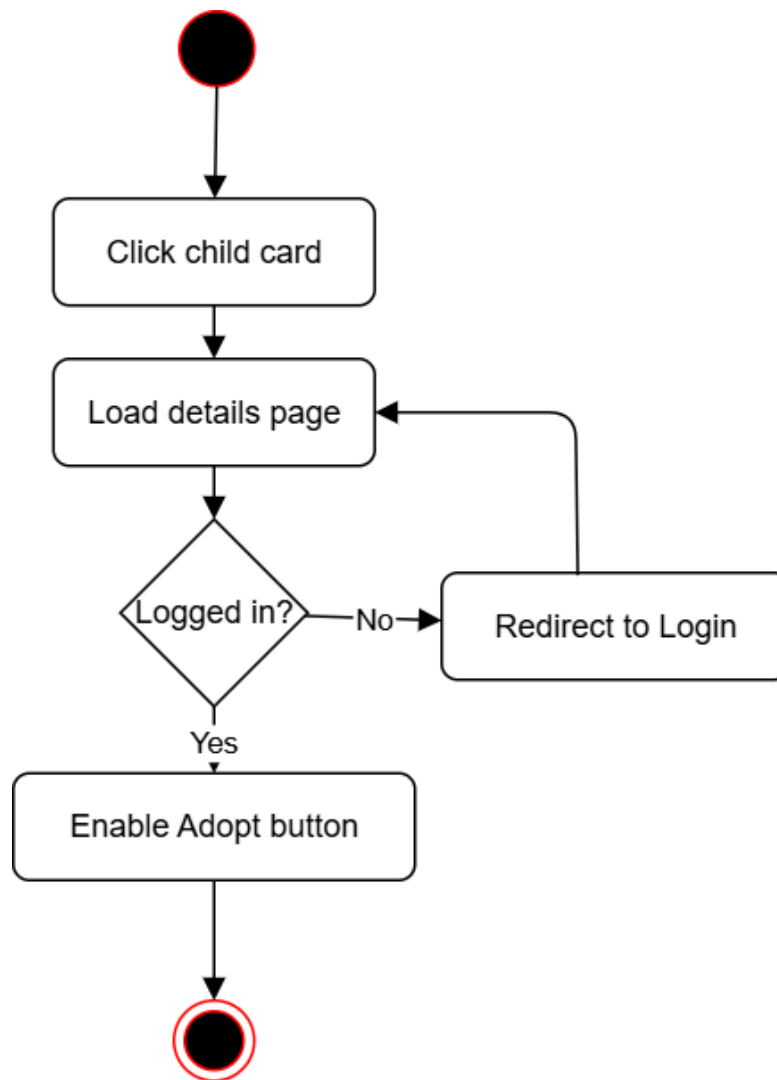


Figure 2.3.2.06: View Child Details

### Activity Diagram-07: Send Adoption Request

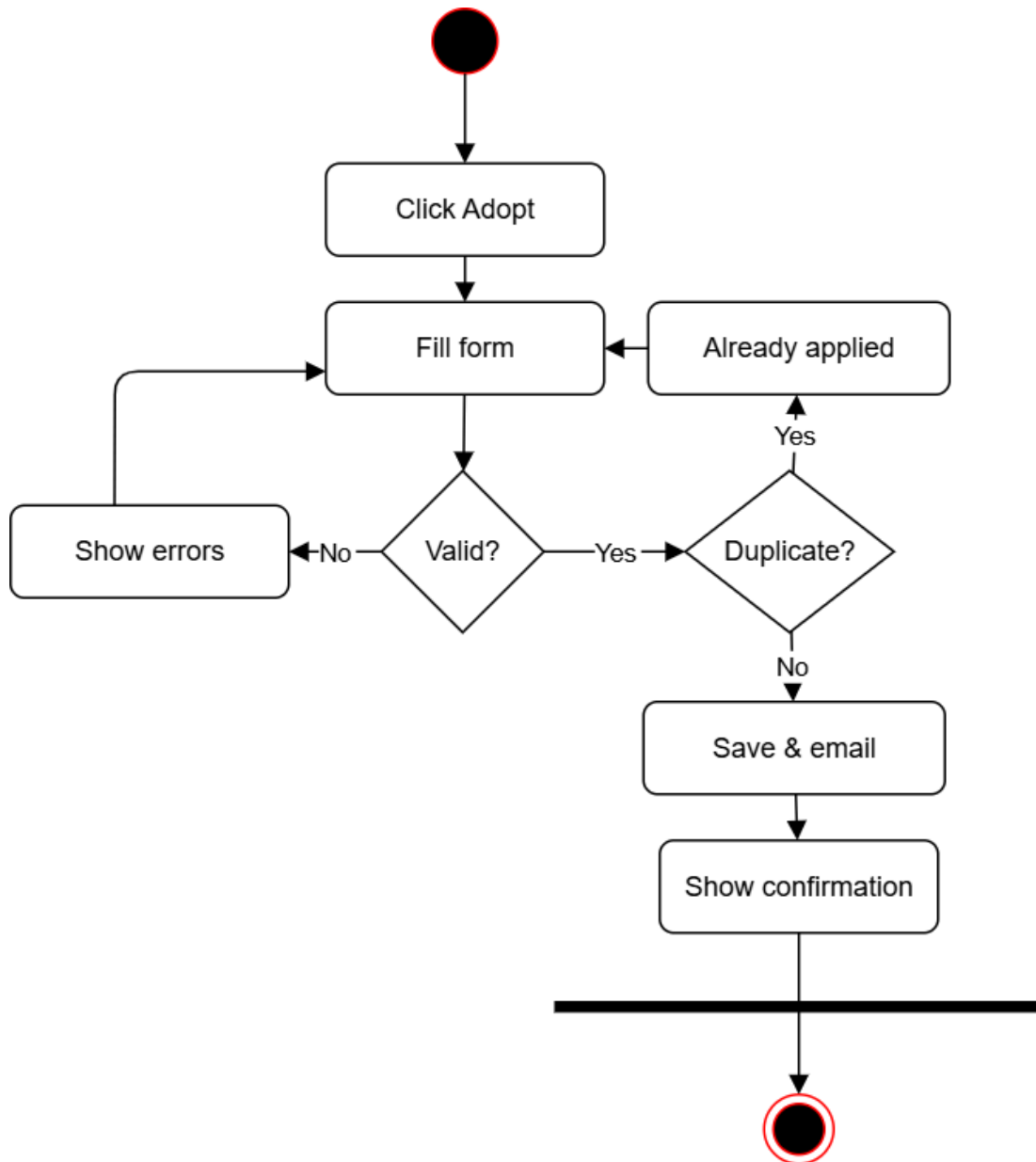


Figure 2.3.2. 07: Send Adoption Request

### Activity Diagram-08: View All Adoption Requests

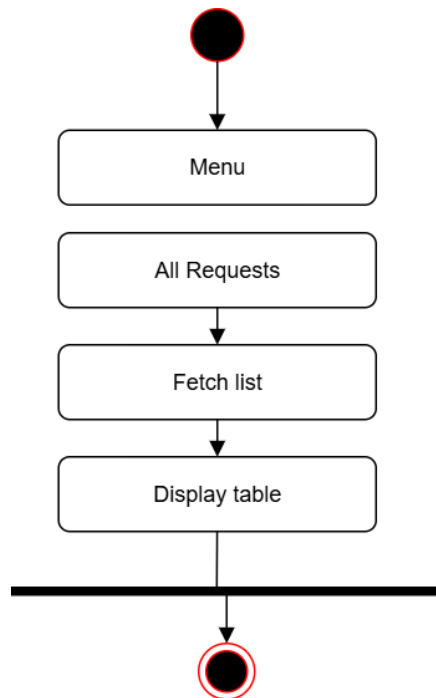


Figure 2.3.2. 08: View All Adoption Request

### Activity Diagram-09: Accept Adoption Request

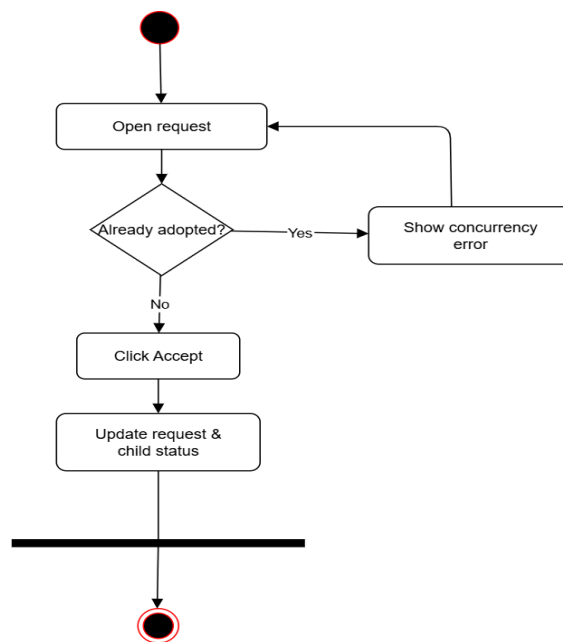


Figure 2.3.2. 09: Accept Adoption Request

**Activity Diagram-10: Reject Adoption Request**

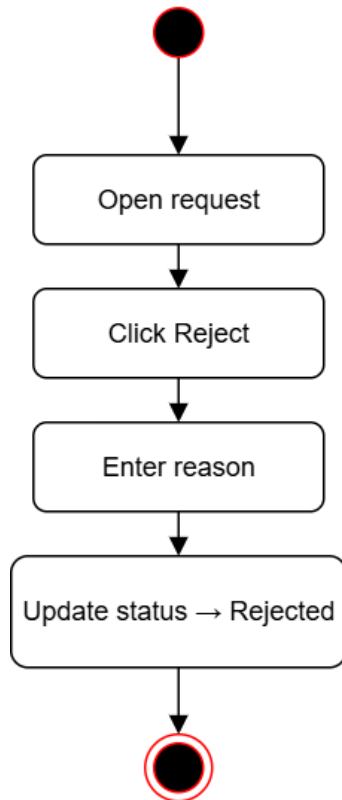


Figure 2.3.2. 10: Reject Adoption Request

### Activity Diagram-11: View News

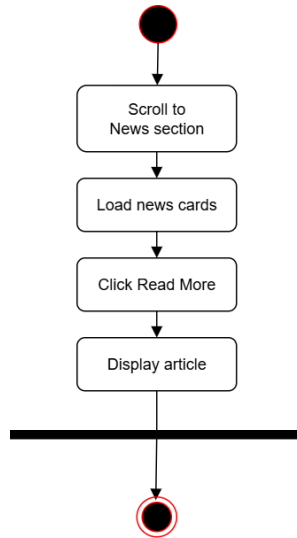


Figure 2.3.2. 11: View News

### Activity Diagram-12: Add News

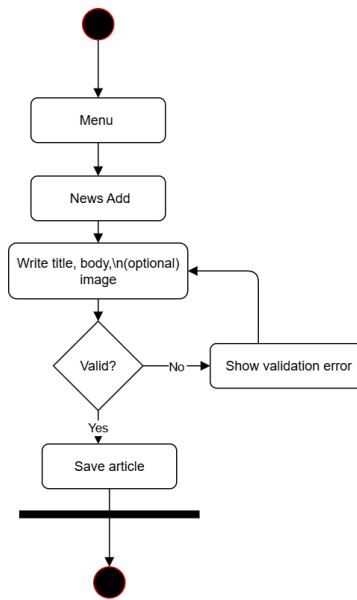
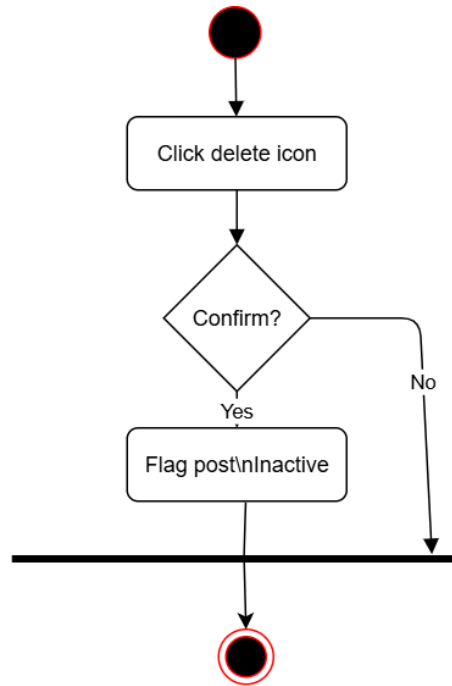


