SANGFood

Requirements Specification and Analysis

1.0

20.11.2016

Ayşe Naz Aydemir

Gamze Saraç

Nazlı Akdağ

Selen Kozanoğlu

Prepared for

SE301 Software Engineering



Table of Contents

[1. Introduction 1](#_Toc431983052)

[1.1. Purpose of the System 1](#_Toc431983053)

[1.2. Scope of the System 1](#_Toc431983054)

[1.3. Objectives and Success Criteria of the Project 1](#_Toc431983055)

[1.4. Definitions, Acronyms, and Abbreviations 1](#_Toc431983056)

[1.5. Overview 1](#_Toc431983057)

[2. Current System 1](#_Toc431983058)

[3. Proposed System 1](#_Toc431983059)

[3.1. Overview 1](#_Toc431983060)

[3.2. Functional Requirements 2](#_Toc431983061)

[3.3. Nonfunctional Requirements 2](#_Toc431983062)

[Usability 2](#_Toc431983063)

[Reliability 2](#_Toc431983064)

[Performance 2](#_Toc431983065)

[Supportability 2](#_Toc431983066)

[Implementation 2](#_Toc431983067)

[Interface 2](#_Toc431983068)

[Packaging 2](#_Toc431983069)

[Legal 2](#_Toc431983070)

[3.4. System Models 2](#_Toc431983071)

[Scenarios 2](#_Toc431983072)

[Use case model 2](#_Toc431983073)

[Object model 2](#_Toc431983074)

[Dynamic model 2](#_Toc431983075)

[User interface—navigational paths and screen mock-ups 3](#_Toc431983076)

[4. Glossary 3](#_Toc431983077)

[5. References 3](#_Toc431983078)

REQUIREMENTS ANALYSIS DOCUMENT[1]

# 

# Introduction

## Purpose of the System

Purpose of the system is to order food easily and quickly from more than one restaurant at the same time. On the other hand, to give an order without going to restaurant. Also purpose of the system includes providing some functionalities to restaurant owner so that they can manage their own page.

## Scope of the System

The System provides service to user, restaurant owner, admin, manager. There are some functionalities for both RestaurantOwner, user, admin and manager. some of them are different too.

The services that the system provides to user is; the user can login own account to edit profile that sets address information. Then they can login own account that contains food and restaurant they are supposed to give an order. They can add which select food to basket and also delete too. In addition, they can search which restaurant is near to them or which food what they want to eat.

The service that the system provides to RestaurantOwners; The RestaurantOwner can send restaurantform to enter the system. Then the RestaurantOwner can login own account to edit restpage that puts the foods menu, also the RestaurantOwner can remove food from the menu and close the order. In addition, the RestaurantOwner can take order which is accepted.

The service that the system provides to admin; The admin can login to system to create RestaurantOwner account when manager accept to restaurant than admin will create account.

The service that the system provides to manager; The manager can login system to accept restaurantform, restaurant owner send form to manager and manager can accept restaurantform and add to the system, or reject the form.

## Objectives and Success Criteria of the Project

The objectives and success criteria of the project is to be usability and saving time that means; shop online instead of going out to buy food. User accesses lots of restaurant at the same time to select food. So this is accessibility for restaurants.

## Definitions, Acronyms, and Abbreviations

RAD: Requirements Analysis Document

RestaurantOwner: Restaurant Owner

UI: User Interface

## Overview

Our requirement analysis document contains Current System section, Proposed System section, overview of SANGFood. Functional requirements, Nonfunctional requirement section, system model section, Object model section, Dynamic Model section and Glossary.

In Current system section of our RAD documentation, we explained about system functions and features of new SANGFood system.

SANGFood has a friendly interface and design to be short response time. All functions provide do task quickly and easy payment system.

In Proposed System section, we talked about our SANGFood system advantages and lates developments. Thus giving an order is very easy by using our system on the other hand the system is very efficient and provide easy payment.

In overview section, we defined overview of features, functions and details of SANGFood.

In Functional requirements section, we talked about functions of SANGFood and also we defined functions on the part of user, RestaurantOwner, admin and manager.

In Nonfunctional requirements we described nonfunctional part of SANGFood such as usability, performance, reliability or availability.

In System model section, we described scenarios and use cases of SANGFood. We defined scenarios, actor, and use cases’ flow events.

In Object model section, we explained class diagrams of SANGFood and we define relationship with classes of my system.

In dynamic model section, we mentioned sequence diagram that is to say we explained methods and functions and their operations with actors of my food.

In Glossary section, we described all things of our system. it is like a dictionary. we explained all words that we used in RAD documentation.

# Current System

SANGFood is an online order system. It is developed for usage purpose of the all people who access and login to website, and RestaurantOwner. On user’s perspective, user can see and search all restaurant and select one of them, user can add food that is selected, on the basket. Also user can remove food on the basket. user can advanced search such as place, name.

On RestaurantOwner’s perspective, RestaurantOwner can send restaurant form to have login on the system, so RestaurantOwner can edit restaurant page such as close order, edit meal. RestaurantOwner can take order which is accepted.

On manager’s perspective, manager can login system. Manager can accept and reject restaurant request form. then manager can add restaurant on the system. manager can control order and then accept or decline and go throw RestaurantOwner system.

On admin’s perspective, admin can login system and create restaurant account.

finally, SANGFoog has a lots of functions and properties for using easy.

# Proposed System

According to the current food order, online food ordering system that will be more efficient and easy for new users. For example, customers with the new system the menus of all the restaurants view all restaurants in can easily order from the closest place to their present address. Restaurants can register to the site, edit the menus, customers can send their orders by using the system easily. In this way, customers, wherever they are, they want the restaurants they want to dine on the dishes easily. Users perform tasks with fewer clicks and movements than using online food Ordering you will be able to. In addition, the interaction between users and online food ordering, it will be better than before. Interface will be improved with new technologies, so that users would be impressive.

