SOFTWARE REQUIREMENT

SPECIFICATION

OF

SELF GARSON

In accordance with IEEE 830

Önder İlke SEVER 1881473

Onur Tosun 1881549

Emir Kaan Perek 1881416

Version Number	Date	Number of section	A-Added M-Modified D-Deleted	Title or brief description
1.0	08.01.2016			Original

TABLE OF CONTENTS

Table of Figures

1. Introduction

This Software Requirement Specification document expresses complete description about our design which named Self Order System. This document includes all the functions and specifications with their explanations to solve related problems as a graduation project of Middle East Technical University Computer Engineering Department.

1.1 Problem Definition

Since our system is a link between businesses and clients, the projects aims to solve problems of both side.

Client side problems are basically arises from needs of human nature. Firstly, our system will help the customers to get served faster in businesses. Most of the time, customers waste their time to catch attention of server to give the order or pay the bill. Main principle of our system is wiping off the waiting time of a customer in a business, by letting them do the job on their own from their smartphones. Secondly, our system will give freedom to user in many ways. Checking the orders or the current bill of the table and looking up the menu whenever they want are most important functionalities that we are planning to implement with our Self Order System.

Business side problems may be handled in three major headline. First of all, working with high individuals generally leads to deal with variety of personal problems which makes scheduling very hard. Even though there will be no personal problems scheduling is challenging process in business decision. Secondly, when it comes to attitudes of employees, customers generally suffer from not getting proper interaction with them. We are planning to cut off the problems with attitude and behaviors. Lastly and the most important part is reducing costs which enables business owners gain higher profit. Since SOS (Self Order System) decreases the necessity of employee, business owners will get rid of many type of costs like salary or insurance.

1.2 System Overview

The system has two main parts as software and hardware. The hardware part will have a special chip (Beacon) which uses BLE (Bluetooth Low Energy) technology and will communicate with smartphones. On the other side, software part is composed of a mobile

application which is in touch with server and beacon, a desktop application and information panel which are in touch with server.

This project will split users into four types and each type of user will be in contact with the specific application that is designed for that type of users.

First and the major part of users will be the restaurant and café customers who has mobile application in their smart phones. After customers are in touch with the beacon on the table, our application first determine the exact place at the restaurants/cafes in which the customers use the application, and then the application will specify the content to show the customers. After this step, customers will be able to see the menu, to follow his/her order status, to have detailed check, and also pay the bill with their smart phones.

Second type users will be the employees who works in the kitchen side of the business (chefs) and they will be in contact with our information panel application. The mobile application will send a notification to information panel application to inform them about the orders of customers.

Third type users will be the employees who work as waiter/waitress in the business. They will also use the smartphone application with a special formed authentication system. Our system will be able to determine the user type of the smartphone user according to their login credentials. They will be as a master user and they will be able to do most of the functionalities that the customers may do, but with a wider authorization which enable them to do these operations for more than one table.

Fourth and last type of users will be the managers at restaurants/cafes and they will be in contact with our desktop application. The manager will have a panel which make him/her be able to modify the menu, to check and control the general status of tables, to get the statistical data of their sales at the end of day, etc. This make the manager as the highest level controller in their company database.

1.3 Definitions, acronyms, and abbreviations

SRS	Software Requirement Specification
BLE	Bluetooth Low Energy
Java	Java is a computer programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible.
PHP	Server-side scripting language designed for web development
Beacon	Hardware tool which uses Bluetooth low energy
Windows	Windows is a series of graphical interface operating systems developed, marketed, and sold by Microsoft.
NetBeans	NetBeans is an integrated development environment (IDE) for developing primarily with Java, but also with other languages, in particular PHP, C/C++ and HTML5.
GUI	Graphical User Interface
Android Studio	Android Studio is the official integrated development environment (IDE) for developing for the Android platform.
SOS	Self Order System

Table 1.3.1: Definitions, acronyms, and abbreviations

1.4 Assumptions and dependencies

Self Order System has some dependencies for each of the application types.

Mobile application depends on an Android based smartphone (with API level 18+) and enough space in the hard drive for installation of our mobile application.

