**Insurance Management System**

**Milestone 02**



**Spring 2025**

Submitted by:

**Khalil hussain**

**Waqas atta**

**muqtadir**

Class Section: **B**

“On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work.”

Submitted to:

Eng. Sumayyea Salahuddin

Month Day, Year (26 05, 2025)

Department of Computer Systems Engineering

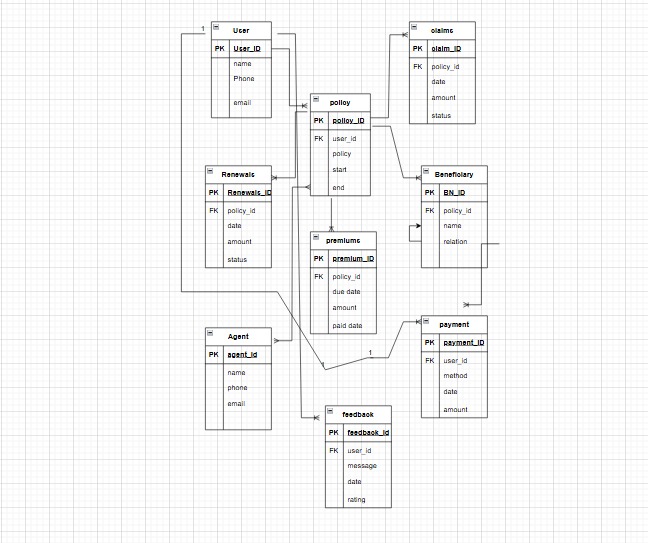
University of Engineering and Technology, Peshawar

**Milestone 02 – Conceptual Schema and Normalization**

**1. Relationship Table**

| **Relationship** | **Type** | **Rule Explanation** |
| --- | --- | --- |
| Users — Policies | 1 : M | One user (Users.user\_id) can register for multiple policies |
| Policies — Claims | 1 : M | One policy (Policy.policy\_id) can have multiple claims |
| Policies — Premiums | 1 : M | One policy can have many premium payments |
| Policies — Renewals | 1 : M | A policy can be renewed many times |
| Policies — Beneficiaries | 1 : M | One policy may have multiple beneficiaries |
| Policies — Agents | M : N | One policy can be handled by multiple agents; one agent can handle many policies |
| Users — Payments | 1 : M | A user can make many general payments |
| Users — Feedback | 1 : M | One user can submit multiple feedback entries |
| Claims — Policies | M : 1 | Each claim belongs to a registered policy |

**2. Relational Schema**



**USER: (1:M )** UserID name email phone

**POLICY: (1:M)** PolicyID UserID policy\_name start\_date end\_date

**CLAIM: (M:1)** ClaimID PolicyID date claim\_amount status

**PREMIUM: (1:M)** PremiumID PolicyID due\_date amount paid\_date

**RENEWAL: (1:M)** RenewalID PolicyID renewal\_date amount status

**BENEFICIARY: (1:M)** BeneficiaryID PolicyID name relation

**AGENT: (M:N)** AgentID name email phone

**PAYMENT: (1:M)** PaymentID UserID method date amount

**FEEDBACK: (1:M)** FeedbackID UserID date message rating

**Normalization Summary**

1NF: Each attribute is atomic (no multi-valued fields like multiple phone numbers or relations).

2NF: All non-key fields depend on the entire primary key.

3NF: No transitive dependencies exist.