Large System Design

Carspot for SE 3A04, Tutorial 2

Yasaswi Gopalkrishnan Sharon Platkin Abhijit Singh Dhoat

Joseph Cole Huot David Eric Hemms Yuchen Liu

Monday March 7th, 2016

Contents

List of Tables

1 Introduction

This section should provide an brief overview of the entire document.

1.1 Purpose

a) Delineate the purpose of the document

b) Specify the intended audience for the document

1.2 System Description

a) Give a brief description of the system. This could be a paragraph or two to give some context to this document.

1.3 Overview

a) Describe what the rest of the document contains

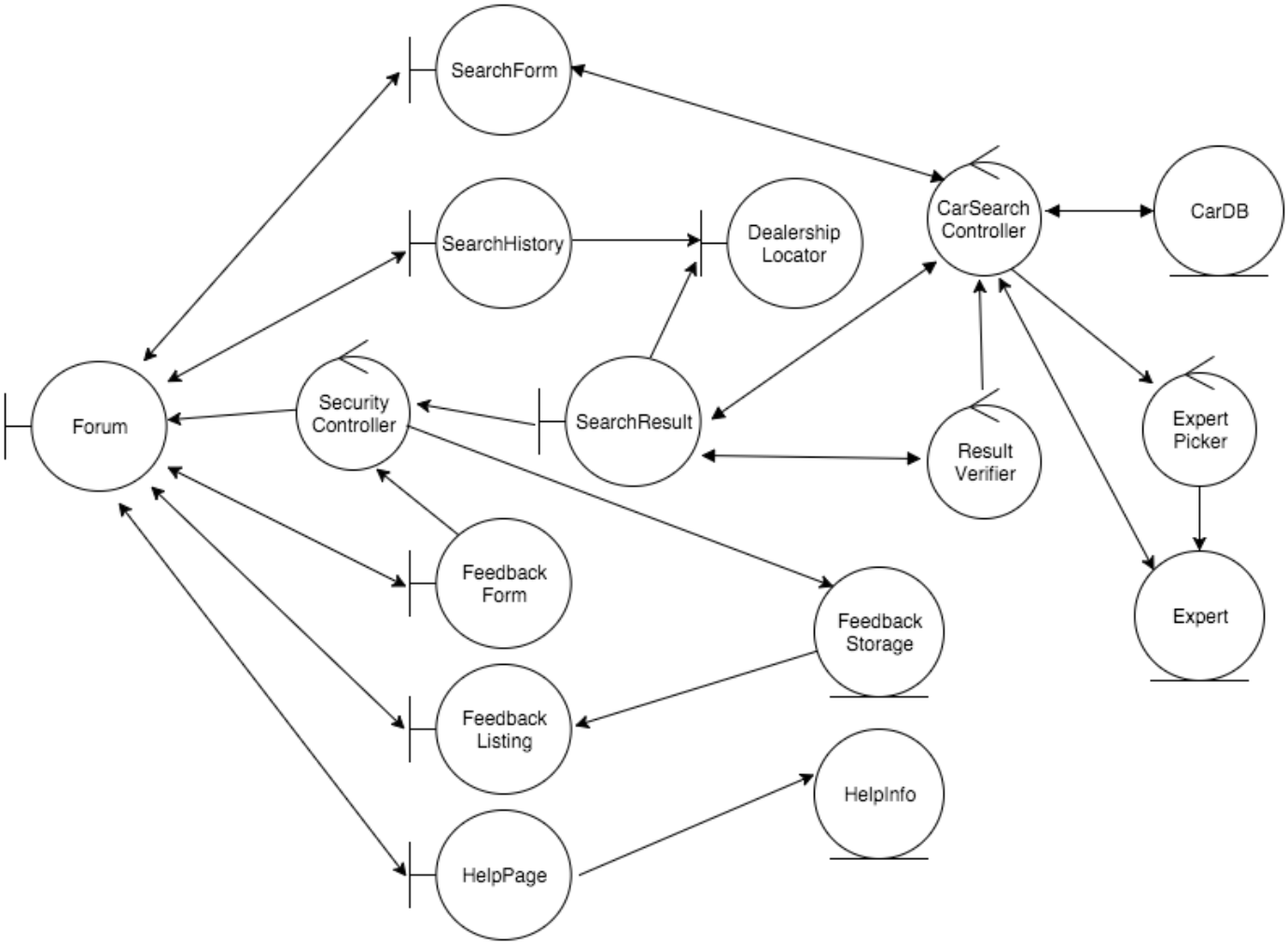
b) Explain how the document is organised

2 Use Case Diagram

This section should provide a use case diagram for your application.

a) Each use case appearing in the diagram should be accompanied by a text description.