Desktop application depends on a PC and enough space in the hard drive for installation.

Information panel application depends on an Android based device (may change with the choice of the business) with API level 16+ and enough space in the hard drive for installation.

Moreover, since the system crucially leans on the Beacon and mobile application communication, every business that is going to use our system needs to have Beacon device on

each table which they want to present Self Order System to their customers. Moreover, the communication with the server is another critical part of our system and the user needs to have an internet connection. Otherwise, the customer will not be able to use further functionalities of our system.

2. Overall description

This section will give an overview of the whole system. The purpose of this section is to give general factors that affect the system and its requirements.

2.1 Product functions

In this section, the main functionalities are going to be provided to give a general insight to the system.
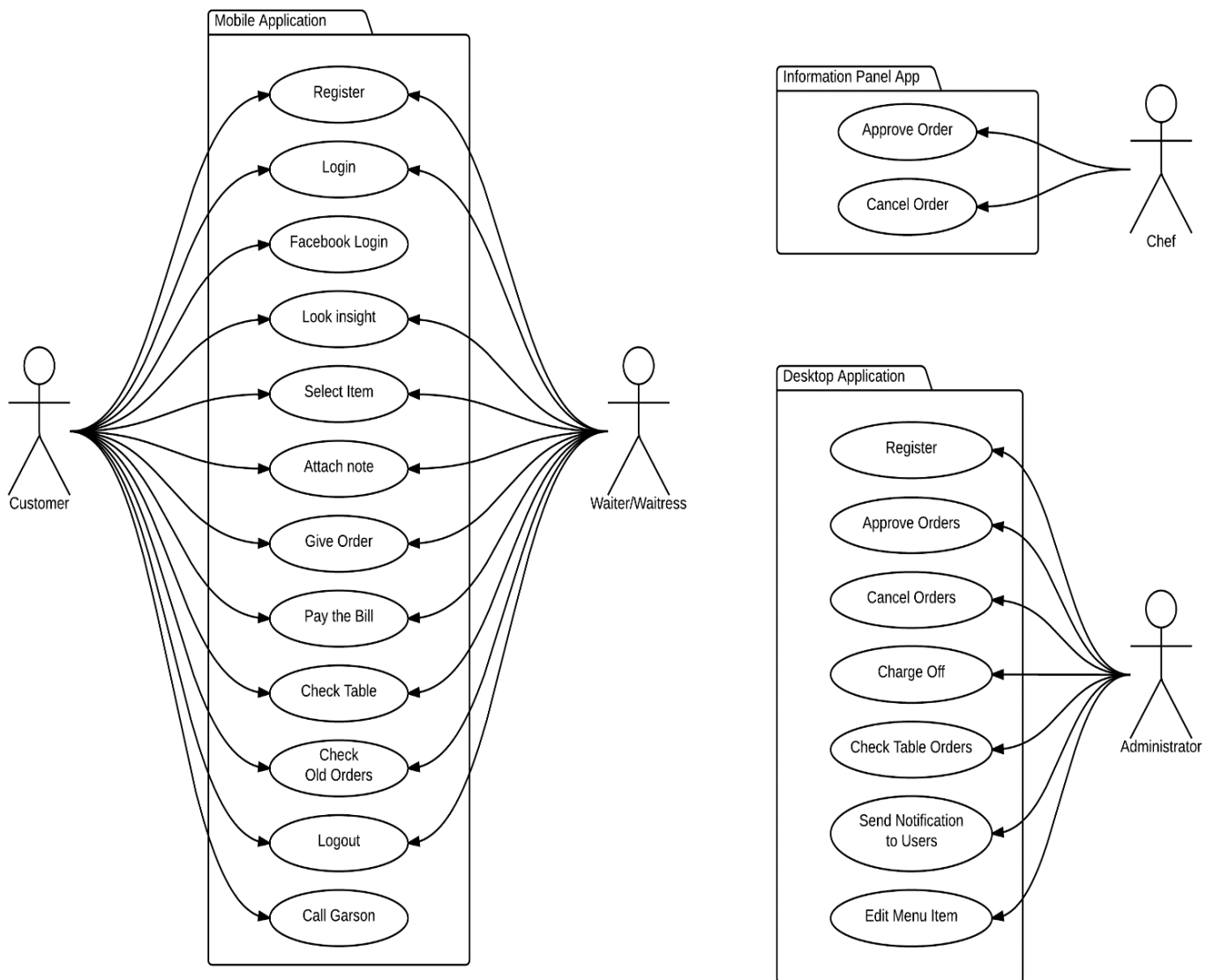

