

Fuel Aviation	Version: <1.0>
Analysis and Design Document	Date: <19/Apr/18>
<document identifier>	

Fuel Aviation
Analysis and Design Document
Student: Stan David Mihai
Group: 30442

Fuel Aviation	Version: <1.0>
Analysis and Design Document	Date: <19/Apr/18>
<document identifier>	

Revision History

Date	Version	Description	Author
<19/APR/19>	<1.0>	First iteration	Stan David Mihai

Fuel Aviation	Version: <1.0>
Analysis and Design Document	Date: <19/Apr/18>
<document identifier>	

Table of Contents

I.	Project Specification	4
II.	Elaboration – Iteration 1.1	4
1.	Domain Model	4
2.	Architectural Design	4
2.1	Conceptual Architecture	4
2.2	Package Design	4
2.3	Component and Deployment Diagrams	4
III.	Elaboration – Iteration 1.2	4
1.	Design Model	4
1.1	Dynamic Behavior	4
1.2	Class Design	4
2.	Data Model	4
3.	Unit Testing	4
IV.	Elaboration – Iteration 2	4
1.	Architectural Design Refinement	4
2.	Design Model Refinement	4
V.	Construction and Transition	5
1.	System Testing	5
2.	Future improvements	5
VI.	Bibliography	5

Fuel Aviation	Version: <1.0>
Analysis and Design Document	Date: <19/Apr/18>
<document identifier>	

I. Project Specification

The aim of this project is to create an application, a little application that behaves like an online store for fuels for airplanes. It will make the airplane fuel buying more easy and intuitive for any customer.

The first page will be a presentation page for the business and about the managing team. another page will be with the list of engines and the description for each. When clicked, each engine has a personal page where we can see a list of fuels and the description more evaluated.

Each Fuel has a special page, where is the description of each fel(what is made of) and an Buy now button and a textfield where you put the quantity in gallons.

II. Elaboration – Iteration 1.1

1. Domain Model

The store has 3 big actors Customer, Account Manager, Manager.

Every actor will have am account associated, where there will be information about them. This is also a table in the DB. The actors will have : Name(first and second), Email, Phone number.

The Customer, can interrogate the engine lists, to see which type of fuel is compatible with each engine, can order some types of fuel, and can modify only his personal data.

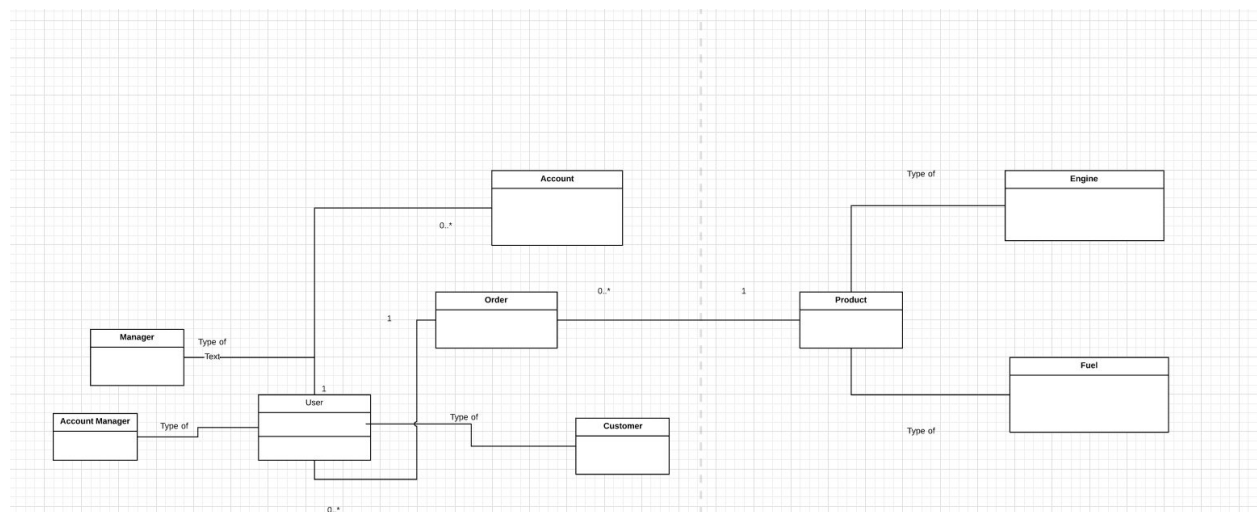
The Account manager, has a list of all the accounts and manages the list. Sends E-mails to the users with the bill, add/delete users if it is necessary.

