EAST WEST UNIVERSITY

Usage Case Scenario & System Specification: AmarShashtho

Group Number : 06

Submitted By:

1.Kamrul Islam (2021-2-60-041)
2.Sourav Roy (2020-3-60-028)
3.Afifa Tabassum (2023-1-60-242)
4.Meheran Sahedi Evan (2022-1-60-025)

Submitted To:

Md sabbir Hossain

Usage Case Scenario & System Specification: AmarShashtho

Date: August 12, 2025
Document Purpose: To provide a description of all functional requirements, initiating roles, system processes, and exception handling for the AmarShashtho platform. This serves as the authoritative blueprint for the application's lifecycle, ensuring consistent understanding between developers, testers, designers, and project managers.

1. System Overview

AmarShashtho is an intelligent health companion platform designed to make complex medical information understandable and actionable. It features:

- **Modern dark-themed full-width UI** with immersive 3D homepage background.
- **Multimodal AI (Med-Gemma)** for medical report analysis and symptom checking.
- **Therapeutic Chat** for confidential emotional and mental well-being conversations.
- **Doctor directory** of verified medical professionals in Bangladesh.
- **Role-based access control** for Users and Administrators.
- **Quota and Pro membership system** for AI features.

2. Roles

- **Visitor:** Not logged in; can only view informational pages and access sign-up/login.
- **Registered User:** Logged-in account holder (Free or Pro tier).
- **Administrator:** Privileged account for platform oversight and data management.

3. Account Management

3.1 Create Account

Initiated By: Visitor

Preconditions:

- Not logged in.
- Username not already in the database.

Flow:

1. Navigate to `/signup`.
2. Enter username and password.
3. If registering as admin, reveal the secret key field and enter the key.
4. System validates:
    - The username is unique.
    - Secret key (if admin) is correct.
5. The system hashes passwords with bcrypt.
6. New record created:
7. Redirect to `/login` with success message.

Failure Conditions:

- Username taken → error message.
- Incorrect admin key → error message.

3.2 Authenticate User (Login)

Initiated By: Visitor

Flow:

1. Navigate to `/login`.
2. Enter username and password.
3. System validates password against stored hash.
4. If correct:
   - Create secure session cookie.
   - Redirect to `/dashboard` (user) or `/admin/dashboard` (admin).
5. If incorrect → error message.

3.3 Upgrade to Pro

Initiated By: Registered User (Free Tier)

Flow:

1. Click “Upgrade to Pro” on dashboard.
2. System shows Pro benefits.
3. User confirms via mock payment form.
4. System updates:
5. Transaction logged in **`payments`** table.
6. Dashboard updated with Pro status.

4. AI-Powered Analysis Tools

4.1 Perform AI Report Analysis

Initiated By: Registered User

Preconditions:

- Logged in.
- Free tier must have `upload_quota > 0`.

Flow:

1. Navigate to `/ai/query`.
2. Upload file (PDF/JPG/PNG) and/or enter text.
3. Quota check; block if zero (for free users).
4. Pre-process file (PDF → image).
5. Build AI prompt with strict JSON structure + specialty list constraint.
6. Send request to LM Studio API.
7. Parse and clean AI JSON response.
8. Store interaction in `queries` table.
9. Match suggested specialties to doctors in database.
10. Display structured analysis + matched doctors (or Google Maps link if none).

Failure Conditions:

- Malformed AI response → error message.
- Quota exceeded → block request.

4.2 Use Interactive Symptom Checker

Initiated By: Registered User

Flow:

1. Navigate to `/symptom_checker`.
2. Initialize empty chat history.
3. AI asks for primary symptom.
4. User responds → AI returns single clarifying question (JSON).
5. Repeat for 3 turns.
6. Final AI response returns JSON with:
   - `POSSIBLE_CAUSES`
   - `SUGGESTED_SPECIALTIES`
   - `NEXT_STEPS`
7. Display structured results + filtered doctor search link.

5. Doctor Directory & User History

5.1 Find a Doctor

Initiated By: Registered User

Flow:

1. Navigate to `/doctors`.
2. Display search form + full doctor list.
3. Apply filters (name, specialty, location).
4. Click doctor card for detailed profile.

5.2 View Past Analysis

Initiated By: Registered User

Flow:

1. On `/dashboard`, click a past analysis entry.
2. Retrieve `queries` record (ownership verified).
3. Parse AI response.
4. Fetch matched doctors.
5. Render complete result view.

6. User Support & Emergency Tools

6.1 Therapeutic Chat

Initiated By: Registered User

Flow:

1. Navigate to `/therapeutic_chat`.
2. Load existing chat from session or start new with welcome message.
3. User sends message → append to chat history.
4. Send conversation to AI with empathetic, plain-text response prompt.
5. Append AI response to history and display.
6. If “Clear Conversation” clicked → delete history and reload fresh.

6.2 Emergency Button

Initiated By: Registered User

Flow:

1. Click “Emergency” button.
2. Browser requests location access.
3. If granted → open Google Maps with “Ambulance near me” centered on user.
4. If denied → open general “Ambulance service” search.

7. Administrative Management

7.1 Administrative Access

Initiated By: Administrator

Flow:

1. Navigate to `/admin/dashboard`.
2. Verify admin role in session.
3. Display dashboard with:
   - User stats
   - Doctor stats
   - Recent AI activity
   - System health

7.2 Monitoring & Reporting

- View all users: username, role, Pro status, quota, last login, join date.
- View all doctors: name, specialty, location, contact.
- Search/filter both lists.

- Export lists to CSV.

7.3 User Management

Edit User Account

- Change role, Pro status, quota.
- Save changes → log in `admin_actions`.

Reset User Quota

- Reset quota to default and extend reset date by 30 days.

Delete User Account

- Confirmation required.
- Deletes user + related records.
- Block deletion of other admin accounts.

7.4 Doctor Database Management

Bulk Update (Offline)

1. Replace `BD_Doctor_Search.json`.
2. Stop server.
3. Run `python init_db.py` → clears table + imports JSON.
4. Restart server.