Online Auction

Object Design

1

23.12.2018

Berkay Yılmaz

Tolga Güldütuna

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

Object Design Document (ODD) describes object design trade-offs made by developers, guidelines they followed for subsystem interfaces, the decomposition of subsystems into packages and classes, and the class interfaces. The ODD is **used** to exchange interface information among teams and **as a reference during testing**. The audience for the ODD includes system architects (i.e., the developers who participate in the system design), developers who implement each subsystem, and testers.

Among three approaches to generate ODD, we follow “**ODD embedded into source code**” approach.

# Introduction

Describes the general trade-offs made by developers (e.g., buy vs. build, memory space vs. response time), guidelines and conventions (e.g., naming conventions, boundary cases, exception handling mechanisms), and an overview of the document. Interface documentation guidelines and coding conventions are the single most important factor that can improve communication between developers during object design. These include a list of rules that developers should use when designing and naming interfaces.

## Object Design Trade-offs

* + 1. Functionality vs Staffing

As our development group have lost more than half of its member throughout its development cycle, we had to cut some functionality from the project to deliver it by delivery date.

* + 1. Use vs Build

We have ‘bought’ component for our project say ‘Buy vs Sell. As delivery on time is the single most import aspect of our project, we have used external modules wherever possible.

* + 1. Delivery Time vs. Functionality

As mentioned in 1.1.1, due to planned works not done, our development is behind schedule and thus cut down on some functionality.

## Interface Documentation Guidelines

In this system, object-design principle had applied. There are two groups of graphical design in this project. First graphical design was made for User side and the other was made for Admin side. User panel has also two groups; User and Registered User. These two designs are similar to each other.

On the user side, there are some of criterias. For instance; user may understand the app easily and do his/her actions easily on the application, also can done their auctions easily. The side should be understandable. In addition, Registered User can see the auctions that are done by a user and The Registered User can check the user’s information who had done a reservation. Registered User also can add a auctions to the system by an admin approval.

The Admin panel, is more complex according to user-side. However, it is also easy to use for an admin. Menus are clearly identified on the panel. An admin can check all users’ information, can approve/reject auctions.

## Definitions, Acronyms, and Abbreviations

- ID : Identification

- React Native : A JavaScript framework for writing real, natively rendering mobile applications for iOS and Android.

-Laravel : a free, open-source PHP web framework, created by Taylor Otwell and intended for the development of web applications following the model–view–controller (MVC) architectural pattern.

- Login: to get access to an operating system or application, usually in a remote computer

- ODD : Object Design Document

- UI : User Interface

- Server: is a computer,provides services to other computers

- User: a person who use a computer

- Exception: Represents errors that occur during application execution.

- Error: The condition of having incorrect or false knowledge.

## References

<https://facebook.github.io/react-native/>

<https://laravel.com/>

# Packages

Front End (React Native - JavaScript)

Components

\*Auctions

\*Buy

\*Login

\*OwnProduct

\*Picker

\*Product

\*Profile

\*Register

\*Sell

\*Splash

Back End (Laravel – Php - Mysql)

* Admin

+++ AllAuctions

+++unprovedAuction

* Auth

+++Password

+++login

+++register

* Buy Product
* Home
* Productview
* Sell Product
* Sold Product
* Welcome

# Class Interfaces

Front End (React Native - JavaScript)

Back End (Laravel – Php - Mysql)