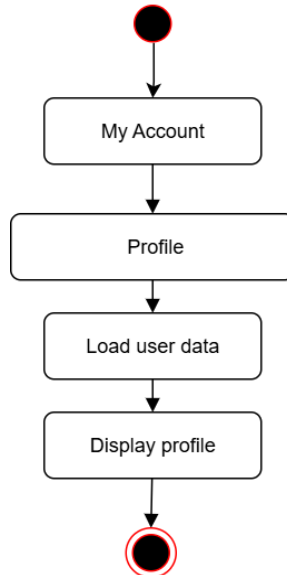
Figure 2.3.2. 02: Add News

**Activity Diagram-13 : Delete News**



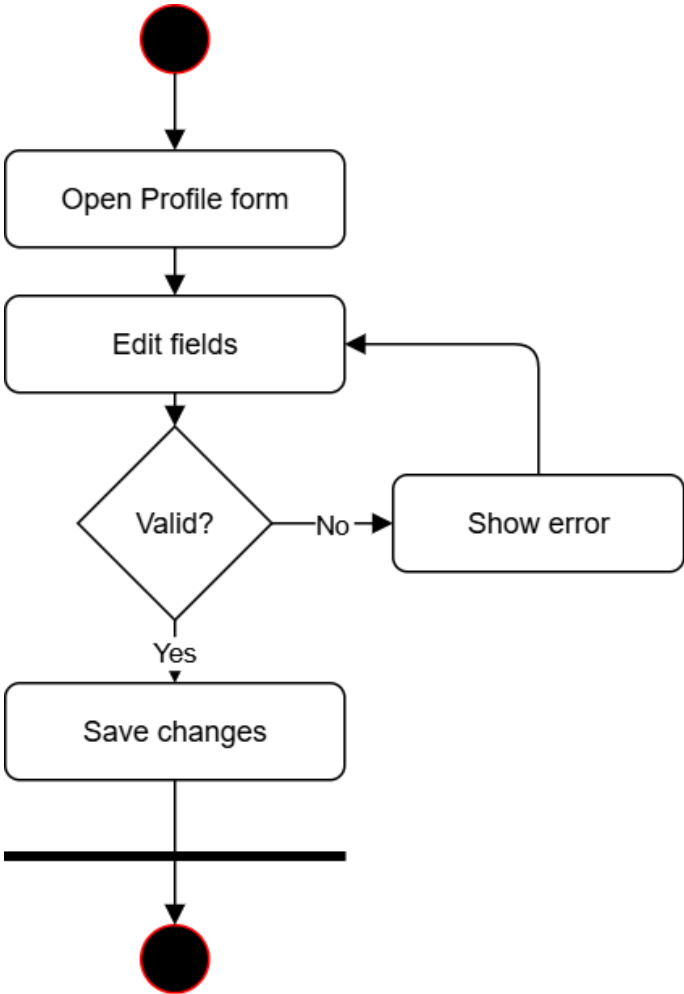
*Figure 2.3.2. 13: Delete News*

**Activity Diagram-14: View Profile**



*Figure 2.3.2. 14: View Profile*

**Activity Diagram-15: Update Profile**



*Figure 2.3.2. 15: Update Profile*

### Activity Diagram- 16: Forgot Password

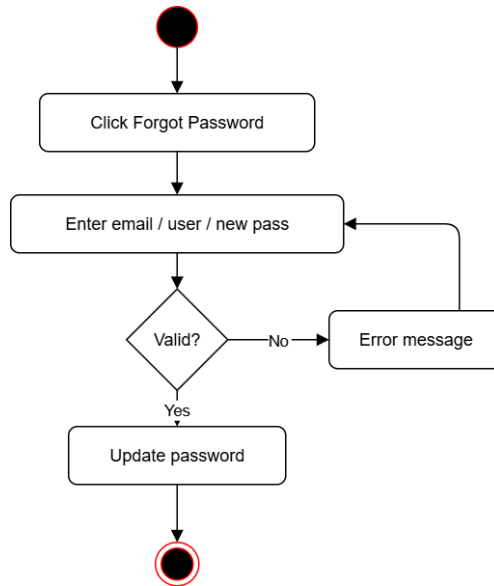


Figure 2.3.2. 16: Forget Password

### Activity Diagram- 17: Log Out

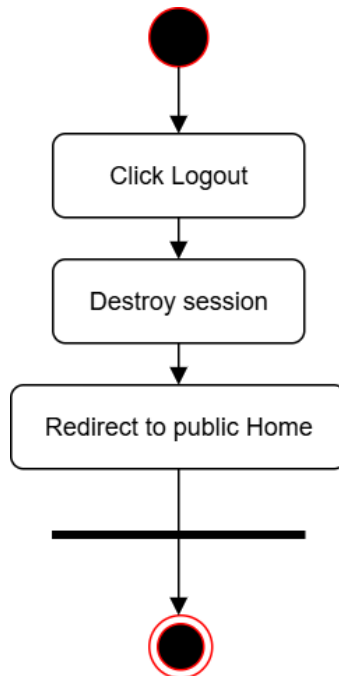


Figure 2.3.2. 07: Log Out

## d. Sequence Diagram

### Sequence Diagram-01: User Registration

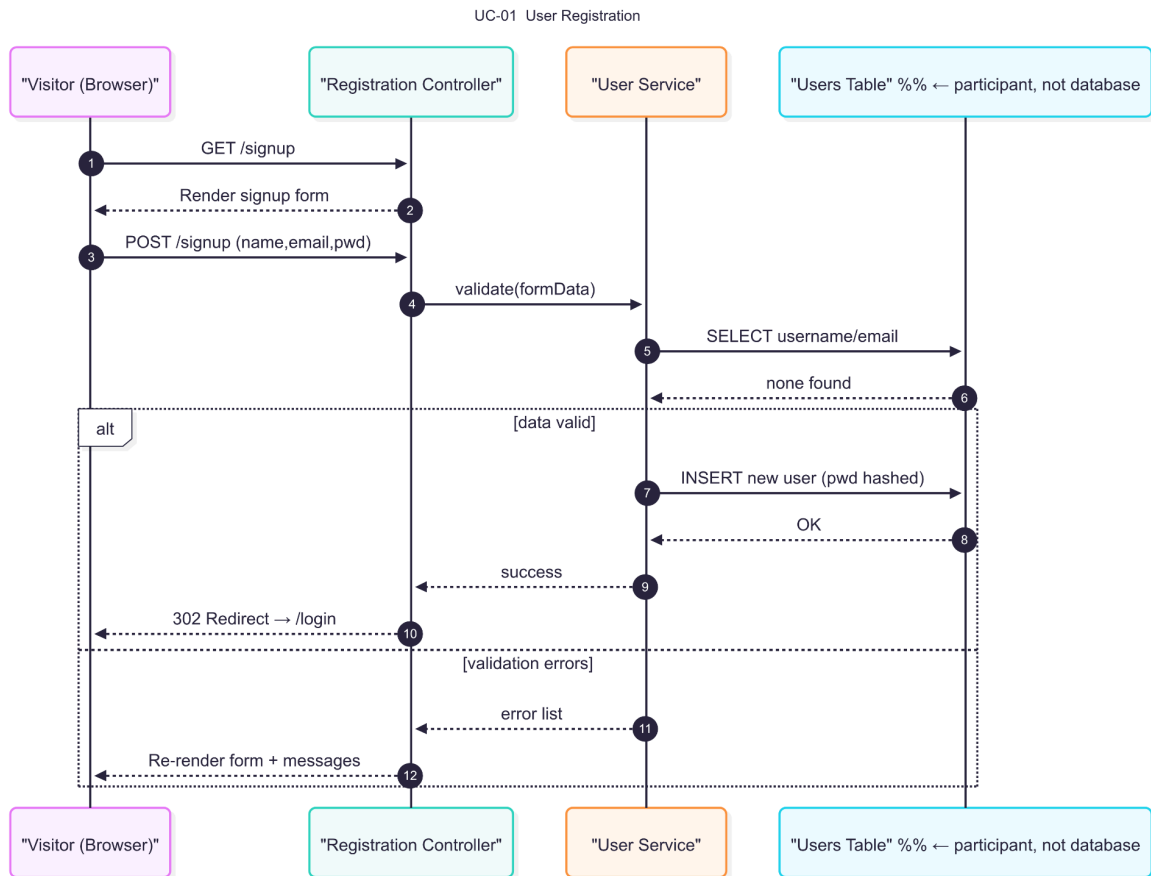


Figure 2.3.3. 01: User Registration

