Smarter Reconciliation and Anomaly Detection using GenAI

- Team Murphy

AGENDA

- Problem Statement
- Solution
- Design Consideration
- Solution Approaches
- Our choice
- Technology Stack
- UX Envisioned

Problem Statement

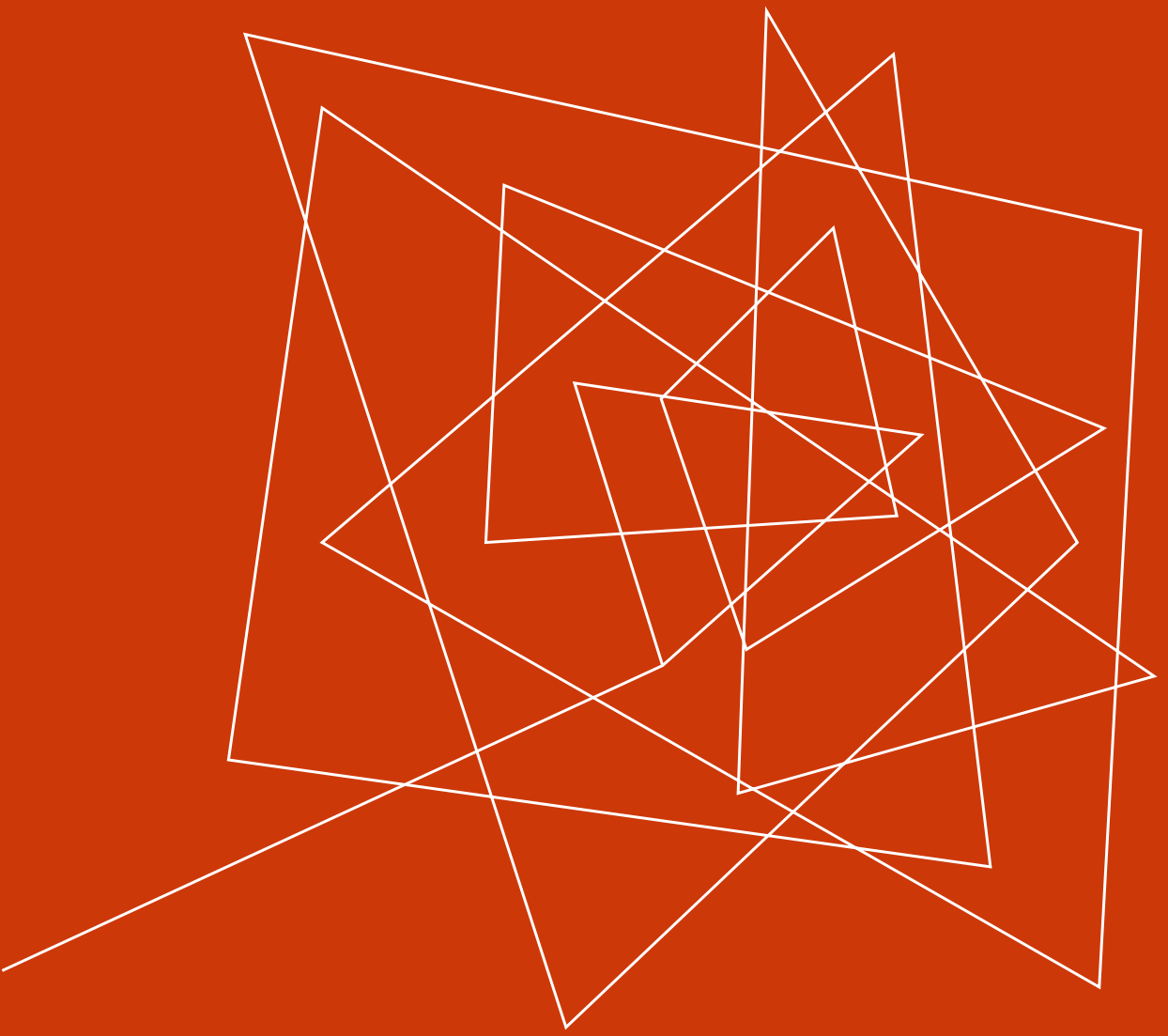
- Reconcilers spend huge time and effort in reconciling transaction discrepancies
- Most of the effort goes into identifying anomalies based on historical patterns, recording them and resolving them
- Every anomaly may need a different resolution