## Overview

SANGFood is food ordering system for users. Customers registering with the system you can view all the restaurants and all they want in the city. Their address is also in close proximity to all the restaurants they want can order from the menu. In order to make changes or additions to their menu if they want to, they can. After the arrival of orders, you are able to pay at the gate. Restaurants can register by submitting the form to the site. Can edit their own pages. May make changes to inventory. You can view the orders that sends customers. The manager of the restaurant accepts or rejects the registration form sent to the site displays. If the system adds to your restaurant. Do you want to reject the restaurant's page also displays the changes Admin creates users who want to register a new account.

## Functional Requirements

SANGFood is an online food order platform for all people and customers. For Customers, SANGFood can list all restaurant in live city, show different types of restaurant, select food and add basket, then give food order, pay at the door or with their credit card, order is made to the address by using SANGFood. In addition, they can register, login, edit profile change password and update their profiles by using SANGFood.

For RestaurantOwner can send restaurant form for display them of customers, they can be login then accept form and create account. RestaurantOwner can edit Restaurant page. They can be take order and accept order of customers by using SANGFood.

For Manager, login as manager in SANGFood, show form of the wanted to be added to the site restaurants. Then, accept restaurant and add restaurant or reject restaurant.

For Admin, login as admin in OnlineFoodOrdering, create account for to be added to the restaurants.

The main and important function of OnlineFoodOrdering is give food order, pay at the door or with credit card. Customers arrive at their address of food orders.

## Nonfunctional Requirements

**Usability**

* + Customers should make 7 clicks to make order to our system which including login and search part for clicks
  + Customers need to register first to use anything on our system.
  + The UI must be user-friendly.
  + The system only usable via web.

**Reliability**

* + Our system must available at any time on web browsers.
  + The system must be running 90% of the time.
  + The system must never crash.

**Performance**

* The system must allow at least 500 users at the same time.
* The system update basket within a few seconds.
* System should response less than 1 second.
* All queries must return a response in less than 3 seconds.

**Supportability**

* + The system also usable from mobile application.
  + The system must be easily to maintenance.
  + The security system must be upgradable.
  + The system must be 100% English.

**Implementation**

* The system must be controlled every month to maintenance.
* There is no need to an install program to use.
* The maintenance group must be able to maintain the software.

**Interface**

* System has user-friendly interface to use.
* All actors can use our system so easily.

**Packaging**

* There is no need to any building, installing to computer.
* All users should use our system via web browser.

**Legal**

* Copyrighted by www.SANGfood.com

## System Models

### **Scenarios**

**Scenario 1**

|  |
| --- |
| *Scenario Name:* MakeOrder |
| *Participant actor instances:* Selen:User |
| *Flow of events:*   1. Selen feels hungry. She wants to eat chicken wings but she feels lazy to go out. She wants to eat at her home. 2. She decided to make an order online from the site [www.sangfood.com](http://www.sangfood.com). She opens her computer and enters the site url to the browser. 3. She made an order before from the site , and that’s why she has a account. 4. She enters her e-mail address and password to login the site. 5. She lives in Suadiye she selects Suadiye and enters chicken wings to the search area. 6. She looks results and sees Baldır XL Caddebostan in search results, she decides to make an order from Baldır XL. 7. She chooses Piri Piri Chicken Wings and makes order by adding basket. |

**Scenario 2:**

|  |
| --- |
| *Scenario Name:* EditItems |
| *Participant actor instances:* Nazlı:User |
| *Flow of events:*   1. Nazlı works in a company. She is back home late and she is always busy. 2. She mostly choose ready to serve food and she always makes order by using [www.sangfood.com](http://www.enjoyyourmeal.com) 3. She login to the site and wants to eat hamburger from Burger King 4. She lives in Altıntepe and always makes order from Bostancı Burger King 5. She enters Burger King to the search area and chooses the closest place to her from the search result which is Bostancı Burger King 6. She Chooses Whopper Menu from Burger King’s page. 7. She also chooses extra materials; standard size as a size of the menu, mustard and ketchup as a extra sauces and coke as a selection of drink. 8. After she adds basket the menu that she wants to eat. 9. Before making an order she changes her mind and she changes sauce as a mayonnaise in order to mustard by going back to the basket and changes the items. |

**Scenario 3:**

|  |
| --- |
| *Scenario Name:* ListRestaurant |
| *Participant actor instances:* Faruk:User |
| *Flow of events:*   1. Faruk invites his friend to his home. He watches Galatasaray – Fenerbahçe derby with his friends. 2. Faruk and his friends wants to eat pizza while they are watching the match. 3. Faruk decides to make an order online. 4. His friend Tanju recommends the online food ordering site name [www.sangfood.com](http://www.enjoyyourmeal.com). 5. Faruk enters the site name to the browser 6. He enters pizza to the search area by selecting city İstanbul and Feneryolu as place. 7. He selects Dominos Pizza Feneryolu and they decides to chooses 3 Cazip Pizza and enters add basket 8. He enters make an order. 9. He has not have account to make an order, and also his friend Tanju does not remember his password. 10. Faruk has to register to the site. He enters sign up. 11. He fills the registration form by entering his e-mail as a faruk@hotmail.com, name-surname as a Faruk Kozanoğlu, password ,birth of date –year as 07.10.1957, place as a Feneryolu. 12. He enters sign up, and then he finishes his order. |