## Sequence Diagram-02: Login

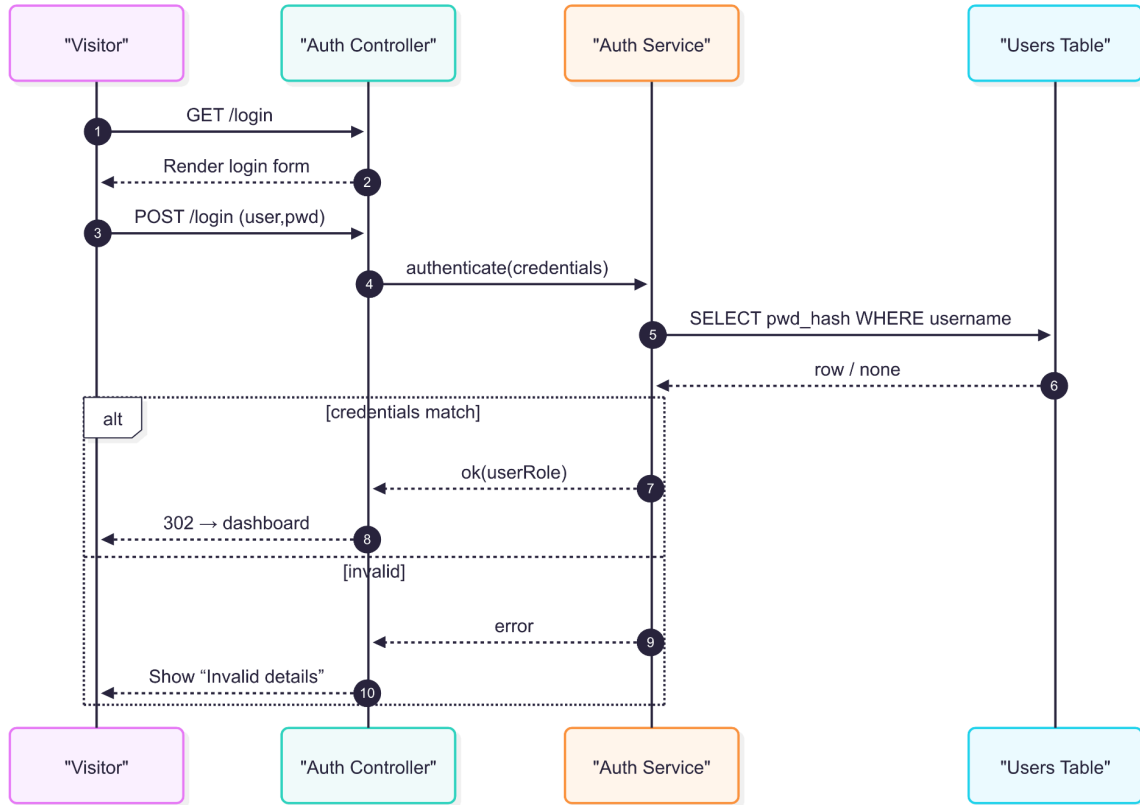


Figure 2.3.3.02 : Login

### Sequence Diagram-03: Add Child Details

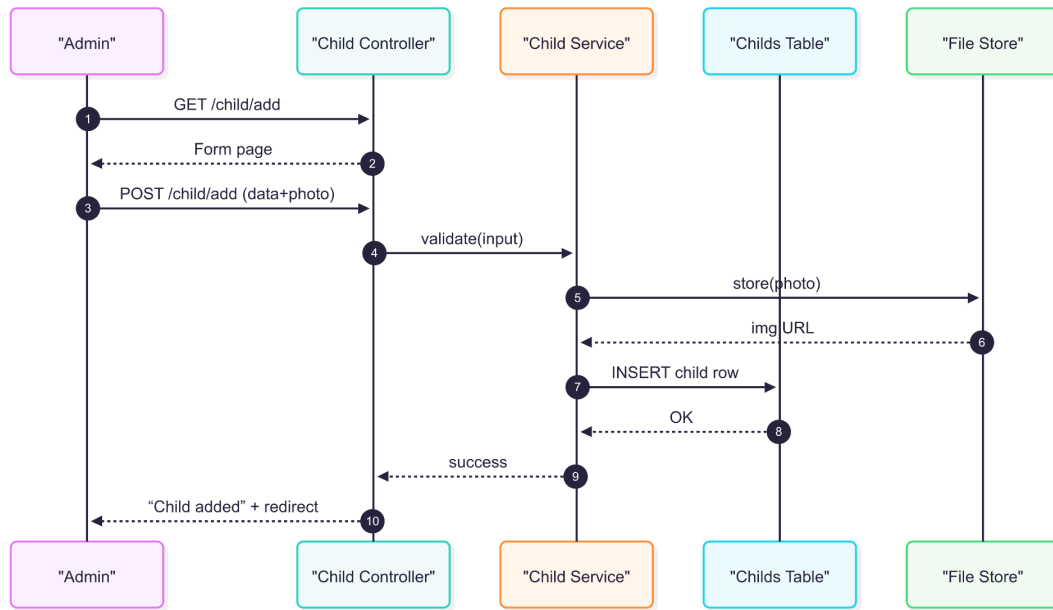


Figure 2.3.3.03 : Add Child Details

## Sequence Diagram-04: Edit Child Details

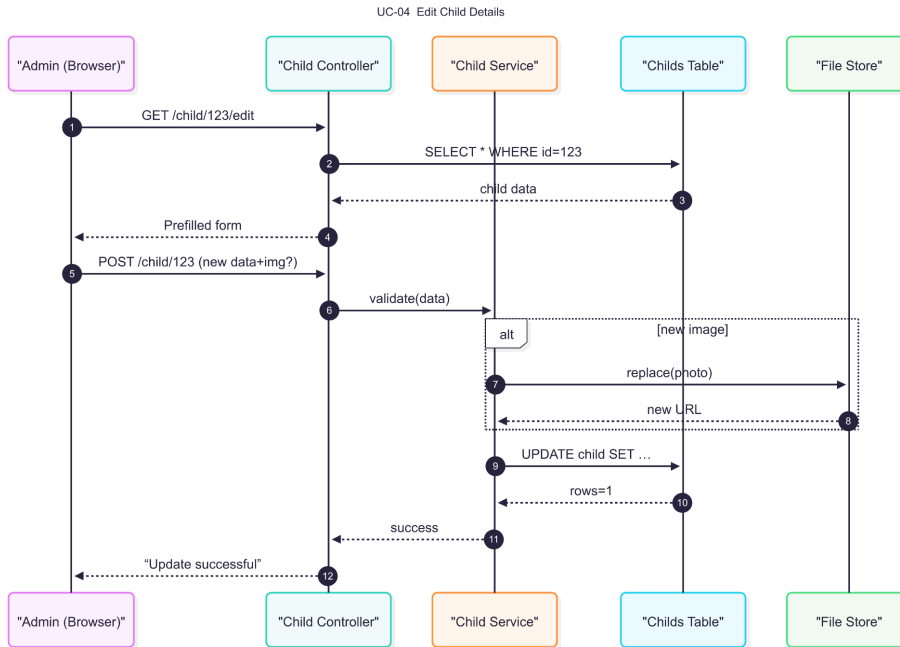


Figure 2.3.3.04 : Edit Child Details

### Sequence Diagram-05: Delete Child Details

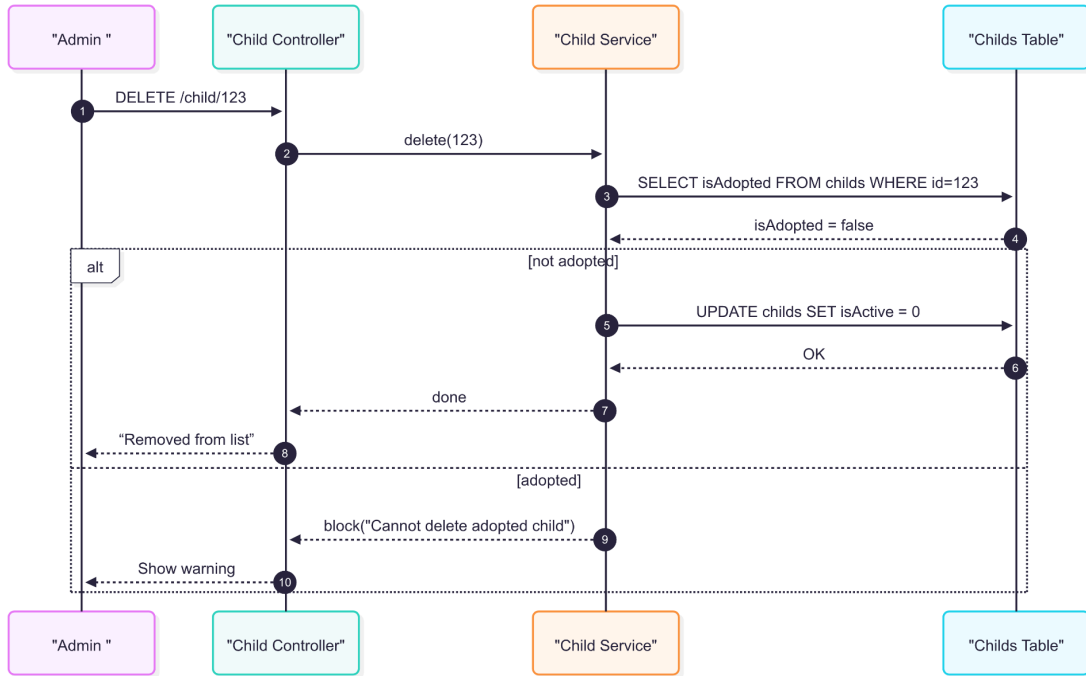


Figure 2.3.3.05: Delete Child Details

### Sequence Diagram-06 : View Child Details

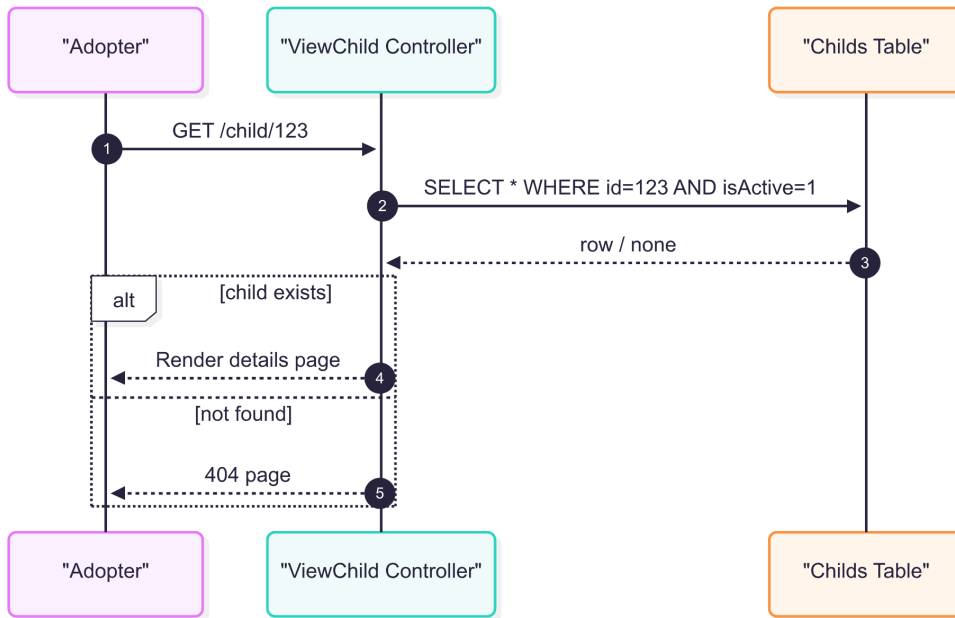


Figure 2.3.3.06: View Child Details

### Sequence Diagram-07: Send Adoption Request

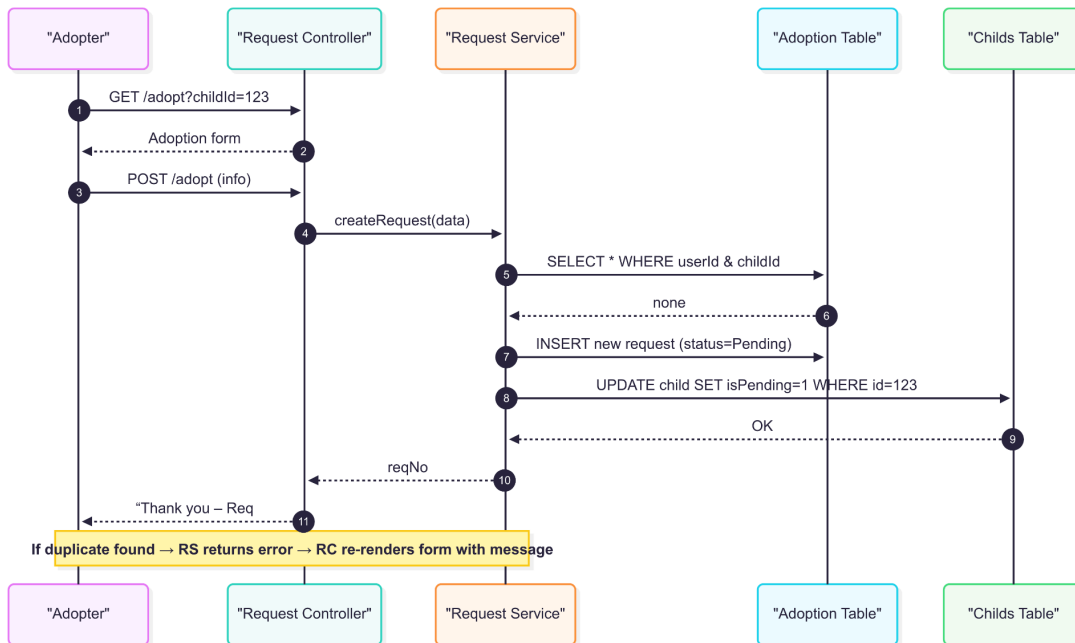
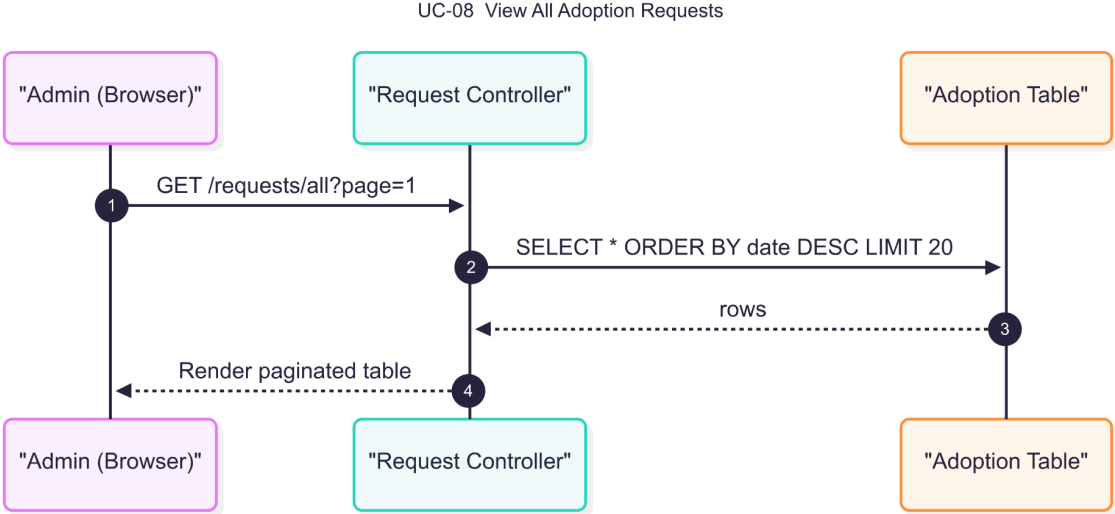