Solution

- With the advent of GenAI, its Agentic capabilities can be employed to
  - Analyze the transactions
  - Detect anomalies
  - Decide on the resolution
  - Execute the resolution

Design Considerations

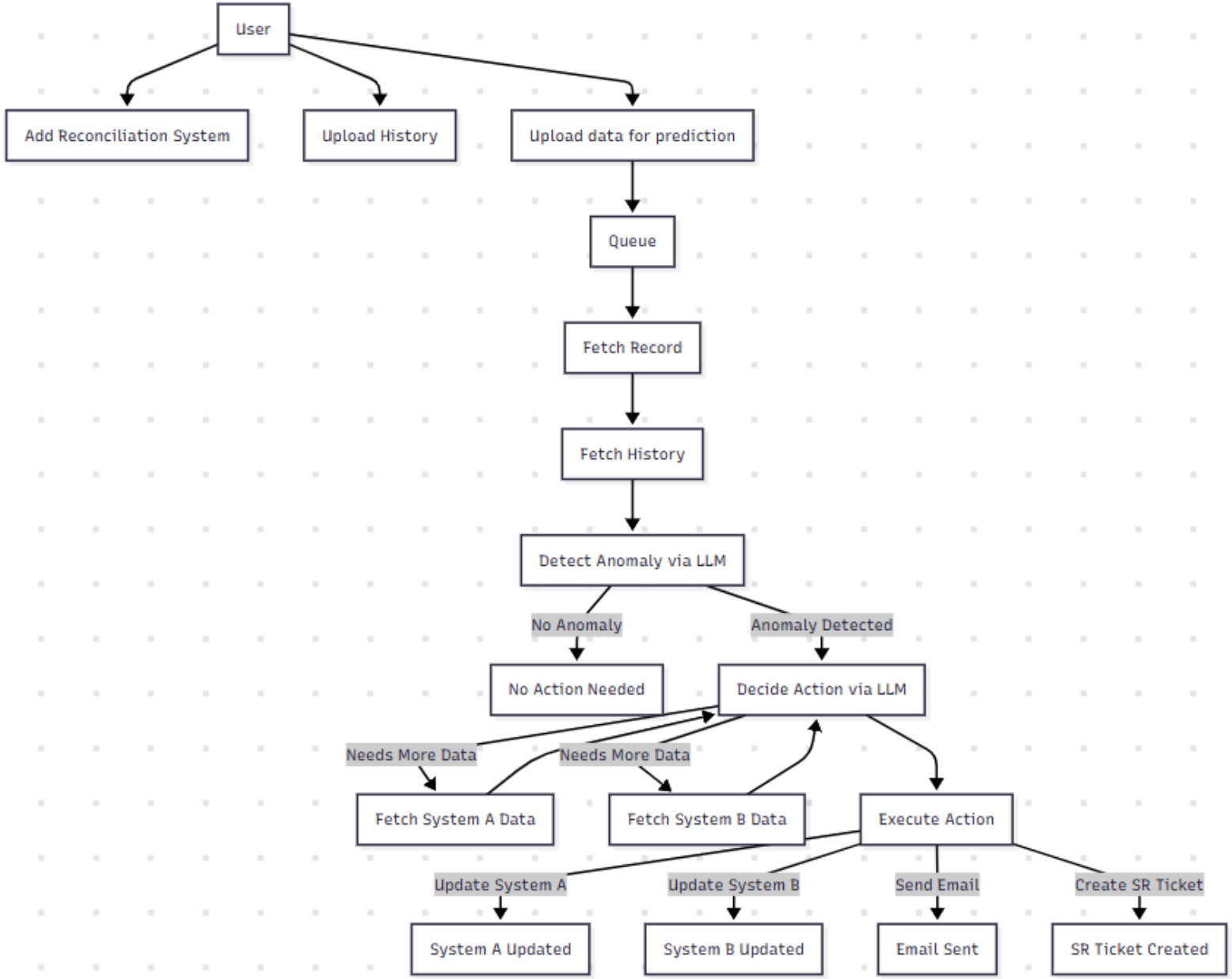
- Solution to be highly Configurable to plug in various reconciliation systems
- AI Agent should be able to distinguish and provide anomalies appropriately for each system
- AI Agent should be able to learn with Human in the loop
- AI Agent should be able to suggest or take actions autonomously to correct the anomalies