**Scenario 4:**

|  |
| --- |
| *Scenario Name:* SelectPurchase |
| *Participant actor instances:* Gamze:User |
| *Flow of events:*   1. Gamze lives in İstanbul. She works in a company. 2. She finishes the day of work and stuck in a heavy traffic when she is in the way of her home Küçükyalı. 3. She feel really hungry she can not wait for the meal. 4. She decides to make an order from [www.sangfood.com](http://www.enjoyyourmeal.com) system. 5. She makes an order by using her cell phone when she is in the heavy traffic she estimates the time to reach her home as 20 minutes, so when she comes home the order probably comes. 6. She logs in to system and decides to make an order from Bizim Dürüm in Bostancı which close enough to her home. 7. She selects the menu Kaşarlı Adana Dürüm with coke 8. She presses Add Basket and enters Approve Basket. 9. She has a 2 address in the system one of them is her work address the other one is a home address. 10. She selects home address as order address and selects paying at the door by using Credit Card as made of payment. 11. She finishes her order by clicking make order. |

**Scenario 5:**

|  |
| --- |
| *Scenario Name:* SendRestaurantForm |
| *Participant actor instances:* Ayşenaz:RestaurantOwner |
| *Flow of events:*   1. Ayşenaz is a agent of Polonez Restaurant. 2. She wants to add her restaurant to the [www.sangfood.com](http://www.enjoyyourmeal.com) system. 3. She enters the site name to the browser, and enters suggest restaurant from the main page. 4. She sees the restaurant suggestion form page , which includes personal info and restaurant info. 5. She fills her personal info that are Ayşenaz as a name Aydemir as surname and and personal phone number. 6. She also fills the restaurant info by entering restaurant information ; Owner of the restaurant , name of the restaurant , telephone number , place and explanation. 7. She sends the form by clicking send. |

**Scenario 6:**

|  |
| --- |
| *Scenario Name:* AcceptRestaurant |
| *Participant actor instances:* Seda: Manager |
| *Flow of Events:*   1. Seda logs in to the system and looks for incoming restaurants. 2. She reviews the restaurant Polonez Restaurant which is send by Ayşenaz Aydemir. 3. She checks all information about the restaurant and can not find anything negative. 4. She decides to add the restaurant to the SangFood system. 5. She clicks add button and the restaurant added. |

**Scenario 7:**

|  |
| --- |
| *Scenario Name:* EditRestaurantPageStock |
| *Participant actor instances:* Büşra: RestaurantOwner |
| *Flow of Events:*   1. Büşra is a owner of the Polonez Restaurant 2. She logs in to the system 3. She wants to edit the restaurant page. 4. The Polonez Restaurant does not have enough burrito at the time. 5. She wants to change the stock of the burrito and brings the limitation for it 6. She clicks edit button and make it’s limitation 5. |

**Scenario 8:**

|  |
| --- |
| *Scenario Name:* EditRestaurantPageOrder |
| *Participant actor instances:* Büşra: RestaurantOwner |
| *Flow of Events:*   1. Büşra has a restaurant named Polonez Restaurant. 2. The week of “eid” , she wants to close the restaurant’s order for weeklong all afternoon. 3. She logs in to the system. 4. She clicks “Edit Restaurant Page” and choose “Close Restaurant” 5. She enters “Save Changes” and finishes her work with the system by clicking logout. |

### **Use case model**

### 

**Use Case 1:**

|  |
| --- |
| *Use case name:* Login |
| *Participant actors:* CUSTOMER, OR Admin, OR RestaurantOwner OR Manager |
| *Flow of events:*   1. CUSTOMER enters the site name on the browser.   2.SYSTEM shows the main page which includes login, sign in, and food searching area according to city that user choice.   1. CUSTOMER selects the restaurant he/she would like to make order.   3.The SYSTEM requests that the user enter his/her name and password.  4. The CUSTOMER enters his/her name and password.  5.The System validates the entered name and password and logs the user into the  System. |
| *Entry Condition:* The CUSTOMER enters to login screen. |
| *Exit Condition:* The CUSTOMER is logged in, OR,  The CUSTOMER has received an explanation indicating why he/she could not  login |

**Use Case 2:**

|  |
| --- |
| *Use case name:* ForgetPassword |
| *Participant actors:* CUSTOMER, OR Admin, OR RestaurantOwner OR Manager |
| *Flow of events:*  1. CUSTOMER enters the site name on the browser.  2 .SYSTEM shows the main page which includes login, sign in, and food searching area according to city that user choice.  3. CUSTOMER enters login  4. SYSTEM displays user name and password area.  5. CUSTOMER enters user name and password.  6. SYSTEM shows the message “Invalid username or password”  7. CUSTOMER enter “Forget Password?”  8. SYSTEM displays forget password page which includes e-mail area and send button.  9. CUSTOMER enters his/her e-mail.  10. SYSTEM sends the password to the e-mail. |
| *Entry Condition:* CUSTOMER clicks “Forget Password?” |
| *Exit Condition:* SYSTEM sends a password to USER’s entered e-mail address. |

**Use Case 3:**

|  |
| --- |
| *Use case name:*  Register |
| *Participating actors:* Initiated by CUSTOMER |
| *Flow of events*:   1. CUSTOMER enters www.sangFood.com on the browser.   2.SYSTEM shows the main page which includes restaurant search area, login and sign in button.  3. CUSTOMER enters the sign in button..   4.The SYSTEM displays register page.   1. CUSTOMER enters his/her information to the register page.   6.The SYSTEM request that the user enter his/her hane, surname, e-mail address and password.  7. The CUSTOMER enter his/her name, surname, e-mail, address and password.  8.The SYSTEM register the entered name, surname, e-mail address and password. |
| *Entry Condition*:CUSTOMER enters sing in button. |
| *Exit Condition*:SYSTEM registers |

