ព្រះរាជាណាចក្រកម្ពុជា

ជាតិ សាសនា ព្រះមហាក្សត្រ

Report Week 3: Course Enrollment & Classroom Scheduling System

Name Of Students	ID Of Students	Score
1. LOU Chumdararith	e20220360	……
2. CHEANG Seyha	e20221098	……
3. HENG Nguonhour	e20221565	……
4. DIN Rasy	e20220514	……
5. CHORK Ratanakdavide	e20221642	……

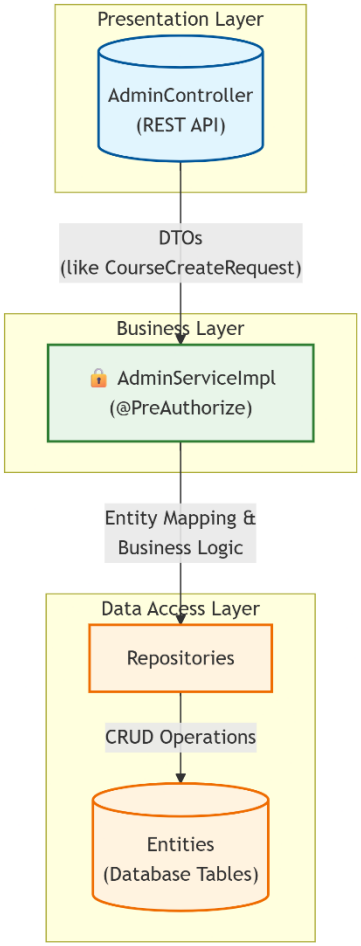
Lecturer: ROEUN Pacharoth (TP)

Academic year 2025-2026

Developer: Seyha

Role: Admin Module Developer

Data Transfer Layer Architecture



- Designed and implemented a comprehensive set of Data Transfer Objects (DTOs) specifically for administrative operations
- Established standardized data structures that ensure consistent information flow between frontend and backend systems
- Implemented validation mechanisms within DTOs to prevent invalid data propagation throughout the system
- Reduced direct dependencies between controllers and database models, enhancing system flexibility

Service Layer Implementation

- Developed the complete Admin Service layer containing all core business logic for administrative operations
- Centralized critical data processing functionality including user management, course oversight, and enrollment administration
- Implemented comprehensive validation routines to ensure data integrity across all administrative functions
- Created maintainable service patterns that separate business logic from controller responsibilities

Security Integration

- Integrated Spring Security annotations throughout the admin service layer to enforce role-based access control
- Implemented method-level security using @PreAuthorize annotations to restrict sensitive administrative functions
- Ensured proper authorization checks occur before any administrative action can be executed
- Designed secure data access patterns that prevent privilege escalation vulnerabilities

Controller & API Development

- Built fully functional controller endpoints for all administrative operations
- Implemented proper error handling to provide meaningful feedback during administrative operations

- Created RESTful API patterns that follow industry best practices for resource management
- Ensured seamless integration between frontend admin dashboard and backend services