Figure 2.3.3. 07: Send Adoption Request

**Sequence Diagram-08: View All Adoption Requests**



*Figure 2.3.3. 08: View All Adoption Request*

### Sequence Diagram-09: Accept Adoption Request

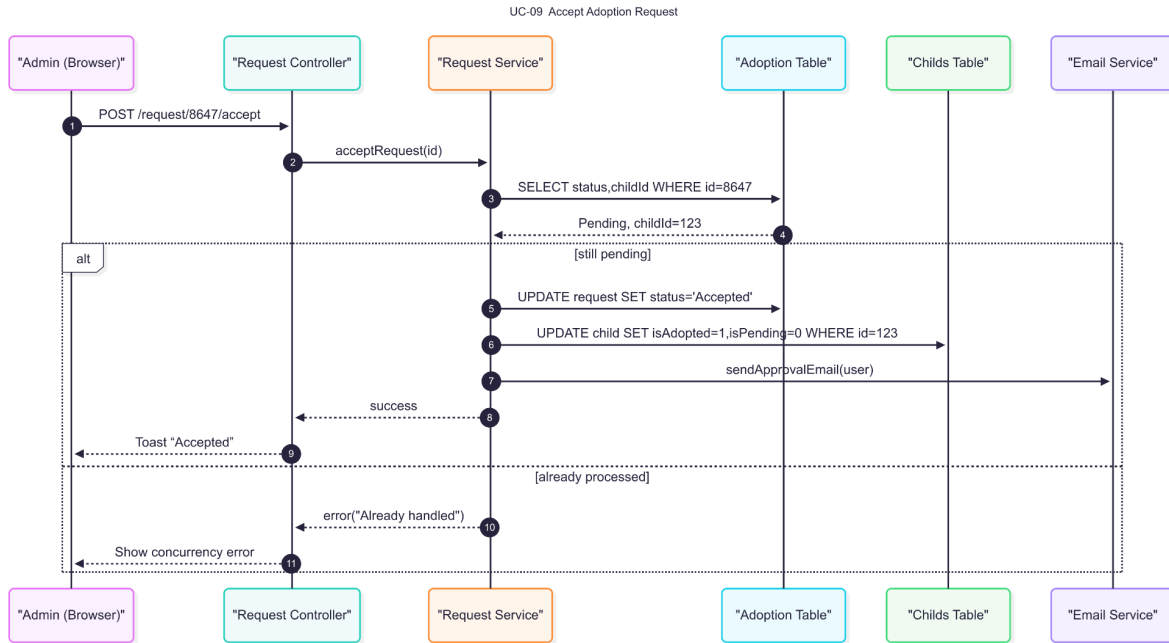


Figure 2.3.3. 09: Accept Adoption Request

### Sequence Diagram-10: Reject Adoption Request

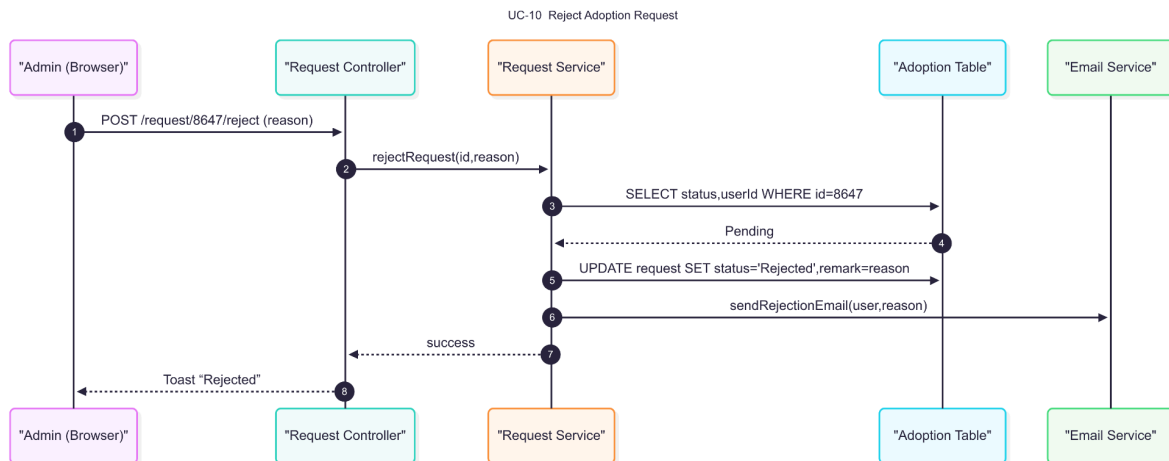


Figure 2.3.3. 10: Reject Adoption Request

## Sequence Diagram-11: View News

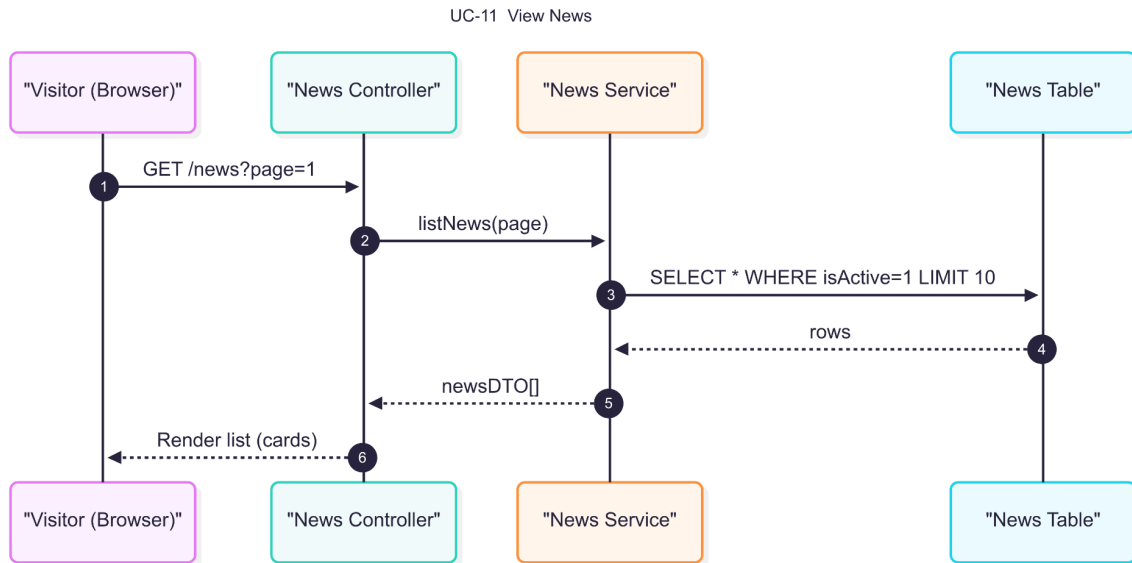


Figure 2.3.3. 11: View News

## Sequence Diagram-12: Add News

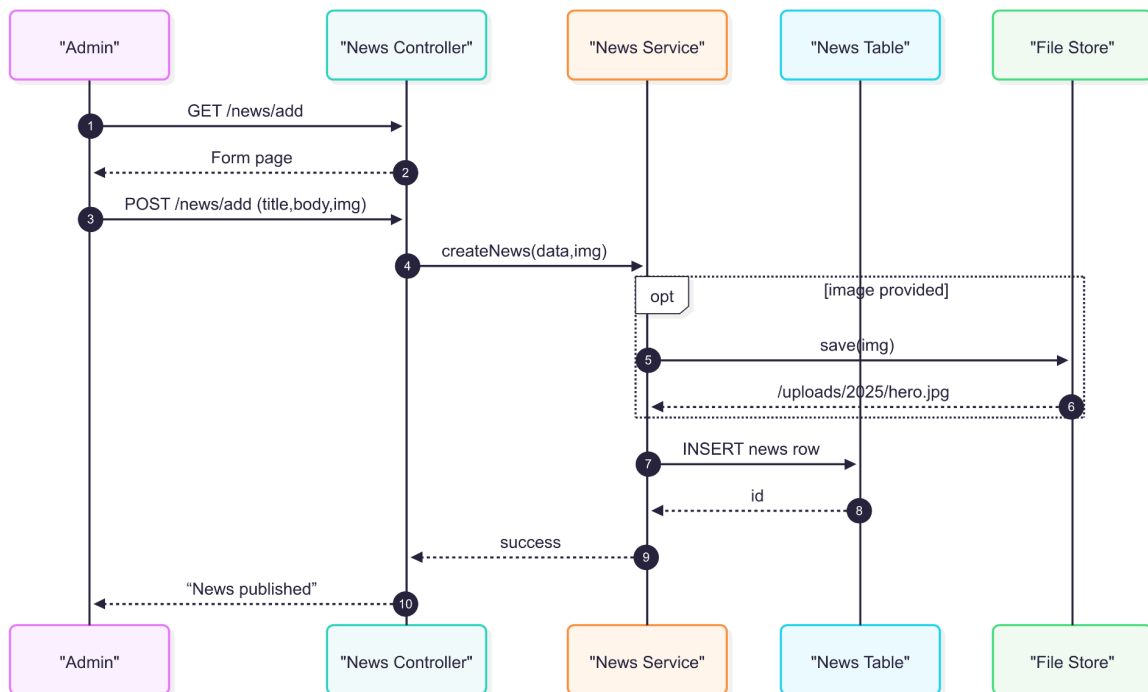


Figure 2.3.3. 02: Add News

### Sequence Diagram-13 : Delete News

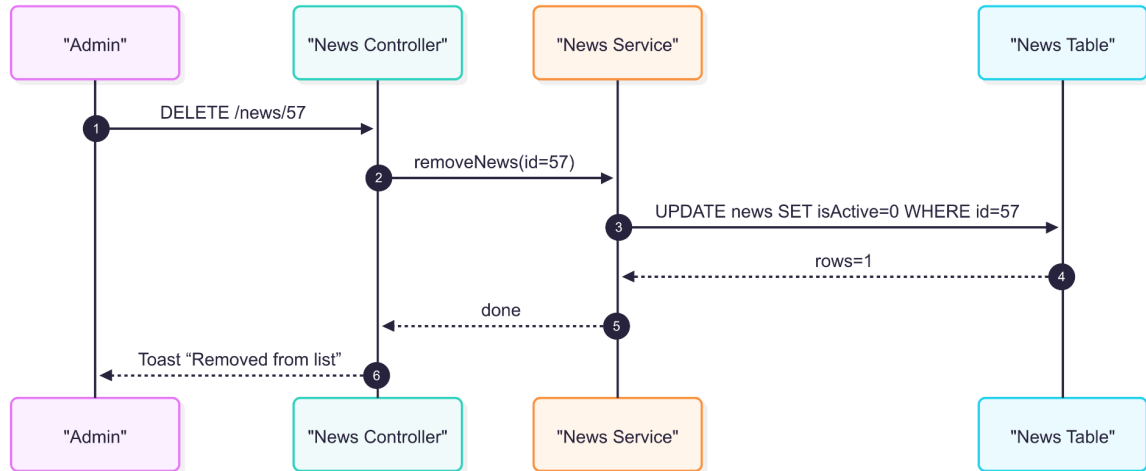


Figure 2.3.3. 13: Delete News

### Sequence Diagram-14: View Profile

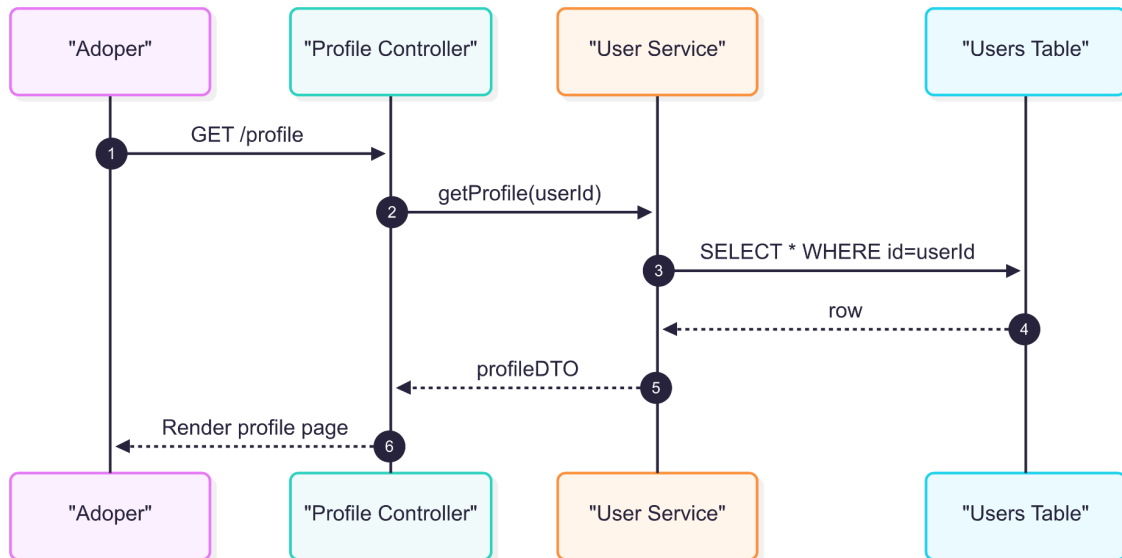


Figure 2.3.3. 14: View Profile

### Sequence Diagram-15: Update Profile

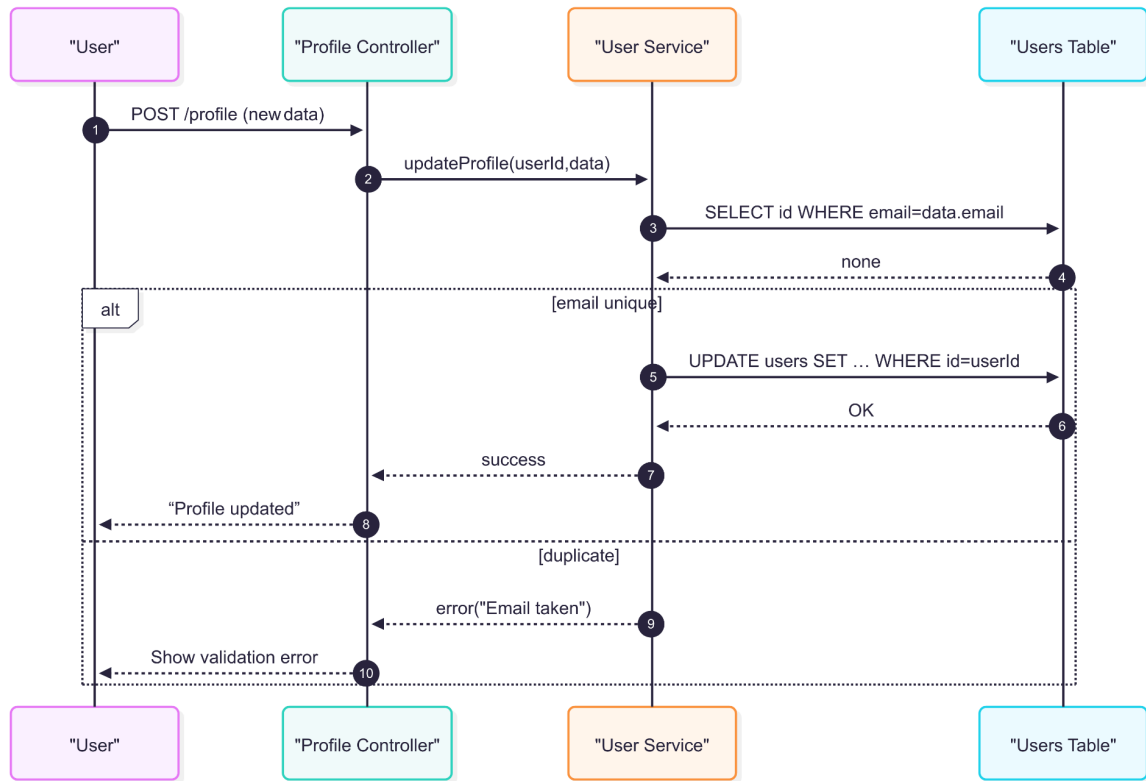


Figure 2.3.3. 15: Update Profile

### Sequence Diagram- 16: Forgot Password

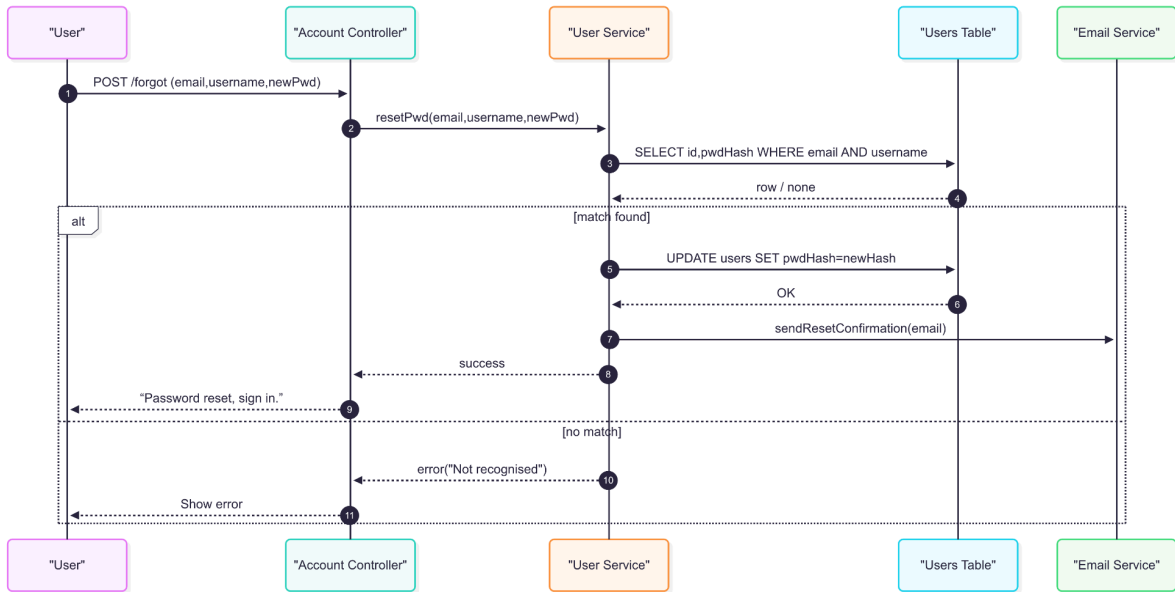


Figure 2.3.3. 16: Forget Password

### Sequence Diagram- 17: Log Out

UC-17 Log Out



Figure 2.3.3. 07: Log Out

