**RestaurantFinder**

**Makes Easy Restaurant Reservations**

Requirements Specification and Analysis

26.06.2016

H. Selen Kozanoğlu – 210CS2022

Aslı Tuba Metin – 210CS2023

Arda Yazkan – 210SE2213

Emine Yıldız- 212CS2176

Prepared for

SE301 Software Engineering



Table of Contents

[1.Introduction 3](#_Toc454745421)

[1.1 Purpose of the System 3](#_Toc454745422)

[1.2 Scope of the System 3](#_Toc454745423)

[1.3 Objectives and Success Criteria of the Project 3](#_Toc454745424)

[1.4 Definitions, Acronyms and Abbreviations 4](#_Toc454745425)

[1.5 Overwiew 4](#_Toc454745426)

[2.Current System 5](#_Toc454745427)

[3. Proposed System 6](#_Toc454745428)

[3.1. Overview 6](#_Toc454745429)

[3.2. Functional Requirements 6](#_Toc454745430)

[3.3. Nonfunctional Requirements 7](#_Toc454745431)

[3.4. System Models 8](#_Toc454745432)

[3.4.1. Scenarios 8](#_Toc454745433)

[Object Model 21](#_Toc454745434)

[Dynamic Model 22](#_Toc454745435)

[Dynamic Model 23](#_Toc454745436)

[4.Glossary 30](#_Toc454745437)

[5.References 31](#_Toc454745438)

REQUIREMENTS ANALYSIS DOCUMENT

# 1.Introduction

## 1.1 Purpose of the System

The main purpose of this system is to create a web site for the restaurants which they are representing for the user who they can make a reservation. It helps users to easily find restaurants and take a look at informations about restaurant. And also users can keep their past reservations on the system.

## 1.2 Scope of the System

RestaurantFinder is a web site designed for people who wants to make a easily reservation for their favorite restaurants and also they can try and discover new restaurants. Our users will be able to view and read restaurants such as their telephones, photos, addresses and descriptions. Also users can be search restaurants using by categories and find restaurants using cities.

## 1.3 Objectives and Success Criteria of the Project

Below are the few objectives of the system;

* Manage restaurants by categories and by cities.
* A search function on the web site by a restaurant name or a city name.
* Let users make reservation to restaurants and save them.
* Users using login to access their past reservations.

Success criteria of RestaurantFinder is a web site where users can search restaurants to make a reservation which is very effectively. Users might be use this web site fluently.

## 1.4 Definitions, Acronyms and Abbreviations

RAD: Requirements Anaysis Document

## 1.5 Overwiew

Rest of the RAD will contain information about the aim of the web site. Followed by the functional and nonfunctional requirements, the web site models and some examples of the user interface can be set.

Sections of the RAD are described:

* **Proposed System**

Main functions and features about web site is described. Both functional and nonfunctional features are explained.

* **Overview**

Web site's information and functions are described.

* **Functional Requirements**

What our users doing on the web site and what they don't.

* **Nonfunctional Requirements**

Parts of our web site which are Usability, Reliability and Performance.

* **System Models**

This part about scenarious and use cases to make a ideas of web site which name is RestaurantFinder.

* **Object Model**

This part about class diagram of the web site which is the relations of the classes.

* **Dynamic Model**

This part about sequence and state diagrams to show rlations in the web site and how the system launch in which order.

* Glossary

This part about uncertain, ambiguous and undefined words which includes in the web site.

# 2.Current System

In the Internet there are a lot of restaurant reservation systems. The purpose of doing new one is to make a better web site from anothers. After researching other websites, The project team which name is RestauranTeam make a decision. In our websites we can ignore adds which is help user to discover our website in effectively. Because of the adds, the users tap wrong spaces and they do not happy for that. And also the other websites use lots of buttons which they can confused the users. Some websites have not used google maps, so when the user make a reservation to the restaurant, he/she cannot find the restaurant easily. It has not got location service on their websites. On the other hand, design also is the most important part of the web sites. When the web design is very original and professional, then users can use the web site happily and also they can trust all restaurant because of the best vision of the web design.

# 3. Proposed System

RestaurantFinder provides find a restaurant which users want to and make a reservation for it. Users can search new restaurants and take information about them. Also, after register, they can login the system and look their past reservations. In addition restaurants owners can send a form to admin for adding their restaurants into the system and after admin’s confirmation, new restaurants can be added into the system. Also, restaurant owners can edit their restaurants page and information.

## 3.1. Overview

RestaurantFinder allows users easily find a restaurant and make reservation for it. However, users must register and login the system to make reservation. After login the system, they can make reservation and cancel or edit their reservation. Also, there is users profile and they can enter their profile page and look past reservations. Further more, users can search a restaurant by category or city and access its telephones, photos, addresses and descriptions without logging the system. In addition, restaurant owners can arrange their restaurant page.

## 3.2. Functional Requirements

**Register:** To login the system, users must register the system. Users can access our website register form. They complete the form and they register the system.

**Login:** The system does not allow entering without login. To make reservation, users must log in, after that they can be redirected home page. Database has users’ information which is password and username. System should be able to check at each login process user’s username and password from database, so one user has to have one unique

username and password.

**Change password:** All user kind; users, restautantOwner and admin can change password. They login and enter profile page. They click change password and firtly they fill old password after they fill new password.

