Software Architecture Document

Flight Fare prediction System

By

Staffin

1. Introduction

1.1 Purpose

This document provides a comprehensive architectural overview of the system, using a number of different architectural views to depict different aspects of the system. It is intended to capture and convey the significant architectural decisions which have been made on the system.

1.2 Scope

Currently, there are many fields where prediction-based services are used such as stock price predictor tools used by stockbrokers and service like Zestimate which gives the estimated value of house prices. Therefore, there is requirement for service like this in the aviation industry which can help the customers in booking tickets. There are many research works that have been done on this using various techniques and more research is needed to improve the accuracy of the prediction by using different algorithms. More accurate data with better features can also be used to get more accurate results

1.4 References

Applicable references are:

1. Course Billing Interface Specification, WC93332, 1985, Wylie College Press.
2. Course Catalog Database Specification, WC93422, 1985, Wylie College Press.
3. Vision Document of the C-Registration System, WyIT387, V1.0, 1998, Wylie College IT.
4. Glossary for the C-Registration System, WyIT406, V2.0, 1999, Wylie College IT.
5. Use Case Spec - Close Registration, WyIT403, V2.0, 1999, Wylie College IT.
6. Use Case Spec – Login, WyIT401, V2.0, 1999, Wylie College IT.
7. Use Case Spec - Maintain Professor Info, WyIT407, Version 2.0, 1999, Wylie College IT.
8. Use Case Spec - Register for Courses, WyIT402, Version 2.0, 1999, Wylie College IT.
9. Use Case Spec - Select Courses to Teach, WyIT405, Version 2.0, 1999, Wylie College IT.
10. Use Case Spec - Maintain Student Info, WyIT408, Version 2.0, 1999, Wylie College IT.
11. Use Case Spec - Submit Grades, WyIT409, Version 2.0, 1999, Wylie College IT.
12. Use Case Spec - View Report Card, WyIT410, Version 2.0, 1999, Wylie College IT.
13. Software Development Plan for the C-Registration System, WyIT418, V1.0, 1999, Wylie College IT.
14. E1 Iteration Plan, WyIT420, V1.0, 1999, Wylie College IT.
15. Supplementary Specification, WyIT400, V1.0, 1999, Wylie College, IT.

2. Architectural Representation

This document presents the architecture as a series of views; use case view, logical view, process view and deployment view. There is no separate implementation view described in this document. These are views on an underlying Unified Modeling Language (UML) model developed using Rational Rose.

3. Architectural Goals and Constraints

There are some key requirements and system constraints that have a significant bearing on the architecture. They are:

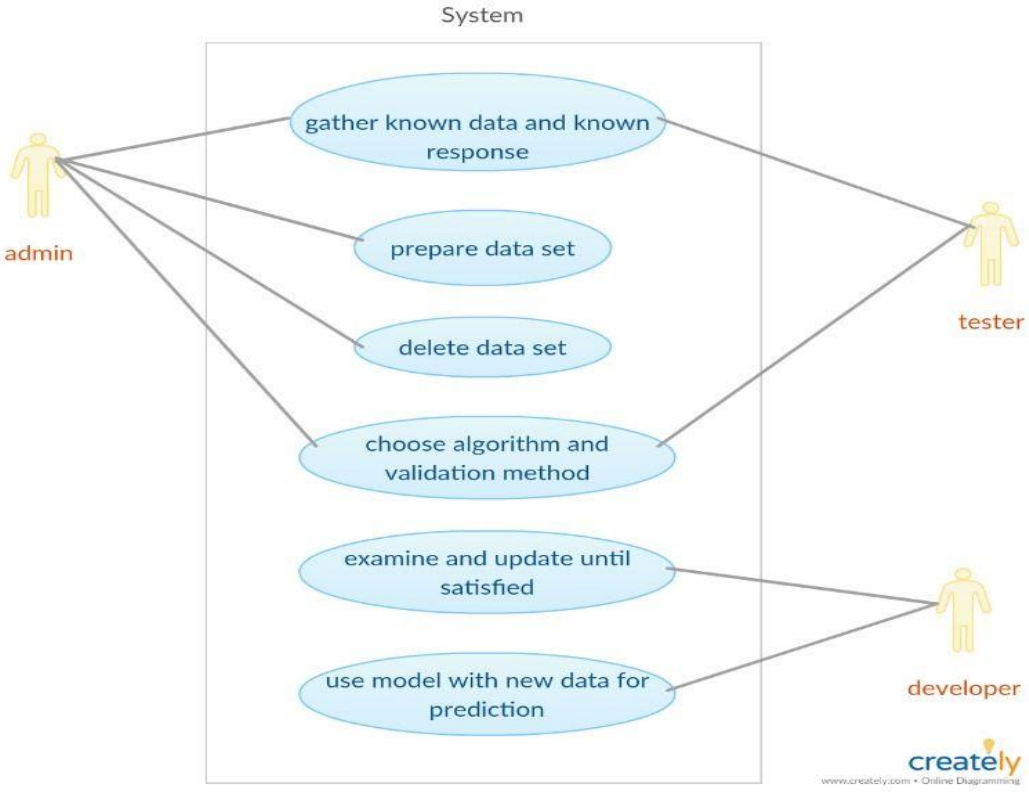
1. The existing system is very manupilative. Flight fares are hugly dynamic
2. Such dynamic and variable pricing causes inconvenience to the users
3. There is a need to develop an efficient flight fare prediction system
4. This will help passengers to plan their travel in an efficient way.

