CS4.301 Data and Applications: Phase 2 Submission

Team Number: 13

October 30, 2025

Contents

1 Mini-world Overview

Our mini-world models the State Alchemist Research and Mission Management System for the Amestrian military. It tracks State Alchemists, their missions, and their coordination with associated military units. The system also manages research labs, including experiments (transmutation attempts), ingredients, registered alchemical devices, and case files on recovered artifacts and captured homunculi.

Eashaan Thakur, Hemang Joshi, Shardul Joshi

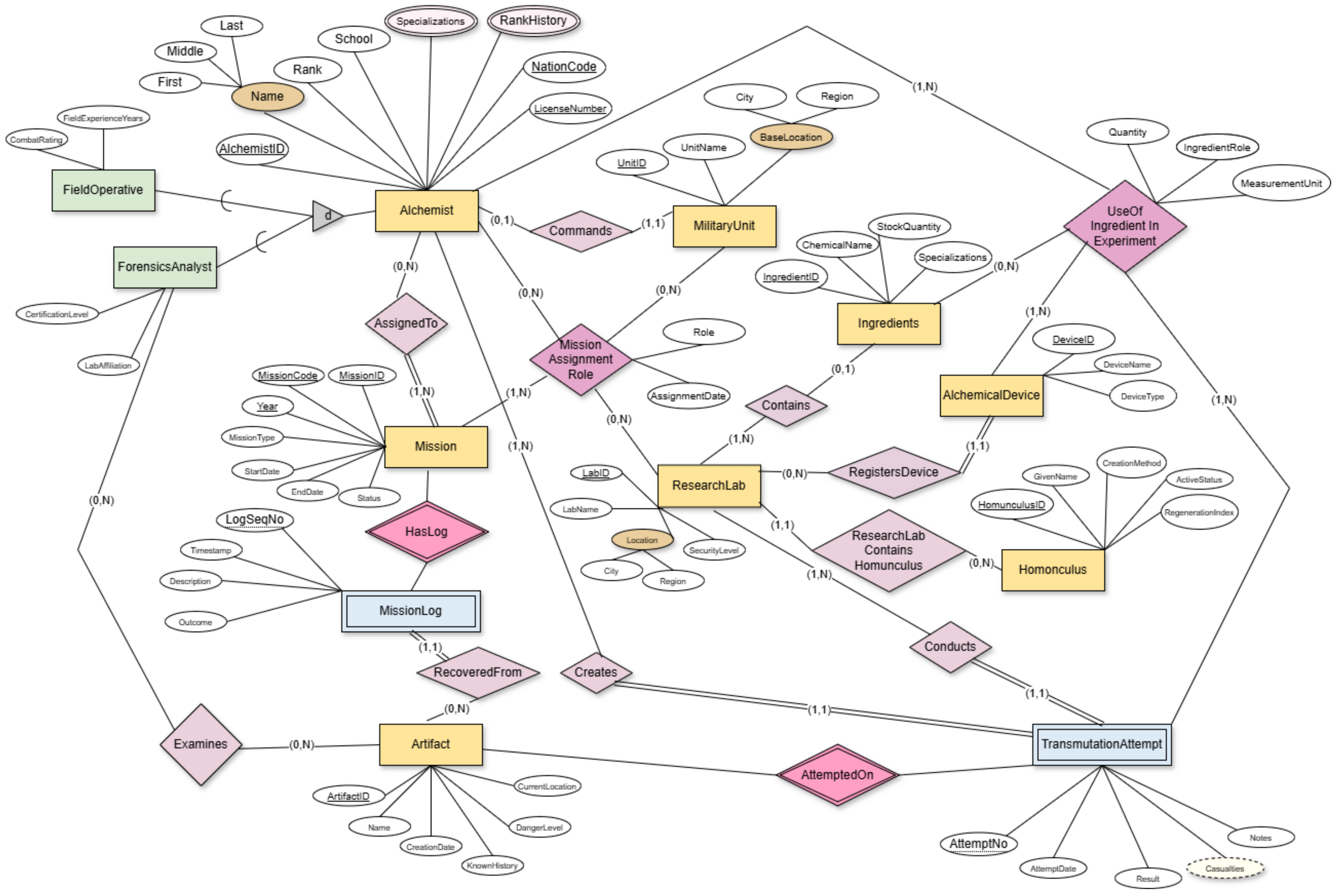

Figure 1: Entity-Relationship Diagram for the State Alchemist Management System

2 Notations & Conventions Used

- **Notation Style:** Chen notation with (min, max) cardinality.
- **Strong Entities:** Yellow boxes (e.g., `Alchemist`).
- **Subclasses:** Green boxes (e.g., `FieldOperative`).
- **Weak Entities:** Blue boxes (e.g., `MissionLog`).
- **Relationships:** Pink diamonds.
- **Attributes:** White ovals.
- **Primary Key (PK):** Underlined attribute (e.g., `AlchemistID`).
- **Partial Key:** Dashed-underline attribute (e.g., `LogSeqNo`).
- **Multi-valued Attribute:** Double oval (e.g., `RankHistory`).
- **Derived Attribute:** Dashed oval (e.g., `Casualties`).
- **Participation:** (min, max) notation. `min=1` indicates total participation; `min=0` indicates partial.

