Use Case Diagram - AI-Powered Pothole Detection System

UML Use Case Diagram

```
┌────────────────────────────────────────────────────────────┐
                                    │
│
                                    │     AI-Powered Pothole Detection & Reporting
System       │
                                    │
│
└────────────────────────────────────────────────────────────┘

     ┌───────────┐
┌───────────┐
     │           │
│           │
     │   Guest   │
│   Admin   │
     │   User    │
│  (Future) │
     │           │
│           │
     └─────┬─────┘
└─────┬─────┘
           │
│
           │
│
           │  ┌──────────────────────────────────────────────────────────────────┐
│
           │  │                                                                  │
│
           ├──┤  Register Account                                                │
│
           │  │  (Email/Password + OTP Verification)                             │
│
           │  │                                                                  │
│
           │  └──────────────────────────────────────────────────────────────────┘
│
           │
│
           │  ┌──────────────────────────────────────────────────────────────────┐
│
           │  │                                                                  │
```

```
│
           ├──┤  Login to System                                       │
│
           │  │  «extends» OAuth Login (Google)
│◄────────┤
           │  │  «extends» OAuth Login (Microsoft)                     │
│
           │  │                                                         │
│
           │  └─────────────────────────────────────────────────────────┘
│
           │
│
           │  ┌─────────────────────────────────────────────────────────┐
│
           │  │                                                         │
│
           ├──┤  Verify Email OTP                                       │
│
           │  │  «includes» Resend OTP                                  │
│
           │  │                                                         │
│
           │  └─────────────────────────────────────────────────────────┘
│
           │
│
           │
│
      ┌─────┴─────┐
┌─────────┴─────────┐
      │           │                                                    │
│
      │Registered │                                                    │
Municipal Admin │
      │   User    │                                                    │
(Future)        │
      │           │                                                    │
│
      └─────┬─────┘
└───────────────────┘
           │
│
           │  ┌─────────────────────────────────────────────────────────┐
│
           │  │                                                         │
│
           ├──┤  Upload Road Image                                     │
│
           │  │  (JPEG/PNG/WebP, Max 10MB)                             │
│
           │  │                                                         │
│
           │  └────────────────────┬────────────────────────────────────┘
```

```
   │
              │                       │
   │
              │                       │ «includes»
   │
              │                       │
   │
              │                       ▼
   │
              │  ┌──────────────────────────────────────────────────────────────────┐
   │
              │  │                                                                  │
   │
              │  │  ┌────────────────────────────────────────────────┐             │
   │
              │  │  │   AI Detection System                           │             │
   │
              │  │  │   • Analyze Image with CNN Model                │             │
   │
              │  │  │   • Calculate Confidence Score (0-100%)         │◄───────┐    │
   │
              │  │  │   • Measure Prediction Time                     │        │    │
   │
              │  │  │   • Generate Repair Recommendation              │        │    │
   │
              │  │  └────────────────────────────────────────────────┘       │    │
   │
              │  │                                                           │    │
   │
              │  └───────────────────────────────────────────────────────────┼────┘
   │
              │                       │                                       │
   │
              │                       │ «includes»                            │
   │
              │                       │                                       │
   │
              │                       ▼                                       │
   │
              │  ┌──────────────────────────────────────────────────────────────────┐
   │
              │  │                                                                  │
   │
              ├──┤  View Detection Results                                         │
   │
              │  │  • Pothole Status (Detected/Not Detected)                       │
   │
              │  │  • Confidence Percentage                                        │
   │
              │  │  • Processing Time                                              │
   │
              │  │  • Recommendation                                               │
   │
              │  │                                                                  │
```

```
│
           │  └─────────────────────────────────────────────────────────────────┘
│
           │
│
           │  ┌─────────────────────────────────────────────────────────────────┐
│
           │  │                                                                 │
│
           ├──┤  Submit Complaint                                               │
│
           │  │  «requires» Detection Complete                                  │
│
           │  │  «includes» Add Location                                        │
│
           │  │  «includes» Add Description                                     │
│
           │  │  «includes» Attach Image Evidence                               │
│
           │  │                                                                 │
│
           │  └─────────────────────────────────────────────────────────────────┘
│
           │                        │
│
           │                        │ «includes»
│
           │                        │
│
           │                        ▼
│
           │  ┌─────────────────────────────────────────────────────────────────┐
│
           │  │                                                                 │
│
           │  │  Store Complaint in Database                                    │
│
           │  │  • Save Location                                                │