**List restaurants:** All users can list all restaurants without login. In home page, there is search field by city as well as by category. Users can search restaurant and look its telephones, photos, addresses and descriptions.

**Make reservation:** After users login, they can make reservation. They select a restaurant and enter reservation date, number of people, table option and description. Thus, they make a reservation and the system send mail which contain reservation information to users .

**Cancel reservation:** There is user reservations page. This page list past and future reservation. Cancel button locates next to all future reservation and if users want to cancel reservation, they click this button and reservation is deleted from the system.

**Edit reservation:** If users want to change reservation ınformation, they can edit reservation their reservations page. Edit button locates next to all future reservation. By click this button, users arrange reservation information.

**Request registerForm:** Restaurant owner fill the register form to request to admin for add their restaurant into the system. They click the “join us” in home page and fill the form.

**Approval-Reject restaurant:** There ara request form in request page for admin. Admin enter this page and he/she can approval or reject button. If he/she approvals request, restaurant is added the systam. If he/she rejects request, restaurant can’t be added the sytem.

**Edit RestaurantProfile:** Restaurant owners can arrange their restautant information. After they login the system and enter restaurants page, if they click edit, they change or add information or they arrange restaurant page.

## 3.3. Nonfunctional Requirements

There is some kind of user who are admin, restaurantOwner and user in the

system. Admin has permission for add new restaurant. RestaurantOwners can edit their restaurants page and information. Lastly, users can login the system and make reservation.

***Usability***

The functionality of the system sufficiently maps to general user-requirements. Usability also refers to how well the user interface is considered easy to use. Our system is simple and easy to use. To make reservation are not complicated for users.

***Reliability***

The system should not be allowed entry without input username and password for the site. User’s and restaurant’s information is kept in database system. Also admin has permission for restaurants. Admin can delete or add restaurants.The protection against unauthorized use password protection.

***Supportability***

Admin creates new restaurant, restaurant owner arranges their restaurant page and users make reservation and look past reservation. However, such additions may require the system. New modules can added to the system or some operations can be shut down.

***Performance***

Performance requirements are concerned with quantifiable attributes of the system. Response time for our system should be very low. The system have more than one admin for quick response.

***Implementation***

All users should be able to access their profile page after login the system. Admins functions used by the user are not available through the web.

***Legal***

Users can access after register the system.

## System Models

* + 1. **Scenarios**

**Scenario 1:**

|  |
| --- |
| *Scenario name:* Login |
| *Participant actor instances:* Murathan:User |
| *Flow of events:*   1. Murathan has a date ,he wants to look some restaurant in İstanbul to make a reservation 2. He writes www.restaurantfinder.com on browser 3. He has an account 4. He wrote his username and password in order to login to the site. 5. Murathan sees “Successfully login” message on the screen. |

**Scenario 2:**

|  |
| --- |
| *Scenario name:* Register |
| *Participant actor instances:* Kaan:User |
| *Flow of events:*   1. Kaan’s best friend Murat who is in USA for his language education comes back from there after a 2 years. 2. Kaan decided to make a Welcome Home surprise to him ,he goes in to the site named RestaurantFinder and finds a restaurant that makes a good sushi which Murat’s favorite food. 3. He chooses Sushico and wants to make a reservation, but he has not have account 4. He registers from main page by clicking sign in buton. 5. He writes all information which necessary to create an account (name, surname, phonenumber, e-mail , etc ) from register page 6. Kaan sees “Successfully registered” message on the screen and now he can make a reservation. |

**Scenario 3:**

|  |
| --- |
| *Scenario name:* SelectRestaurant |
| *Participant actor instances:* Arda:User |
| *Flow of events:*   1. Arda login to the site named RestaurantFinder 2. He decides to make a reservation for his 4th anniversary with his girlfriend 3. He selects the city İstanbul and clicks the category named world cuisine to look results 4. He see many restaurant as a result from them he chose La Mancha Restaurant and by clicking make reservation button , he makes reservation for two. 5. System approves the reservation by itself. |

**Scenario 4:**

|  |
| --- |
| *Scenario name:* MakeReservation |
| *Participant actor instances:* Hayri :User |
| *Flow of events:*   1. Hayri login to the site named RestaurantFinder 2. He decides to make a reservation for his mother’s birthday. 3. He selects the city İstanbul and clicks the category named world cuisine to look results 4. He see many restaurant as a result from them he chose Suada Restaurant 5. He clicks make reservation button by selecting 3 people (his father, mother and himself) as a number of person and selects date&time for 17.02.2017 , 19:00. 6. RestaurantFinder receives his reservation and makes the reservation by itself. |

**Scenario 5 :**

|  |
| --- |
| *Scenario name:* RequestRegisterForm |
| *Participant actor instances:* Arda: RestaurantOwner Aslı:Admin |
| *Flow of events:*   1. Arda has a restaurant named GoMongo. 2. He wants to add his restaurant to the site named RestaurantFinder to get more customer and to known more. 3. He clicks Wanna Join Us? Button and go to into Restaurant Register form page. 4. He fills all spaces with his information. His first name , surname, address, phone number, work history, and restaurant information. 5. He clicks OK button and send his form to Aslı. 6. Aslı evaluates Arda’s restaurant information and approves it. 7. Arda has a restaurant owner status right now he can edit his restaurant profile from edit profile page. |

