<Restaurant Management System>

Analysis and Design Document

Student:Popa Diana-Maria

**Group:30234**

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

‘Hospital Management System’ is an application designed for the management of a restaurant. The application has four types of users ( administrator, client, doctor, ) . which have to provide a username and a password in order to use the application.

The administrator can perform the following operations:

* Create/update/delete an user accounts.
* View all the users
* View all orders
* View all orders for a client
* View all orders for a waiter
* Generate new daily offer
* Send email to a client

The chef can perform the following operations:

* Send a notify to the waiter when an order is ready

The waiter can perform the following operations:

* Modify order’s status

The client can perform the following operations:

* View all the products
* Add a product to the order

# Elaboration – Iteration 1.1

# Domain Model

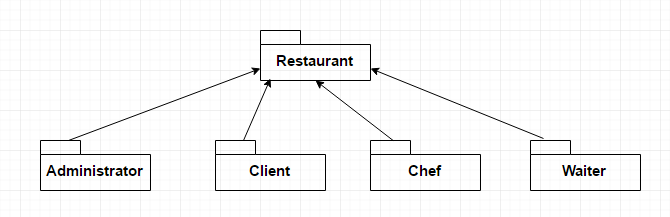
# Architectural Design

## Conceptual Architecture

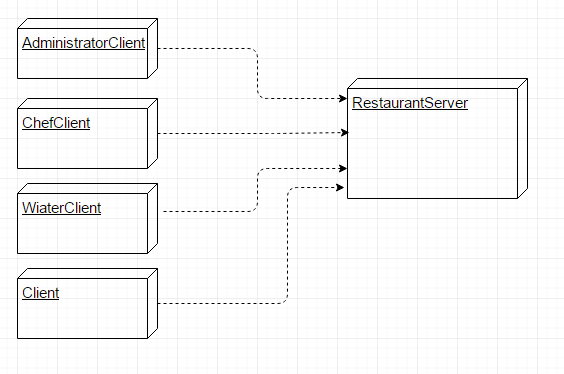
This application implements a client-server functionality. The system is segregated into five application: one server – the database with application logic represented as stored procedures, and three clients: administrator, client, waiter and chef which make requests to the server.

Using a .Net Web API to expose the server functionality to the client applications, there are basically two programs running at the same time: the code that lives on the server and responds to HTTP requests and the code that lives in the browser and responds to user input.

## Package Design



## Component and Deployment Diagrams

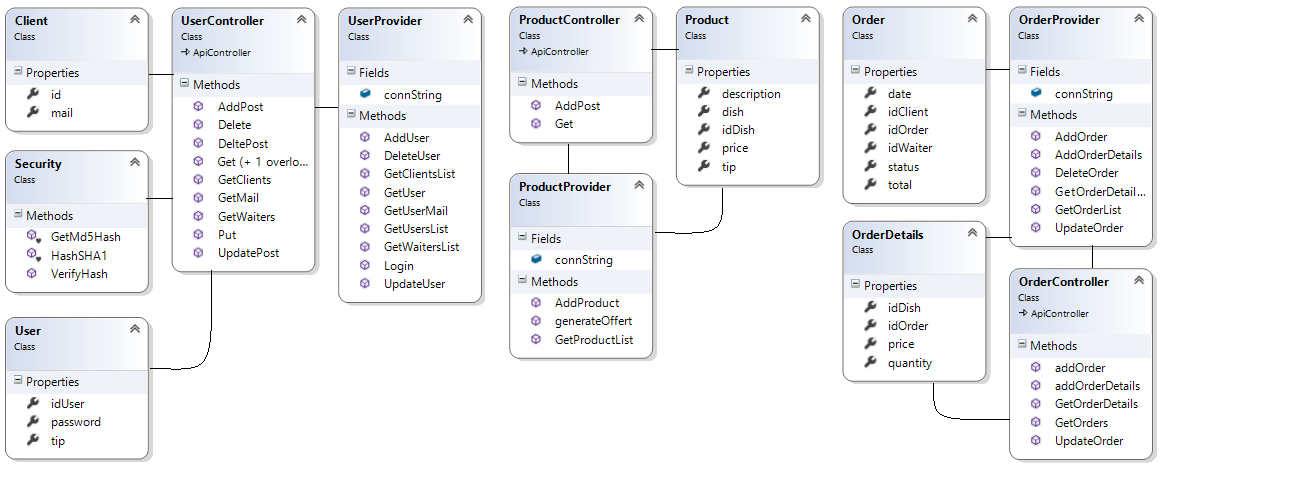


# Elaboration – Iteration 1.2

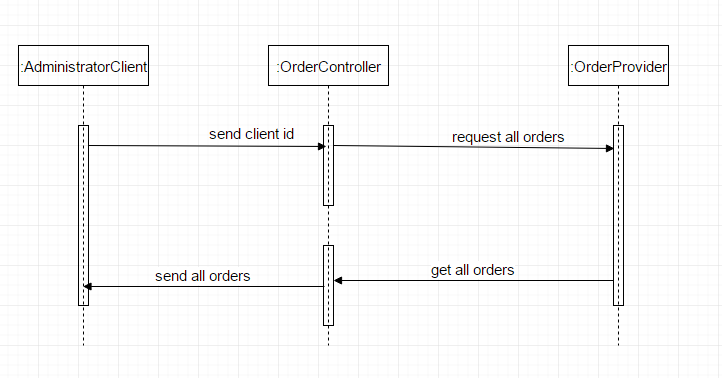
# Design Model

## Dynamic Behavior

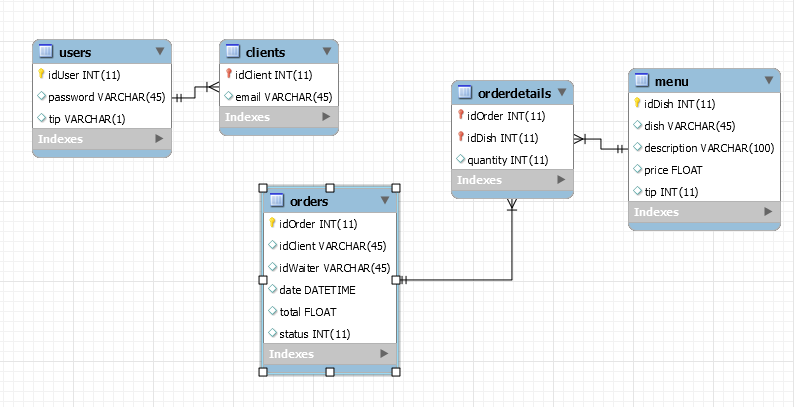
## Class Design



* 1. **Sequence Diagram**



# Data Model



# Bibliography

[**http://www.codeproject.com/Articles/769671/Web-API-without-MVC**](http://www.codeproject.com/Articles/769671/Web-API-without-MVC)

<http://www.asp.net/web-api/overview/getting-started-with-aspnet-web-api/tutorial-your-first-web-api>