│
           │  │  • Save Description                                             │
│
           │  │  • Save Base64 Image                                            │
│
           │  │  • Save Confidence Score                                        │
│
           │  │  • Link to User Account                                         │
│
           │  │  • Timestamp Creation                                           │
│
           │  │                                                                 │
│
           │  └─────────────────────────────────────────────────────────────────┘
│
           │
```

```
    │
              │  ┌──────────────────────────────────────────────────────────┐
    │
              │  │                                                          │
    │
              ├──┤  View Submitted Complaints                               │
    │
              │  │  (History of User's Reports)                             │
    │
              │  │                                                          │
    │
              │  └──────────────────────────────────────────────────────────┘
    │
              │
    │
              │  ┌──────────────────────────────────────────────────────────┐
    │
              │  │                                                          │
    │
              ├──┤  View About & Information Pages                         │
    │
              │  │  • Learn About Project                                   │
    │
              │  │  • View Workflow Diagram                                 │
    │
              │  │  • Understand Life-Saving Impact                         │
    │
              │  │                                                          │
    │
              │  └──────────────────────────────────────────────────────────┘
    │
              │
    │
              │  ┌──────────────────────────────────────────────────────────┐
    │
              │  │                                                          │
    │
              ├──┤  Send Contact Message                                    │
    │
              │  │  (Contact Form Submission)                               │
    │
              │  │                                                          │
    │
              │  └──────────────────────────────────────────────────────────┘
    │
              │
    │
              │  ┌──────────────────────────────────────────────────────────┐
    │
              │  │                                                          │
    │
              ├──┤  Logout from System                                      │
    │
              │  │  (Clear Session & JWT Token)                             │
```

```
│
          │  │                                                              │
│
          │  └──────────────────────────────────────────────────────────────┘
│
          │
│
          │
│
          │
┌─────────────┴──────────┐
          │                                                          │
│
          │                                                     ┌────┤
View All Complaints    │
          │                                                     │    │
(Dashboard)             │
          │                                                     │    │
│
          │                                                     │
└────────────────────────┘
          │                                                     │
          │                                                     │
┌────────────────────────┐
          │                                                     │    │
│
          │                                                     ├────┤
Filter Complaints       │
          │                                                     │    │
(By Date/Location)      │
          │                                                     │    │
│
          │                                                     │
└────────────────────────┘
          │                                                     │
          │                                                     │
┌────────────────────────┐
          │                                                     │    │
│
          │                                                     ├────┤
Verify Complaint        │
          │                                                     │    │
(Mark as Valid)         │
          │                                                     │    │
│
          │                                                     │
└────────────────────────┘
          │                                                     │
          │                                                     │
┌────────────────────────┐
          │                                                     │    │
│
          │                                                     ├────┤
Mark Complaint Resolved │
```

```
           │                                                             │    │
  (Update Status)           │
           │                                                             │    │
│
           │                                                             │
└───────────────────────────┘
           │                                                             │
           │                                                             │
┌───────────────────────────┐
           │                                                             │    │
│
           │                                                             └────┤
  Generate Reports          │
           │                                                                  │
  (Analytics Dashboard)     │
           │                                                                  │
│
           │
└───────────────────────────┘
           │
           │

┌─────┴───────────────────────────────────────────────────────────────────────┐
     │
│
     │                          EXTERNAL SYSTEMS
│
     │
│
     │  ┌─────────────────┐  ┌─────────────────┐  ┌─────────────────┐
┌───────────────┐ │
     │  │                 │  │                 │  │                 │  │
│ │
     │  │  Google OAuth   │  │ Microsoft Auth  │  │  Brevo Email    │  │   MongoDB
│ │
     │  │   Service       │  │   Platform      │  │     API         │  │   Database
│ │
     │  │                 │  │                 │  │                 │  │
│ │
     │  └─────────────────┘  └─────────────────┘  └─────────────────┘
└───────────────┘ │
     │
│

└──────────────────────────────────────────────────────────────────────────────
─┘
```

Actor Descriptions

1. Guest User (Unauthenticated)

Description: Any visitor to the website without an account.