3 Analysis Class Diagram



4 Architectural Design

This section should provide an overview of the overall architectural design of your application. You overall architecture should show the division of the system into subsystems with high cohesion and low coupling.

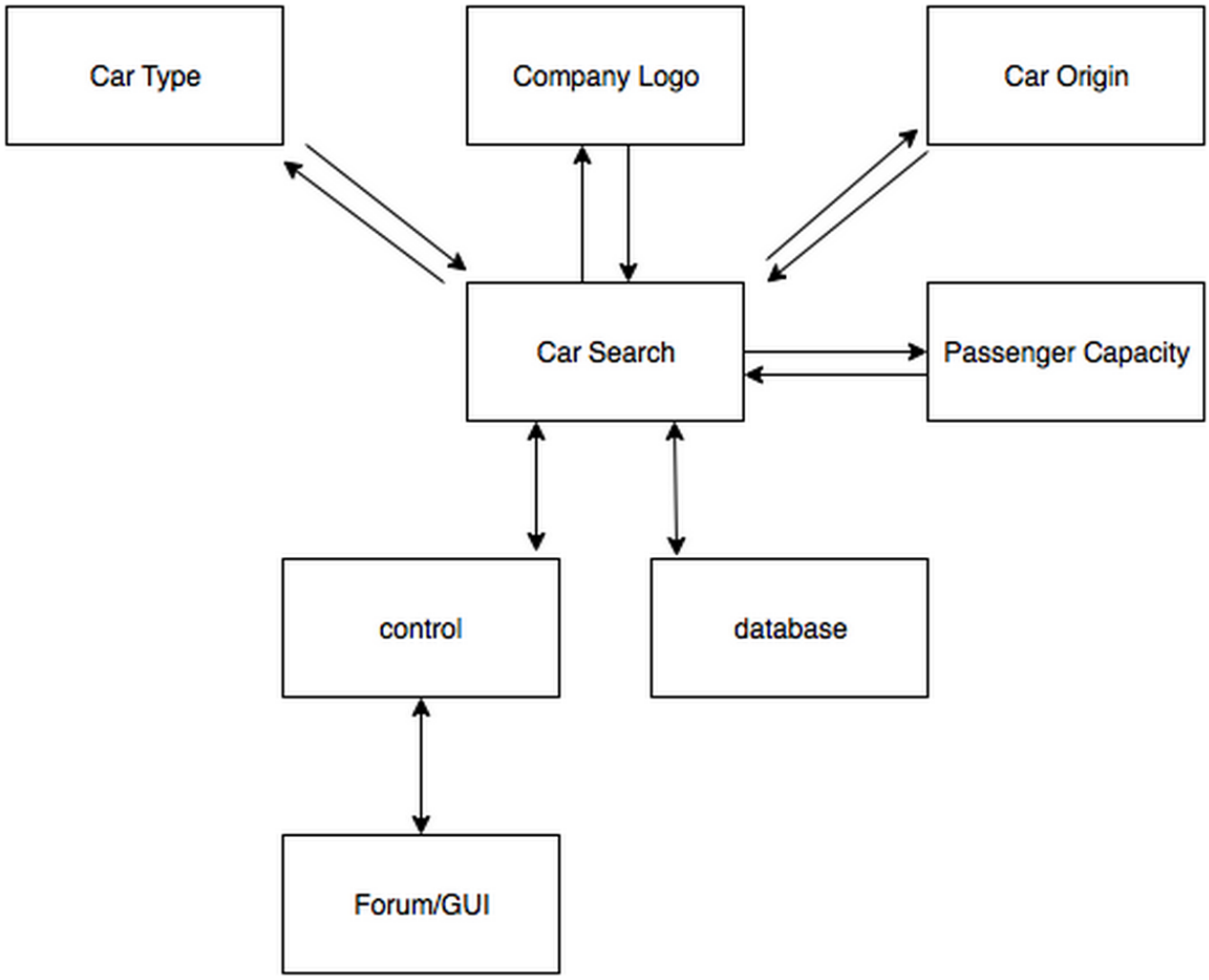
4.1 System Architecture

a) The system is based on a blackboard architecture. There are four separate experts who can provide information independently using their expertise. Each expert identifies a different car property. A car search uses the information provided by the experts to search the car database, finding cars which have the identified properties.

