<Batu Auto>
Design Document

Authors:
Batuhan Ersan

Group: 0

The purpose of this document is to provide you with a guideline for writing the software design document for your project.

Points to remember:

- *Content is important, not the volume.* ***Another team should be able to develop this system from only this document.***
- *Pay attention to details.*
- *Completeness and consistency will be rewarded.*
- *Readability is important.*

This page intentionally left blank.

Document Revision History

Date	Version	Description	Author
12/01/2023	0.0	Initial draft	Batuhan Ersan
12/02/2023	0.1	Created Main Frame	Batuhan Ersan
12/05/2023	0.2	Created BrandManagementPanel	Batuhan Ersan
12/06/2023	0.3	Created BrandModelManagementPanel	Batuhan Ersan
12/06/2023	0.4	Created DetailManagementPanel	Batuhan Ersan
12/15/2023	0.5	Created UserLoginPanel	Batuhan Ersan
12/20/2023	0.6	Connect the panels	Batuhan Ersan
12/30/2023	1.0	Final version	Batuhan Ersan

This page intentionally left blank.

Contents

1 Introduction

This section addresses the purpose of this document including the intended audience, an introduction to the problem and a detailed view of the project's design. In the discussion, the design of the final system including several detailed diagrams will be described in detail.

1.1 ***Purpose***

This application, which allows all new vehicles to be collected in one place, offers the opportunity to compare with price lists.

1.2 System Overview

The user who wants to buy a new vehicle can select the relevant brands and access the desired price information.

1.3 Design Objectives

The system only provides price and hardware information, you need to contact the authorized dealer for purchases.

1.4 References

I visited their website to find out about the relevant brands, their models and prices, and what equipment packages they offer these vehicles with.

For example:

https://fiyatlistesi.audi.com.tr/?utm_source=google&utm_medium=search&utm_campaign=audi_alwayson_fiyatlistesi&gad_source=1&gclid=CjwKCAiA4smsBhAEEiwAO6DEjcY2UlOMhqoFmmR8i4SAZoVw_e5o7Js6FVj_p1rKmRZz PctZfaREsRoCTZkQAvD_BwE

1.5 Definitions, Acronyms, and Abbreviations

1.6 Introduction

The vehicle price information application is designed as a desktop application based on object-oriented programming. During the design process, Java language was preferred and swing was used. The preferred IDE in this process is Eclipse.

1.7 Environment Overview

The JAR in the sent file works together with the db folder next to it. It enables the program to run by reading from here,

The read information is presented to the user in the system.

1.8 System Architecture

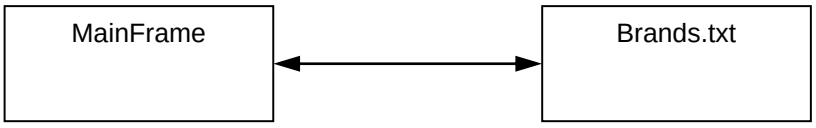
MainFrame reads from brands.txt If the brand is clicked, it brings up the models of that brand with the selectedBrand.txt extension.

Clicked model returns detailed information of the relevant model from the file with selectedBrand+selectedModel.txt extension.

This information is done via the admin panel without the need to fill in the txt files manually..

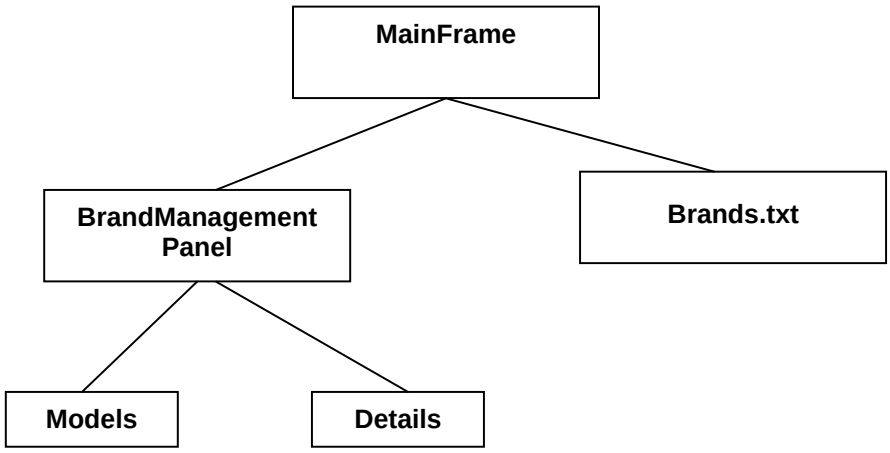
The following is provided as an example:

1.8.1 Top-level system structure of Fuel level controller



The system consists of two major components, a level monitoring subsystem and a fuel subsystem. The interaction between the systems is bi-directional. Current sensor measurements are sent to the level monitoring subsystem from the fuel subsystem and instructions to various actuators are sent back based on the current fuel level.