Capabilities:

- Register new account
- Login with email/password
- Login with Google OAuth
- Login with Microsoft OAuth
- View public pages (Home, About, Workflow)

Limitations:

- Cannot upload images
- Cannot submit complaints
- Cannot view complaint history

---

2. Registered User (Authenticated)

Description: Verified user with an active account and valid JWT token.

Capabilities:

- All Guest User capabilities, plus:
- Upload road images for analysis
- View AI detection results
- Submit complaints with evidence
- View own complaint history
- Send contact messages
- Logout from system

Authentication Methods:

- Email/Password (with OTP verification)
- Google OAuth 2.0
- Microsoft OAuth 2.0

---

3. AI Detection System (Automated Actor)

Description: Python/Flask microservice with trained CNN model.

Responsibilities:

- Receive uploaded images
- Preprocess images (resize, normalize)
- Perform inference with PyTorch CNN
- Calculate confidence scores
- Measure prediction time
- Generate repair recommendations
- Return JSON results to frontend

Technology:

- PyTorch 2.0 CNN model
- Flask REST API
- Deployed on Heroku

---

4. Backend Server (System Actor)

Description: Node.js/Express REST API handling business logic.

Responsibilities:

- User authentication and authorization
- JWT token generation and validation
- OTP generation and verification
- OAuth integration (Google, Microsoft)
- Complaint data persistence
- Email delivery via Brevo API
- Rate limiting and security enforcement

Technology:

- Node.js 18 + Express
- MongoDB (Mongoose ODM)
- Passport.js (OAuth)
- Deployed on Render

---

5. Municipal Admin (Future Enhancement)

Description: Government official managing submitted complaints.

Planned Capabilities:

- View all user complaints
- Filter complaints by date/location/status
- Verify complaint authenticity
- Mark complaints as resolved
- Generate analytics reports
- Export data for planning

Status: Not yet implemented (Phase 2)

Use Case Specifications

UC-001: Register Account

Attribute	Description

Attribute	Description
Use Case ID	UC-001
Use Case Name	Register Account
Actor	Guest User
Description	New user creates an account with email and password
Preconditions	User has valid email address
Postconditions	Account created, OTP sent to email
Normal Flow	1. User navigates to signup page 2. User enters email and password 3. System validates input 4. System hashes password with bcrypt 5. System generates 6-digit OTP 6. System sends OTP via Brevo API 7. System displays OTP verification page
Alternative Flows	**3a. Email already exists:** 1. System displays error "Email already registered" 2. User redirects to login page **6a. Email delivery fails:** 1. System logs error 2. System displays generic success message 3. User can request resend
Exception Flows	**E1. Invalid email format:** Display inline error "Enter valid email" **E2. Weak password:** Display error "Password must be 8+ characters"

UC-002: Verify Email OTP

Attribute	Description
Use Case ID	UC-002
Use Case Name	Verify Email OTP
Actor	Guest User
Description	User verifies email ownership with OTP
Preconditions	User registered, OTP sent to email
Postconditions	Account activated, user logged in

Attribute	Description
Normal Flow	1. User receives OTP email 2. User enters 6-digit OTP 3. System validates OTP hash 4. System checks expiration (10 min) 5. System activates account 6. System generates JWT token 7. System redirects to dashboard
Alternative Flows	**3a. Invalid OTP:** 1. Increment attempt counter 2. Display error "Invalid OTP" 3. Allow 3 total attempts **4a. OTP expired:** 1. Delete expired OTP 2. Display error with resend option **UC-002a. Resend OTP:** 1. User clicks "Resend OTP" 2. System generates new OTP 3. System sends new email 4. Display success message
Exception Flows	**E1. Maximum attempts exceeded:** Lock account for 15 minutes

---

UC-003: Login to System

Attribute	Description
Use Case ID	UC-003
Use Case Name	Login to System
Actor	Registered User
Description	User authenticates with credentials
Preconditions	User has verified account
Postconditions	User authenticated, JWT issued