   This architecture structure works well for this system because it is a knowledge based system. Each expert can provide information which is then used to make a decision. Experts can also be added or removed very easily which gives the system flexibility. The

experts are independent of one another, giving the system low coupling. An individual expert has one property which it will identify, giving high cohesion.

b) Structural architecture diagram of the system:



¿¿¿¿¿¿¿ origin/master

4.2 Subsystems

a) **Blackboard Subsystems**

Car Search:

This subsystem uses car properties provided by the experts to find car models in the database which have the provided properties.

b) **Knowledge Source Subsystems**

Car Type:

An expert which identifies the type of car (Sedan, SUV, Minivan, etc).

Company Logo:

An expert which identifies the company that made the car based on their logo.

Car Origin:

An expert which identifies the origin of the car (North American, European, etc).

Passenger Caoacity:

An expert which identifies the number of passengers the car can hold.

Database:

A database containing car models and their properties. The database can be searched to find models which fit certain criteria.

c) **Controller Subsystem**

Control:

This subsystem can initiate a car search and supervise the overall identification process.

5 Class Responsibility Collaboration (CRC) Cards

Class Name: CarDB	
Responsibility:	**Collaborators:**
Contain a listing of all car models and their attributes	-
Allow insertion and deletion of entries	-
Allow editing of entries	-
Provide information to CarSearchController	CarSearchController

Class Name: FeedbackStorage	
Responsibility:	**Collaborators:**
Contain a list of all feedback forms completed by users with anonymity, stored in a file	-
Receive feedback from feedback form for storage	FeedbackForm

Class Name: FeedbackForm	
Responsibility:	**Collaborators:**
Allow user to enter feedback about the application	-

Class Name: CarSearchController	
Responsibility:	**Collaborators:**
Contains algorithm to identify a car given some attributes	-
Extract information from the SearchForm and compile it into a search query	SearchForm
Send result of search to SearchResult for display and verification	SearchResult
Query car database and experts as part of search algorithm to identify the car	CarDB, Expert
Control experts to be used in identification based on attributes given	ExpertPicker

Class Name: SearchResult	
Responsibility:	**Collaborators:**
Receive search result and send it to the forum to be displayed	Forum,CarSearchController
Once a car identification is confirmed, result sent to search history	SearchHistory
Send result for verification before sending to search history	ResultVerifier

Class Name: ExpertPicker	
Responsibility:	**Collaborators:**
Control which experts will be used to identify the car based on attributes that are inputted	Expert
Set experts to "passive" or "active" for identification process	Expert

Class Name: HelpPage	
Responsibility:	**Collaborators:**
Provide information about the application, and how to use it	-

Class Name: Forum	
Responsibility:	**Collaborators:**
Central hub of application to allow navigation to various pages	SearchForm, SearchHistory, HelpPage, FeedbackForm
Display result of car identification	SearchResult

Class Name: SearchForm	
Responsibility:	**Collaborators:**
Allow user to input characteristics of the car they want to identify	-
Send inputted attributes to car identification algorithm	CarSearchController

Class Name: SearchHistory	
Responsibility:	**Collaborators:**
Store previous five confirmed identification results	-
When a new result enters the history, pushes out fifth most recent confirmed identification	-

Class Name: DealershipLocator	
Responsibility:	**Collaborators:**
Interface with Google Maps API to locate dealerships that sell a specific car from the search history	SearchHistory

Class Name: SecurityController	
Responsibility:	**Collaborators:**
Contains encryption and decryption mechanisms for transmitted messages	-
Decrypt search result once it arrives at the forum	Forum
Encrypt the search result before sending it to the forum	SearchResult

Class Name: ResultVerifier	
Responsibility:	**Collaborators:**
Provide the user with the ability to confirm or deny the identified car result	-
Restart car identification if identified car is incorrect	CarSearchController
Restart search form if the identified car is incorrect three times	CarSearchController, SearchForm

Class Name: Expert	
Responsibility:	**Collaborators:**
Know potential car identifications given certain attribute combinations in respective domain of expertise	-
Provide expertise to identify a car given some attributes of its domain	CarSearchController
Provide functionality to be set as "active" or "passive" when trying to identify a car	ExpertPicker

A Division of Labour

Team Member:	**Sections Completed:**
Abhijit	Section 1, 4
Cole	Section 3, 4, Reviewed and Reworked Business Events
David	Section 3, 5, Reviewed and Reworked Business Events
Sharon	Section 2, 3, Reviewed and Reworked Business Events
Yash	Section 3, 5, Reviewed and Reworked Business Events
Yuchen	Section 4, Reviewed and Reworked Business Events