**Scenario 6:**

|  |
| --- |
| *Scenario name:* EditRestaurantProfile |
| *Participant actor instances:* Selen:Restaurant Owner |
| *Flow of events:*   1. Selen has a restaurant in [www.restaurantfinder.com](http://www.restaurantfinder.com) site.. 2. She changed his restaurant’s location to another address. 3. She also wants to change the restaurant adres from the system. 4. She logs in the system from main page by entering her username and password. 5. She enters Edit button from her profile 6. She updates the restaurant’s address from restaurant’s page by writting new adres to the address to the adres area. |

**Scenario 7:**

|  |
| --- |
| *Scenario name:* RejectRequestRegisterForm |
| *Participant actor instances:* Faruk:User , Nuran:Admin |
| *Flow of events:*   1. Faruk has a restaurant named EminEv Yemekleri 2. He is new in food community. He wants to spurt. 3. He thinks that if his restaurant becomes a part of RestaurantFinder community he can grows up easily. 4. He writes [www.restaurantfinder.com](http://www.restaurantfinder.com) on the browser, and he clicks Wanna Join US? Button. 5. He fills all form with his information and clicks OK. 6. Nuran takes Clark’s form and she views his form. 7. Nuran thinks that EminEv Yemekleri is not good for make a reservation, because it is small restaurant and serves home cooking which is not in RestaurantFinder’s category. 8. Nuran clicks reject button and sending mail to the Faruk which explains this rejection. 9. Faruk gets the mail and understands situation. He gives up this idea maybe he can try another time when he has big restaurant. |

**Scenario 8:**

|  |
| --- |
| *Scenario name:* CancelReservation |
| *Participant actor instances:* Mert:User |
| *Flow of events:*   1. Mert has a date in Valentine’s Day 2. He has a account in restaurantfinder 3. He select Istanbul city and lloks results. 4. He decides to make reservation in La Boom Restaurant according to results by selecting the date 14.02.2017 and time is 19.00 5. Mert’s date Hande appears urgent job and she calls david to the that she would’t come to the date. 6. Mert has to cancel reservation because he can not go date by himself 7. He login to system again and by clicking cancel reservation button he cancels reservation from his profile’s reservations page. 8. RestaurantFinder receives an acknowledgement and indicates that the process has been completed successfully. |

**Scenario 9:**

|  |
| --- |
| *Scenario name:* EditUserProfile |
| *Participant actor instances:* Selena:User |
| *Flow of events:*   1. Selena has a account in [www.restaurantfinder.com](http://www.restaurantfinder.com) system. 2. Selena’s mobile phone number change to another telephone line. 3. She wants to updates her mobile phone number from the system. 4. She clicks edit profile button and changes her mobile phone number from the mobile phone number’s space. 5. System updates the information successfully. |

**Scenario 10:**

|  |
| --- |
| *Scenario name:* SelectDifferentTime |
| *Participant actor instances:* Emine:User |
| *Flow of events:*   1. Emine takes new job in Google. It is her dream’s job. 2. She wants to celebrate it with his family and some friends 3. She wants to make reservation from [www.restaurantfinder.com](http://www.restaurantfinder.com) 4. She logs in the web site and looks for some restaurant by selecting the city Ankara 5. She decides to make reservation in Big Chefs Restaurant 6. She selects the date as 05.07.2016 and the time as 18:00 by selecting 6 people 7. RestaurantFinder displays that the selected time and number of people not avaliable at that time. 8. She only changes the time by go back the reservation page and looks for avaliable times 9. She selects the time 20:00 10. System approves the reservation. |

**Use Case Model**

****

**Use case 1:**

|  |
| --- |
| *Use case name:* Login |
| *Participating actors:* Initiated by The User |
| *Flow of events*:   1. USER enters the site name on the browser.    1. SYSTEM shows the main page which includes login, sign in, and restaurant searching area according to city that use choice. 2. USER selects the restaurant he/she would like to make reservation.    1. The SYSTEM requests that the user enter his/her name and password.    2. The User enters his/her name and password.    3. The System validates the entered name and password and logs the user into the system. |
| *Entry Condition*:USER enters login button. |
| *Exit Condition*:SYSTEM returns user’s page |

**Use case 2:**

|  |
| --- |
| *Use case name:* Register |
| *Participating actors:* Initiated by The User |
| *Flow of events*:   1. USER enters [www.restaurantfinder.com](http://www.restaurantfinder.com) onthe browser.    1. SYSTEM shows the main page which includes restaurant search area, login and sign in button. 2. USERenters the sign in button..    1. The SYSTEM displays register page. 3. USER enters his/her information to the register page.    1. The SYSTEM request that the user enter his/her hane, surname, e-mail address and password.    2. The USER enter his/her name, surname, e-mail, address and password.    3. The SYSTEM register the entered name, surname, e-mail address and password. |
| *Entry Condition*:USER enters sing in button. |
| *Exit Condition*:SYSTEM registers |