3 Changes and Refinements from Phase 1

- **Multi-valued Attributes:** Represented `RankHistory` and `Specializations` as multi-valued attributes of `Alchemist` for conceptual clarity. In Phase 1, these were modeled as separate tables, which will be derived from these multi-valued attributes during normalization.
- **N-ary Relationships:** The multi-entity relationships were expanded to 4-ary as required:
  - **UseOfIngredientInExperiment:** A 4-ary relationship (`TransmutationAttempt` × `Ingredient` × `Alchemist` × `AlchemicalDevice`) with attributes for `Quantity`, `IngredientRole`, and `MeasurementUnit`. `ResearchLab` was removed as it's redundant (both `Device` and `Attempt` link to a lab).
  - **MissionAssignmentRole:** A 4-ary relationship (`Mission` × `Alchemist` × `MilitaryUnit` × `ResearchLab`) with attributes for `Role` and `AssignmentDate`. `ResearchLab` was added to track lab support for the mission.
- **Derived Attributes:** Refined derived attributes to include `TransmutationAttempt.Casualties`, which is derived from outcome records.
- **Entity Additions:** Fully incorporated the `MilitaryUnit` and `AlchemicalDevice` entities and their relationships as specified in the Phase 1 requirements.

4 Design Quality Analysis

4.1 Redundancy Analysis

- **Potential Redundancy 1 – Rank Information:** The attribute `Rank` is stored directly in `Alchemist`, but a detailed `RankHistory` is also kept. **Effect:** If both are updated inconsistently, an alchemist could appear with two different current ranks. **Resolution:** Keep `RankHistory` as the authoritative table; `Rank` in `Alchemist` should be derived or updated via a trigger.

- **Potential Redundancy 2 – Location Data:** Both ResearchLab and MilitaryUnit store City and Region. **Effect:** Repeated storage may cause inconsistent spellings (e.g., "Central City" vs "Central Town"). **Resolution:** Optionally normalize locations into a Location entity shared by both.
- **Potential Redundancy 3 – Lab Reference:** Homunculus, AlchemicalDevice, and Ingredient each include a reference to LabID (via relationships). **Effect:** If a lab closes or renames, multiple tables require synchronized updates. (This is a necessary and controlled redundancy for referential integrity).

4.2 Update Anomaly

Example: An application-level integrity constraint states: "Artifacts with DangerLevel = 'Critical' must be stored in a ResearchLab with SecurityLevel = 'High' or 'Critical' ". **Anomaly:** If a lab's SecurityLevel is downgraded from 'High' to 'Medium', a simple UPDATE on ResearchLab would succeed, but this could leave a 'Critical' Artifact in a now-non-compliant lab. This violates policy and yields inconsistent analytics.

4.3 Insertion Anomaly

- **Scenario 1:** You cannot insert a TransmutationAttempt record until the corresponding Artifact exists, because TransmutationAttempt is a weak entity depending on Artifact. Hence, a researcher cannot log a new attempt while an artifact record is still pending creation.
- **Scenario 2:** A MissionAssignmentRole requires an existing Mission, Alchemist, MilitaryUnit, and ResearchLab. If any of those are not yet entered, the assignment tuple cannot be created—blocking partial scheduling.

4.4 Deletion Anomaly

- **Example 1:** Deleting an Artifact would cascade-delete all dependent TransmutationAttempt records. Important experimental results could vanish permanently.
- **Example 2:** If a Mission is deleted, all MissionLog entries tied to it may also disappear (due to ON DELETE CASCADE). This would erase the operational history even though logs are valuable for audit.

Anomaly Summary Table

Anomaly Type	**Example Entity / Relation**	**Risk**	**Mitigation**
Redundancy	`Alchemist` ↔ `RankHistory`	Inconsistent ranks	Use `RankHistory` as source of truth; derive `Rank`.
Redundancy	`ResearchLab` / `MilitaryUnit`	Inconsistent location data	Normalize into a shared `Location` entity.
Update	`ResearchLab` ↔ Artifact Policy	Out-of-sync security levels	Add constraint or trigger to check `Artifact` locations before lab update.
Insertion	`TransmutationAttempt` ↔ `Artifact`	Cannot insert attempt without artifact	Use a default ”pending” artifact template record.
Deletion	`Mission` ↔ `MissionLog`	Loss of historical records	Use ”Archive” flags or triggers instead of `DELETE`.

Table 1: Summary of Design Quality Analysis