Technical Impact

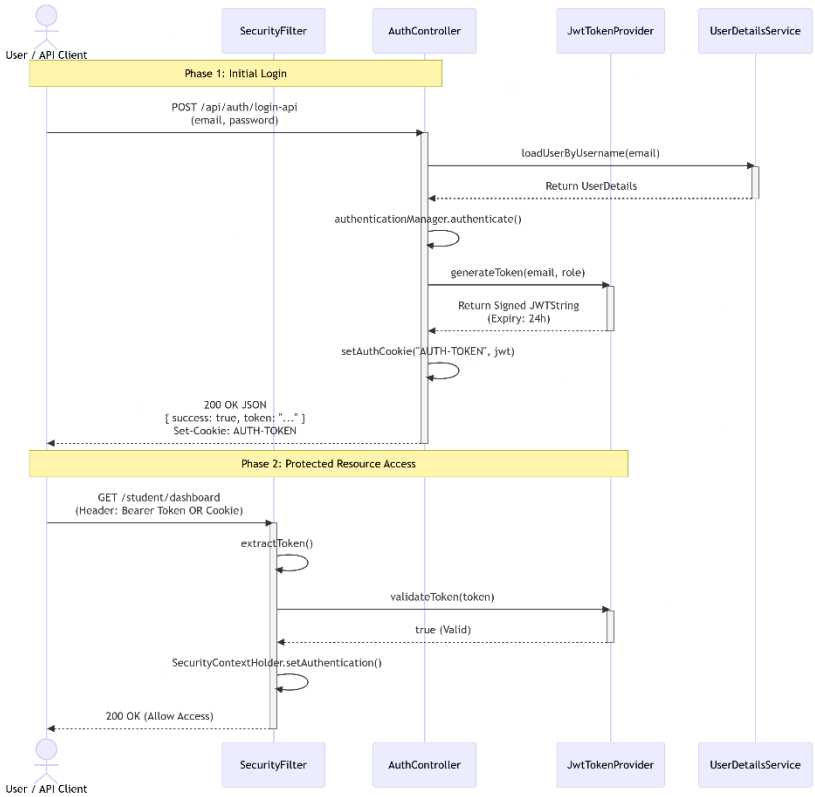
- Established a maintainable architecture pattern that can be extended for future administrative features
- Created clear separation of concerns between data access, business logic, and presentation layers
- Implemented security-by-design principles throughout the administrative subsystem
- Provided comprehensive documentation for all service interfaces and data structures

Quality Assurance

- All administrative functions include proper validation before data persistence
- Comprehensive error handling ensures system stability during edge cases
- Security annotations prevent unauthorized access attempts at the service level
- Data integrity checks prevent inconsistent states in the administrative workflow

Developer: David

Role: Authentication & Security Developer



Token-Based Authentication

- Developed a complete JSON Web Token (JWT) implementation for stateless authentication
- Designed secure token generation with industry-standard encryption practices
- Implemented automatic token expiration after 24 hours to minimize security exposure windows
- Created efficient token validation mechanisms to verify user identity on each request

Authentication System

- Successfully integrated both form-based login for web users and token-based authentication for API clients
- Maintained seamless user experience across both authentication methods
- Implemented proper session management that handles both authentication paradigms
- Created unified security configuration that manages both access patterns efficiently

User Experience Enhancements

- Designed and implemented a professional access-denied error page with clear user guidance
- Added intuitive navigation elements to help users recover from permission errors
- Implemented consistent error messaging across all authentication failure scenarios
- Created visual feedback mechanisms for successful and failed authentication attempts

Security Configuration Modernization

- Updated core security configuration to support modern authentication patterns
- Implemented comprehensive error handling for access denied scenarios
- Enabled detailed security debugging capabilities while maintaining production readiness
- Configured proper token storage mechanisms using secure browser cookies

Technical Impact

- Established industry-standard authentication practices for both web and API clients
- Created a security foundation that scales with future feature development
- Implemented proper token lifecycle management to minimize security vulnerabilities
- Designed system architecture that separates authentication concerns from business logic