1.8.2 Level Monitoring Sub-system



MainFrame gets the information from Brands.txt and it uses it in BrandManagementPanel to show models and details

2 Interfaces and Data Stores

Files with .txt extension consisting of brands.txt and the names of the elements added to it.

2.1 System Interfaces

There is an admin panel where new brands, models and details will be added. Only authorized persons can access this panel with the passwords they have previously created, there is no option to register.

Files created in this panel are saved in the db folder.

The following is provided as an example:

2.1.1 Admin Panel

It is used to log in as admin to this panel.Pump

2.1.2 Brand Management Panel

It edits the selected brand or opens the model management panel to view the models of the brand, in addition, a new brand can be added.

2.1.3 Model Management Panel

It edits the selected model or opens the detail management panel to view the details of the model, in addition, a new models can be added.

2.1.4 Detail Management Panel

It edits the selected detail, in addition, new details can be added.

2.2 Data Stores

The db folder included with the sent rar file provides the necessary architecture for the program to run.

3 Structural Design

3.1 Design Explanation and Rationale

One of the features of the application that makes it stand out is that the interface used is in a single frame, eliminating panel confusion and providing faster use.

3.2 Class Diagram

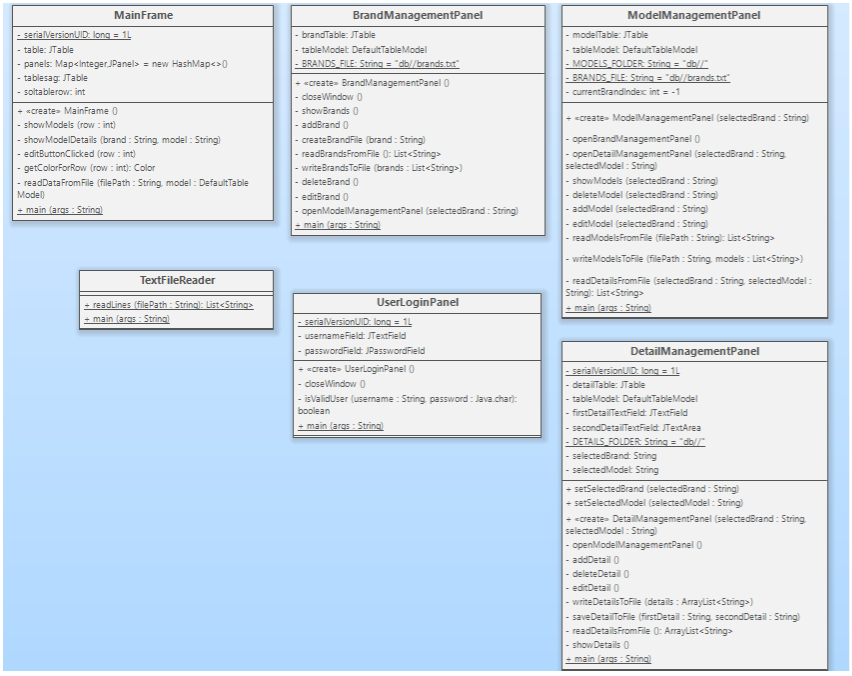
BrandManagement panel, ModelManagement panel and DetailManagement panel are three classes that are intertwined with each other.

Brand management depends on the elements in brand.txt.

Each element has its own txt file and marks are created this way.

The brand.txt file contains the models of that brand, and as in brands.txt, each element has its own file in the form of brand+model.txt.

The brand+model.txt file provides us with the details of that model.



3.3 Class Descriptions

Added to brands.txt file from admin panel

The "audi" brand saves the audi element in brands.txt and creates the file with the audi.txt extension.

Then, if you want to add a model to the Audi brand, the ModelManagementPanel is opened and added as a new element to the audi.txt file.

If the added element is "a3" model

(model can be any desired value) A file with audia3.txt extension is created.

In the DetailManagementPanel, the added detail is written into the audia3.txt file.

The following is provided as an example:

3.4 Classes in the Batu Auto

3.4.1 Class: BrandManagementPanel

- Purpose: *Add or edit brands*
- Constraints: *None*
- Persistent: *No (created at system initialization from other available data)*

3.4.1.1 Method Descriptions

1. Method: *AddBrand*
Return Type: *String*
Parameters: *Brand for panel*
Return value: *success or failure*
Pre-condition: *have a brands.txt*
Post-condition: *String saved to brands.txt*
Attributes read/used: *Brand, brands.txt, String*
Methods called: *getValueAt,exists,.delete,remove()*

Processing logic:
It allows adding, removing and updating brands, and the working logic of model management and retail management is the same.

Test case 1: *Add audi brand. It creates an audi.txt and writes audi to brands.txt*
Expected output is: Audi element at brands.txt

---

4 Dynamic Model

Admin who wants to add a new model

First logs in from UserLoginPanel, then opens BrandManagementPanel

selects the relevant brand

ModelManagementPanel opens, add model button is pressed the relevant model entry is made,

Click on the entered model and the DetailManagementPanel opens and details of the relevant model are entered..

Admin returns to the brandManagementPanel with buttons and all details are saved when it exits.