## e . UML Diagram

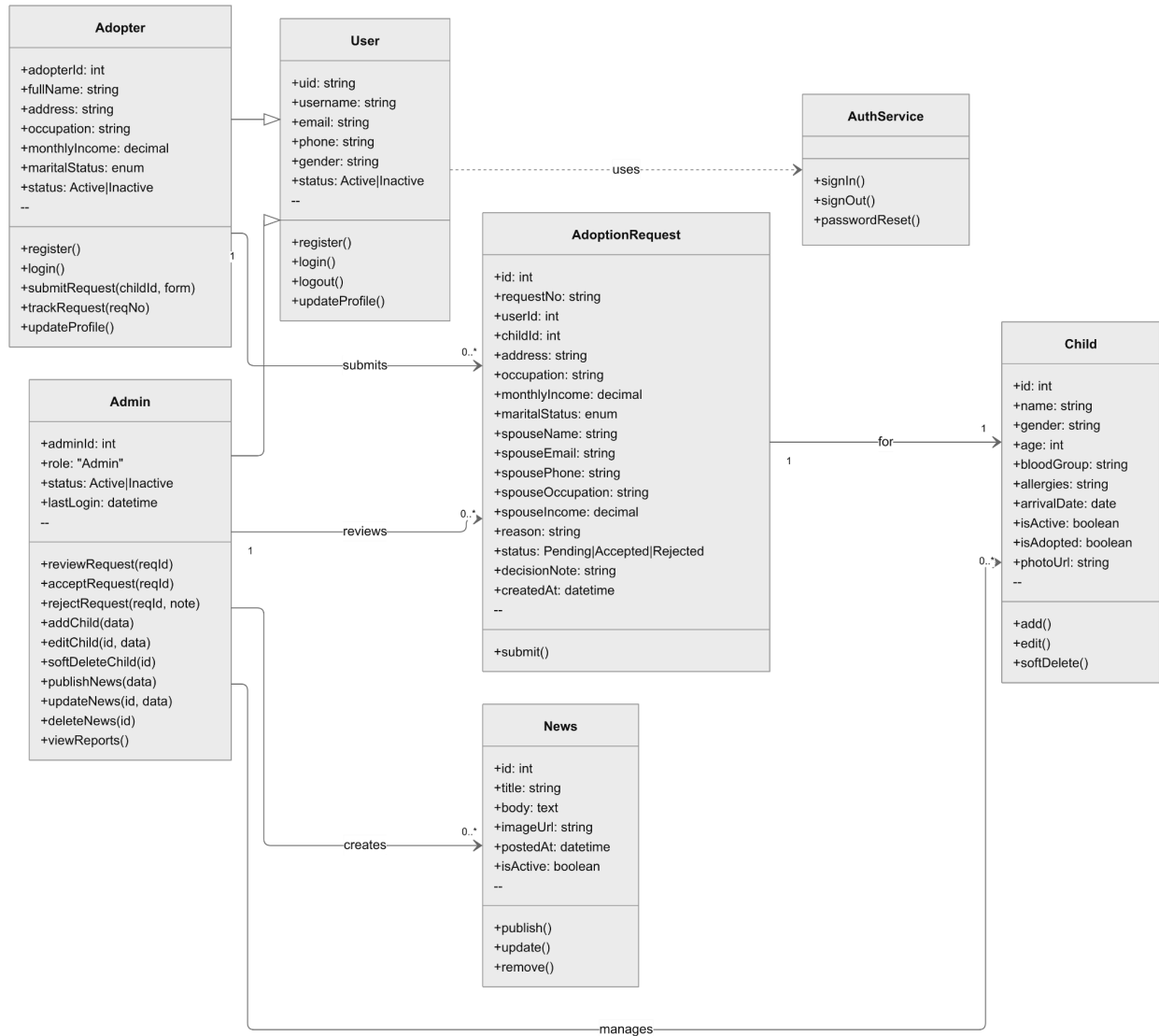
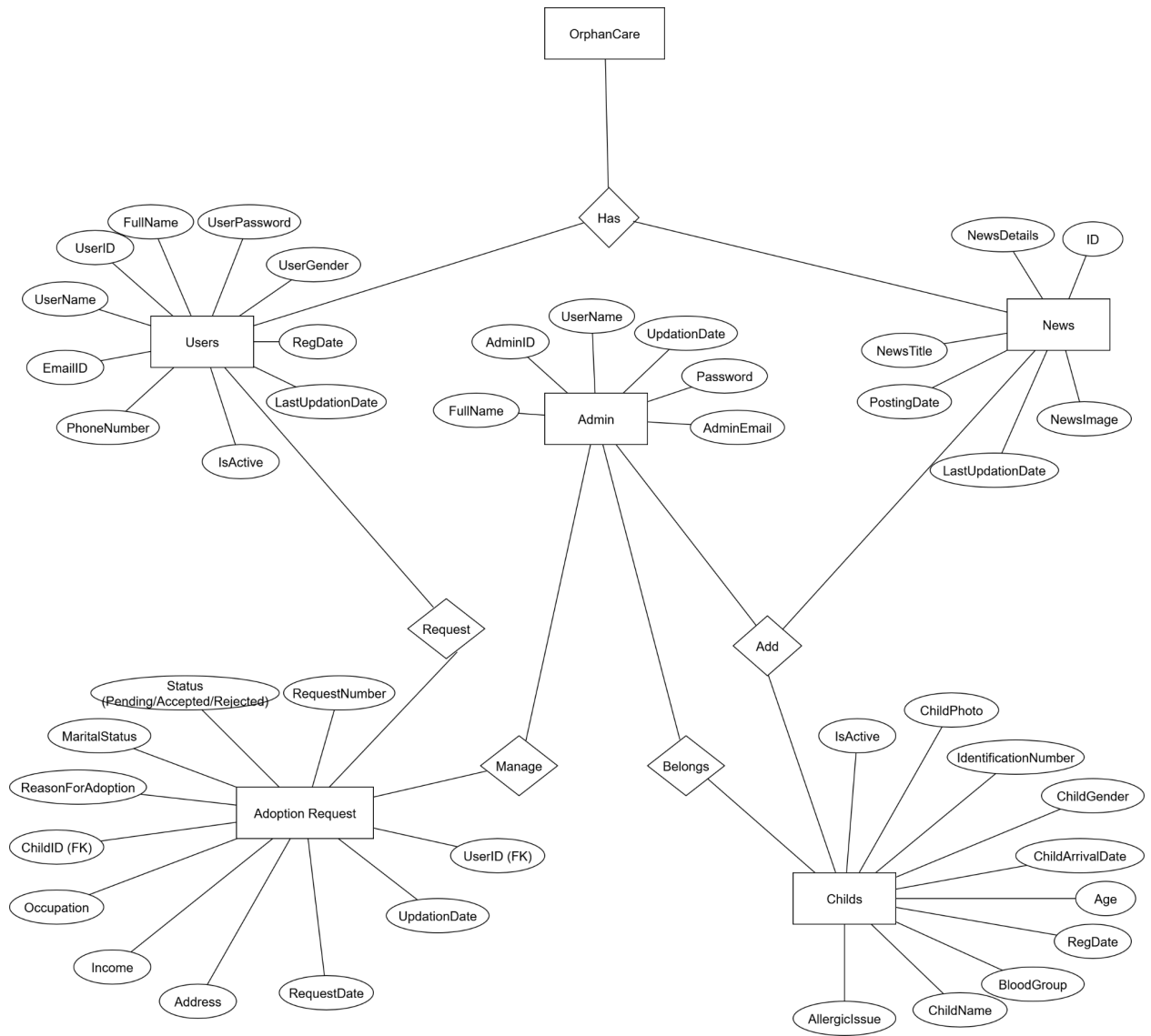


Figure 2.3.4: Unified Modeling Language (UML) Diagram

**f. ERD**



*Figure 2.3.5 : Entity Relationship Diagram (ERD)*

## Chapter 3: Software Testing

Table 3.1.1: Testing Features:

Feature Area	Specific Functionality Tested
User Authentication	Sign-Up, Login, Logout, Role-based access, Invalid-credential handling
Child Management (Admin)	Add Child, Edit Child, Soft Delete, Photo upload validation
Adoption Workflow	Submit request, Duplicate-request prevention, Accept, Reject, Status update
News Module	Create article, Edit article, Delete article, Image size check
Reporting	Generate date-range report, Search by status, Export to CSV

### 3.2 Testing Strategies

- **Unit Testing** – PHP functions (validators, data-access) tested with PHPUnit.
- **Integration Testing** – End-to-end flows run on local XAMPP to ensure DB + UI + email all interact correctly.
- **Manual Exploratory** – Two staff users and two adopter users followed real-world scenarios and logged UI issues.
- **Regression Sweep** – Quick retest of critical paths after each sprint.

