ST10022295

Group 3

Documentation

## Introduction

The **Claiming System** is a software application that facilitates the management of claims. It aims to automate the process of submitting, reviewing, and resolving claims, providing a seamless experience for both users and administrators.

## System Architecture

The system is built using:

* **Database**: SQL-based database to manage and store claims data.
* **Backend**: Likely powered by .NET, indicated by the solution file (Claiming\_System.sln).
* **Frontend**: A user interface for interacting with the system (specific framework not specified).

## Database Design

### SQL File: claimingsystem.sql

This file contains the schema and initial data setup for the Claiming System database. It defines:

1. **Tables**:
   * Users: Stores user information and roles.
   * Claims: Tracks the details and status of claims.
   * Statuses: Lists possible claim statuses (e.g., Pending, Approved, Rejected).
2. **Relationships**:
   * Claims are linked to users who submit them.
   * Claims reference statuses to indicate progress.
3. **Indexes**:
   * For faster queries on key columns like user ID or claim ID.
4. **Constraints**:
   * Foreign key constraints ensure data integrity between users and claims.

### Key SQL Features:

* **Data Types**: Defined to optimize storage and ensure validity.
* **Triggers/Stored Procedures** (if applicable): Automate specific actions in the database.
* **Initial Data**: May include sample users, statuses, and claims.

## Setup and Installation

### Prerequisites

* Database Management System (e.g., MySQL, SQL Server).
* .NET development environment (e.g., Visual Studio).

### Steps

1. **Database Setup**:
   * Import the claimingsystem.sql file into your DBMS.
   * Configure the connection string in the application.
2. **Application Setup**:
   * Open the Claiming\_System.sln file in Visual Studio.
   * Restore dependencies using NuGet.
   * Build and run the solution.

## Functionality Overview

1. **Claim Submission**:
   * Users can file new claims by providing required details.
2. **Claim Status Management**:
   * Admins can update the status of claims (e.g., Approved, Rejected).
3. **Notifications**:
   * Automatic updates sent to users about claim status changes.
4. **Audit Trails**:
   * Record all interactions for transparency and compliance.

## Roles and Permissions

### Users

* **Submit Claims**: Add new claims and view existing ones.
* **Track Progress**: See status updates.

### Administrators

* **Manage Claims**: Review, approve, or reject claims.
* **Configure Settings**: Modify user roles and claim parameters.

## Technical Details

### Database Schema Highlights

* Primary keys: Ensure unique identifiers for users and claims.
* Foreign keys: Maintain relationships between tables.
* Indexes: Improve performance on frequently queried columns.