**Use Case 4:**

|  |
| --- |
| *Use case name:*  EditProfile |
| *Participant actors:* User, OR Admin, OR RestaurantOwner OR Manager |
| *Flow of events*:   1. CUSTOMER enters [www.sangfood.com](http://www.enjoyyourmeal.com) on the browser.   2. SYSTEM display main page.  3. CUSTOMER click login button.  4. SYSTEM display login page which includes username and password.   1. CUSTOMER enters his/her username and password.     6. SYSTEM display the user’s profile  7. CUSTOMER clicks Edit Profile button.    8. SYSTEM displays users ’s information which are name surname, address, e-mail  address, and phone number.  9. CUSTOMER makes some changes and clicks Save Changes button.  10. SYSTEM makes changes. |
| *Entry Condition*: CUSTOMER login the system and USER clicks edit profile |
| *Exit Condition*:SYSTEM makes changes. |

**Use Case 5:**

|  |
| --- |
| *Use case name:* ChangeCity |
| *Participant actors:* Initiated by the CUSTOMER |
| *Flow of events*:   1. CUSTOMER logs into the system   2.SYSTEM displays the user’s page.  3. CUSTOMER wants to selects another city to search and clicks change city button.  4. SYSTEM returns all restaurants according to the city that he/she choose. |
| *Entry Condition:* CUSTOMER clicks “Change City” |
| *Exit Condition*: SYSTEM returns restaurants |

**Use Case 6:**

|  |
| --- |
| *Use case name:*  ListRestaurant |
| *Participant actors:* Initiated by Customer |
| *Flow of events*:  1. CUSTOMER enters the site name to the browser    2. SYSTEM displays main page.  3. CUSTOMER selects place and selects restaurant that he/she wants to order    4. SYSTEM displays all restaurant which are at the same place  5. CUSTOMER selects the restaurant which is closest place to his/her.    5. SYSTEM shows the restaurant menu page. |
| *Entry Condition*: CUSTOMER selects the place |
| *Exit Condition*: SYSTEM returns restaurants |

**Use Case7:**

|  |
| --- |
| *Use case name:* SelectFood |
| *Participant actors:*  Initiated by the CUSTOMER |
| *Flow of events*:   1. CUSTOMER logs in to the system.   2. SYSTEM shows user’s page.  3. CUSTOMER selects place and search for a food which he/she wants to order.    4. SYSTEM displays the results(food or Menu) according to the customer’s search  5. CUSTOMER selects the food which he/she want to order.    6. SYSTEM displays the food page with its additional properties.    7. CUSTOMER enters/selects the properties and clicks “Add Basket”.    8. SYSTEM adds the food to the basket. |
| *Entry Condition:*  CUSTOMER logs into the system and searches for food |
| *Exit Condition*: SYSTEM adds the food and shows the message “The food added basket successfully”  OR  “The food is not available right now, please select another” message. |

**Use Case 8:**

|  |
| --- |
| *Use case name:* AddBasket |
| *Participant actors:* Initiated by Customer |
| *Flow of events*:  1. CUSTOMER logs into the SangFood system.  2.SYSTEM displays main page.  3. CUSTOMER selects the place that she/he wants to order.  4. SYSTEM displays all restaurants which belongs the place that the user search.  5. CUSTOMER chooses the restaurant that he/she wants to order from.    6. SYSTEM displays the restaurant’s menu page  7. CUSTOMER chooses the food from the restaurant’s menu page.    8. SYSTEM display the food’s page which includes what it includes , what the user  wants to add in it etc.  9. CUSTOMER clicks add basket.    10. SYSTEM display the basket with the total price. |
| *Entry Condition*: The CUSTOMER is logged into the SANGFOOD SYSTEM.  The customer selects the food and then clicks “Add Basket” button. |
| *Exit Condition*: The CUSTOMER sees the food in his/her basket and clicks “Approve Basket”. |

**Use Case 9:**

|  |
| --- |
| *Use case name:* DiscardBasket |
| *Participant actors:* Initiated by Customer |
| *Flow of events*:  1. CUSTOMER clicks “ Add Basket” button  2. SYSTEM displays the basket with prices and the items in it.  3. CUSTOMER changes his/her mind and wants to order another food and clicks “Discard  Basket”  4. SYSTEM empties the basket. |
| *Entry Condition*: CUSTOMER clicks “Discard Basket” |
| *Exit Condition*: SYSTEM empties the basket. |

**Use Case 10:**

|  |
| --- |
| *Use case name:* EditItems |
| *Participant actors:* CUSTOMER |
| *Flow of events*:  1. CUSTOMER selects the food and with the what he/she wants with it and clicks “Add  Basket”.  2. SYSTEM displays the basket.  3. CUSTOMER wants to change the food’s item which he/she wants to come with his/her  food.  4. SYSTEM displays the food’s page with materials  5. CUSTOMER changes the material again clicks “Add Basket”  6. SYSTEM adds the updated food to the basket and displays it. |
| *Entry Condition*: CUSTOMER clicks “Add Basket” |
| *Exit Condition*: CUSTOMER clicks “Approve Basket” |

**Use Case 11:**

|  |
| --- |
| *Use case name:* MakeOrder |
| *Participant actors:* Initiated by Customer |
| *Flow of events*:  1. CUSTOMER logs into the SangFood system.  2. SYSTEM displays the main page.  3. CUSTOMER selects the restaurant and the food which he/she wants to order, and clicks  “Add Basket”  4. SYSTEM displays the basket with “Approve Basket” button.  5. CUSTOMER clicks “Approve Basket” button.  6. SYSTEM displays the purchase page with address information ,the user information  and made of payment.  7. CUSTOMER selects the made of payment and fills the other needs for necessary for  making an order , and clicks “Make Order”.  8. SYSTEM receive the order. |
| *Entry Condition*: CUSTOMER clicks “Approve Basket” |
| *Exit Condition*: CUSTOMER sees Your Order is Received OR  The Customer has received an explanation indicating why the order could not be bought.  System extends like timeout |