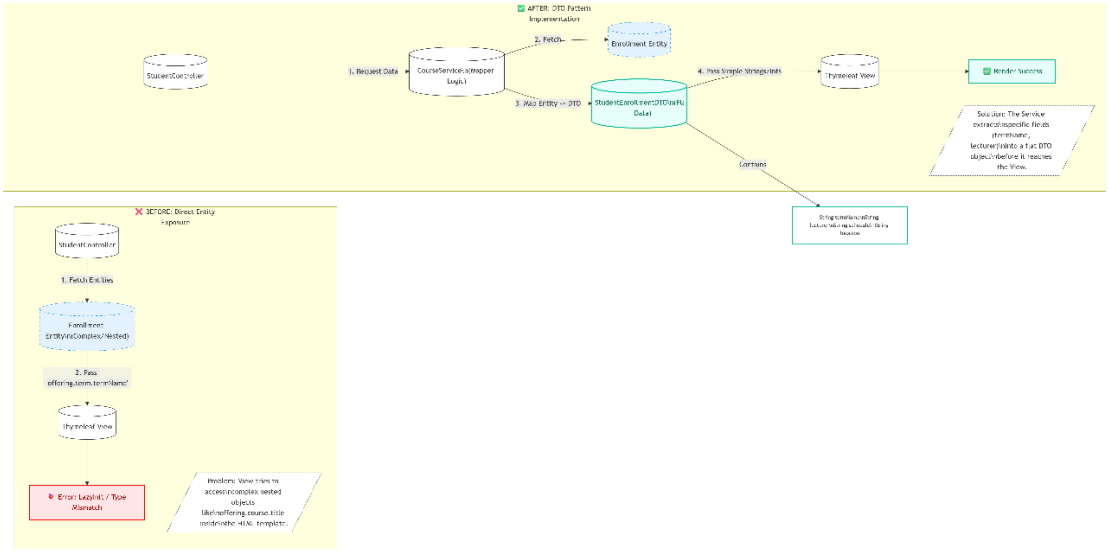
Quality Assurance

- Comprehensive token validation prevents invalid or expired tokens from gaining access
- Proper error handling provides meaningful feedback without exposing sensitive system details
- Security debugging capabilities enable rapid resolution of authentication issues

- Dual authentication support ensures all client types can securely access system resources

Developer: Rith

Role: Database & DevOps Lead



Security Module Stabilization

- Resolved critical authentication failures by correcting Spring Security parameter configuration
- Fixed "Invalid Credentials" errors by aligning form submission fields with security expectations
- Merged competing implementations of user detail services while preserving security requirements
- Ensured inactive users cannot access the system despite having valid credentials
- Reconciled conflicting security configurations while maintaining proper user redirection logic

Administrative Module Completion

- Standardized the entire Admin service interface across all management functions
- Implemented comprehensive administrative capabilities for users, courses, enrollments, and scheduling
- Integrated proper role-based access control throughout all administrative endpoints
- Fixed critical view path errors that were preventing administrative pages from rendering
- Resolved template processing errors that were breaking data display in administrative interfaces

Student Module Refactoring

- Redesigned the Student controller architecture to eliminate type conversion errors
- Created specialized Data Transfer Objects (DTOs) to simplify complex data relationships
- Resolved persistent template errors by flattening nested object structures
- Updated service layer logic to support new data presentation requirements
- Fixed numeric conversion errors that were breaking enrollment calculations

User Experience Standardization

- Developed a universal sidebar component for student navigation across all pages
- Implemented consistent naming conventions throughout all templates and controllers
- Added helper methods to user entities to simplify display logic system-wide
- Created standardized data formatting for enrollment and schedule information
- Ensured consistent error handling and user feedback across all student-facing features

Technical Impact

- Successfully merged two significantly divergent development branches into a single cohesive codebase
- Eliminated all compilation errors that were blocking development progress
- Resolved runtime authentication failures that were preventing user access
- Created a stable foundation that supports all planned system features
- Established patterns for future development that prioritize maintainability and error prevention