The Manager has responsibility upon the engine and fuel lists. Adds/Updates/Deletes the engines, and the fuels.

Each Engine has a Name, Description and a list of compatible Fuels.

Each Fuel has a Name, Price, Description, and The total amount available in the Stock.

This being said, I have identified 6 tables: Customer, Account Manager, Manager, Profile, Fuel and Engine.



Fuel Aviation	Version: <1.0>
Analysis and Design Document	Date: <19/Apr/18>
<document identifier>	

2. Architectural Design

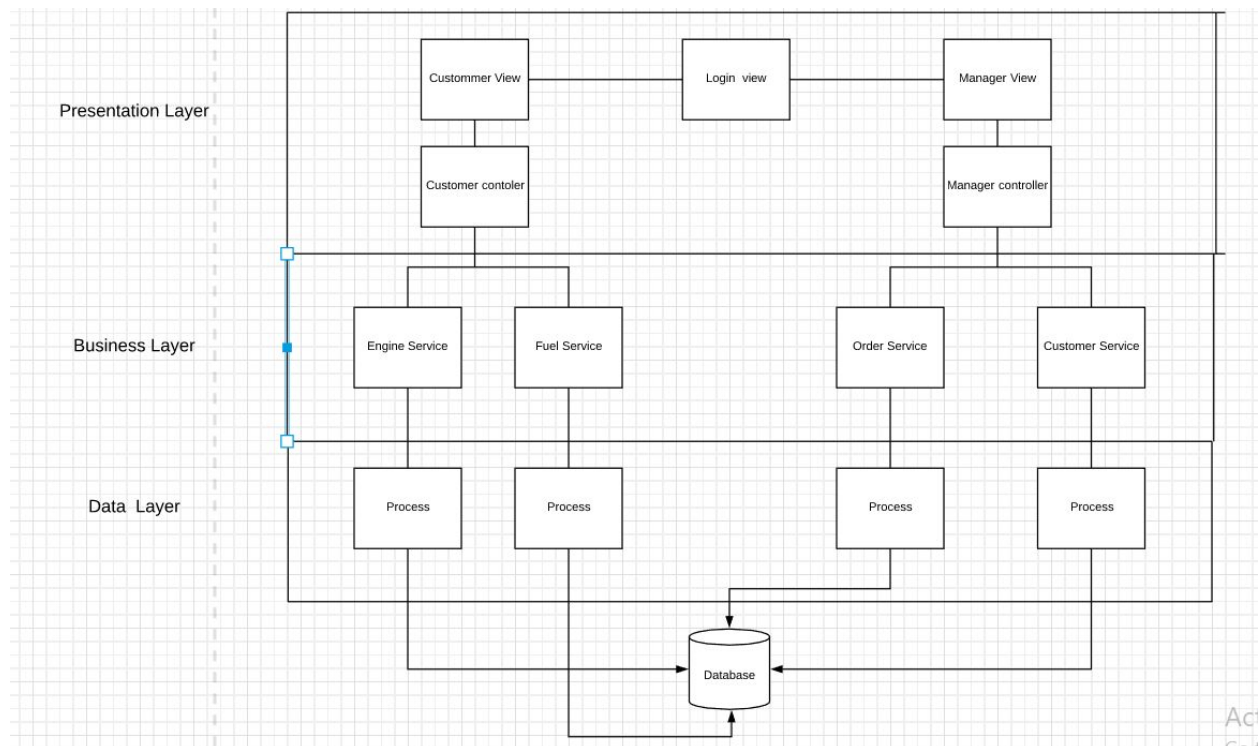
3. Conceptual Architecture

I opted for an MVC architecture. Model–view–controller is an architectural pattern commonly used for developing user interfaces that divides an application into three interconnected parts. This is done to separate internal representations of information from the ways information is presented to and accepted from the user.

The Model represents a mapping of the DB, so the representation of the data.