Attribute	Description
Normal Flow	1. User navigates to login page 2. User enters email and password 3. System validates credentials 4. System compares bcrypt hash 5. System generates JWT (24h expiry) 6. System stores token in localStorage 7. System redirects to pothole detection page
Alternative Flows	**UC-003a. OAuth Login (Google):** 1. User clicks "Sign in with Google" 2. System redirects to Google consent 3. User authorizes app 4. Google returns auth code 5. Backend exchanges for user profile 6. System creates/links account 7. System generates JWT 8. Frontend stores token and redirects **UC-003b. OAuth Login (Microsoft):** (Same flow as Google, different provider)
Exception Flows	**E1. Invalid credentials:** Display error "Invalid email or password" **E2. Unverified account:** Redirect to OTP verification page **E3. Rate limit exceeded:** Return 429 error, wait 15 minutes

---

UC-004: Upload Road Image

Attribute	Description
Use Case ID	UC-004
Use Case Name	Upload Road Image
Actor	Registered User
Description	User uploads image for pothole detection
Preconditions	User authenticated with valid JWT
Postconditions	Image uploaded, displayed in preview

Attribute	Description
Normal Flow	1. User clicks upload area or selects file 2. User chooses image (JPEG/PNG/WebP) 3. System validates file type 4. System validates file size (<10MB) 5. System creates base64 preview 6. System displays image preview 7. User clicks "Detect Pothole" button 8. System includes UC-005 (AI Detection)
Alternative Flows	**3a. Invalid file type:** Display error "Only JPEG, PNG, WebP allowed" **4a. File too large:** Display error "Maximum file size is 10MB"
Exception Flows	**E1. No file selected:** Keep detect button disabled

---

UC-005: AI Detects Pothole

Attribute	Description
Use Case ID	UC-005
Use Case Name	AI Detects Pothole
Actor	AI Detection System (Automated)
Description	CNN model analyzes image for potholes
Preconditions	Image uploaded to Flask service
Postconditions	Detection results returned as JSON
Normal Flow	1. Flask receives POST /predict request 2. System reads image file from FormData 3. System resizes image to model input size 4. System normalizes pixel values 5. System converts to PyTorch tensor 6. System performs forward pass through CNN 7. System applies softmax activation 8. System extracts confidence score 9. System measures processing time 10. System determines binary classification 11. System generates recommendation 12. System returns JSON response
Alternative Flows	None (automated process)

Attribute	Description
Exception Flows	**E1. Invalid image format:** Return 400 error "Invalid image" **E2. Model inference failure:** Return 500 error "Detection failed" **E3. Timeout (60 seconds):** Return 504 Gateway Timeout

JSON Response Schema:

```json
{
  "is_pothole": true,
  "confidence": 0.9576,
  "prediction_time": 1.23
}
```

UC-006: View Detection Results

Attribute	Description
Use Case ID	UC-006
Use Case Name	View Detection Results
Actor	Registered User
Description	User views AI analysis results
Preconditions	AI detection complete (UC-005)
Postconditions	Results displayed, complaint form enabled
Normal Flow	1. System receives JSON from AI service 2. System parses detection data 3. System displays prediction status ("Pothole Detected" or "No Pothole") 4. System displays confidence percentage 5. System displays processing time 6. System displays recommendation 7. System enables complaint form (if pothole detected) 8. User reviews results
Alternative Flows	**7a. No pothole detected:** Keep complaint form disabled Display message "No pothole found"

Attribute	Description
Exception Flows	**E1. AI service unavailable:** Display error "Service temporarily unavailable" **E2. Invalid response format:** Display generic error message

---

UC-007: Submit Complaint

Attribute	Description
Use Case ID	UC-007
Use Case Name	Submit Complaint
Actor	Registered User
Description	User reports pothole with evidence
Preconditions	- User authenticated - Pothole detected (UC-005) - Detection results viewed (UC-006)
Postconditions	Complaint stored in database
Normal Flow	1. User enters location (text field) 2. User enters description (textarea) 3. User clicks "Submit Complaint" 4. System validates required fields 5. System extracts JWT from localStorage 6. System prepares complaint payload: - location - description - imageData (base64) - confidence (raw 0-1 value) 7. System sends POST /api/complaints 8. Backend validates JWT 9. Backend creates complaint document 10. MongoDB stores complaint 11. Backend returns success response 12. System displays success message 13. System redirects to "Life Saver" page