Quality Assurance

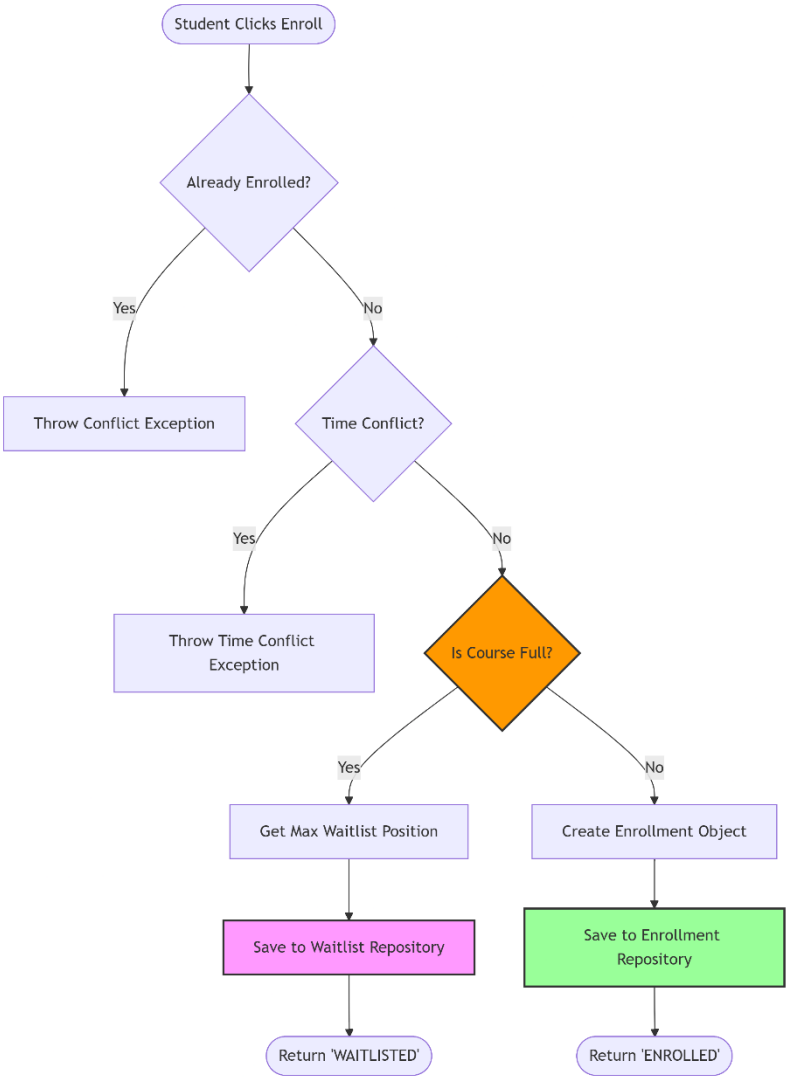
- All security configurations properly validate user credentials and roles
- Administrative functions enforce strict access controls at multiple system levels
- Student interfaces display accurate enrollment data without template errors
- System handles edge cases gracefully with appropriate user feedback
- Codebase maintains consistent architecture patterns across all modules

System Integration

- Coordinated efforts across five developers to produce a unified deliverable
- Resolved complex merge conflicts while preserving critical functionality from all branches
- Verified complete system functionality across all user roles and scenarios
- Documented architectural decisions to facilitate future maintenance and extension
- Ensured the final system meets all specified requirements for the two-week development cycle

Developer: Rasy

Role: Main Entity CRUD Developer (Work on Student features)



Enrollment Service Layer

- Developed comprehensive **EnrollmentService** interface defining all **enrollment** operations
- Implemented core business logic in **EnrollmentServiceImpl** including capacity checks and waitlist processing
- Created automatic waitlist promotion when courses become available
- Prevented invalid **enrollment** operations through systematic validation

Data Access & Transfer

- Built **WaitlistRepository** with proper ordering logic for student priority management
- Designed **EnrollmentRequestDTO** for secure data transfer between system layers
- Enhanced entity models with proper academic relationships and constraints

User Interface Integration

- Developed my-courses.html UI component allowing students to view and manage their **enrollments**
- Connected frontend actions with backend services for seamless drop course functionality
- Implemented user feedback mechanisms for **enrollment** status changes

Technical Impact

- Established a centralized **enrollment** management system that enforces all academic rules
- Created automatic waitlist handling that improves student experience during high-demand periods
- Implemented proper separation of concerns between service, repository, and controller layers
- Provided students with intuitive course management capabilities through clean UI integration

Quality Assurance

- All **enrollment** operations include proper validation before execution
- Waitlist promotion occurs automatically when seats become available
- Course dropping follows academic policies with appropriate status tracking
- System prevents duplicate **enrollments** and maintains data integrity across all operations