**OBJECT DESIGN DOCUMENT**

1. **Introduction**

This Object Design Document (ODD) defines the object-level design of the Restaurant Finder project to be developed. The goal of this project is to provide users to easily make reservation for a restaurant which they want.

* 1. **Object Design Trade-Offs**

In the object design phase some trade-off decisions may needed to be made:

* + 1. **Understandability versus Cost**

One of the biggest trade-off during the object design phase has been made between understandabilityand cost. Understandability of the code is too important especially during the testing phase. Understandability of the code is maximized on the sacrifice of cost. Each class and method has designed by readability, so number of methods have increased instead of merging different functionalities in the system and functions have been implemented in a clear way. These cause an additional cost in the developing phase.

* + 1. **Development Cost versus Functionality**

Our project provides a lot of functions for users such as listing restaurants by a category or city, looking restaurant’s information and making reservation for this restaurant etc. Each function of the system requires extra design and this causes an extra cost for the development.

**1.1.3 Security versus Cost**

In our Project, each user will be able to login to the system by using the username and password. This brings an additional cost to the system.

**1.1.4 Efficiency versus Portability**

This project is efficiently designed by which porting from one environment to another is easy. With server based database and file tools, portability is much easier to support in various environment .

* + 1. **Reusability versus Cost**

The adapters and other components (i.e XML,HTML Parser) can be used within the system to implement the functionality. This increases and negatively effect the cost while increases the reusability.

* + 1. **Memory Space versus Response Time**

The project mainly focuses on to achieve minimum response time for their users. The informations and pictures are all stored in the memory but since they are all consists of characters there is no huge usage of memory. Speed of the project mainly depends on the speed of the internet connection.

* 1. **Interface Documentation Guidelines**

Interface documentation guidelines and coding conventions are the most important factors that can improve communications between developers during object design. Naming and coding conventions make programs more understandable by making them easier to read.

* + 1. **Naming and Coding Conventions**

**Packages**

The prefix of a unique package name have been written in capital and all lowercase  
  
**Classes**

Class names are nouns, in mixed case with the first letter of each internal word capitalized. We kept our class names simple and descriptive  
  
**Interfaces**

Interface names have been capitalized like class names.

**Methods**

Our methods are verbs, in mixed case with the first letter lowercase, with the first letter of each internal word capitalized.

**Variables**

Variable names are short yet meaningful.They are lowercase first letter. Internal words start with capital letters. Variable names start with dollar sign $.

**Constants**

Constants are all uppercase with words separated by underscores.

* 1. **Definitions, Acronyms and Abbreviations**
* ODD : Object Design Document
* PHP : Hypertext Preprocessor
* XML: Extensible Markup Language
* HTML: Hypertext Markup Language