Solution Approaches

Approach 1 – LLM for anomaly detection

- Consider LLM for anomaly detection and action decisioning
- Agentic AI using LangGraph for workflow
- Tool calling from LLM when more data is needed
- Action decision based on response from LLM
- When no action can be taken record will be sent for human intervention
- Reconciler updates the record with the action
- RAG is adopted for sending historical data and reconciler feedback



Approach 1

Pros

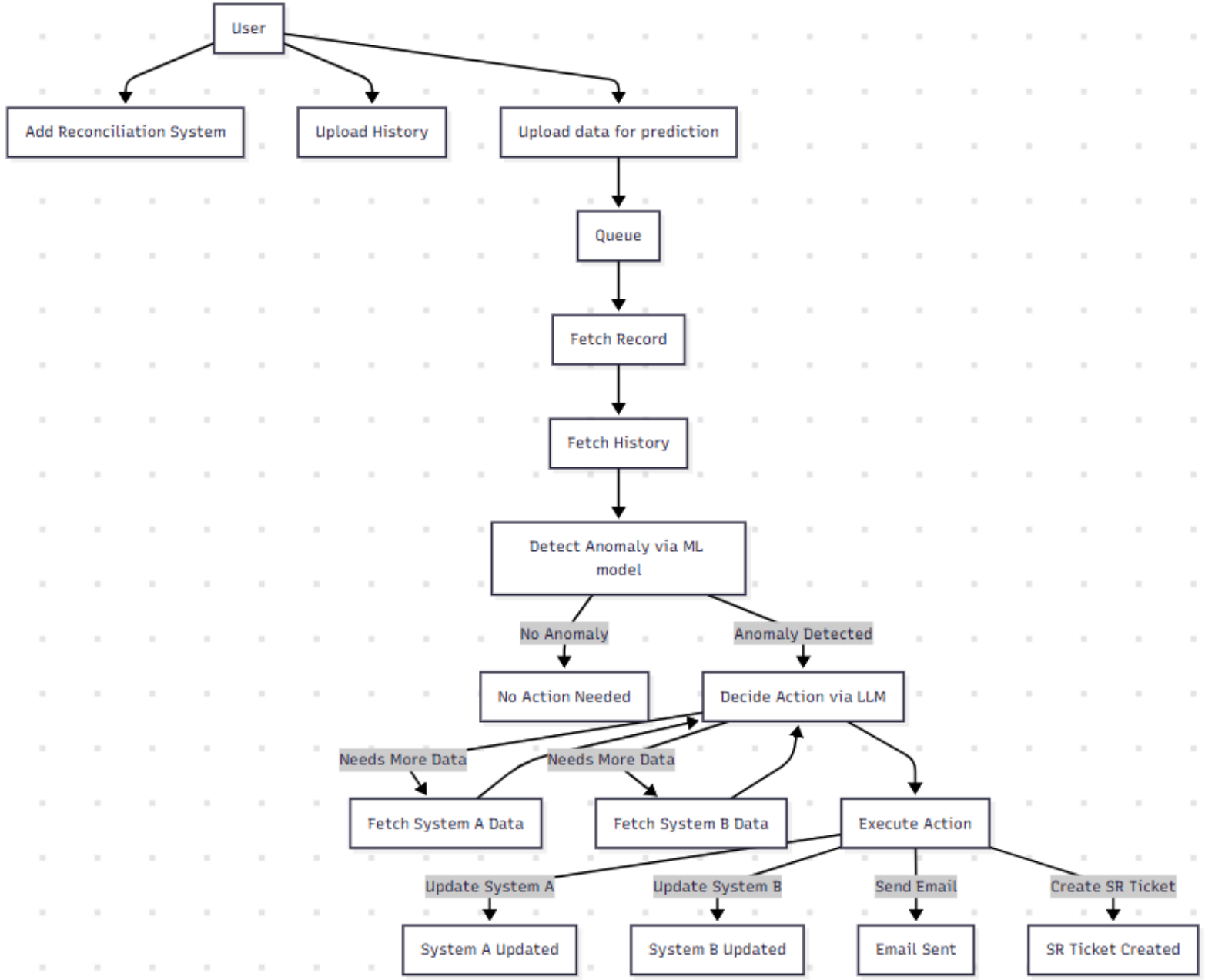
- LLM is pretrained for complex scenarios that can be utilized for anomaly detection
- LLM can be finetuned for more accurate detection
- With RAG, LLM can dynamically work with real-time data
- Tool call capabilities can be leveraged for additional data analysis

