CampusLearn™ Software Architecture Documentation

Document Overview

This document provides comprehensive technical documentation for the CampusLearn™ platform, detailing the database architecture, entity relationships, and microservices design.

System Architecture Overview

CampusLearn™ follows a **microservices architecture** with three independent bounded contexts, each with its own database and DbContext:

1. **User Management Microservice** - Core user data and authentication
2. **Topics & Q&A Microservice** - Tutor-led learning and academic support
3. **Forum Microservice** - Public anonymous discussions

---

1. User Management Microservice

Database Context: UserManagementDbContext

Purpose: Manages all user-related data, authentication, and academic module information.

Entity Definitions:

UserProfile - Core User Information

```
public class UserProfile

{

   public int UserProfileID { get; set; }        // Primary Key

   public UserRole UserRole { get; set; }        // Student, Tutor, or Admin

   public string Name { get; set; } = string.Empty;

   public string Surname { get; set; } = string.Empty;

   public string Email { get; set; } = string.Empty; // @belgiumcampus.ac.za

   public string ProfilePictureUrl { get; set; } = string.Empty;

   public Qualification Qualification { get; set; } // DIP, BIT, BCOM

   public int StudentNumber { get; set; }        // Campus student number

   public DateTime CreatedAt { get; set; }

   public DateTime? LastLogin { get; set; }

   public bool IsActive { get; set; }

}
```

Relationships:

- **1:1** with Login (Authentication details)
- **1:1** with Student (if role is Student)
- **1:1** with Tutor (if role is Tutor)

Login - Authentication System

```
public class Login

{

   public int LoginID { get; set; }             // Primary Key

   public string Email { get; set; } = string.Empty; // Unique identifier

   public string Password { get; set; } = string.Empty; // Hashed via external service

   public DateTime CreatedAt { get; set; }

   public DateTime? LastPasswordChange { get; set; }

   public bool IsActive { get; set; }

   public int FailedLoginAttempts { get; set; }   // Security lockout

   public DateTime? LockoutEnd { get; set; }      // Temporary lockout

   public int UserProfileID { get; set; }        // Foreign Key


   // Navigation

   public UserProfile UserProfile { get; set; } = null!;

}
```

Security Implementation:

- Passwords are hashed using an external HashingService
- Account lockout after multiple failed attempts
- Timestamp tracking for security auditing

Student - Student Specialization

```
public class Student
{
   public int StudentID { get; set; }          // Primary Key
   public int UserProfileID { get; set; }       // Foreign Key to UserProfile

   // Navigation
   public UserProfile UserProfile { get; set; } = null!;
}
```

Purpose: Extends UserProfile for students. Contains student-specific relationships and behaviors.

Tutor - Tutor Specialization

```
public class Tutor
{
   public int TutorID { get; set; }            // Primary Key
   public int UserProfileID { get; set; }       // Foreign Key to UserProfile
   public bool IsAdmin { get; set; }           // Administrative privileges
   public DateTime? AdminSince { get; set; }     // Admin promotion date

   // Navigation
   public UserProfile UserProfile { get; set; } = null!;
}
```

Purpose: Extends UserProfile for tutors. Includes admin capabilities and tutor-specific relationships.

Module - Academic Course Management

```
public class Module

{

   public int ModuleID { get; set; }           // Primary Key

   public string ModuleName { get; set; } = string.Empty; // e.g., "Web Programming"

   public string ModuleCode { get; set; } = string.Empty; // e.g., "WPR181" (Unique)

   public Qualification ProgramType { get; set; }  // DIP, BIT, BCOM

   public string Description { get; set; } = string.Empty;

   public bool IsActive { get; set; }

}
```

Purpose: Represents academic courses/modules at Belgium Campus.

