Solution Architecture Document

T316 – Capstone Project - AutoGuru

A Report

Written For

QUT

Prepared by

Alvin Luu

Submitted in partial fulfillment of project requirements for

QUT - T316 - Capstone Project - AutoGuru

Document Control

Document Title	Solution Architecture Document
Project	Vantage
Revision	
Date	25/08/25
Prepared By	Alvin Luu
Checked By	
Approved By	

Revision History

Revision	Date	Author	Description	Approved

Table of Contents

1 Scope

This document outlines the boundary of the solution, defining the solution capabilities. The solution will include a user faced front end for data entry and a backend responsible for the operation of the model and its preprocessing processes. This solution will not serve as a definitive answer to approval or disapproval cases, but instead as a guide for AutoGuru's future development.

2 Executive Summary

The scope of the solution includes a backend system that supports new data entry and manages both model and its processing components. The system generates a predicted price based on the ML model's training with the provided input data. This prediction will not be treated as a final solution to the business problem however it will serve as guidance to support AutoGuru's future development of the automated processes.

The final solution will contribute to future automation processes by AutoGuru, through use of the model price predictions. This price prediction outputs a confidence score with different explainability to ensure a trust factor when AutoGuru implements this for customers.

To implement this solution, the development process was split into 2 phases, a planning and development phase. The planning phase consisted of the initial exploration of other models as well as other real-world scenarios within the problem space. This testing narrowed down the potential final models. The second phase than saw an in-depth development of these models as well as their subsequent fine tuning. Through this investigation a final model of Catboost was selected due to its potential in the problem space as well as its improved findings. Secondary in this phase, was the development of integration documents to assist in AutoGuru's future implementation of the solution to their systems.

To mitigate potential risk such as data drift, inclusion of cpi indexes was applied to raw prices to reduce the effect of data drift over time on the model. Within the same context, to ensure all data is handled correctly within the model, a single data entry was created in the front end to ensure inputted data is formatted correctly for the model to operate as intended. These mitigate potential risk that may affect the model in the future.

In conclusion, this solution will serve as a future guide for AutoGuru to enhance their current automation process. Through mitigation strategies this price predictor solution will serve as a predictor to ensure customer trust with the solution due to its accuracy and interpretability.

3 Requirements Summary

For our solution the following business, functional and nonfunctional requirements were considerations in our development process.

3.1 Business Requirements

Description	Priority	Notes
AutoGuru must provide customers with an automated approval or disapproval system	Must Have	Must contribute to the current automation system
The solution provides an interface for developers to upload and incorporate new data into the training pipeline	Must Have	Required for up-to-date model performance
Requires transparent and explainable pricing predictions to ensure trust and compliance	Must Have	Required to evaluate and analyse the prediction
AutoGuru will be handed over the final solution as a GitHub repository	Must Have	Ensures projects transferability and long-term viability

