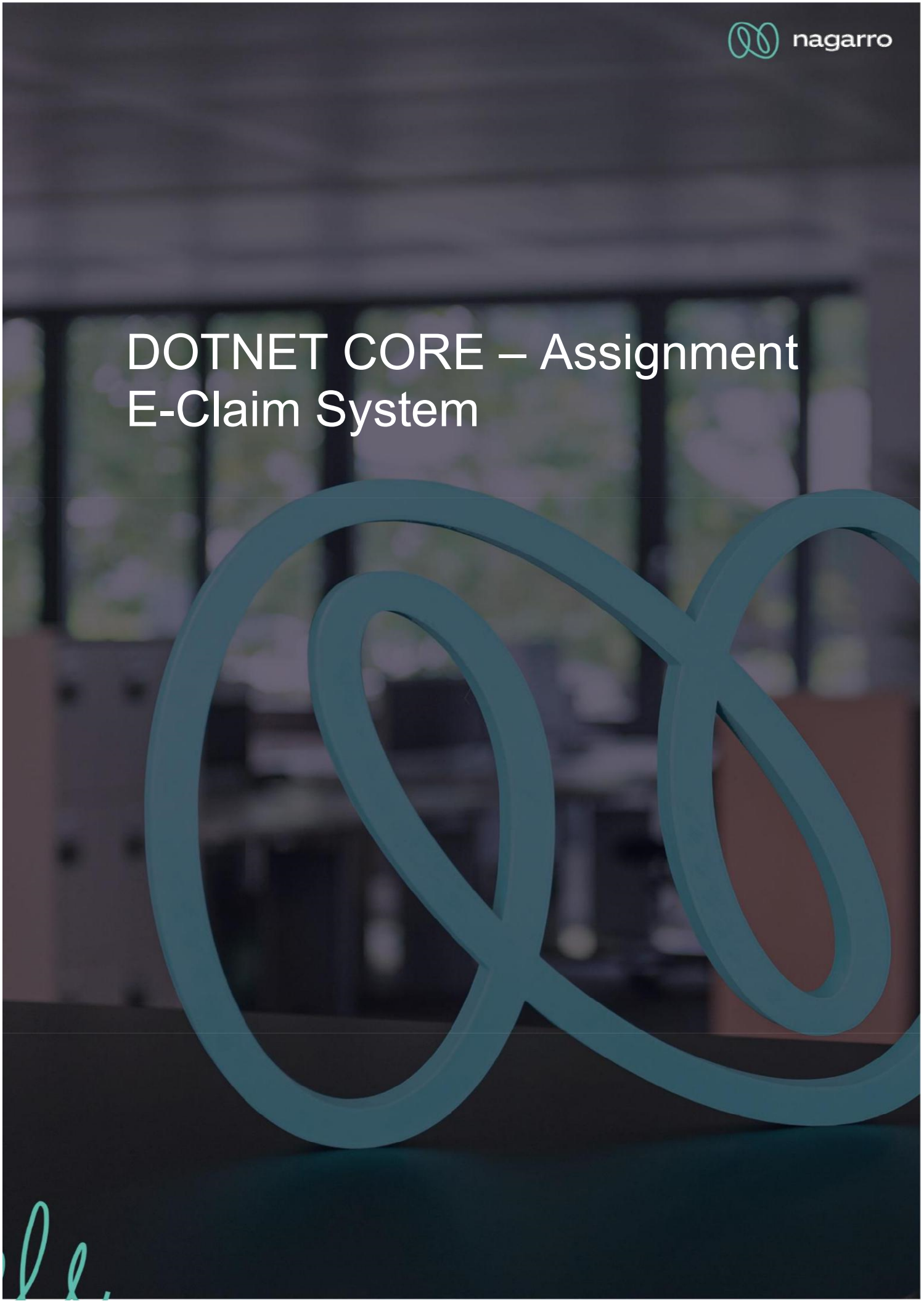
nagarro
DOTNET CORE – Assignment
E-Claim System

nagarro

.NET Core

Contents

Assignment

Develop a web application for an E-Claim system using ASP.NET Core. The system should handle the submission, processing, tracking and management of insurance claims. The system caters to various user roles such as claimants, adjusters, and administrators. The application should demonstrate advanced skills in software architecture, security, performance, and modern development practices.

Requirement

Business Requirements

Build an e-claim application. Key features should be as follows.

1. **User Management**
   - Registration and Login (with email verification)
   - Role-based authorization (Admin, Claimant, Adjuster, Approver)
   - Profile management

2. **Claim Management**
   Design a user-friendly interface for claim submission with various claim types.
   Implement a workflow engine to automate claim processing with configurable approval steps based on claim type or severity
   - Claim submission (with file uploads for supporting documents)
   - Claim status tracking
   - Claim history and details
   - Workflow for claim review, adjustment, and approval.

3. **Notification System**
   Develop notification functionalities for users regarding claim updates, approvals, and required actions.
   - Email and SMS notifications for claim status updates
   - Notifications for required actions (e.g., additional documentation)

4. **Reporting**
   - Generate reports on claim statuses, types, and other relevant metrics
   - Export reports to PDF, Excel, etc.

5. **Payment Integration (optional)**
   - Integrate with payment gateways for claim payouts

6. **Security**
   - Authentication: Implement JWT-based authentication.
   - Authorization: Implement role-based authorization.
   - Input Validation: Validate and sanitize user inputs to prevent SQL Injection, XSS, and other attacks.

- Audit Logs: Track and log all critical actions for auditing purposes.

7. **Search and Filtering**
    - Implement search functionality for claims
    - Filter claims by status, type, date range, etc.

Technical requirements

1. You can use framework/libraries of your choice for user interface design.
2. Backend of the application should be designed as a .NET Web API.
3. SQL Server should be used for storing all the data.
4. App should be portable. Database migrations/seed data initialization should be implemented. Other than changing the host name for SQL server no additional changes should be required to run the application on a different machine.
5. Entity framework should be used for all database operations.
6. Dependency injection should be used.
7. Global Exception Handling should be implemented. The exceptions should be logged to database.
8. Logging of additional information and warnings should be done in database.
9. Relevant validations should be implemented and mentioned in the documentation during submission.
10. Write unit tests for critical components using xUnit.

Deliverables

You need to share the code without Dlls on SharePoint and share the link accordingly. Make sure it should be public.

- A well-structured and documented codebase.
- Unit and integration tests covering core functionalities.
- Deployment configuration for a chosen cloud platform (optional).
- User guides for different roles within the system.

Bonus Question

Containerize the application using Docker. Use Kubernetes for container orchestration.