**Use case 3:**

|  |
| --- |
| *Use case name:* ListRestaurant |
| *Participating actors:* Initiated by The User |
| *Flow of events*:   1. USER enters [www.restaurantfinder.com](http://www.restaurantfinder.com) onthe browser.    1. SYSTEM shows the main page which includes restaurant search area, login and sign in button. 2. USER enters which city button..    1. SYSTEM shows cities with select option. 3. USER enters his/her information to the register page.    1. SYSTEMreturns restaurants according to the city. |
| *Entry Condition*:User enters which city? button |
| *Exit Condition*:SYSTEM returns restaurants |

**Use case 4:**

|  |
| --- |
| *Use case name:* ChooseRestaurants |
| *Participating actors:* Initiated by The User |
| *Flow of events*:   1. USER enters [www.restaurantfinder.com](http://www.restaurantfinder.com) onthe browser.    1. The USER selects city that he/she wants to search.    2. SYSTEM returns restaurants. 2. USER select the restaurant which he/she wants according to result.    1. SYSTEM shows the restaurant’s page which includes some information about the restaurant. |
| *Entry Condition*:USER select city. |
| *Exit Condition*:SYSTEM returns selected restaurant page. |

**Use case 5:**

|  |
| --- |
| *Use case name:* MakeReservation |
| *Participating actors:* Initiated by The User |
| *Flow of events*:   1. USER enters [www.restaurantfinder.com](http://www.restaurantfinder.com) onthe browser.    1. SYSTEM shows the main page which includes restaurant search area, login and sign in button. 2. USER selects city and finds the restaurant that he/she wants to make reservation.    1. SYSTEM requests that the user enter his/her name and password.    2. USER enters his/her username and password.    3. SYSTEM request that the user enter number of persons and selects,date & time.    4. USER enter number of persons and selects date and time    5. SYSTEM confirms the reservation. |
| *Entry Condition*:User clicks make reservation from selected restaurant page and USER should be login first. |
| *Exit Condition*:SYSTEM confirms the reservation. |

**Use case 6:**

|  |
| --- |
| *Use case name:* CancelReservation |
| *Participating actors:* Initiated by The User |
| *Flow of events*:   1. USER enters reservations field from his/her profile    1. SYSTEM returns all reservations making by the USER. 2. USER selects current reservation and clicks cancel. |
| *Entry Condition*:User should be login first and USER clicks cancel button |
| *Exit Condition*:SYSTEM shows “ successfully cancelled” message to the USER |

**Use case 7:**

|  |
| --- |
| *Use case name:* EditProfile |
| *Participating actors:* Initiated by The User |
| *Flow of events*:   1. USER enters [www.restaurantfinder.com](http://www.restaurantfinder.com) onthe browser.    1. SYSTEM display mainpage. 2. USER click login button.    1. SYSTEM display login page which includes username and password.    2. USER enters his/her username and password.    3. SYSTEM display the user’s profile 3. USER clicks EditProfike button.    1. SYSTEM displays users’S information which are name surname, address, e-mail address, and phone number    2. USER makes some changes and clicks Save Changes button.    3. SYSTEM makes changes. |
| *Entry Condition*:USER login the system and USER clicks edit profile |
| *Exit Condition*:SYSTEM makes changes. |

**Use case 8:**

|  |
| --- |
| *Use case name:* EditRestaurantProfile |
| *Participating actors:* Initiated by The RestaurantOwner |
| *Flow of events*:   1. RESTAURANTOWNER login to the site by entering his/her username and password.    1. SYSTEM displays his/her restaurant page 2. RESTAURANTOWNER clicks Edit Restaurant Profile button.    1. SYSTEM shows the restaurant’s information page    2. RESTAURANTOWNER makes some changes about his/her restaurant information    3. SYSTEM saves changes. |
| *Entry Condition*:RESTAURANTOWNER loging the system and clicks edit restaurant profile |
| *Exit Condition*:SYSTEM makes changes. |

**Use case 9:**

|  |
| --- |
| *Use case name:* RequestRegisterForm |
| *Participating actors:* Initiated by The RestaurantOwner and Admin |
| *Flow of events*:   1. RESTAURANT OWNER wants to add his/her restaurant to the www.restaurantfinder.com    1. RESTAURANTOWNER clicks Wanna Join Us? Button from main page.    2. SYSTEM displays participation form and request that the RestaurantOwner enters participation form    3. RESTAURANTOWNER fills the form which includes owner name – surname, phone number, job , e-mail address, restaurant history , restaurant category and restaurant information.    4. RESTAURANTOWNER clicks send form 2. ADMIN takes the form    1. ADMIN views the form and she/he evaluates restaurant that it can be part of restaurantfinder.com    2. ADMIN gives response to the owner. |
| *Entry Condition*:RESTAURANTOWNER clicks Wanna Join US? button |
| *Exit Condition*:RESTAURANTOWNER sends the form |

**Use case 10:**

|  |
| --- |
| *Use case name:* ApproveRestaurant |
| *Participating actors:* Initiated by The Admin |
| *Flow of events*:   1. ADMIN log in the System    1. He/she looks coming request    2. He/She evaluates the form    3. He/She clicks approve at the end of the form |
| *Entry Condition*:Admin clicks APPROVE button. |
| *Exit Condition*:SYSTEM added the restaurant |