Table 3.3.1 : System Testing (Test cases with results)

Test Case ID	Feature	Test Steps (summary)	Expected Result	Status
TC-01	Positive – User Registration	Fill form with valid data → Click Register	User created, redirect to Login	Pass
TC-02	Negative – Duplicate Email	Register with email already in DB	Error “Email already in use”	Pass
TC-03	Positive – Add Child	Admin logs in → Add Child form → Save	Record saved, visible in list	Pass

TC-04	Negative – Oversize Photo	Upload image 4 MB	Error “File too large”	Pass
TC-05	Positive – Submit Adoption	Adopter selects child → Submit form	Request # generated, status “Pending”	Pass
TC-06	Positive – Accept Request	Admin clicks <b>Accept</b>	Child flagged Adopted, email sent	Pass

## Chapter 4: Deployment and Maintenance

### 4.1 Software Release Life Cycle

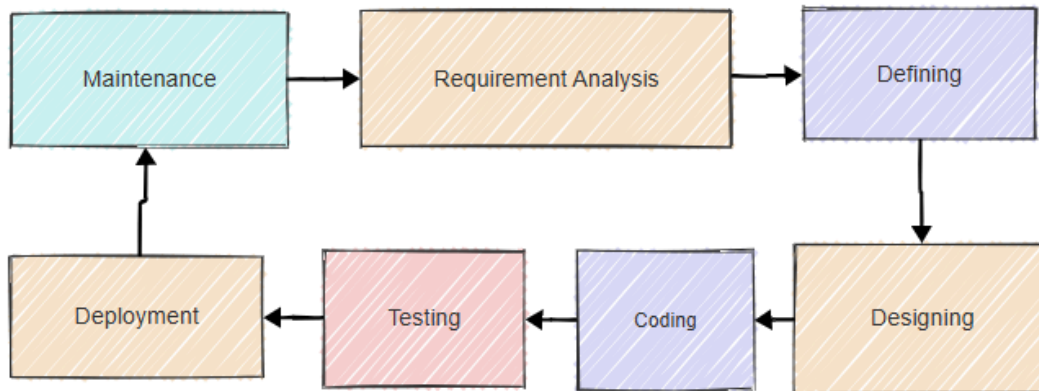


Figure 4.1 Life cycle

The project followed the **Agile** methodology with an **iterative Software Development Life Cycle (SDLC)** approach.

1. **Requirement Analysis** – Collected and refined functional and non-functional requirements for the OrphanCare system.
2. **Defining** – Created technical specifications, database schema, and UI/UX wireframes.
3. **Designing** – Developed detailed class diagrams, ER diagrams, and system architecture.
4. **Coding** – Implemented core modules (User Management, Adoption Requests, Child Records, News Management) in PHP/MySQL.
5. **Testing**- Unit testing, integration testing and user acceptance testing to correct errors and to stabilize.
6. **Deployment** – Uploaded the final build to the hosting server, configured the database, and ensured all features worked as expected.

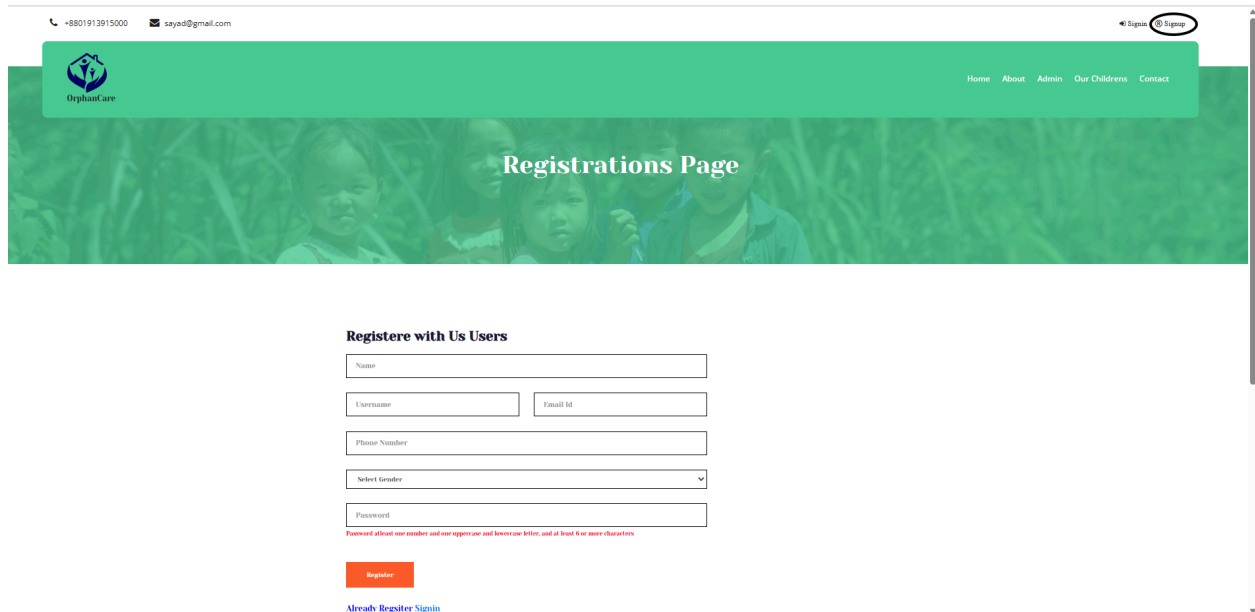
7. **Maintenance** – Ongoing monitoring, bug fixing, and minor enhancements based on adopter and admin feedback.

## Chapter 5: User Manual

### For Adopter

#### 5.1 User Registration

New users enter name, username, contact details, gender, and password; the system validates fields before creating an account.



The screenshot shows the 'Registrations Page' of the OrphanCare website. The page features a green header with the OrphanCare logo and navigation links: Home, About, Admin, Our Childrens, and Contact. Below the header, the main content area is titled 'Registrations Page' and contains a registration form. The form is titled 'Registere with Us Users' and includes the following fields: Name, Username, Email Id, Phone Number, Select Gender (a dropdown menu), and Password. A red error message below the password field reads: 'Password atleast one number and one upper case and lower case letter, and at least 6 or more characters'. Below the form is an orange 'Register' button and a blue link that says 'Already Register Signin'.

*Figure 5.1 : User Registration*

## 5.2 Log in

Existing users provide username and password; on success, they are redirected to the appropriate dashboard.

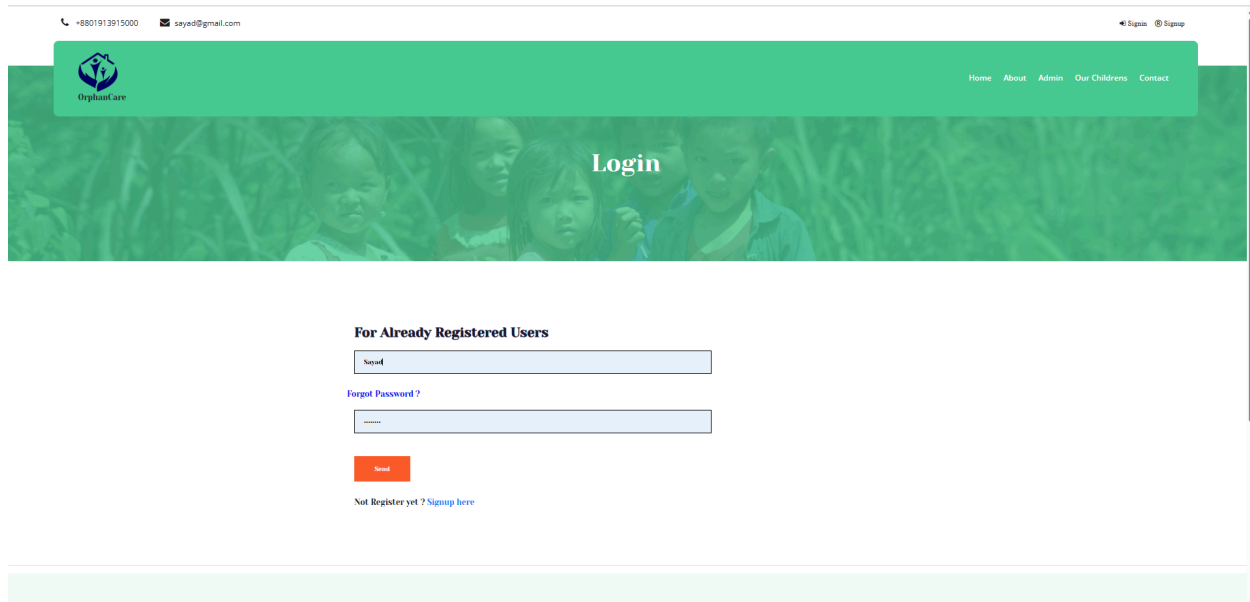


Figure 5.2 : Login

## 5.3 View Child Details

A details page shows a child's photo and profile; the **Adopt** button is enabled only for logged-in users.

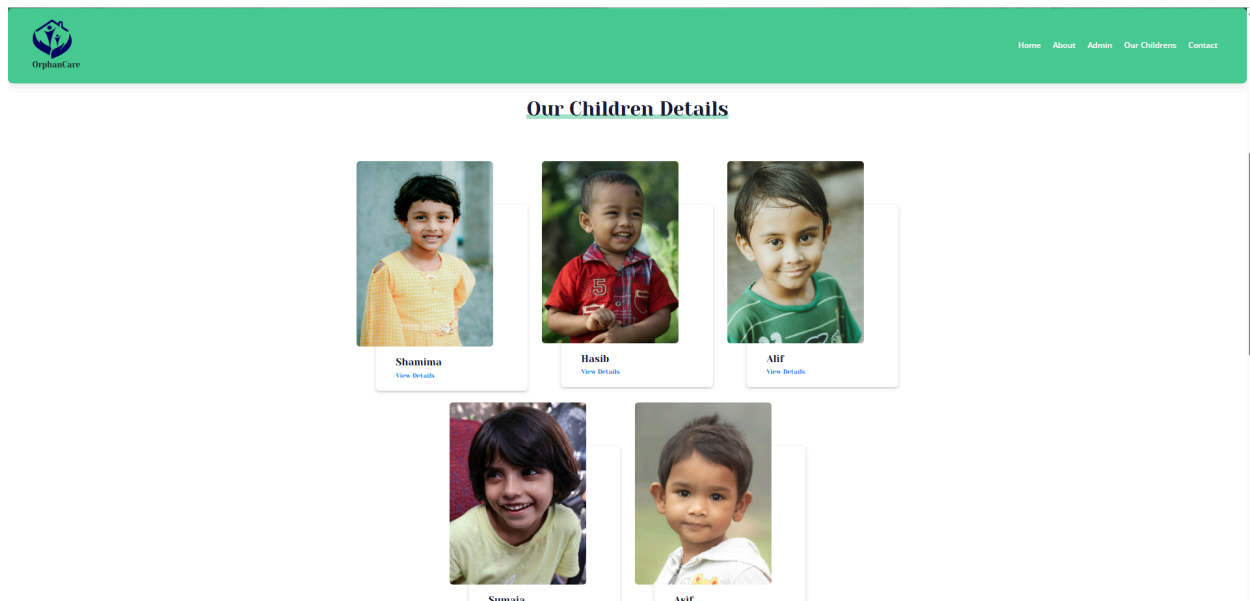
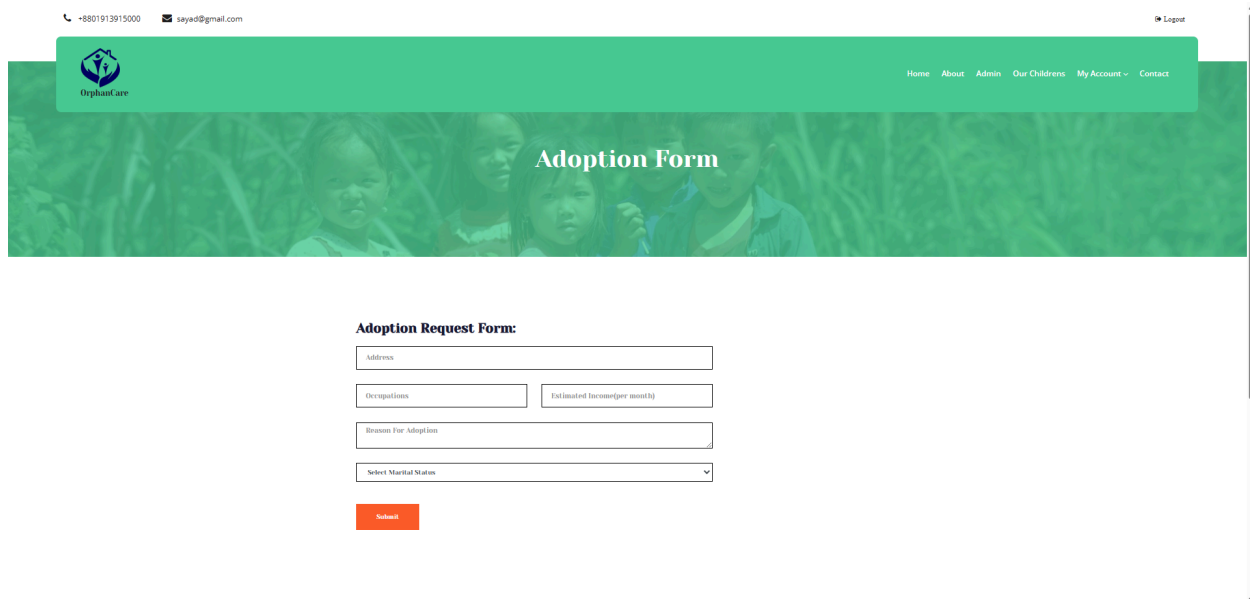


Figure 5.3: View Child Details

## 5.4 Send Adoption Request

The adoption form collects address, occupation, income, reason, and—if applicable—spouse information; after submission a unique request number is issued.