StudentModule - Student Subscriptions (Join Table)

```
public class StudentModule

{

   public int StudentID { get; set; }          // Composite Key Part 1

   public int ModuleID { get; set; }           // Composite Key Part 2

   public DateTime SubscribedAt { get; set; }     // Subscription timestamp


   // Navigations

   public Student Student { get; set; } = null!;

   public Module Module { get; set; } = null!;

}
```

Purpose: Many-to-Many relationship between Students and Modules. Tracks which students need help in which modules.

TutorModule - Tutor Qualifications (Join Table)

```
public class TutorModule
{
   public int TutorID { get; set; }            // Composite Key Part 1
   public int ModuleID { get; set; }           // Composite Key Part 2
   public DateTime QualifiedSince { get; set; }   // Qualification date
   public bool IsActive { get; set; }          // Active status

   // Navigations
   public Tutor Tutor { get; set; } = null!;
   public Module Module { get; set; } = null!;
}
```

Purpose: Many-to-Many relationship between Tutors and Modules. Defines which tutors are qualified to help in which modules.

2. Topics & Q&A Microservice

Database Context: TopicsDbContext

Purpose: Manages tutor-led academic support, including FAQs, student queries, and tutor responses.

Entity Definitions:

FAQs - Frequently Asked Questions

```
public class FAQs
{
   public int FAQID { get; set; }              // Primary Key
   public string FrequentlyAskedQuestion { get; set; } = string.Empty;
   public string Answer { get; set; } = string.Empty;
   public DateTime CreatedAt { get; set; }
   public DateTime? UpdatedAt { get; set; }
   public bool IsPublished { get; set; }
   public int ViewCount { get; set; }

   // Cross-Service References
   public int TutorID { get; set; }            // Reference to UserManagement.Tutor
   public string ModuleCode { get; set; } = string.Empty; // Reference to Module
}
```

Purpose: Pre-defined common questions and answers created by tutors for quick student reference.

QueryTopic - Student Help Requests

```
public class QueryTopic
{
   public int QueryTopicID { get; set; }        // Primary Key
   public string QueryTopicTitle { get; set; } = string.Empty;
   public string QueryTopicDescription { get; set; } = string.Empty;
   public string RelatedModuleCode { get; set; } = string.Empty;
   public DateTime TopicCreationDate { get; set; }
   public DateTime? LastActivity { get; set; }
   public bool IsResolved { get; set; }         // Mark as solved
   public bool IsUrgent { get; set; }           // Priority flag

   // Cross-Service Reference
   public int StudentID { get; set; }           // Reference to UserManagement.Student

   // Navigation (Within Service)
   public ICollection<QueryResponse> Responses { get; set; } = new List<QueryResponse>();
}
```

Purpose: Individual help requests created by students. Forms the core of the tutor-student Q&A system.

QueryResponse - Tutor Responses

```
public class QueryResponse
{
    public int QueryResponseID { get; set; }       // Primary Key
    public string Comment { get; set; } = string.Empty;
    public string? MediaContentUrl { get; set; }   // PDFs, videos, images
    public DateTime ResponseCreationDate { get; set; }
    public bool IsSolution { get; set; }          // Marks as correct answer
    public int HelpfulVotes { get; set; }         // Community voting

    // Cross-Service References
    public int TutorID { get; set; }              // Reference to UserManagement.Tutor

    // Foreign Key (Within Service)
    public int QueryTopicID { get; set; }

    // Navigation (Within Service)
    public QueryTopic QueryTopic { get; set; } = null!;
}
```

Purpose: Responses from tutors to student queries. Supports rich media content and community voting.

3. Forum Microservice

Database Context: ForumDbContext

Purpose: Manages public anonymous discussions and community interactions.