**Use case 11:**

|  |
| --- |
| *Use case name:* RejectRestaurant |
| *Participating actors:* Initiated by The Admin |
| *Flow of events*:   1. ADMIN log in the System    1. He/she looks coming request    2. He/She evaluates the form    3. He/She clicks reject at the end of the form |
| *Entry Condition*:Admin clicks REJECT button. |
| *Exit Condition*:- |

**Use case 12:**

|  |
| --- |
| *Use case name:* AddRestaurant |
| *Participating actors:* Initiated by The Admin |
| *Flow of events*:   1. ADMIN log in the System. 2. ADMIN clicks Add Restaurant button    1. SYSTEM request that the admin creates restaurant’s profile    2. ADMIN enters restaurant information which are restaurant name, restaurant location, address, restaurant information and telephone number    3. ADMIN enters OK. |
| *Entry Condition*:Admin adds restaurant |
| *Exit Condition*:SYSTEM added the restaurant |

**Use case 13:**

|  |
| --- |
| *Use case name:* AddCategory |
| *Participating actors:* Initiated by The Admin |
| *Flow of events*:   1. ADMIN selects add new category button from his/her profile. 2. SYSTEM adds new category in to the restaurantfinder database. |
| *Entry Condition*:Admin clicks add category |
| *Exit Condition*:SYSTEM added the category |

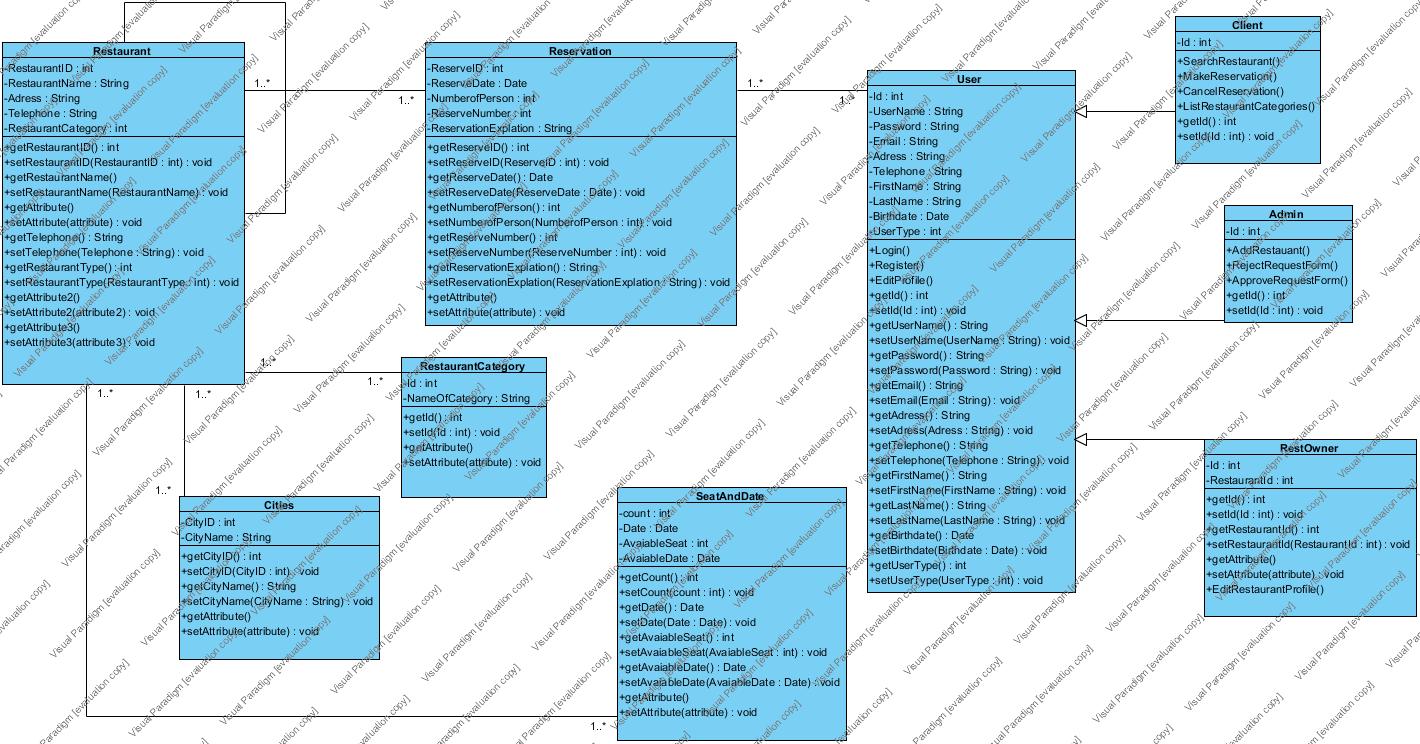
**Use case 14:**

|  |
| --- |
| *Use case name* ***:*** SelectDifferentDateTime |
| *Participating actors:* Initiated by The User |
| *Flow of events:*   1. USER makes reservation according to the restaurant which he/she wants    1. USER’s selected time not avaliable at that day.    2. SYSEM displays that the user needs to selects different time    3. USER looks avaliable time and select the time which is looks okay for him/her.   2.SYSTEM makes the reservation. |
| *Entry Condition:* The USER clicks makes reservation button. |
| *Exit Condition:* SYSEM displays “not avaliable try different time” message to the user. |