**Use Case 12:**

|  |
| --- |
| *Use case name:* CancelOrder |
| *Participant actors:* Initiated by The CUSTOMER |
| *Flow of events*:   1. CUSTOMER logs into the system.   2. SYSTEM displays the user’s page.  3. CUSTOMER makes order by doing all steps to making an order.  4. SYSTEM takes the order.  5. A few minutes later CUSTOMER wants to cancel his/her order and enters “My Orders”.  6. SYSTEM displays the active and non-active order.  7. CUSTOMER enters the active order and clicks “Cancel”.  8. SYSTEM deletes the order from the system. |
| *Entry Condition*: CUSTOMER logs into the system and clicks “Cancel” |
| *Exit Condition*: SYSTEM cancels the order. |

**Use Case 13:**

|  |
| --- |
| *Use case name:* SelectPurchase |
| *Participant actors:* Initiated by The CUSTOMER |
| *Flow of events*:   1. CUSTOMER logs in to the system.   2. SYSTEM displays the user page.  3. CUSTOMER selects place and restaurant.  4. SYSTEM displays foods which belongs the restaurant.  5. CUSTOMER selects the food that he/she wants to order and clicks “Add Basket”.  6. SYSTEM adds the food to the basket and displays it.  7. CUSTOMER clicks “ Approve Basket”  8. SYSTEM displays the made of payment.  9. CUSTOMER selects the purchase type that he/she want to make.  10. SYSTEM wants to user fills the credit card information if the made of payment is not  paying at the door  11. CUSTOMER enters credit card no and type and enters “Make Order”.  12. SYSTEM takes the order. |
| *Entry Condition*: CUSTOMER logs in to the system AND clicks “Approve Basket” |
| *Exit Condition*: SYSTEM displays “Your Order is taken successfully” message OR  “Timeout” OR “No Limit” messages. |

**Use Case 14:**

|  |
| --- |
| *Use case name:* SendRestaurantForm |
| *Participant actors:* Initiated by t RESTAURANTOWNER |
| *Flow of events*:   1. USER / RESTAURANTOWNER enters the system url to the browser.   2. SYSTEM displays main page.  3. USER / RESTAURANTOWNER clicks “Suggest Restaurant”.  4. SYSTEM displays empty form which includes personal information the restaurant  information’s area.  5. USER / RESTAURANTOWNER fills the form and clicks “Send”  6. SYSTEM displays the message “Thank you for your suggestion”. |
| *Entry Condition*: USER / RESTAURANTOWNER clicks “Suggest Restaurant” |
| *Exit Condition*: SYSTEM takes the form and display the message . |

**Use Case 15:**

|  |
| --- |
| *Use case name:*  TakeOrder |
| *Participant actors:* Initiated by the RestaurantOwner |
| *Flow of events*:   1. RESTAURANTOWNER logs in to the system.   2. SYSTEM displays restaurant page.  3. RESTAURANTOWNER sees incoming orders and takes them  4. SYSTEM returns the message “Order is given” |
| *Entry Condition:* RESTAURANTOWNER logs into the system and takes an order |
| *Exit Condition*: SYSTEM returns the message “Order is given” |

**Use Case 16:**

|  |
| --- |
| *Use case name:* EditRestaurantPage |
| *Participant actors:* Initiated by the RESTAURANTOWNER |
| *Flow of events*:   1. RESTAURANTOWNER logs in to the system by using restaurant login.   2. SYSTEM displays Restaurant page.  3. RESTAURANTOWNER clicks “Edit Restaurant Profile”  4. SYSTEM displays the profile info  5. RESTAURANTOWNER makes some changes and clicks “Save Changes”.  6. SYSTEM makes changes. |
| *Entry Condition:* RESTAURANTOWNER logs in to the system AND,  Clicks “Edit Restaurant Profile” |
| *Exit Condition*: RESTAURANTOWNER clicks “Save Changes” AND  SYSTEM returns “Your changes made successfully” message. |

**Use Case 17:**

|  |
| --- |
| *Use case name:* AcceptRestaurant |
| *Participant actors:* Initiated by the Manager |
| *Flow of events*:   1. MANAGER logs in to the system.   2. SYSTEM displays Manager’s page.  3. MANAGER clicks “See Incoming Restaurant’s Form”  4. SYSTEM shows all incoming Restaurant’s Form.  5. MANAGER reviews the forms and clicks “Approve the Restaurant”  6. SYSTEMS sends the approved restaurant to the Admin. |
| *Entry Condition:* MANAGER logs in to the system  MANAGER clicks “See Incoming Restaurant’s Form” |
| *Exit Condition*: SYSTEM returns the message “The Restaurant successfully sended ” |

**Use Case 18:**

|  |
| --- |
| *Use case name:* RejectRestaurant |
| *Participant actors:* Initiated by the Manager |
| *Flow of events*:  1. MANAGER logs in to the system.  2. SYSTEM displays Manager’s page.  3. MANAGER clicks “See Incoming Restaurant’s Form”  4. SYSTEM shows all incoming Restaurant’s Form.  5. MANAGER reviews the forms and clicks “Reject the Restaurant”  6. SYSTEM displays the message “The restaurant successfully rejected”. |
| *Entry Condition:* MANAGER logs in to the system  MANAGER clicks “See Incoming Restaurant’s Form” |
| *Exit Condition*: SYSTEM returns the message “The restaurant successfully rejected”. |

**Use Case 19:**

|  |
| --- |
| *Use case name:* CreateAccount |
| *Participant actors:* Initiated by the Admin |
| *Flow of events*:   1. ADMIN logs into the system.   2. SYSTEM displays Admin Page.  3. ADMIN clicks “See Coming Restaurants” and clicks “Add Restaurant”.  4. SYSTEM display Adding Restaurant Page.  5. ADMIN enters the restaurant information and enters create and account for it.  6. SYSTEM adds the restaurant to the system restaurants. |
| *Entry Condition:* ADMIN logs into the system AND  Clicks “Add Restaurant” |
| *Exit Condition*: SYSTEMS add the restaurants |