The screenshot shows the 'Adoption Form' page on the Orphanare website. The page has a green header with the Orphanare logo and contact information (+8801913915000, sayad@gmail.com) on the left, and a 'Logout' link on the right. Below the header is a navigation menu with links for Home, About, Admin, Our Childrens, My Account, and Contact. The main content area features a large green banner with the text 'Adoption Form' and a background image of children. Below the banner is the 'Adoption Request Form' with the following fields:

- Address
- Occupations
- Estimated Income(per month)
- Reason For Adoption
- Select Marital Status (dropdown menu)

A red 'Submit' button is located at the bottom of the form.

*Figure 5.4 : Send Adoption Request*

## 5.5 View News :

The news page lists recent posts with images and dates; clicking **Read More** opens the full article.



The screenshot shows a news article on a website. At the top, there is a green banner with the text "News Details" and a background image of children. Below the banner, the main article features a photo of five young boys in white Islamic attire sitting on a staircase. The article title is "Growth of Islamic Orphanage Initiatives in Bangladesh (2018-2025)(2025-08-08 22:47:58)". The text discusses the growth and quality of Islamic orphanages in Bangladesh, mentioning the Islamic principle of kafalah and the role of local mosques and charities. To the right of the main article is a "Recent News" sidebar with three smaller article thumbnails and titles: "Growth of Islamic Orphanage Initiatives in Bangladesh (2018-2025)", "Decade Progress Report on Orphanage Development in Bangladesh (2015-2025)", and "Father, Domestic Help Physically Abuse Adopted Child In Assam, Arrested".

*Figure 5.5 : View News*

## For Admin

### 5.6 Dashboard

Summary cards display counts (children, users, new, accepted, and rejected requests) with links to detailed views.

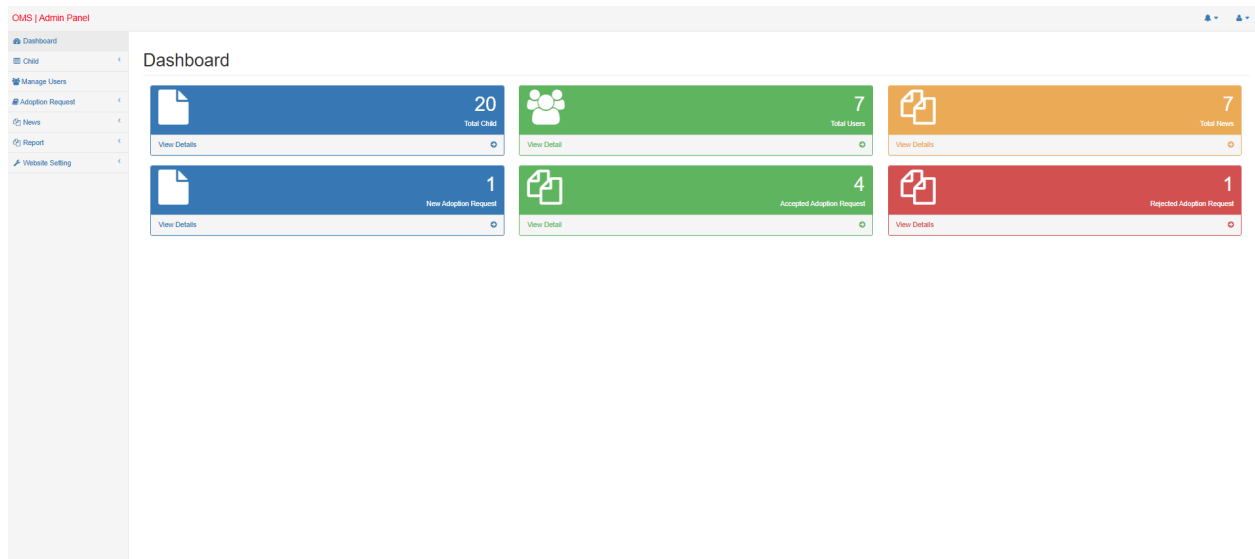


Figure 5.6 :Dashboard

### 5.7 Add Child Details

Admin completes a form (name, age, blood group, allergies, arrival date, gender) and uploads a photo to create a profile.

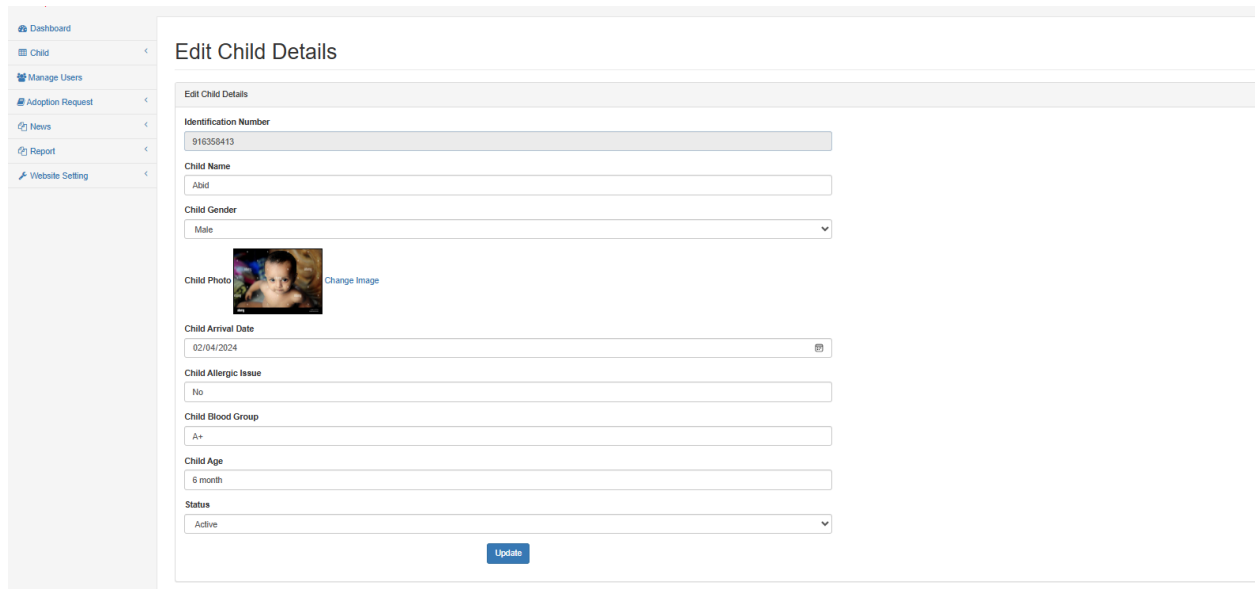
The screenshot shows the CMS Admin Panel 'Add Child Details' form. The form is titled 'Add Child Details' and contains the following fields:

- Child Name:
- Child Gender:
- Child Photo:
- Child Arrival Date:
- Child Allergic Issue:
- Child Blood Group:
- Child Age:
-

Figure 5.7 :Add Child Details

## 5.8 Edit Child Details

Existing child information can be updated; saving persists the changes and records the update time.



The screenshot shows a web application interface for editing child details. On the left is a sidebar menu with options: Dashboard, Child, Manage Users, Adoption Request, News, Report, and Website Setting. The main content area is titled 'Edit Child Details' and contains the following form fields:

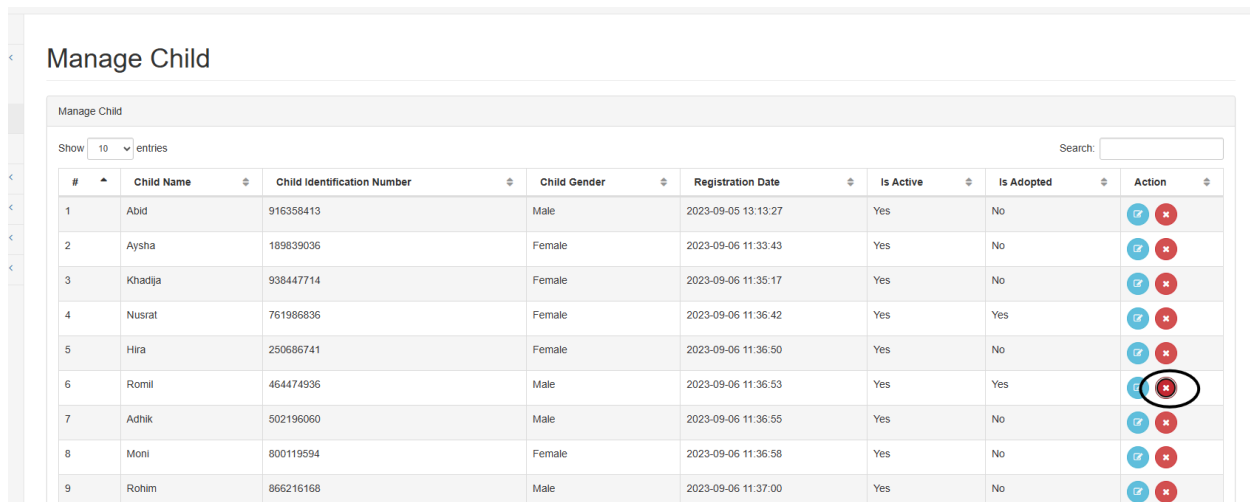
- Identification Number:** 916358413
- Child Name:** Abid
- Child Gender:** Male (dropdown menu)
- Child Photo:** A small image of a child with a 'Change Image' link next to it.
- Child Arrival Date:** 02/04/2024 (calendar icon)
- Child Allergic Issue:** No
- Child Blood Group:** A+
- Child Age:** 6 month
- Status:** Active (dropdown menu)

An 'Update' button is located at the bottom center of the form.

Figure 5.8 :Edit Child Details

## 5.9 Delete Child Details

An inactive (soft-delete) action hides the profile from public view; adopted records cannot be removed.





















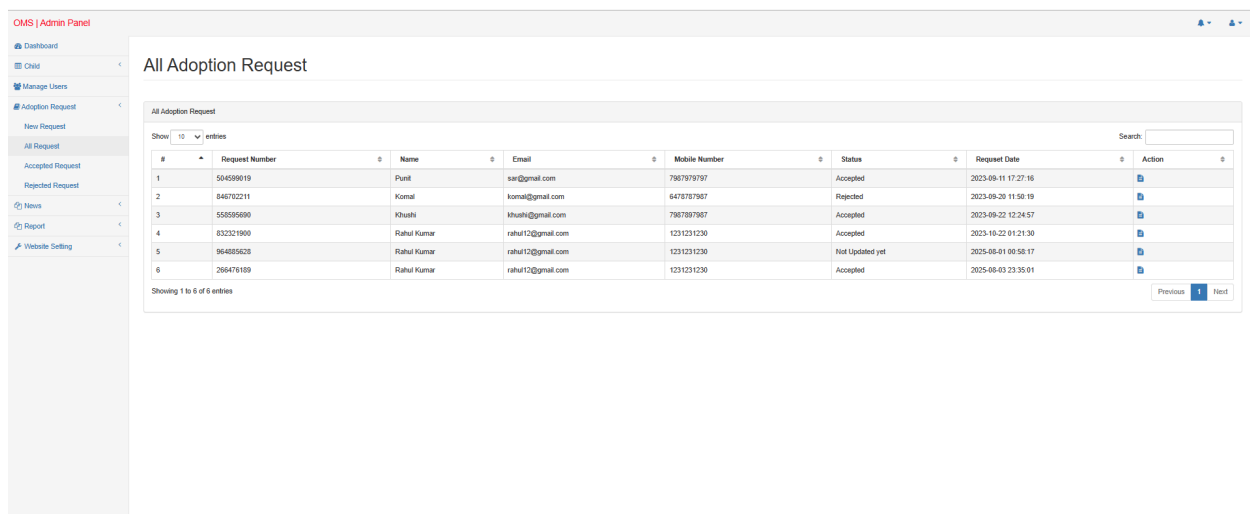
#	Child Name	Child Identification Number	Child Gender	Registration Date	Is Active	Is Adopted	Action
1	Abid	916358413	Male	2023-09-05 13:13:27	Yes	No	 
2	Aysha	189839036	Female	2023-09-06 11:33:43	Yes	No	 
3	Khadija	938447714	Female	2023-09-06 11:35:17	Yes	No	 
4	Nusrat	761986836	Female	2023-09-06 11:36:42	Yes	Yes	 
5	Hira	250686741	Female	2023-09-06 11:36:50	Yes	No	 
6	Romil	464474936	Male	2023-09-06 11:36:53	Yes	Yes	 
7	Adhik	502196060	Male	2023-09-06 11:36:55	Yes	No	 
8	Moni	800119594	Female	2023-09-06 11:36:58	Yes	No	 
9	Rohim	866216168	Male	2023-09-06 11:37:00	Yes	No	 

Figure 5.9 :Delete Child Details

## 5.10 View all Adoption Request

A paginated table lists requests with search and a link to each record's details.