4. Use-Case View

A description of the use-case view of the software architecture. The Use Case View is important input to the selection of the set of scenarios and/or use cases that are the focus of an iteration. It describes the set of scenarios and/or use cases that represent some significant, central functionality. It also describes the set of scenarios and/or use cases that have a substantial architectural coverage (that exercise many architectural elements) or that stress or illustrate a specific, delicate point of the architecture.

These use cases are initiated by the student, professor, or the registrar actors. In addition, interaction with external actors; Course Catalog and Billing System occur.

4.1 Architecturally-Significant Use Cases



4.1 Input data

Th user give the necessary data as input

4.2 Output

Based on the given input data the model predicts the flight fare.

5. Logical View

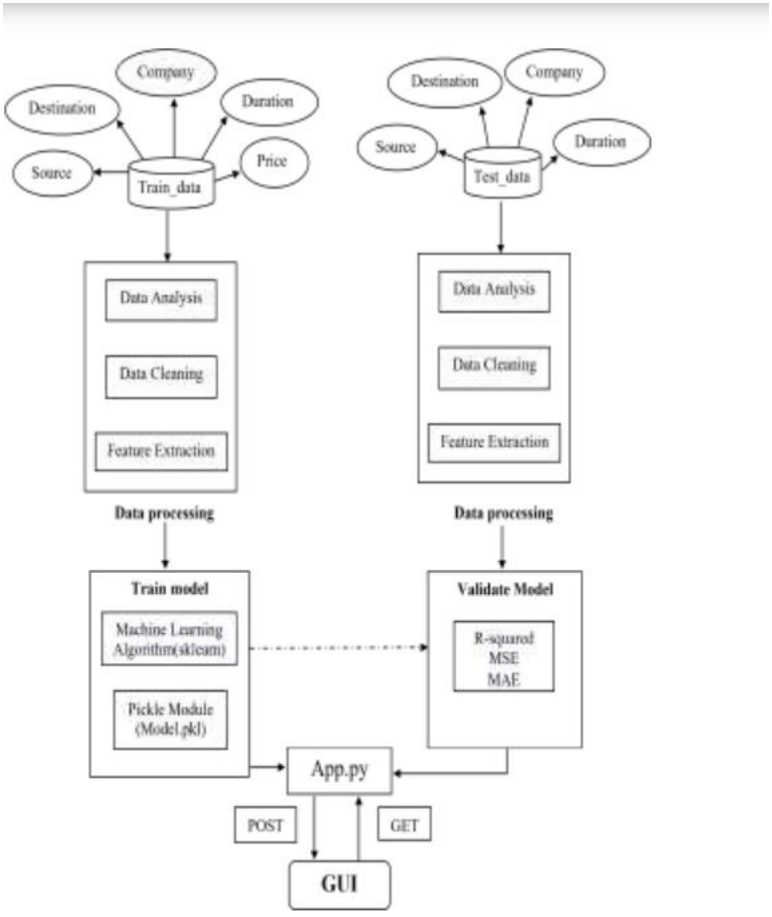
A description of the logical view of the architecture. Describes the most important classes, their organization in service packages and subsystems, and the organization of these subsystems into layers. Also describes the most important use-case realizations, for example, the dynamic aspects of the architecture. Class diagrams may be included to illustrate the relationships between architecturally significant classes, subsystems, packages and layers.

