**GRENT**

Object Design Document

22.12.2019

Ahmet Can Terlikçi

Duygu Genç

Tunç Bora Tamsan

Bora Arseven

Prepared for

SE301 Software Engineering



Table of Contents

[1. Introduction 1](#_Toc436772639)

[1.1. Object Design Trade-offs 1](#_Toc436772640)

[1.2. Interface Documentation Guidelines 1](#_Toc436772641)

[1.3. Definitions, Acronyms, and Abbreviations 1](#_Toc436772642)

[1.4. References 1](#_Toc436772643)

[2. Packages 1](#_Toc436772644)

[3. Class Interfaces 1](#_Toc436772645)

OBJECT DESIGN DOCUMENT

# Introduction

## Object Design Trade-offs

While designing classes and interfaces, we mainly focused the security, response time and usability. Since we are following the 3-Tier architectural style, we have one central web server. This style automaticly brings a secure system because all user inputs are validated before reach to the database and also authorization is done by one central server properly. Usability criteria is mainly followed on the UI design. We designed functions in the interfaces with less possible steps to reach a fun to use UI. Also, we made a tradeoff between response time and memory space. We actually chose the response time with a balance to the space of course. This is done by using static variables in storage classess and session scoped beans to reach faster response time in the pages.

## Interface Documentation Guidelines

* Classess are named by using singular nouns.
* Methods are named with verb phrases.
* Errors are returned as exceptions.

## Definitions, Acronyms, and Abbreviations

**Bean:** The name of the controller classes on the JavaServer Faces framework. Each view has a corresponding bean class in java.

**Storage class:** Java classess to store static data of the beans.

## References

We mainly take some examples from the following sites:

* [www.enterprise.com.tr](http://www.enterprise.com.tr)
* www.europecar.com

# Packages

All packages and their content is specified in the javadoc file of the project. Here, we will examine their definitions and usages.

**Authentication:** Includes the controller class of the Login page and also Login Manager class. Mainly provides the classess related authentication of users.

**Chart:** Includes the controller class of the Chart page. Provides all classess related to the chart page and its depended classess.

**Database:** Includes the Database Manager class. Provides connection and initialization for the database.

**Entity:** Includes all entity classess in the GRENT.

**Main:** Includes the controller of the main page and navigation bar. Provides all classess and functions related to the main page and also navigation bar.

**Office:** Includes the controller of the offices page. Offers all classes and functions related to the office page and its depended classess.

**Search:** Includes the controller of the search page and static storage class for that controller. Static storage class stores the corresponding variable which come from other controllers to the controller of the search page. Offers all classess and functions related to the search page and it s depended classess.

**Myprofile:** Includes the controller of the my profile page and a java class for my profile content. Offers all classess and functions related to the my profile page and its depended classess.

# Class Interfaces

Class and interface definitions are specified in the javadoc file of the project.