**Use case 15:**

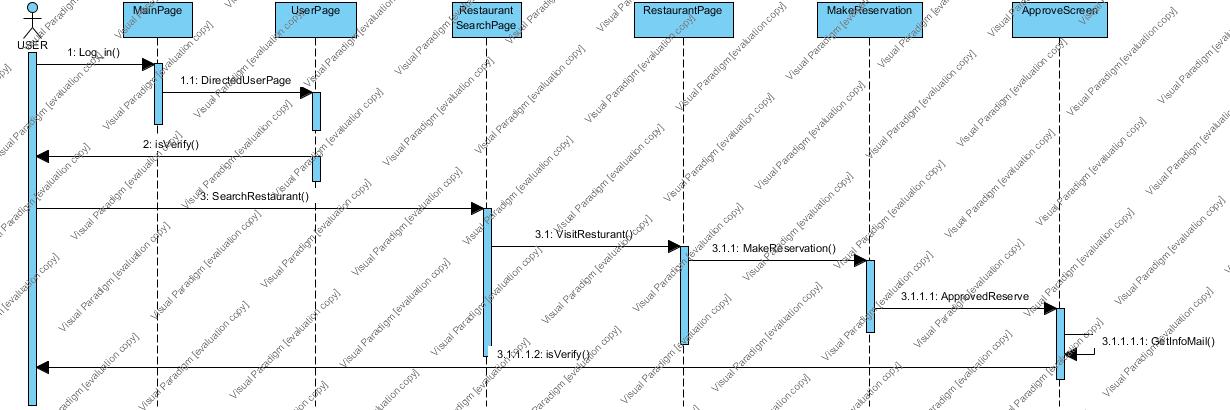
|  |
| --- |
| *Use case name:* SeeAvaliableTimes |
| *Participating actors:* Initiated by the USER |
| *Flow of events:*   1. USER makes reservation but the selected time is in full at that date 2. SYSTEM shows the selected time is not avaliable please try another message to the user. 3. USER go back to the reservation page and clicks see avaliable times. 4. USER selects the time which is okay for his/her. 5. SYSTEM makes the reservation. |
| *Entry Condition:*  The USER clicks see avaliable times |
| *Exit Condition:* SYSTEM shows avaliable times |