The logical view of the course registration system is comprised of the 3 main packages: User Interface, Business Services, and Business Objects.

The User Interface Package contains classes for each of the forms that the actors use to communicate with the System.

.

5.1 Architecture Overview – Package and Subsystem Layering



5.1.1 Application

This application layer has all the boundary classes that represent the application screens that the user sees. This layer depends upon the Process Objects layer; that straddles the separation of the client from mid-tier.

5.1.2 Business Services

The Business Services process layer has all the controller classes that represent the use case managers that drive the application behavior. This layer represents the client-to-mid-tier border. The Business Services layer depends upon the Process Objects layer; that straddles the separation of the client from mid-tier.

5.1.3 Middleware

The Middleware layer supports access to Relational DBMS and OODBMS.

5.1.4 Base Reuse

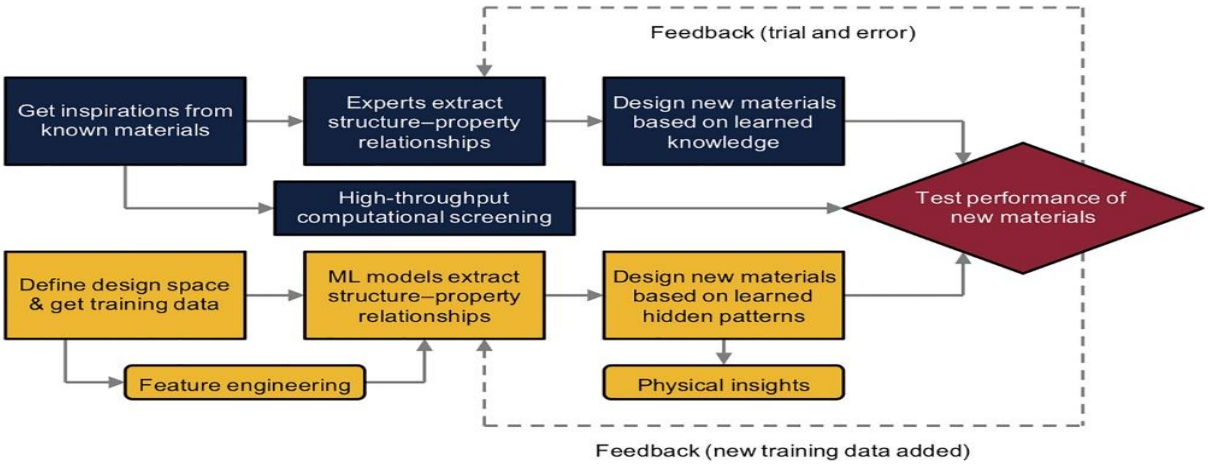
The Base Reuse package includes classes to support list functions.

6. Process View

A description of the process views of the architecture. Describes the tasks (processes and threads) involved in the system's execution, their interactions and configurations. Also describes the allocation of objects and classes to tasks.

The Process Model illustrates the course registration classes organized as executable processes.