**Use Case 20:**

|  |
| --- |
| *Use case name:* SearchFoodByCity |
| *Participant actors:* Initiated by CUSTOMER |
| *Flow of events*:   1. CUSTOMER enters the site url to the browser.   2. SYSTEM displays the main page.  3. CUSTOMER enters change city button  4. SYSTEM displays all cities.  5. CUSTOMER selects the city that he/she wants to search  6. SYSTEM shows the places and restaurants which belongs the city |
| *Entry Condition:* CUSTOMER enters change city |
| *Exit Condition*: SYSTEM returns places and restaurants |

**Use Case 21:**

|  |
| --- |
| *Use case name:* SearchFoodByPlace |
| *Participant actors:* Initiated by CUSTOMER |
| *Flow of events*:   1. CUSTOMER enters the site url to the browser.   2. SYSTEM displays the main page.  3. CUSTOMER selects the place that he/she wants search  4. SYSTEM returns all restaurant according to the place that he/she select |
| *Entry Condition:* CUSTOMER selects the place |
| *Exit Condition*: SYSTEM returns all restaurants |

**Use Case 22:**

|  |
| --- |
| *Use case name:* SearchFoodByFoodType |
| *Participant actors:* Initiated by USER |
| *Flow of events*:   1. CUSTOMER enters the site url to the browser.   2. SYSTEM displays the main page.  3. CUSTOMER enters the food name / type to the search area  4. SYSTEM returns all restaurants which has the food that he/she wants |
| *Entry Condition:* CUSTOMER enters food name/type |
| *Exit Condition*: SYSTEM returns all restaurant according the food. |

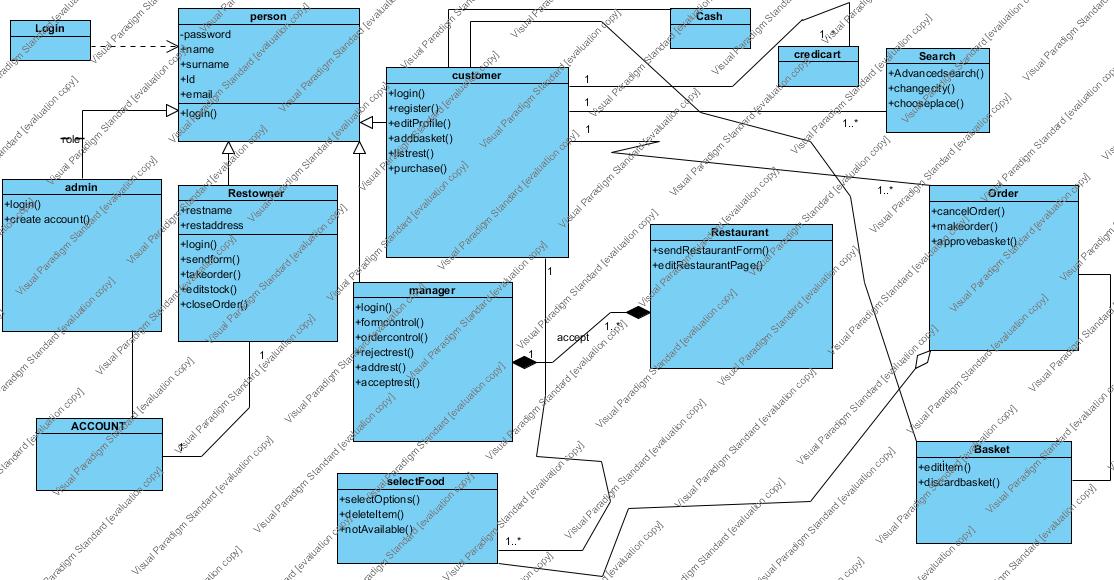
**Use Case 23:**

|  |
| --- |
| *Use case name:* CloseOrder |
| *Participant actors:* Initiated by RestaurantOwner |
| *Flow of events*:   1. RESTAURANT OWNER logs into the system.   2. SYSTEM displays Restaurant Page.    3. RESTAURANT OWNER clicks “Edit Restaurant Profile”    4. SYSTEM displays editable Restaurant Profile’s Page.    5. RESTAURANT OWNER clicks “Close Order” and enters “Save Changes”.    6. SYSTEM closes orders for the restaurant. |
| *Entry Condition:*  RESTAURANT OWNER logs into the system AND  Enters “Edit Profile” |
| *Exit Condition*: SYSTEM closes orders. |

**Use Case 24:**

|  |
| --- |
| *Use case name:* EditStock |
| *Participant actors:* Initiated by the RestaurantOwner |
| *Flow of events*:   1. RESTAURANT OWNER logs into the system.   2. SYSTEM displays Restaurant Page.  3. RESTAURANT OWNER clicks “Edit Restaurant Profile”    4. SYSTEM displays editable Restaurant Profile’s Page.  5. RESTAURANT OWNER makes changes in stocks by entering new stock value and clicks “Save  Changes”.  6. SYSTEM makes changes. |
| *Entry Condition: :*  RESTAURANT OWNER logs into the system AND  Enters “Edit Profile” |
| *Exit Condition*: SYSTEM makes changes. |

### **Object model**



Entity Boundary Objects

-Login:

