Travel Agency Manager

Analysis and Design Document

Student:Vid Alexandru

**Group:30234**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <dd/mmm/yy> | <x.x> | <details> | <name> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

I. Project Specification 4

II. Elaboration – Iteration 1.1 4

1. Domain Model 4

2. Architectural Design 4

2.1 Conceptual Architecture 4

2.2 Package Design 4

2.3 Component and Deployment Diagrams 4

III. Elaboration – Iteration 1.2 4

1. Design Model 4

1.1 Dynamic Behavior 4

1.2 Class Design 4

2. Data Model 4

3. Unit Testing 4

IV. Elaboration – Iteration 2 4

1. Architectural Design Refinement 4

2. Design Model Refinement 4

V. Construction and Transition 5

1. System Testing 5

2. Future improvements 5

VI. Bibliography 5

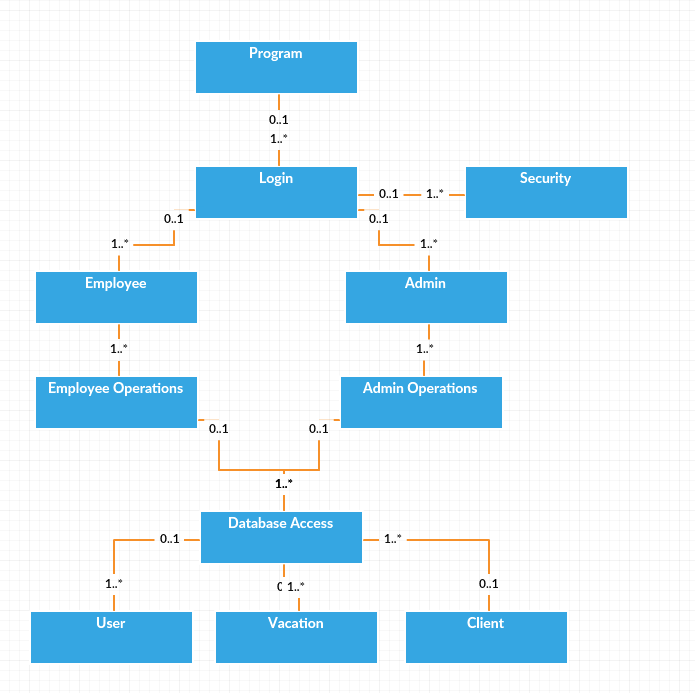
# Project Specification

Design a client-server application for managing the activity in a tourism agency. The application is used by the agency’s employees and allows them to add/modify/delete vacations, add/modify client information and reserve vacations for clients. Vacations can be reserved in the country and abroad. There are three types of vacations that can be reserved: cruises, tours and stays. Within a stay, a client can choose to go on one or more several sightseeing trips. The information about clients and vacations is stored in a database which is updated periodically according to the information (hotels/trips availability, etc.) provided by the operators that collaborate with the agency. This information is provided in XML files, each having a format specific to each operator.

# Elaboration – Iteration 1.1

# Domain Model

# Architectural Design



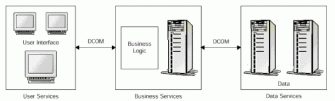
## Conceptual Architecture

**Three-Tier Applications**

Due to the limitations of the 2-tier client-server architecture, distributed applications are often divided up into three or more tiers. Components in each of these perform a specific type of processing – there's a **User Services** (Presentation) tier, a **Business Services** tier, and a **Data Services** tier in a 3-tier application.

The main distinction between this 3-tier architecture and your traditional 2-tier client-server architecture is that, with a 3-tier architecture, the business logic is separated from the user interface and the data source.

Breaking up applications into these separate tiers or sections can reduce the complexity of the overall application, and results in applications that can meet the growing needs of today's businesses. **n-tier** applications are just 3-tier applications that might further sub-divide the standard User Services, Business Services, or Data Services tiers. In any case, an application with more than two tiers can be considered an n-tier application.

[](https://msdn.microsoft.com/en-us/library/bb727121.image002_big(l=en-us).gif)

In this type of application, the client should never access the data storage system directly. If it did, it would be circumventing the business rules of the application and would thus be unable to ensure that the data on display to the client was correct.

## Package Design

## Component and Deployment Diagrams

# 

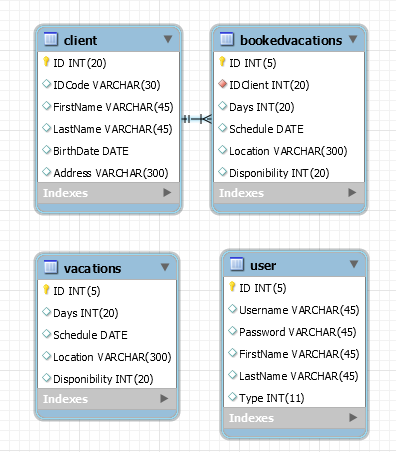
# Elaboration – Iteration 1.2

# Design Model

## Dynamic Behavior

# Untitleddd

# 2.Data Model



# 

# Construction and Transition

# Future improvements

The application could be improved by adding more features for the employee user.

Another improvement that could be done is extending the application for mobile use.

Many improvements are possible for the UI and the code behind it.

# Bibliography

<http://wikipedia.com>

<http://stackoverflow.com>