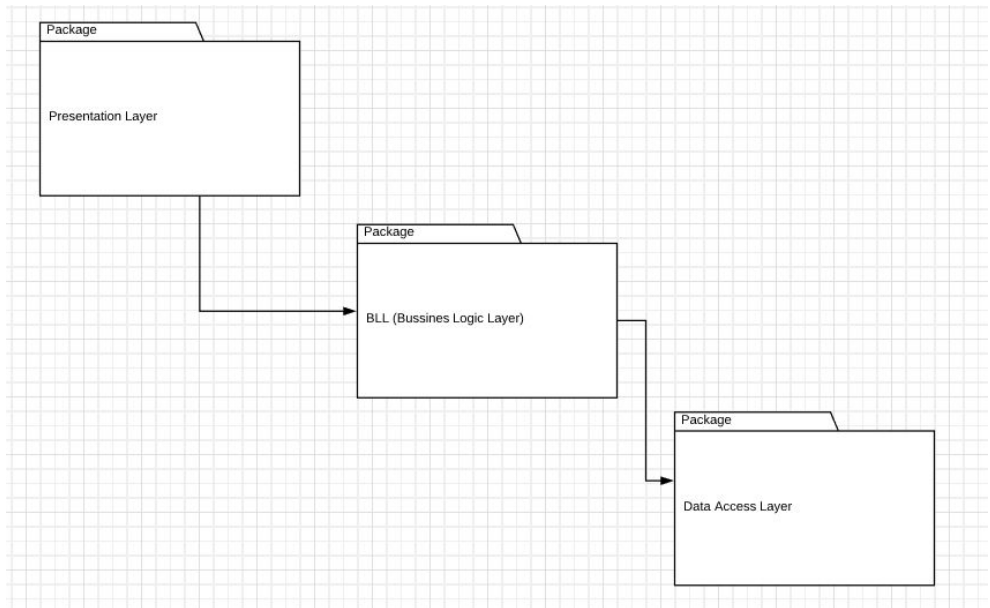
The controller responds to the user input and performs interactions on the data model objects. The controller receives the input, optionally validates it and then passes the input to the model.

The view means presentation of the model in a particular format.

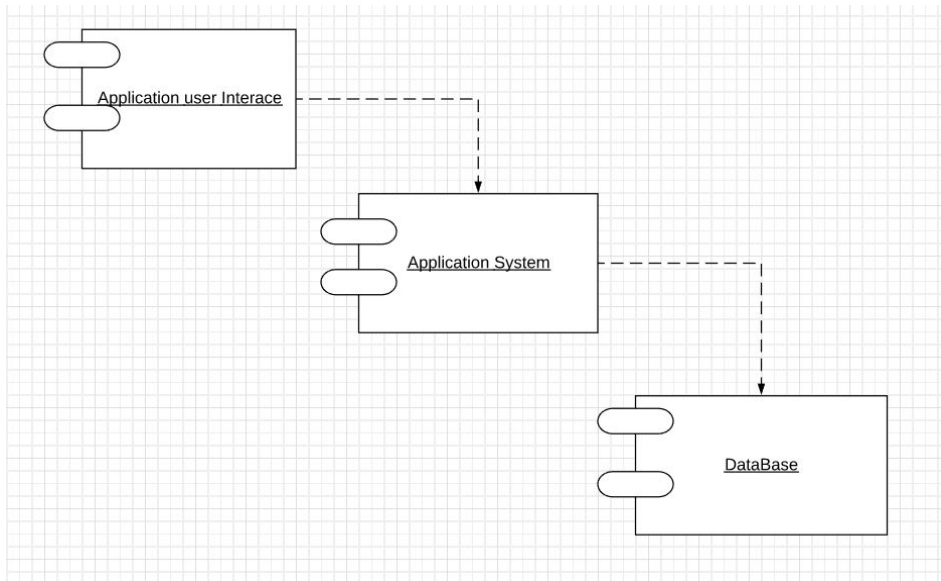


Fuel Aviation	Version: <1.0>
Analysis and Design Document	Date: <19/Apr/18>
<document identifier>	

