A PROJECT REPORT
ON

"TRAVEL RESERVATION SERVICE"

Submitted to
UNIVERSITY BORDEAUX 1

BY

NGUYEN Quang Anh	Team Leader
GROUP MEMBER B	ROLL NUMBER B
GROUP MEMBER C	ROLL NUMBER C
GROUP MEMBER D	ROLL NUMBER D

MASTER INFORMATIC - SOFTWARE ENGINEERING
UNIVERSITY BORDEAUX 1

2014-2015

ABSTRACT

Design pattern is a strong tool in Object Oriented Programming. The purpose of this project is to test the ability of understanding and implementing the design patterns.

In this project, I have implemented 11 design patterns. They are Singleton, Abstract Factory, Builder, Proxy, Composite, Decorator, Observer, Iterator, Visitor, Template

Contents

List of Figures

1 INTRODUCTION

In this project, we immitated a system that work for a travel agency. This system can book two main types of travel for the clients: travels without service and travels with services. Travel without service, means that the reservation is made only for the flight, where as travel with services allow a demand for checking in an hotel an renting a car.

For the purpose of simplify the input process, we prepared some input in advance. Below is the tables that present the flights available betweens the cities.

Departure	Destination				
	Paris	Bordeaux	Canberra	Tokyo	Delhi
Hanoi	x	x	x	x	x
Hochiminh	x			x	x
Hue	x		x	x	
Haiphong		x	x		
Paris		x	x	x	
Bordeaux	x			x	x
Canberra	x	x		x	x
Tokyo	x		x		x
Delhi	x		x	x	

Between Hanoi, Hochiminh, Hue and Haiphong, there are flights between each other.

2 PROGRAM 'S SPECIFICATION OVERVIEW

2.1 Brief Introduction

Our project 's purpose is to help a travel agency make a booking for clients. There're two kinds of service package supported

- Travel without service: consist of only flight's reservation
- Travel with services: apart from flight's reservation, hotel's reservation and car renting could also be added

There're three kinds of flight's reservation

```
public enum FlightTicketType {
   VIP, Normal, Pool
}
```

The *Pool* ticket is the type of ticket proposed to the travel agency by a flight agency. The only difference between *Pool* and *Normal* is that *Pool*'s price is cheaper than the other.

There're also three kinds of hotel's reservation

```
public enum HotelServiceType {
   Cheap, Normal, Lux
}
```

2.2 Software Usage

The creation of a Travel's object could be done using class *TravelBuilder*. In this class, there're two methods that can create a travel without service

```
public Travel buildTravelNoServiceForOneClient(Client c, FlightTicketType
    type, CityName destination)
public Travel buildTravelNoServiceForGroup(Client c, FlightTicketType type,
    CityName destination)
```

When a client register a service, if there isn't any direct flight between the client's current city and the destination city, a transit flight will be added.

For creating a travel with services, we could use

```
public Travel buildTravelSimpleService(Client c, CityName destination,
    FlightTicketType f, HotelServiceType h)
```