# Object Model

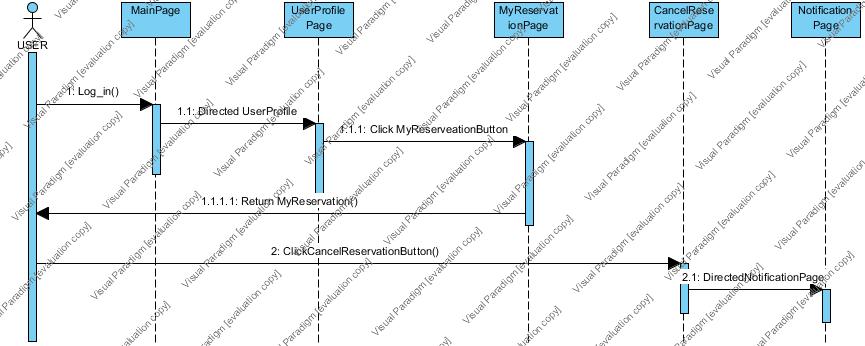
****

# Dynamic Model

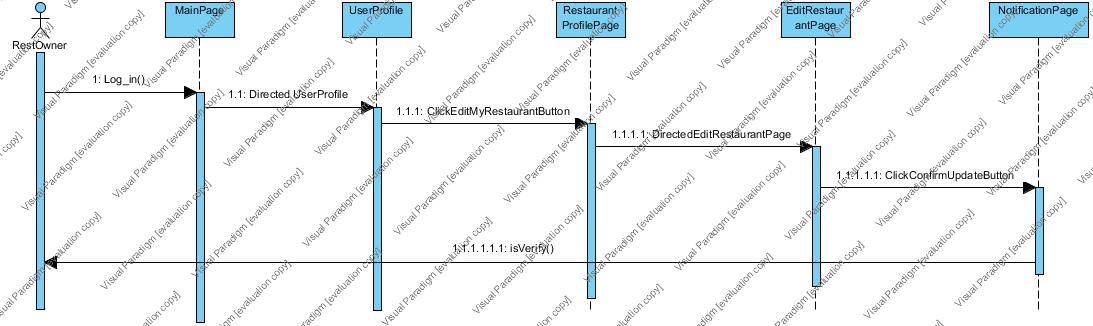
MakeReservation

****

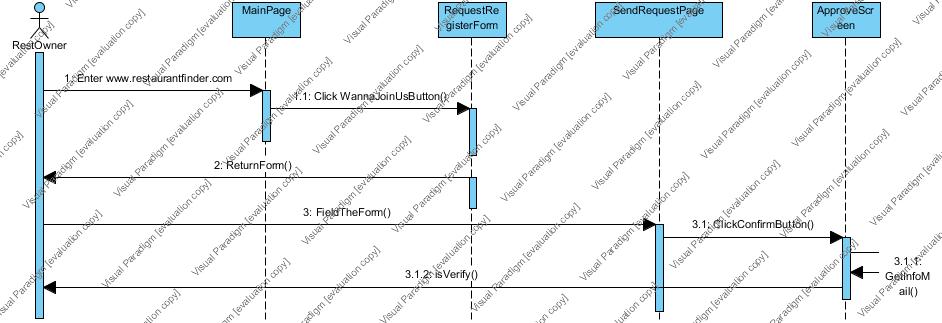
CancelReservation

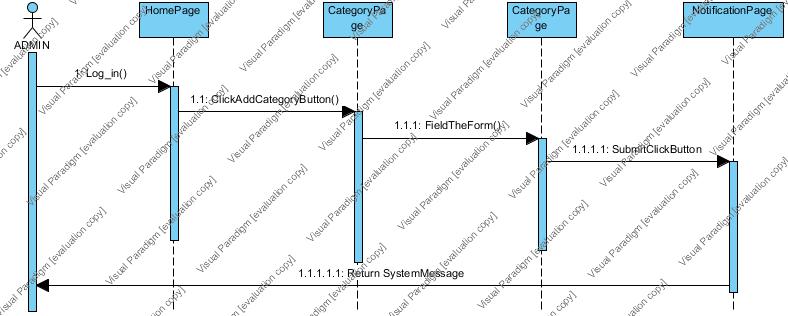


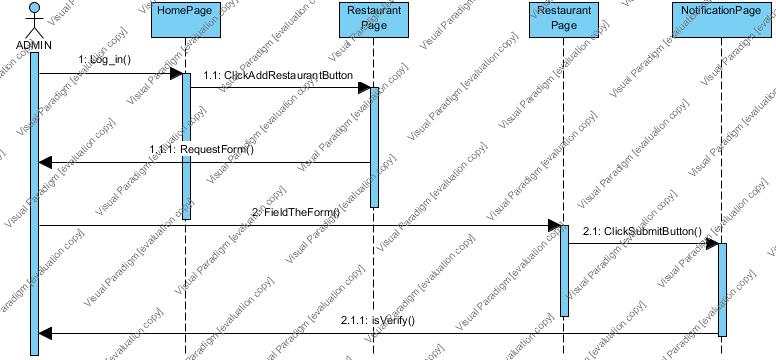
# Dynamic Model

EditRestaurantProfile

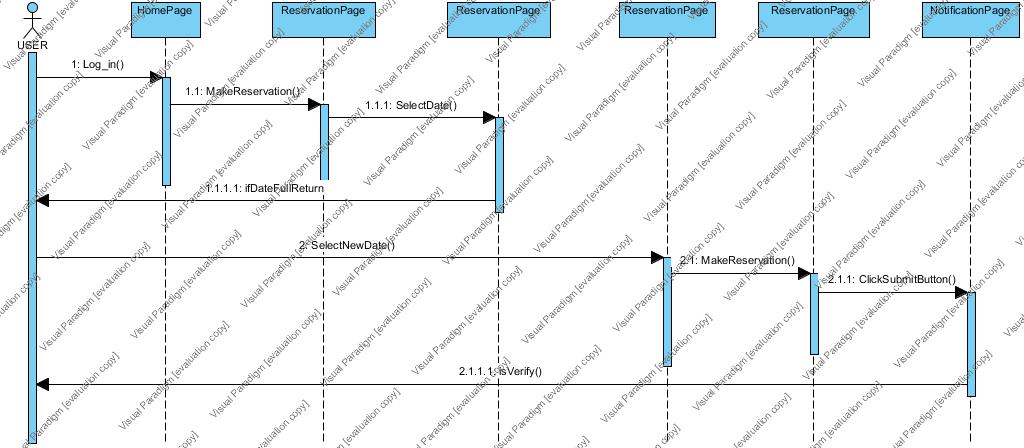
SendRequestForm



AddCategory:

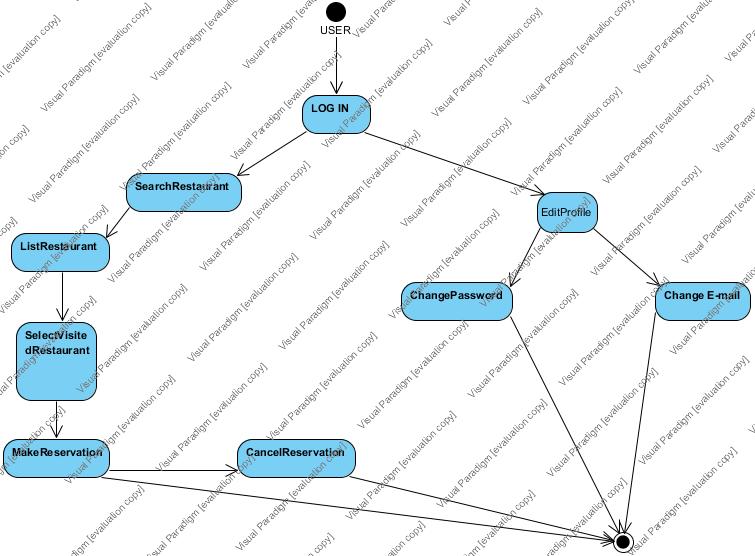
AddRestaurant:

SelectDiffrentDateTime:

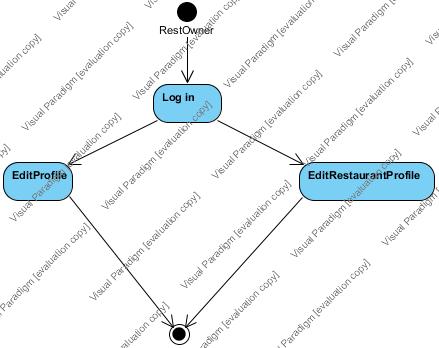


**Activity Diagram**

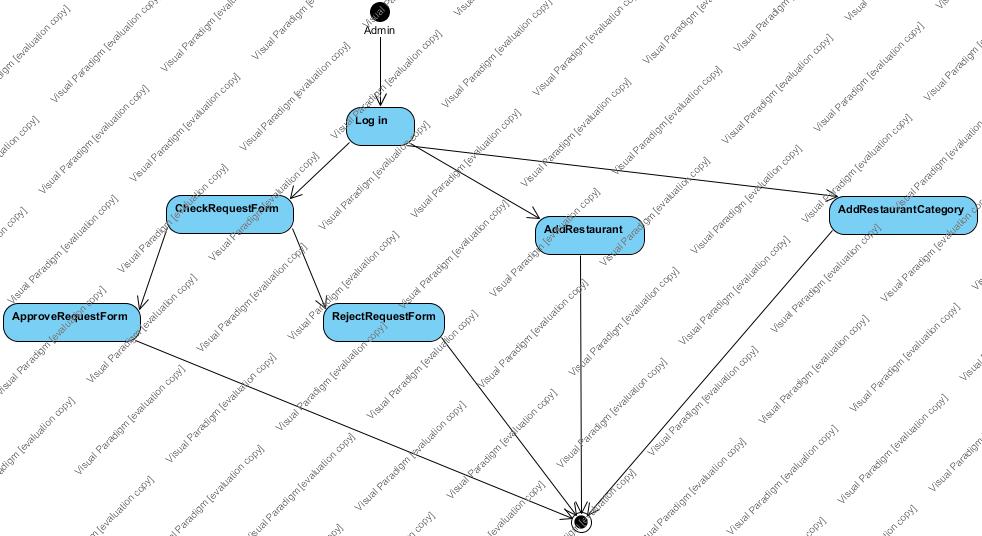
**USER**

****

**RestOwner**

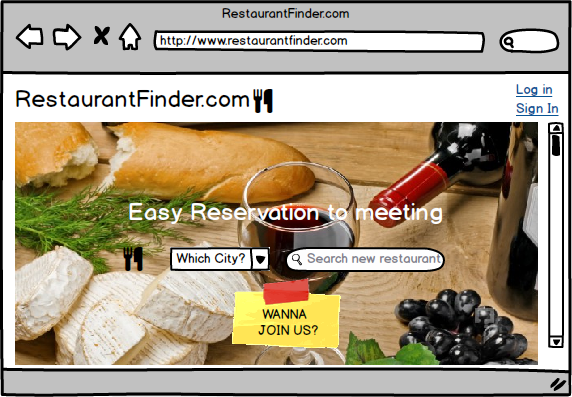
****

**ADMIN**

****

**User Interfaces-Sketches**

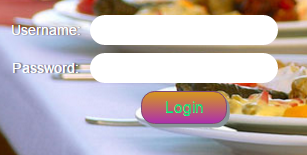
**Home Page**

**Category Page**

****

**User Interfaces-Sketches**

**Login Page**

****

# 4.Glossary

Restaurant Team : The project group’s name.

RestaurantFinder : The System’s name

Request Register Form : If restaurant owner wants to join the restaurantfinder team. He/she needs to fill register form this form called request register form.

Restaurant page: Includes all information about The Restaurant which are Restaurant name, history, phone number and contact address.

User profile: Includes all information about The User which are user name , surname, user’s phone number, user address and user e-mail address.

Login Page: Contains two text fields; one of them is used for username and the other one used for password. Also the page contains login button.

Register Page: Contains five text fields ; They are used for user name, surname , phone number, address and e-mail. At the end of this page there is also Sign in button.

Which City? : It is a combo box which includes all cities in Turkey.

World Cuisine : It includes all food from the world.

Safari : Special web browser for Iphones.

# 5.References

1. Bruegge B. & Dutoit A.H.. (2010). ​*Object­Oriented Software Engineering Using UML,* *Patterns, and Java​*,Prentice Hall, 3​rded.
2. "5 Steps to Draw a Sequence Diagram. Perf. Visual Paradigm. Youtube. Youtube, n.d. Web. <​https://www.youtube.com/watch?v=18\_kVlQMavE​>.
3. "From Class Diagram to Relation Schema." From Class Diagram to Relation Schema. N.p., n.d.Web.24Feb.2016.<​http://www2.amk.fi/digma.fi/www.amk.fi/opintojaksot/0303011/1146 161367915/1146161783414/1146163048742/1146163155917.html​>.
4. SE301 Course’s slayts added by ASSIST. PROF. EMINE EKIN
5. Innovation in Software Engineering Topic: Sequence Diagram of Hotel Management System < http://www.slideshare.net/mishraasushil/se-15862740>