

# DoConnect Project

## Use Case Document with Deliverables of Sprint (1, 2 & 3)

Prepared by: Utsav Dubey  
Date: 07/09/2025

---

# Sprint1

## Foundation & Design

---

### Use Case 1.1: Project Initialization and Requirement Gathering

**Actors:** Product Owner, Development Team

**Goal:** Define the project scope, understand client needs, and build initial documentation

**Precondition:** Project approved and resources assigned

**Trigger:** Project kickoff meeting

**Main Flow:**

- 1.) Product owner presents project goals and scope to the team
- 2.) Development team documents initial requirements and use cases
- 3.) Draft use case documents and user stories created
- 4.) Stakeholders review and approve the documentation

**Deliverables:**

- 1.) Approved project scope document
- 2.) Requirements specification
- 3.) Use case documentation draft

### Use Case 1.2: Database Schema and ER Diagram Design

**Actors:** Backend Developer, Database Architect

**Goal:** Design a clear and normalized database schema supporting project requirements

**Preconditions:** Requirement document ready

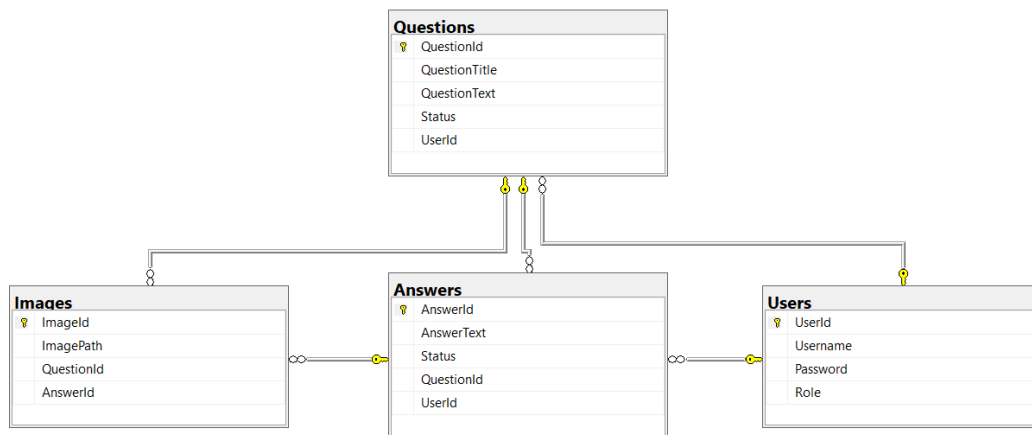
**Trigger:** Completion of initial requirement gathering

## Main Flow:

- 1.)Draft an entity-relationship diagram (ERD)
- 2.)Review schema for integrity and normalization
- 3.)Finalize and document the database design

## Deliverables:

- 1.)Entity Relationship Diagram (ERD)
- 2.)Database schema document



\*Figure 1:Entity Relation Diagram\*

## Use Case 1.3: Backend Project Setup

**Actors:** Development Team

**Goal:** Initialize backend codebase with foundational modules

**Preconditions:** High-level design approved

**Trigger:** Completion of database design

**Main Flow:**

- 1.)Setup ASP.NET Core MVC project structure
- 2.)Configure Entity Framework Core, add DbContext and model classes
- 3.)Create base controllers and test project initialization

**Deliverables:**

- 1.)Backend project repository initialized
- 2.)Codebase with base controller and models
- 3.)Project build and run confirmation

# Sprint 2

## Functionality & Integration

---

### Use Case 2.1: User Registration and Secure Authentication

- **Actors:** End Users (Students/Admins)
- **Goal:** Allow users to safely register, login, and be authenticated using JWT
- **Preconditions:** Backend project functional; database tables created
- **Trigger:** User submits registration or login form
- **Main Flow:**
  1. User inputs credentials and submits form
  2. System hashes passwords, validates credentials
  3. JWT token is generated and returned on successful authentication
  4. User session starts with role-based access control

### Deliverables:

- 1.) JWT authentication mechanism implemented
- 2.) User registration and login endpoints
- 3.) Verified secure login flow with token handling

## Use Case 2.2: CRUD APIs for Questions, Answers, and Image Upload

- **Actors:** Authenticated Users (Standard and Admin)
- **Goal:** Enable users to create, read, update, delete questions and answers, and upload images
- **Preconditions:** User authenticated; backend project running
- **Trigger:** User action to create/edit/delete content or upload images
- **Main Flow:**
  1. User submits data via frontend interfaces
  2. Backend validates and processes CRUD operations and image uploads
  3. Data store updated; confirmation response sent to frontend
- **Deliverables:**
  - Fully functional CRUD API endpoints for questions and answers
  - Image upload API supporting multipart/form-data
  - Integrated frontend services consuming APIs

## Use Case 2.3: Frontend Integration with Backend APIs

- **Actors:** Development Team (Frontend and Backend)
- **Goal:** Connect Angular frontend components with backend APIs effectively
- **Preconditions:** APIs tested and ready; frontend scaffold prepared
- **Trigger:** Beginning of frontend development and integration phase
- **Main Flow:**
  1. Implement Angular services for API calls
  2. Bind UI components to service responses and handle errors/validation
  3. Test end-to-end data flow between frontend and backend
- **Deliverables:**
  - Angular services integrated with backend APIs
  - Working UI components with dynamic data
  - Complete end-to-end tested user workflows

# Sprint 3: Advanced Features & Polish

---

## Use Case 3.1: Search Functionality Implementation

- **Actors:** Authenticated Users
- **Goal:** Allow users to search questions using keywords and filters
- **Preconditions:** Questions data populated; search API endpoint implemented
- **Trigger:** User initiates search via frontend UI
- **Main Flow:**
  1. User enters search criteria
  2. Request sent to backend search API
  3. API returns matching questions
  4. UI displays results dynamically
- **Deliverables:**
  - Search API endpoint with keyword filtering
  - Frontend search component with real-time results
  - Tested and validated search experience



## Use Case 3.2: Admin Approval Workflow and Real-Time Notifications

- **Actors:** Admin Users
- **Goal:** Enable admins to approve or reject user-generated content and receive real-time alerts
- **Preconditions:** Content moderation features enabled; SignalR notifications configured
- **Trigger:** New content submitted requiring approval
- **Main Flow:**
  1. Admin receives real-time notification of pending content via SignalR
  2. Reviews content in admin dashboard
  3. Approves or rejects with optional comments
  4. Content status updated; notifications sent to content creators
- **Deliverables:**
  - Admin approval workflow implemented
  - Real-time admin notifications via SignalR
  - User notification system for status updates

## Use Case 3.3: End-to-End Testing and Performance Validation

- **Actors:** QA Team and Developers
- **Goal:** Verify system stability, security, and performance before release
- **Preconditions:** All features developed and integrated
- **Trigger:** Testing phase initiation
- **Main Flow:**
  1. Conduct unit, integration, and UI tests
  2. Perform security audits including JWT and role-based access validation
  3. Monitor API performance and database efficiency
  4. Record and fix defects
- **Deliverables:**
  - Complete test suite with passing results
  - Security validation report
  - Performance benchmarking documentation
  - Final deployment-ready software