Attribute	Description
Alternative Flows	**4a. Missing required fields:** Display inline errors Prevent submission **8a. Invalid/expired JWT:** Return 401 Unauthorized Redirect to login page
Exception Flows	**E1. Payload too large (>10MB):** Return 413 error "Image too large" **E2. Database connection failed:** Return 500 error "Submission failed" **E3. Network timeout:** Display error "Check connection, try again"

Includes: UC-008 (Store Complaint)

UC-008: Store Complaint in Database

Attribute	Description
Use Case ID	UC-008
Use Case Name	Store Complaint in Database
Actor	Backend Server (System)
Description	Persist complaint data to MongoDB
Preconditions	Valid complaint payload received
Postconditions	Complaint document created
Normal Flow	1. Backend extracts user ID from JWT 2. Backend creates Complaint object: - userId (ObjectId reference) - location (String) - description (String) - imageData (base64 String) - confidence (Number 0-1) - status: "pending" - createdAt (Date) 3. Mongoose validates schema 4. MongoDB inserts document 5. MongoDB returns document ID 6. Backend sends success response

Attribute	Description
Alternative Flows	None (internal system process)
Exception Flows	**E1. Validation error:** Return 400 Bad Request **E2. Database write failure:** Return 500 Internal Error **E3. Duplicate detection:** Log warning, proceed (no unique constraint)

UC-009: View Submitted Complaints

Attribute	Description
Use Case ID	UC-009
Use Case Name	View Submitted Complaints
Actor	Registered User
Description	User views history of their complaints
Preconditions	User authenticated
Postconditions	Complaint list displayed
Normal Flow	1. User navigates to complaints page 2. Frontend sends GET /api/complaints 3. Backend validates JWT 4. Backend extracts user ID 5. MongoDB queries complaints by userId 6. MongoDB returns complaint array 7. Backend sends JSON response 8. Frontend displays complaint list 9. User views complaint details
Alternative Flows	**5a. No complaints found:** Return empty array Display "No complaints yet"
Exception Flows	**E1. Unauthorized access:** Return 401, redirect to login

---

UC-010: Logout from System

Attribute	Description
Use Case ID	UC-010

Attribute	Description
Use Case Name	Logout from System
Actor	Registered User
Description	User ends authenticated session
Preconditions	User logged in
Postconditions	Session cleared, user logged out
Normal Flow	1. User clicks "Logout" button 2. Frontend clears localStorage: - Remove authToken - Remove user object 3. Frontend resets app state 4. Frontend redirects to home page 5. Backend session destroyed (if using cookies)
Alternative Flows	None
Exception Flows	**E1. Already logged out:** No action, redirect to home

---

Future Use Cases (Admin Panel - Phase 2)

UC-ADMIN-001: View All Complaints

Actor: Municipal Admin
Description: Admin views dashboard with all submitted complaints

UC-ADMIN-002: Filter Complaints

Actor: Municipal Admin
Description: Admin filters complaints by date range, location, or status

UC-ADMIN-003: Verify Complaint

Actor: Municipal Admin
Description: Admin marks complaint as verified after field inspection

UC-ADMIN-004: Mark Complaint Resolved

Actor: Municipal Admin
Description: Admin updates complaint status to "resolved" after repair

UC-ADMIN-005: Generate Analytics Report

Actor: Municipal Admin
Description: Admin exports complaint statistics and trends for planning

---

System Integration Use Cases

UC-SYS-001: Authenticate with Google OAuth

Actor: Google OAuth Service (External)
Description: System exchanges authorization code for user profile data

UC-SYS-002: Authenticate with Microsoft OAuth

Actor: Microsoft Identity Platform (External)
Description: System exchanges authorization code for user profile data

UC-SYS-003: Send OTP Email

Actor: Brevo Email API (External)
Description: System sends transactional OTP verification email

