**Contract Monthly Claim System (CMCS) — Part 1 Report (Prototype)**

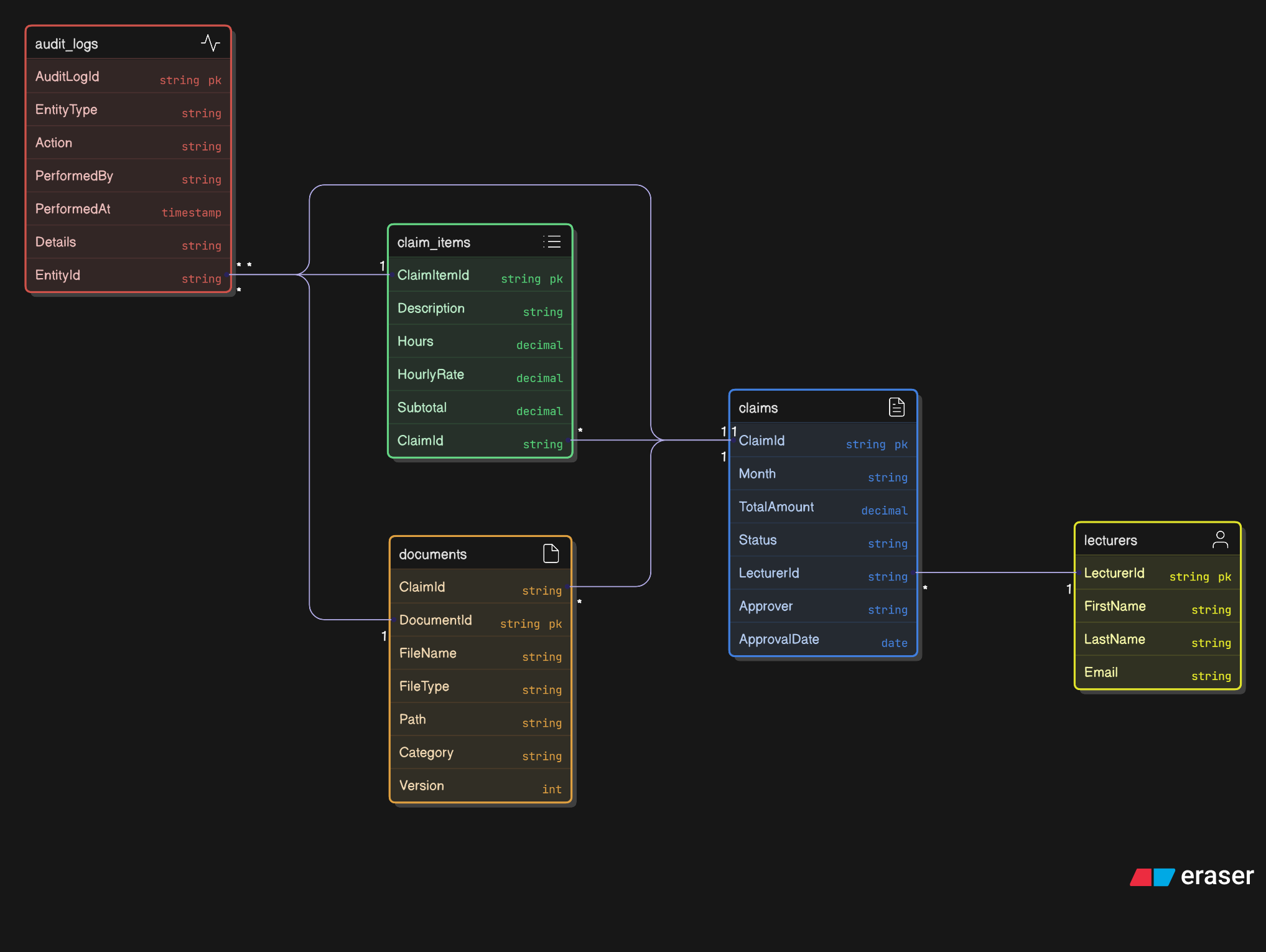
**Overview**

**This document describes the planned prototype for the Contract Monthly Claim System (CMCS). The goal is a non-functional front-end prototype (ASP.NET Core MVC) that demonstrates how lecturers submit monthly claims, upload supporting documents, and how Programme Coordinators and Academic Managers view and approve claims.**

**Design choices & assumptions**

* **Use ASP.NET Core MVC for a web-based prototype (clear separation of concerns: Models, Views, Controllers).**
* **Database model will represent Lecturers, Claims, ClaimItems (hours × rate), Documents, and Users with roles (Lecturer, ProgrammeCoordinator, AcademicManager).**
* **Prototype is front-end only (UI mockups and static placeholders) — no server-side logic or database operations implemented in Part 1.**
* **Assume each claim can contain multiple claim items and multiple supporting documents.**

**Database UML (class diagram)**

**GUI layout (pages)**

* **Home / Dashboard — quick links: Submit Claim, My Claims, Claim Status.**
* **Submit Claim (form) — month selector, dynamic list of claim items (description, hours, rate), add/remove item buttons, file upload control (placeholder), Submit button.**
* **My Claims (table) — list claims with ID, month, total, status, and View Details button.**
* **Approval Panel (Admin) — filterable list for Programme Coordinators and Academic Managers with Approve / Reject buttons (prototype shows static buttons).**
* **All pages use a shared layout with a top navbar and record of current user role (mocked).**

**Project plan (high-level, 4 weeks)**

1. **Week 1: Environment setup (Visual Studio, .NET SDK, Git), create project scaffold, initial commit.**
2. **Week 2: Build UI skeleton: layout, nav, pages (Submit, My Claims, Admin Panel), second & third commits.**
3. **Week 3: Add mock data, placeholder upload UI, prepare UML diagram and documentation, fourth commit.**
4. **Week 4: Polish visuals, finalize report (Word), final commit & push to GitHub.**

**Version control (5 commits - recommended messages)**

1. **init: scaffold project (ASP.NET Core MVC)**
2. **ui: add shared layout + navbar**
3. **feature: add ClaimsController and views (skeleton)**
4. **docs: add UML diagram and project report**
5. **polish: finalize prototype UI and README**

**Screenshots of working project:**

