BorBann
A Real Estate Information Platform
(End-of-Semester Progress Report)

by

Pattadon Loyprasert 6510545608
Sirin Phungkun 6510545730

Submitted as
Software Requirements Specification
for
01219395 Innovative Software Group Project Preparation

Faculty of Engineering
Kasetsart University
Bangkok, Thailand

March 2025

BorBann
A Real Estate Information Platform
(Project Proposal)

by

Pattadon Loyprasert 6510545608
Sirin Phungkun 6510545730

This Project Submitted in Partial Fulfillment of the Requirements
for Bachelor Degree of Engineering
(Software Engineering)

Department of Computer Engineering, Faculty of Engineering
KASETSART UNIVERSITY
Academic Year 2025

Approved by:

Advisor ..... Date.../.../...
(Assoc.Prof.Dr. Kitsana Waiyamai)

1 Executive Summary

This development period, we focused on implementing the **Customizable Automated Data Integration Pipeline**. Approximately 90% of this feature is completed, including:

- Data aggregation and pipeline management systems
- An AI data mapping module for unified dataset creation

Remaining work involves completing the frontend-backend connection and refining the user interface for this module.

2 Detailed Development Description

2.1 Overview

The development scope focused on creating a data integration pipeline that automates the collection, processing, and unification of diverse data sources. Our objective was to implement an system capable of handling heterogeneous datasets from pipeline while minimizing manual configuration requirements. We adopted an iterative development approach, prioritizing core pipeline functionality before addressing user interface components.

2.2 Key Classes and Modules Developed

- **pipeline/****: This core module contains all components related to the Automated Data Integration Pipeline functionality
- **pipeline/ingestion/ingestors/mapping/mapping_ingest.py**: Contains the AI mapping function that consolidates diverse data sources into a unified dataset structure
- **frontend/****: Contains all user interface components, with specific emphasis on the pipeline configuration and monitoring interfaces

2.3 Implementation Details and Challenges

The data integration pipeline was implemented using a fastAPI combine with model fine-tuning on Vertex AI. Here are lists of things we've developed

- Data adaptor which consume data from API, File, and Scraping sources
- Data ingestor to help aggregate the data from each adaptor including AI ingestor module (AI data mapping module)

- Scheduling service to help manage pipeline
- Data store to save the pipeline configuration
- UI interface for pipeline service

We faced several challenges. First, the scraping adapter doesn't work on Windows right now. Second, the scheduling service was poorly designed, which caused many problems and now needs to be redesigned. We also had trouble building stable data connectors for different data sources. It was hard to create a smart mapping system that could understand how different data formats are related. Changing the data while keeping it correct was another challenge. Finally, making sure the system can handle a large amount of data was also difficult.

2.4 Screenshots

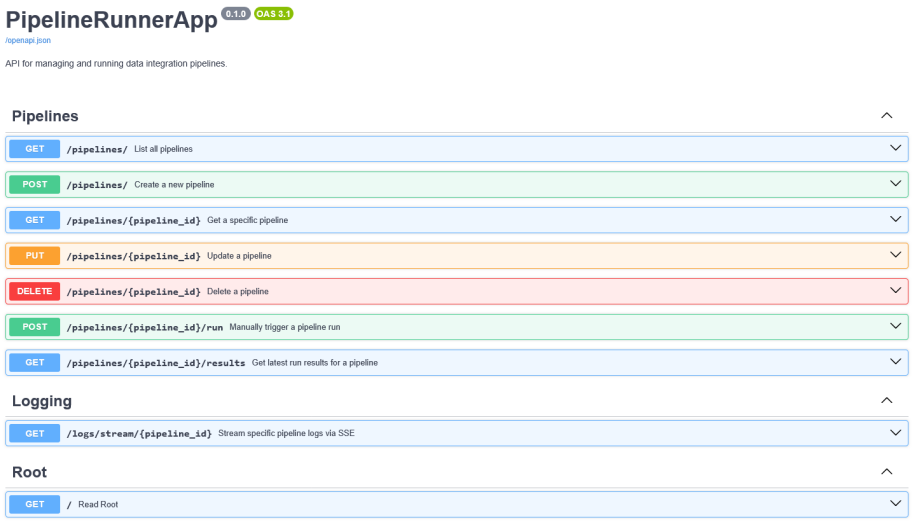

Figure 1: API documentation interface for pipeline management endpoints, providing comprehensive testing capabilities for backend functionality

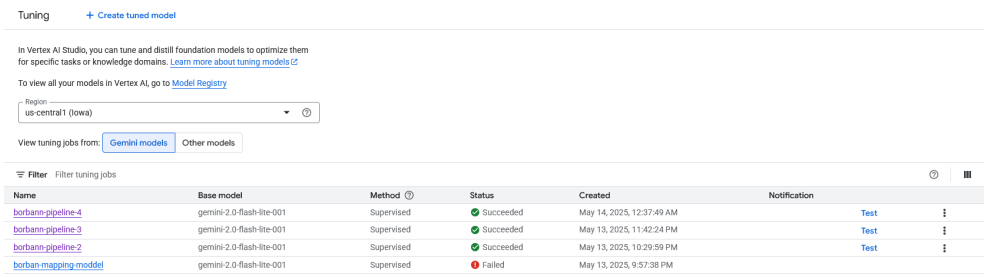

Figure 2: Data Mapping LLM models with multiple versions

Figure 2 display versions of data schema apping model that we fine-tune on Vertex AI platform.