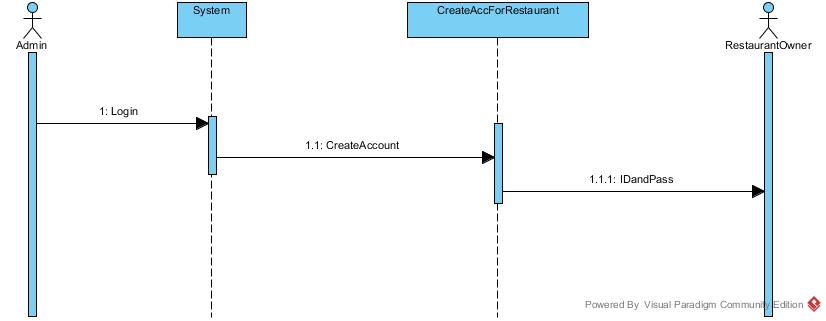
****

**-**Register:

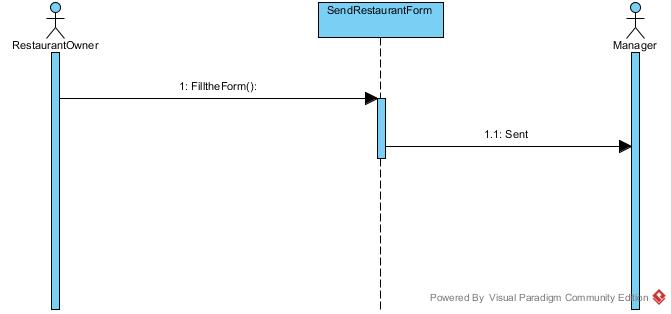
****

### **Dynamic model**

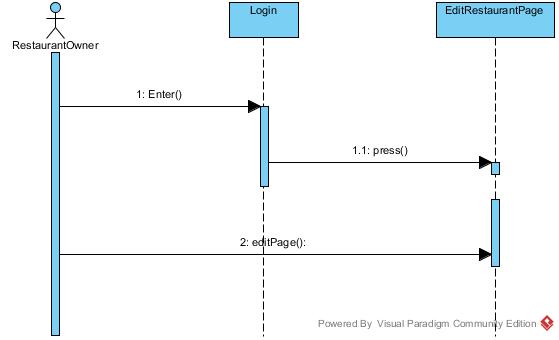
### -Admin:



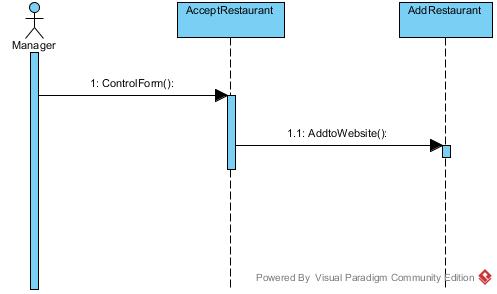
*-RestOwner:*

**

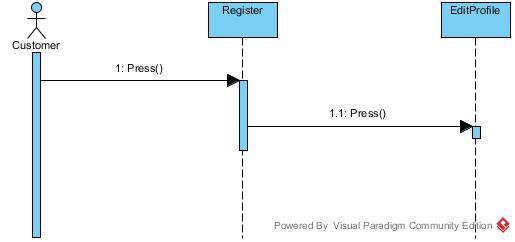
*-RestOwner:*

**

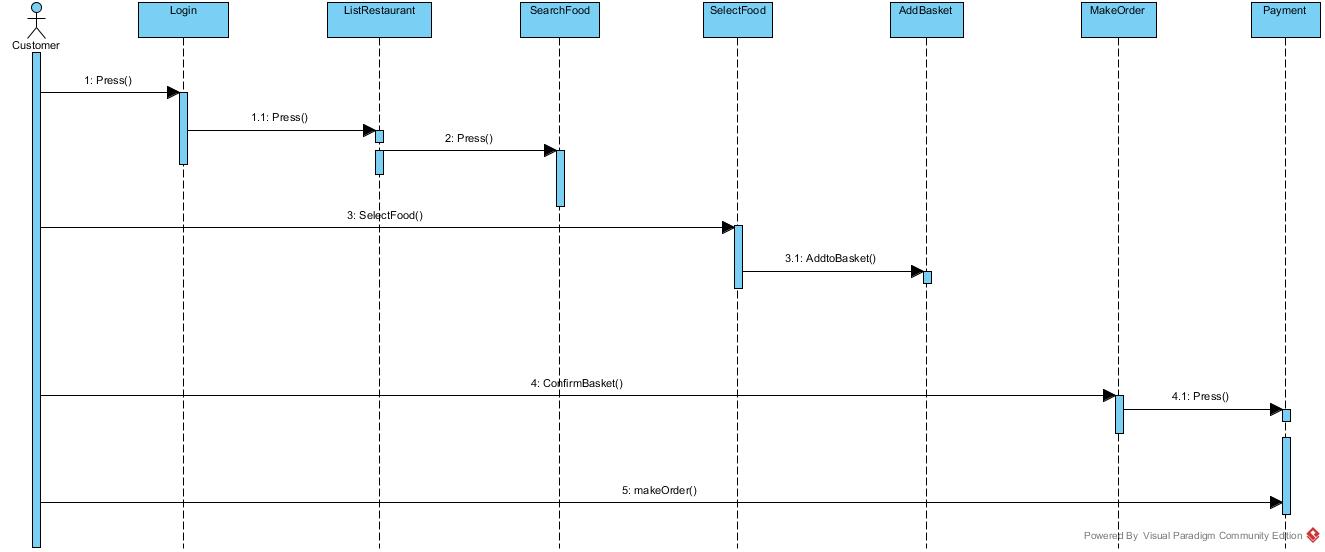
*-Manager:*

**

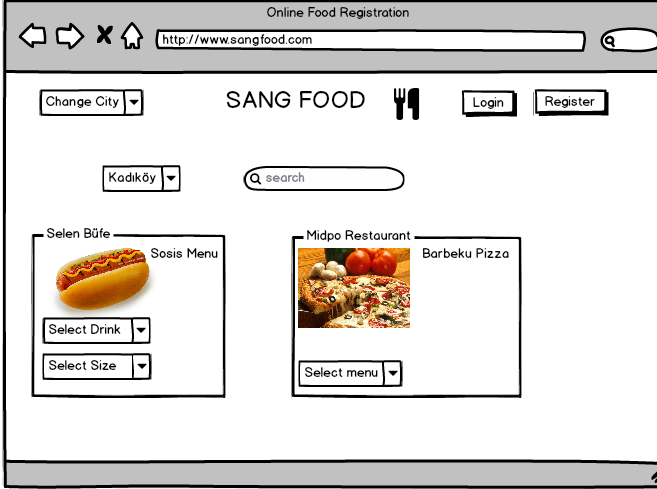
*-Customer (Edit Profile):*

**

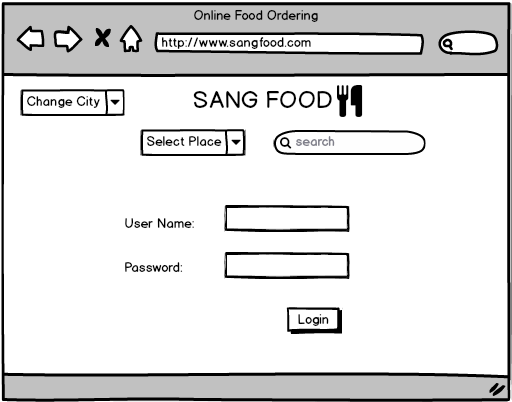
*-Customer (Make Order):*

**

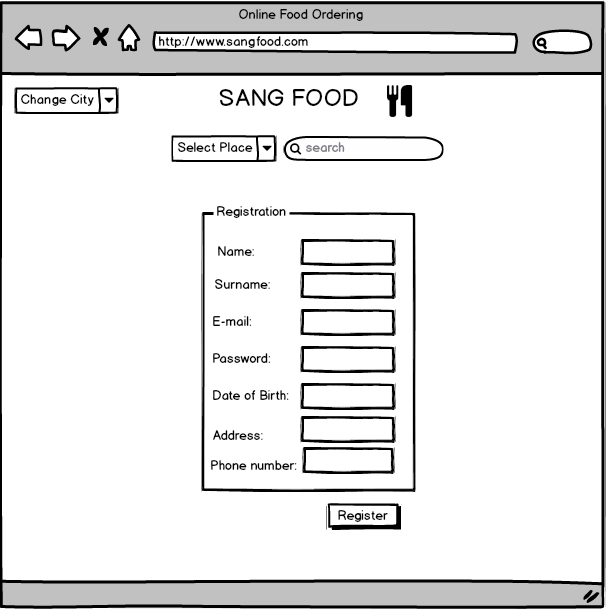
### **User interface**



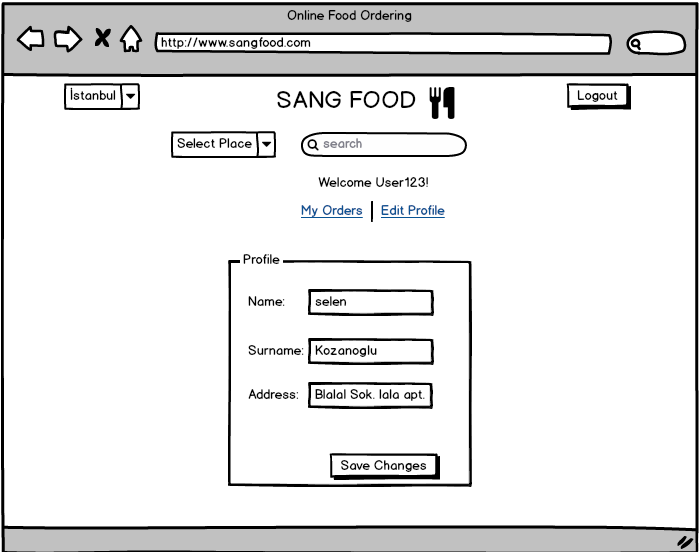
**Guest\_home page**



**Login Page**



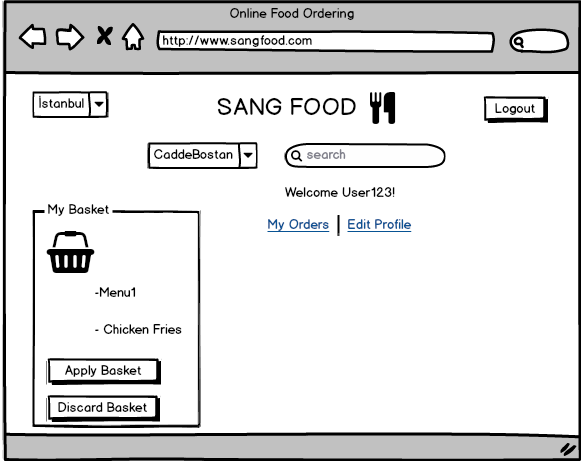
**Register Page**



**Edit Profile Page**



**Add Basket page**



**Apply Basket Page**

# Glossary

**Sangfood:** The website’s name (domain address; [www.sangfood.com](http://www.enjoyyourmeal.com))

**Customer:**  A person who uses the system for making an order.

**Restaurant Owner also Restowner:** A person who has a restaurant in sangfood system.

**Manager:** has a task to accept or reject the incoming restaurants.

**Admin:** It can be person or not. It’s task is creating an account for restaurants.

**Restaurant Form:**  A form which includes restaurant information to suggest a restaurant to the system.

**SearchFoodByPlace:** It is a part of Advanced Search the user can make a search by choosing his/her place.

**SearchFoodByFoodType:** It is a part of Advanced Search the user can make a search by choosing the food type (i.e.; Fast Food, Kebap etc.)

**Restaurant Page:** It is a page which approved by the system.It can be distinguished from everyone when they want to display the restaurant.

# References

* Bruegge B. & Dutoit A.H.. (2010). *Object-Oriented Software Engineering Using UML, Patterns, and Java*, Prentice Hall, 3rd ed.