6.1 Processes



6.2 Process Model to Design Model Dependencies

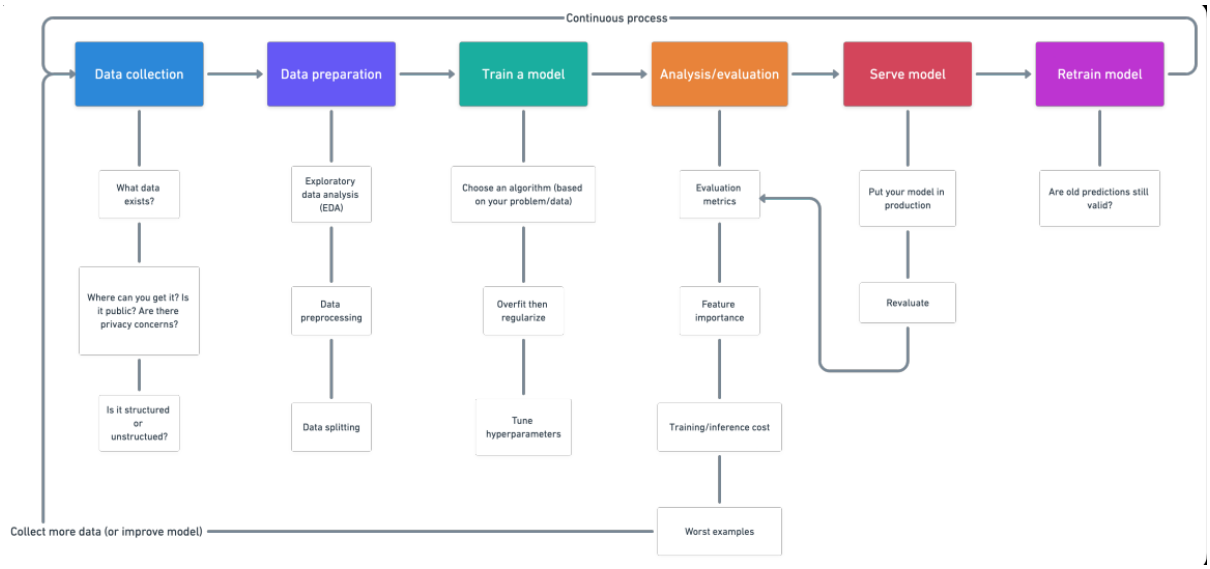
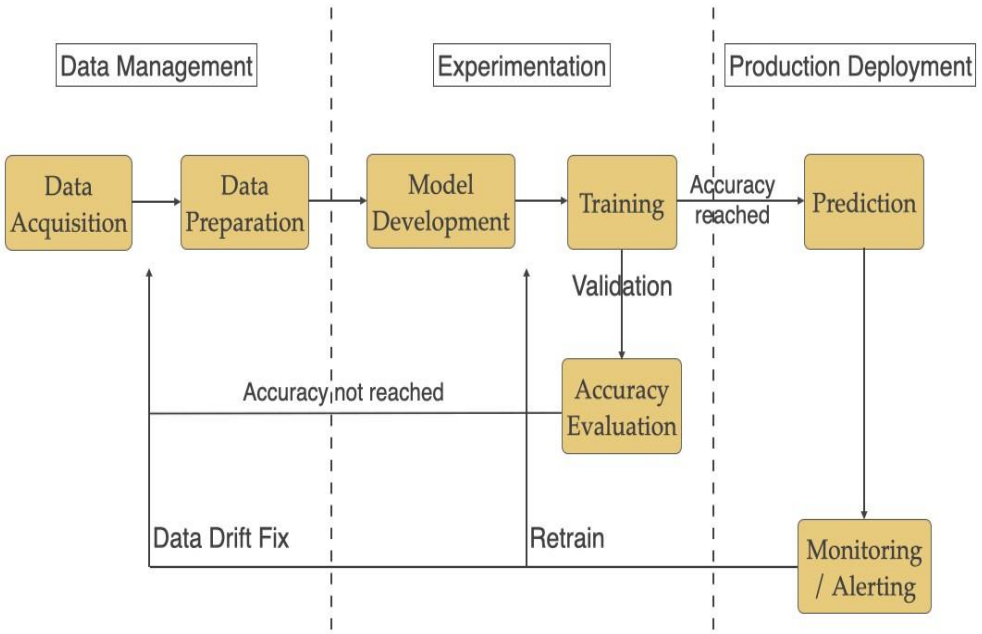


Diagram Name: Process Model to Design Model Dependencies

7. Deployment View

A description of the deployment view of the architecture Describes the various physical nodes for the most typical platform configurations. Also describes the allocation of tasks (from the Process View) to the physical nodes.

This section is organized by physical network configuration; each such configuration is illustrated by a deployment diagram, followed by a mapping of processes to each processor.



8. Size and Performance

The chosen software architecture supports the key sizing and timing requirements, as stipulated in the Supplementary Specification [15]:

1. The system shall support up to 2000 simultaneous users against the central database at any given time, and up to 500 simultaneous users against the local servers at any one time.
2. The system shall provide access to the legacy course catalog database with no more than a 10 second latency.
3. The system must be able to complete 80% of all transactions within 2 minutes.
4. The client portion shall require less than 20 MB disk space and 32 MB RAM.

The selected architecture supports the sizing and timing requirements through the implementation of a client-server architecture. The client portion is implemented on local campus PCs or remote dial up PCs. The components have been designed to ensure that minimal disk and memory requirements are needed on the PC client portion.