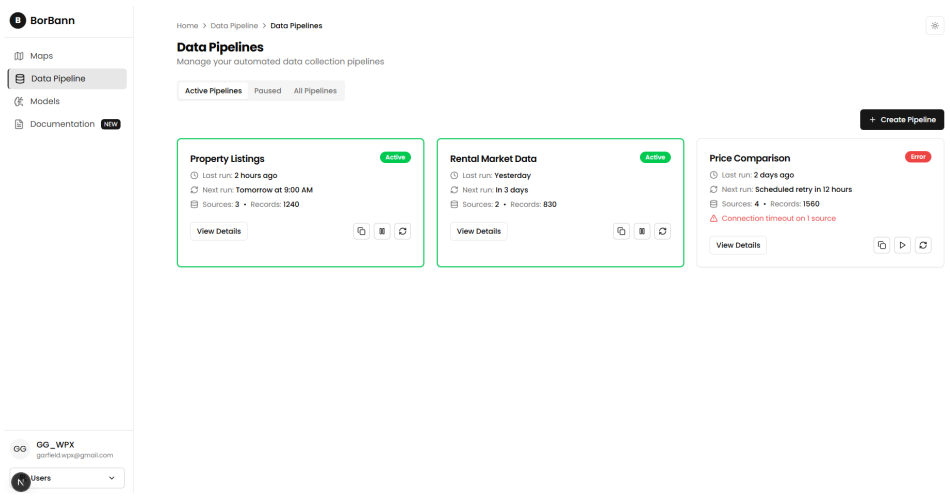

Figure 3: Pipeline status UI interface

Figure 3 is an example of UI interface we have been developed along with the pipeline backend service.

2.5 Experimental Results

Performance metrics are based on 2 metrics:

- **JSON Syntactic Validity**: Parse the output string and check for validity.
- **Pydantic Schema Conformance**: Check with pre-defined pydantic schema to ensure that it output the desire data scheme.

There is a problem on the training data so conformance scores are low, we will fix it.

Table 1: Model Validation Metrics

Model Version	Metric	Value (%)
BORBANN_PIPELINE_2	JSON Syntactic Validity	91.67%
	Pydantic Schema Conformance	63.64%
BORBANN_PIPELINE_3	JSON Syntactic Validity	100.00%
	Pydantic Schema Conformance	0.00%
BORBANN_PIPELINE_4	JSON Syntactic Validity	100.00%
	Pydantic Schema Conformance	0.00%

2.6 Datasets Acquired or Used

We mainly based on two sources; data from implemented pipeline service and generated data from LLM

- **Collected from pipeline service** – Combine the data output from pipeline with specific prompt to create user role and define the target canonical dataset for model role
- **Generate with Gemini 2.5 Flash Preview 04-17 with this prompt** – Craft prompt to more synthetic datas and cover more cases

2.7 Updated Development Plan

Here is the updated gantt chart

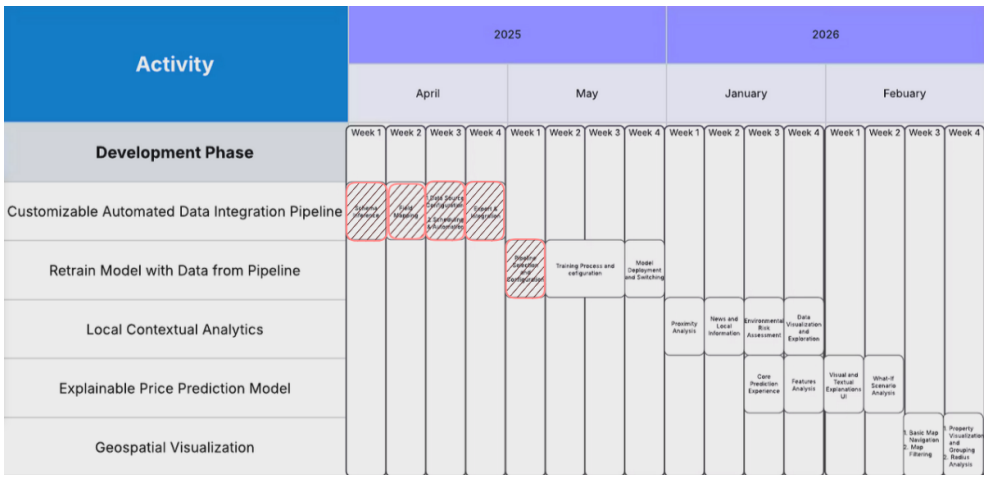

Figure 4: Gantt Chart

2.8 Self-evaluation

I think this pass month we have done a lot of things such as system design and UI design, our progress is faster on the frontend side and we think that on the backend is a bit slower due to challenges such as integrity of data in pipeline, so we need to redesign the pipeline service many times. Also, we have a bit problem on the scheduling of pipeline too, so we need to rewrite it too due to bad design.

Overall, We think that we need to work harder if we want to finish this project in time with good quality.