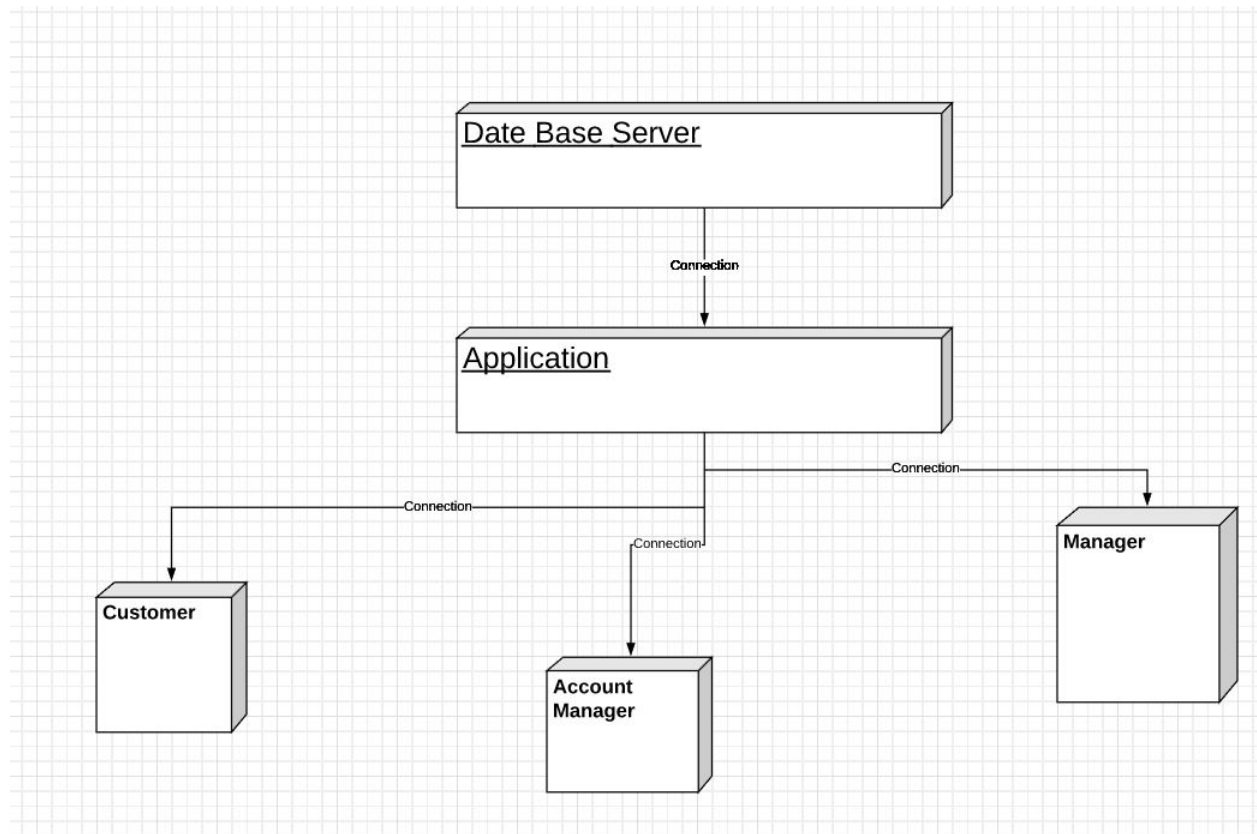
4. Package Design



5. Component and Deployment Diagrams



Fuel Aviation	Version: <1.0>
Analysis and Design Document	Date: <19/Apr/18>
<document identifier>	



III. Elaboration – Iteration 1.2

1. Design Model

1.1 Dynamic Behavior

[Create the interaction diagrams (1 sequence, 1 communication diagrams) for 2 relevant scenarios]

1.2 Class Design

[Create the UML class diagram; apply GoF patterns and motivate your choice]

2. Data Model

[Create the data model for the system.]

3. Unit Testing

[Present the used testing methods and the associated test case scenarios.]

IV. Elaboration – Iteration 2

1. Architectural Design Refinement

[Refine the architectural design: conceptual architecture, package design (consider package design principles),

Fuel Aviation	Version: <1.0>
Analysis and Design Document	Date: <19/Apr/18>
<document identifier>	

component and deployment diagrams. Motivate the changes that have been made.]

2. Design Model Refinement

[Refine the UML class diagram by applying class design principles and GRASP; motivate your choices. Deliver the updated class diagrams.]

V. Construction and Transition

1. System Testing

[Describe how you applied integration testing and present the associated test case scenarios.]

2. Future improvements

[Present future improvements for the system]

VI. Bibliography