Cons

- May not be as accurate as a purpose trained ML model for anomaly detection

Approach 2 – ML Model for anomaly detection

- Consider ML model for anomaly detection as we are dealing with structured data
- Use algorithms best suited for anomaly detection with classified dataset
  - Random Forest
  - XG Boost
- Use the following for unclassified dataset
  - Isolation Forest
  - Autoencoders
- Agentic AI using LangGraph for workflow
- Tool calling from LLM when more data is needed
- Action decision based on response from LLM
- When no action can be taken record will be sent for human intervention
- Reconciler updates the record with the action
- RAG is adopted for sending historical data and reconciler feedback



Approach 2

Pros

- Trained ML model for a given data structure that can detect anomalies with high accuracy

Cons

- May need one model for each reconciliation system
- Needs retraining when data structure changes
- Needs a good dataset for the training
- Additional data cannot be used for the analysis

Our choice: Approach 1 Considerations

- Highly configurable with addition of new reconciliation systems with varying data structures
- Agentic AI with LangGraph provides autonomous agents and flexibility to switch models
- Leveraging RAG provides capabilities to send additional data for reasoning
- Tool calling capabilities leveraged for action decisioning based on external data

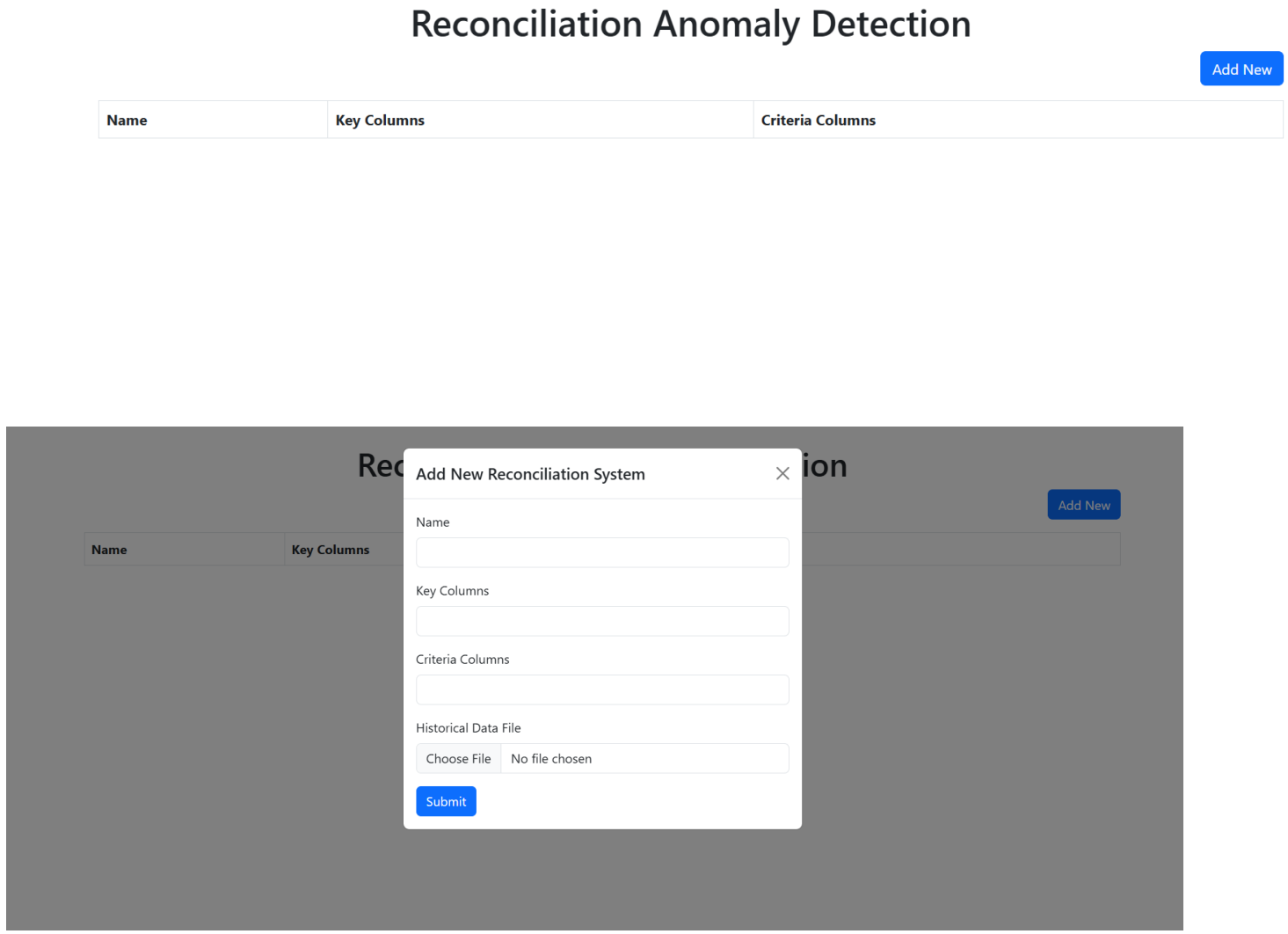
Database Considerations

- Reconciliation system metadata about key columns and criteria columns are stored in database
- Historical data and data for prediction is also persisted to database and a mapping is maintained to lookup the right data

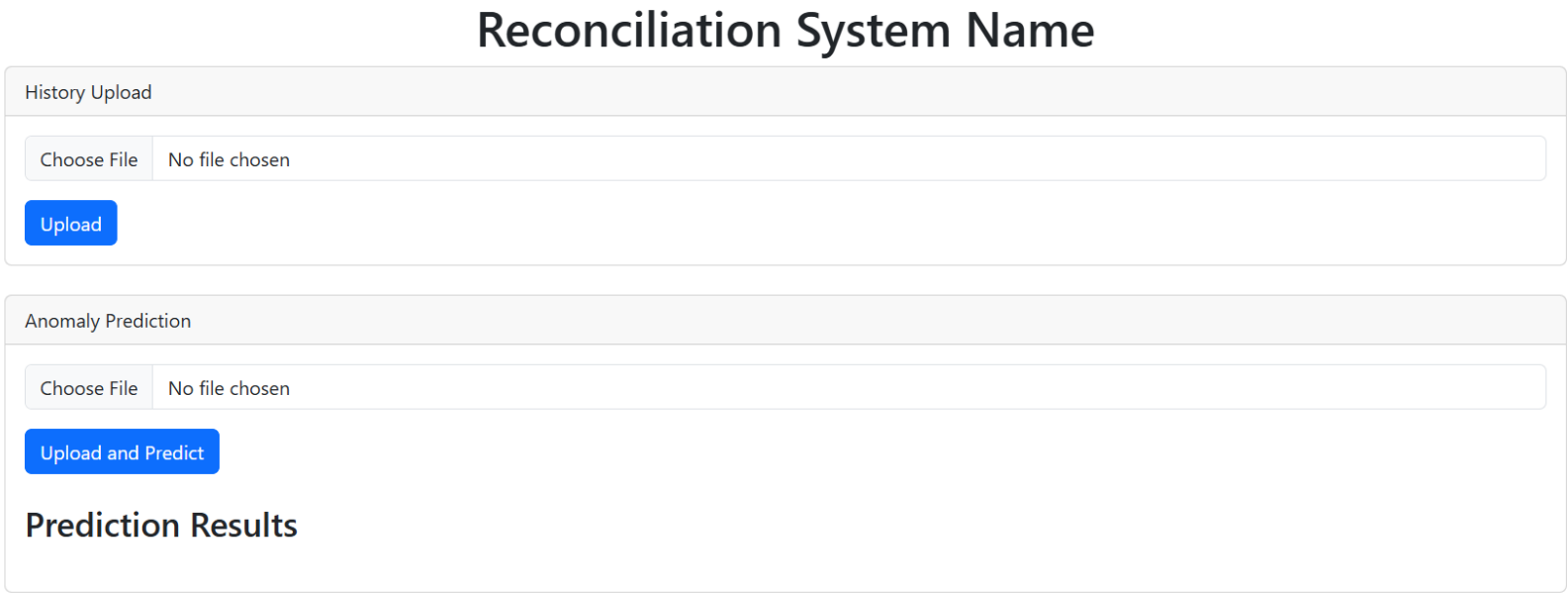
Technology Stack

- Python
- Flask
- LangGraph
- OpenRouter
- SQLiteDB
- Bootstrap, jQuery

UX Envisioned



UX Envisioned



THANK YOU

Team Murphy

Muralidharan Balanandhan

Rajagopalan Krishnamoorthy

MadhanKumar Balakrishnan

Satyendran A

Prakash Duraisamy