3.2 Functional Requirements

Description	Priority	Notes

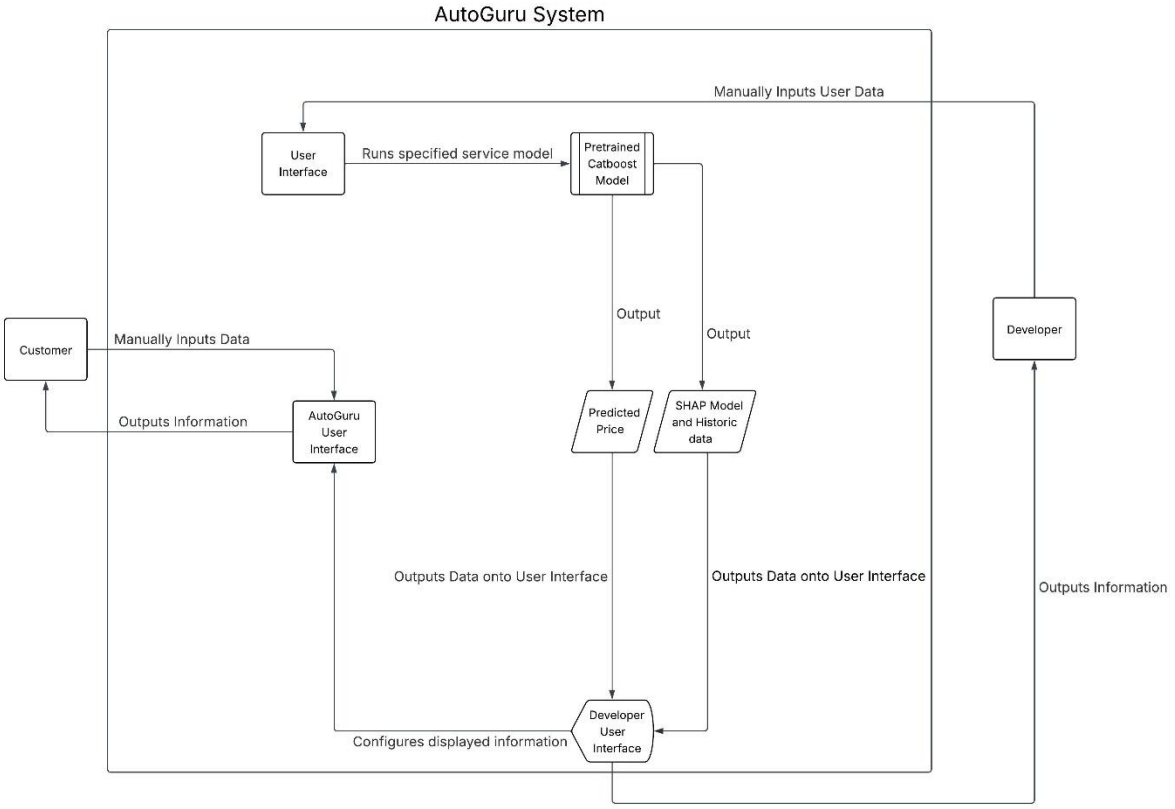
Developers can submit new data	Must Have	Keeps model performance up to date with latest trends
Predicted price	Must Have	Provides a result that can be analysed and evaluated
Explainability of pricing through confidence score and other metrics	Must Have	Ensures the model result is interpretable to developers
Use the most prominent features in the data	Must Have	Required to produce the most accurate results
Front end UI	Should have	Provides developers with a simplistic interface to interact with the model

3.3 Non-Functional Requirements

Description	Priority	Notes
Predictions must be accurate	Must Have	Ensures the solution provides accurate predictions to demonstrate good generalization
Model explainability	Must Have	Allow anybody to interpret and evaluate the models results
Log Predictions	Should Have	Helps developers maintain the model
Scalability	Should Have	Should be designed with future scalability to handle future changes to the model

Portability	Must Have	The final solution should be easily deployable and set up in a new environment

4 Architectural Diagram



5 Implementation Strategy

To implement the solution, the development of the model was broken down into 2 phases. Phase 1, the planning phase featured the initial planning for the solution and its subsequent rough conceptual plans. Phase 2 featured the development of the decided solution in phase 1, and the integration process into AutoGuru's system.

5.1 Phase 1

The objective of phase 1 was to finalise a solution for development in phase 2. To complete this, investigation into real world scenarios like AutoGuru's problem space was done to form

a conceptual plan. After this research, development of numerous models based off the research was used for evaluation and analysis. Using this comparison, the findings decided upon the final solution based off its performance compared to the other developed models

5.2 Phase 2

Upon decision of the final solution, the development was split into 2 groups, one focusing on the solution development and the other on the integration process. The main solution development featured finetuning of the model and feature engineering to improve the data before processed into the model. These actions improved the models results and produced better performance overall. In parallel, the integration team focused on development of integration documentation such as the solution architecture document, quick start guide and the solution architecture document. This was to document the decision choices within the developed solution, as well as explain to stakeholders how the solution solves the problem. Outside of the integration documentation, this also featured the development of a user interface for developers. This is integral for the integration process as it allows developers to test and update the model after handover.

6 Risk and Mitigation

Risk	Mitigation
Data Drift	Incorporate cpi on raw prices to account for inflation over time
Incorrect Data Formatting	Single data entry in the user interface will ensure incorrect data entries won't break the model
Catboost Complexity	Use of SHAP to explain feature importance, allowing for easier debugging and increases trust factor
Sustainability over time	Documentation of retraining processes within the quick start guide will allow model retraining to prevent model decay
Data privacy	To prevent data leakage, a secure private repository with access control on GitHub will ensure confidentiality

7 Conclusion

In conclusion, to resolve AutoGuru's problem of improving their current system of approval and disapproval of booking cases, our solution will result in the development of a price prediction model. This model was developed through thorough investigations into the problem space and numerous testings against other models. This resultant model will be used by AutoGuru to contribute to the current automation process. This will allow AutoGuru to remove static business rules and allow a more dynamic system to take place using predictions based off historical data. This method will also deal with data drift as it incorporates inflation indexes, reducing chance of the model failing over time.