#	Request Number	Name	Email	Mobile Number	Status	Request Date	Action
1	504598919	Punit	san@gmail.com	798797979	Accepted	2023-09-11 17:27:16	
2	846702311	Komal	komal@gmail.com	647879787	Rejected	2023-09-20 11:59:19	
3	558586990	Khushi	khushi@gmail.com	798789787	Accepted	2023-09-22 12:24:57	
4	832321900	Rahul Kumar	rahul12@gmail.com	1231231230	Accepted	2023-10-22 01:21:30	
5	964885628	Rahul Kumar	rahul12@gmail.com	1231231230	Not Updated yet	2025-08-01 00:58:17	
6	266476189	Rahul Kumar	rahul12@gmail.com	1231231230	Accepted	2025-08-03 23:35:01	

Figure 5.10 : View All Adoption Request

## 5.11 Update Profile

Users can modify their own profile information (name, email, phone, etc.).

OMS | Admin Panel

### Edit User Profile

John Doe's Profile

Reg Date : 2023-05-21 17:34:02  
Last Updated at : 2023-05-21 17:34:21

**Username**

**Fullname**

**Email id**

**Phone Number**

**Gender**

**Status**

Figure 5.11 : Update profile

## 5.12 Accept Adoption Request

Setting a request to **Accepted** also marks the related child as **Adopted**.

OMS | Admin Panel

### Accepted Adoption Request

Accepted Adoption Request

Show 10 entries

#	Request Number	Name	Email	Mobile Number	Status	Request Date	Action
1	504599019	Punit	sar@gmail.com	7987979797	Accepted	2023-09-11 17:27:16	<input type="button" value="Edit"/>
2	558595690	Khushi	khushi@gmail.com	7987897987	Accepted	2023-09-22 12:24:57	<input type="button" value="Edit"/>
3	832321900	Rahul Kumar	rahul12@gmail.com	1231231230	Accepted	2023-10-22 01:21:30	<input type="button" value="Edit"/>
4	266476189	Rahul Kumar	rahul12@gmail.com	1231231230	Accepted	2025-08-03 23:35:01	<input type="button" value="Edit"/>

Showing 1 to 4 of 4 entries

Previous 1 Next

Figure 5.12 : Accept Adoption Request

### 5.13 Rejection Adoption request

Admin may reject a request and record an optional reason; the applicant is notified.

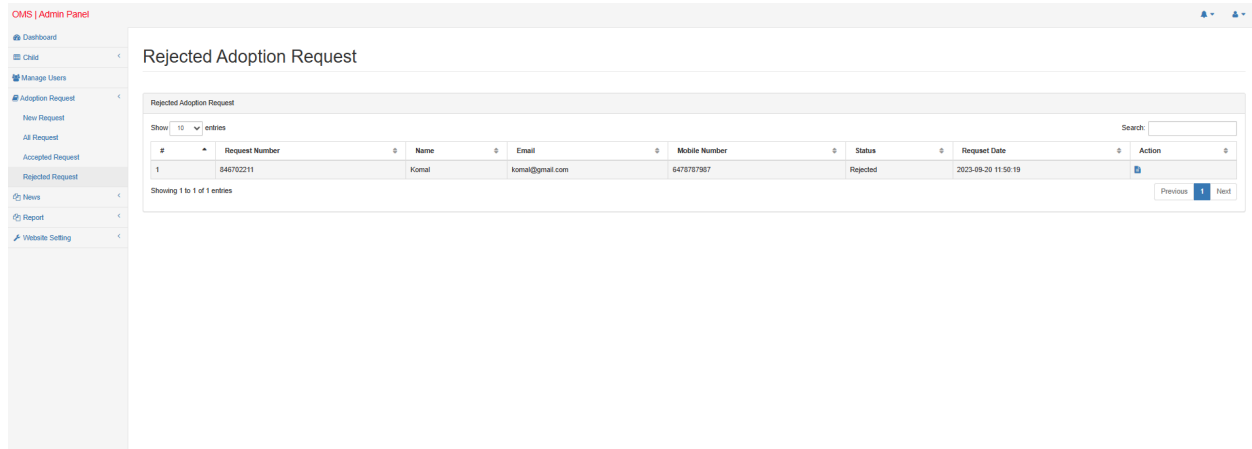


Figure 5.13: Rejection Adoption Request

### 5.14 Add News

Admin publishes a news item with title, body, and optional image.

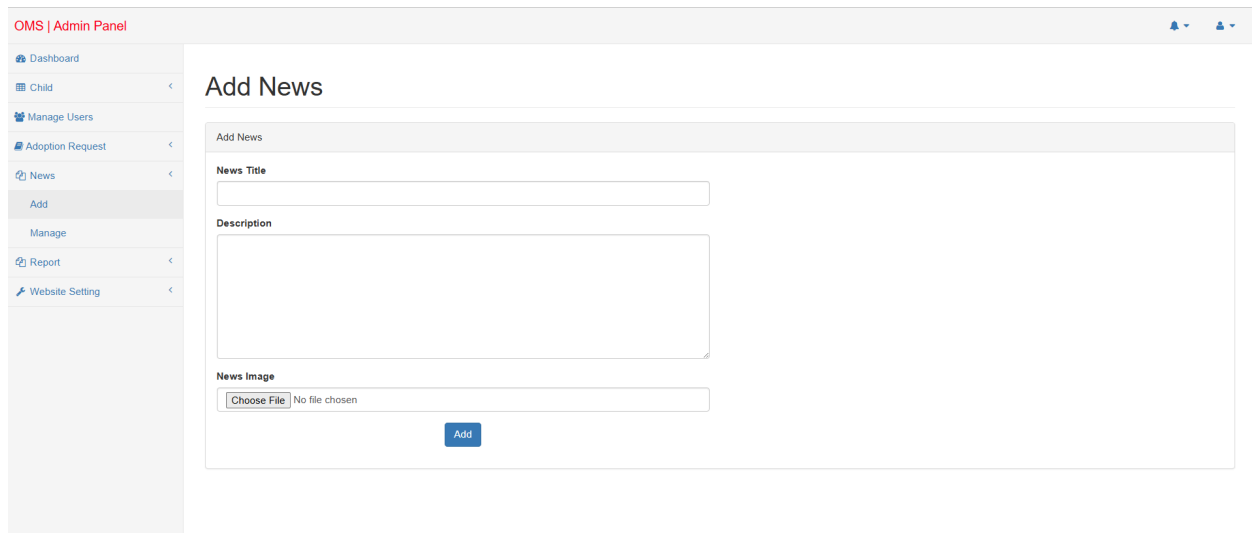


Figure 5.14: Add News

## 5.15 Update news

An existing news post can be edited and re-saved.

The screenshot shows the 'Manage News' interface in the OMS Admin Panel. A success message at the top indicates 'Success : News deleted'. Below this, there is a search bar and a table of news items. The table has columns for '#', 'News Title', 'Description', 'Posting Date', and 'Action'. Four news items are listed, each with a red circular icon containing a white 'u' (update) symbol in the 'Action' column. The first item is titled 'Number Of Orphan, Abandoned Children Rise By 25% Since 2020: Centre'. The second item is 'Father, Domestic Help Physically Abuse Adopted Child In Assam, Arrested'. The third item is 'Decade Progress Report on Orphanage Development in Bangladesh (2015-2025)'. The fourth item is 'Growth of Islamic Orphanage Initiatives in Bangladesh (2018-2025)'. At the bottom of the table, it says 'Showing 1 to 4 of 4 entries' and there are 'Previous' and 'Next' navigation buttons.

Figure 5.15: Update News

## 5.16 Delete News

An outdated post is removed from public view using a soft-delete action.

The screenshot shows the 'Manage News' interface in the OMS Admin Panel. A success message at the top indicates 'Success : News deleted'. Below this, there is a search bar and a table of news items. The table has columns for '#', 'News Title', 'Description', 'Posting Date', and 'Action'. Four news items are listed, each with a red circular icon containing a white 'x' (delete) symbol in the 'Action' column. The first item is titled 'Number Of Orphan, Abandoned Children Rise By 25% Since 2020: Centre'. The second item is 'Father, Domestic Help Physically Abuse Adopted Child In Assam, Arrested'. The third item is 'Decade Progress Report on Orphanage Development in Bangladesh (2015-2025)'. The fourth item is 'Growth of Islamic Orphanage Initiatives in Bangladesh (2018-2025)'. At the bottom of the table, it says 'Showing 1 to 4 of 4 entries' and there are 'Previous' and 'Next' navigation buttons. A red circle highlights the 'x' icon in the 'Action' column of the first news item.

Figure 5.16 :Delete News

## **Chapter 6: Project Summary**

### **6.1 Overview**

OrphanCare is a compact web application that modernises orphanage operations by moving child records, adoption workflows, and announcements into a single, role-based system. Built with PHP 8 and MySQL 8 on a shared LAMP host, it is inexpensive to deploy and straightforward to maintain.

### **6.2 Summary**

The release delivered five core modules—child registry, adoption management, news publishing, profile management, and basic reports—along with an admin dashboard and responsive pages. Replacing paper files with a central database reduced manual effort and made request tracking faster and more transparent for both staff and adopters.

### **6.3 Future Work**

Planned, non-breaking additions include notifications by SMS/email, a small analytics view for monthly trends, multilingual content, and optional integrations such as document-verification or a donation module.

### **6.4 Conclusion**

The project achieves its immediate aim: a secure, usable platform that fits the constraints of a typical Bangladeshi orphanage while improving accuracy and turnaround time. With modest enhancements, the same foundation can support a wider audience and more advanced features without changing the core design.

## References

1. PHP Group, "PHP Manual (v8.x)," *php.net*, Accessed: Aug. 2025.
2. Daffodil International University, "Thesis/Project Report Template and Guidelines," DIU, Accessed: Aug. 2025..
3. draw.io / diagrams.net, "User Guide and Diagramming Best Practices," *diagrams.net/doc*, Accessed: Aug. 2025.
4. Mermaid, "Mermaid Syntax Reference," *mermaid.js.org*, Accessed: Aug. 2025.
5. IEEE, "830-1998 IEEE Recommended Practice for Software Requirements Specifications," 1998 (historical, for terminology).

## Appendix A: Source Code

1. Source Code : [Asif-36/orphanage-management-system](#)

# Accounts Clearance

The screenshot displays a student dashboard with a dark sidebar on the left containing navigation options: Dashboard, Student Profile, Payment Ledger, Registration/Exam Clearance, Registered Course, Result, Routine, Live Result, Teaching Evaluation, Convocation Apply, Certificate & Transcript, Laptop, and Mentor Meeting. The main content area is titled 'Dashboard Student Portal' and features a user profile 'Sayad Md. Asif' with ID '211-15-763'. A summary table shows financial data: Total Payable (741,200.00), Total Paid (741,200.00), Total Due (0.00), and Total Other (1,150.00). Below this, a section for 'Today's Routine - Thursday' indicates 'No routine available for today.' The 'Semester Wise Result' section shows a 'Semester-wise SGPA Performance' chart with a legend for SGPA.

Total Payable	Total Paid	Total Due	Total Other
741,200.00	741,200.00	0.00	1,150.00

Today's Routine - Thursday  
No routine available for today.

Semester Wise Result  
Semester-wise SGPA Performance  
SGPA

## ORIGINALITY REPORT

<b>11</b> %	<b>9</b> %	<b>1</b> %	<b>8</b> %
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

## PRIMARY SOURCES

<b>1</b>	Submitted to Daffodil International University Student Paper	4%
<b>2</b>	<a href="https://dspace.daffodilvarsity.edu.bd:8080">dspace.daffodilvarsity.edu.bd:8080</a> Internet Source	4%
<b>3</b>	Submitted to TAR University College Student Paper	<1%
<b>4</b>	Submitted to Deptford Township High School Student Paper	<1%
<b>5</b>	Submitted to NCC Education Student Paper	<1%
<b>6</b>	Submitted to National Institute of Business Management Sri Lanka Student Paper	<1%
<b>7</b>	Submitted to Middlesex University Student Paper	<1%
<b>8</b>	Submitted to University of Durham Student Paper	<1%
<b>9</b>	Submitted to University of Lancaster-Main Account Student Paper	<1%
<b>10</b>	Submitted to University of New England Student Paper	<1%
<b>11</b>	<a href="http://www.hermanstreet.com">www.hermanstreet.com</a> Internet Source	<1%
<b>12</b>	<a href="http://www1.communitech.ca">www1.communitech.ca</a> Internet Source	<1%

13

Hanoi Pedagogical University 2

Publication

<1%

---

14

vi-seem.eu

Internet Source

<1%

---

Exclude quotes Off

Exclude matches Off

Exclude bibliography Off