Entity Definitions:

ForumTopic - Public Discussion Topics

```
public class ForumTopic
{
    public int ForumTopicID { get; set; }         // Primary Key
    public string ForumTopicTitle { get; set; } = string.Empty;
    public string ForumTopicDescription { get; set; } = string.Empty;
    public string RelatedModuleCode { get; set; } = string.Empty;
    public int TopicUpVote { get; set; }          // Community voting
    public int ViewCount { get; set; }            // Popularity metric
    public DateTime TopicCreationDate { get; set; }
    public DateTime? LastActivity { get; set; }

    // Anonymous Posting System
    public int? UserProfileID { get; set; }       // Optional: Reference to UserManagement
    public bool IsAnonymous { get; set; }
    public string? AnonymousName { get; set; }     // Optional display name

    // Moderation Features
    public bool IsLocked { get; set; }            // Prevent new responses
    public bool IsPinned { get; set; }            // Highlight important topics
    public bool IsFeatured { get; set; }          // Special highlighting

    // Navigation (Within Service)
    public ICollection<ForumTopicResponse> Responses { get; set; } = new List<ForumTopicResponse>();
}
```

Purpose: Public discussion threads that support both identified and anonymous posting.

ForumTopicResponse - Forum Replies

```
public class ForumTopicResponse
{
    public int ResponseID { get; set; }           // Primary Key
    public string Comment { get; set; } = string.Empty;
    public string? MediaContentUrl { get; set; }    // Supporting media
    public int ResponseUpVote { get; set; }        // Community voting
    public DateTime ResponseCreationDate { get; set; }

    // Anonymous Posting System
    public int? UserProfileID { get; set; }        // Optional: Reference to UserManagement
    public bool IsAnonymous { get; set; }
    public string? AnonymousName { get; set; }

    // Foreign Key (Within Service)
    public int ForumTopicID { get; set; }

    // Navigation (Within Service)
    public ForumTopic ForumTopic { get; set; } = null!;
}
```

Purpose: Individual replies to forum topics, supporting the same anonymous posting features.

Cross-Service Relationships & Data Consistency

Foreign Key References Between Microservices

From Entity (Service)	To Entity (Service)	Reference Type	Purpose
Topics.FAQs.TutorID	UserManagement.Tutor.TutorID	Cross-Service	FAQ authorship
Topics.QueryTopic.StudentID	UserManagement.Student.StudentID	Cross-Service	Query ownership
Topics.QueryResponse.TutorID	UserManagement.Tutor.TutorID	Cross-Service	Response authorship
Forum.ForumTopic.UserProfileID	UserManagement.UserProfile.UserProfileID	Optional Cross	Identified posting
Forum.ForumTopicResponse.UserProfileID	UserManagement.UserProfile.UserProfileID	Optional Cross	Identified responses

Data Consistency Strategy

1. **Eventual Consistency**: Cross-service references may have temporary inconsistencies
2. **API Validation**: Services validate foreign key existence via API calls
3. **Caching**: Frequently accessed user data cached locally
4. **Cleanup Jobs**: Periodic cleanup of orphaned references

Database Schema Summary

UserManagement Database Tables:

- UserProfiles - Core user information
- Logins - Authentication credentials
- Students - Student role extensions
- Tutors - Tutor role extensions
- Modules - Academic course catalog
- StudentModules - Student subscriptions (Many-to-Many)
- TutorModules - Tutor qualifications (Many-to-Many)

Topics Database Tables:

- FAQs - Frequently asked questions
- QueryTopics - Student help requests
- QueryResponses - Tutor answers and solutions

Forum Database Tables:

- ForumTopics - Public discussion threads
- ForumTopicResponses - Thread responses and comments

Security & Access Control

Authentication Flow:

1. User credentials validated against Login table
2. HashingService verifies password hashes
3. Successful login updates UserProfile.LastLogin
4. Failed attempts increment Login.FailedLoginAttempts
5. Account lockout after threshold exceeded (LockoutEnd)

Authorization Rules:

- **Students**: Can create queries, subscribe to modules, post in forum
- **Tutors**: Can respond to queries, create FAQs, manage topics in qualified modules
- **Admin Tutors**: Additional moderation capabilities

This documentation provides the complete technical foundation for the CampusLearn™ platform, ensuring clear understanding of the data model, relationships, and architectural decisions for all development team members.