UC-SYS-004: Store User Data

Actor: MongoDB Database (External)
Description: System persists user accounts, complaints, and OTPs

---

Use Case Relationships

«includes» Relationships

- **UC-001 (Register)** includes **UC-SYS-003 (Send OTP Email)**
- **UC-004 (Upload Image)** includes **UC-005 (AI Detection)**
- **UC-005 (AI Detection)** includes **UC-006 (View Results)**
- **UC-007 (Submit Complaint)** includes **UC-008 (Store Complaint)**

«extends» Relationships

- **UC-003a (OAuth Login Google)** extends **UC-003 (Login)**
- **UC-003b (OAuth Login Microsoft)** extends **UC-003 (Login)**
- **UC-002a (Resend OTP)** extends **UC-002 (Verify OTP)**

«requires» Relationships

- **UC-007 (Submit Complaint)** requires **UC-005 (AI Detection)** to complete
- **UC-009 (View Complaints)** requires **UC-003 (Login)** authentication

---

Non-Functional Use Cases

UC-NFR-001: System Performance

Description: All API responses complete within 500ms (read operations)

UC-NFR-002: Security Enforcement

Description: System enforces rate limiting (5 auth attempts per 15 min)

UC-NFR-003: Data Encryption

Description: All communications encrypted with TLS 1.3

UC-NFR-004: Concurrent User Support

Description: System handles 100+ concurrent users without degradation

---

PlantUML Code (For Presentation)

```
@startuml
left to right direction
skinparam packageStyle rectangle

actor "Guest User" as guest
actor "Registered User" as user
actor "AI Detection System" as ai
actor "Backend Server" as backend
actor "Municipal Admin\n(Future)" as admin

rectangle "Pothole Detection System" {
  usecase "Register Account" as UC1
  usecase "Login to System" as UC2
  usecase "Verify Email OTP" as UC3
  usecase "Upload Road Image" as UC4
  usecase "AI Detects Pothole" as UC5
  usecase "View Detection Results" as UC6
  usecase "Submit Complaint" as UC7
  usecase "Store Complaint" as UC8
  usecase "View Complaints" as UC9
  usecase "Logout" as UC10

  usecase "OAuth Login (Google)" as UC2a
  usecase "OAuth Login (Microsoft)" as UC2b
  usecase "Resend OTP" as UC3a

  usecase "View All Complaints" as UCA1
  usecase "Mark Resolved" as UCA2
  usecase "Generate Reports" as UCA3
}

guest --> UC1
guest --> UC2
guest --> UC3

user --> UC4
user --> UC6
user --> UC7
user --> UC9
user --> UC10

ai --> UC5

UC2 <.. UC2a : <<extends>>
UC2 <.. UC2b : <<extends>>
UC3 <.. UC3a : <<extends>>

UC1 ..> UC3 : <<includes>>
UC4 ..> UC5 : <<includes>>
UC5 ..> UC6 : <<includes>>
UC7 ..> UC8 : <<includes>>

admin --> UCA1
admin --> UCA2
admin --> UCA3

@enduml
```

Presentation Tips for Interviewers

🎯 Key Points to Highlight:

1. **Clear Actor Separation:**
   - Guest vs. Registered User (shows access control understanding)
   - AI System as automated actor (shows system design knowledge)
   - Admin for future scalability (shows forward thinking)
2. **Well-Defined Relationships:**
   - «includes» for mandatory sub-processes
   - «extends» for optional variations (OAuth)
   - «requires» for preconditions
3. **Comprehensive Coverage:**
   - 10 core use cases documented
   - 5 future admin use cases planned
   - 4 system integration use cases identified
4. **Enterprise Thinking:**
   - Security-focused (authentication, authorization)
   - Scalable design (admin panel planned)
   - Real-world constraints (rate limiting, timeouts)

📊 What Interviewers Love:

✅ **Clean, professional UML diagrams**
✅ **Detailed use case specifications with flows**

✅ **Clear preconditions and postconditions**
✅ **Exception handling documented**
✅ **Future enhancements planned**
✅ **System integration awareness**

Generated: November 18, 2025
Project: AI-Powered Pothole Detection System
Document Version: 1.0