Figure 1 – Use Case Diagram

- Mobile Application Functions

· Register: When the user creates a new account on the application that the SOS runs on, also implicitly an account for the user inside the server side will be created to store information about the user profile.

· Login: When the user logs in to the application server will be informed and Self Garson session for the user start so customer enable to use enrolled customer functions.

· Facebook Login: Registered user can also login our application by login with Facebook button.

· Get Menu: This function stimulated without user interaction and get the data for related business then user enable to see menu.

· Look Insight: User can analyze the content of menu item by clicking the content part which located under each menu item

· Select Item: User select whatever they want to eat or drink via touching on the item stated at the screen.

· Attach Note: User can attach a note after clicking attach note button before give order.

· Give Order: After user Select Items can give a self-order by clicking give order button.

· Call Garson: When user want to contact with user it is enough to click Call Garson button.

· Pay the Bill: When user decided to leave the restaurant or café he/she can pay his/her bill by using app.

· Check Table/Old Table: User can see the dynamic situation of his/her table with "Check Table" button, also can see the old tables that he/she sit before with "Old Table" button.

· Logout: When the user decides to remove actual authentication from application, s/he may choose to Logout from system.

- Desktop Application Functions

· Login: User Login after we give related business activate code to use SOS.

· Approve Orders: When user saw the order come from table X he click approve order button then smart phone user can see that his/her order approved.

· Cancel Orders: When user saw the order come from table X he click cancel order button (user can add why he/she cancelled order) then smart phone user can see that his/her order cancelled.

· Charge Off: If smart phone user or other clients prefer the pay with classic methods user can change the bill situation of the table by clicking 'Foot the Bill' button.

· Check Table Orders: This methods does not have user interface at SOS desktop application side, system check if there is an order automatically then show the result to user.

· Send Notification to Users Instantly: User can send a push notification to their clients when they launch a campaign. (Ex: Milky Desserts are %30 Sale)

· Edit Menu Item: User can add new menu item, remove menu item and can change the property of menu item like its price or picture.

- Information Panel Application Functions

· Approve order: If the incoming order is appropriate and servable to the customer, employee may approve the order.

· Cancel order: If the incoming order is not appropriate and servable for some reasons, employee may cancel the order.

2.1.1 Use-case model survey

Use Case ID	UC-1
Use Case Name	Register
Primary System Actor	Smart Phone User
Participating Actors	-
Description	This use case describes the event of an Smart Phone User who wants to register the system
Precondition	-
Trigger	This use case is initiated when the user click "Register" Button

Typical Course Of Events	Actor Action	System Response
	Step 1:Smart Phone User click "Register" button Step 2: Smartphone User enters username. Step 3: Smartphone User enters surname. Step 4: Smartphone User enters Phone Number. Step 5: Smartphone User presses "Submit" button. Step 6: Smartphone User determine his/her password	
		Step 7:System Checks the Validations Step 8:System Send a verification code to user
	Step 9: Smartphone User enter the verification code to open pop up and click submit	
		Step 10:System save the profile of user to database

Alternate Courses	Alt Step 7: System checks validations and if is invalid then show error message according to missing part , go to Step 1
Post condition	Smart Phone User registered the system Successfully

Use Case ID	UC-2
Use Case Name	Login
Primary System Actor	Smart Phone User
Participating Actors	-
Description	This use case describes the event of an Smart Phone User who wants to login the system
Precondition	Smart Phone User should be registered in the system
Trigger	This use case is initiated when the Smartphone User click "Login" Button

Typical Course Of Events	Actor Action	System Response
	Step 1: Smart Phone User click "Login" button Step 2: Smart Phone User enters username. Step 3: Smartphone User enters phone number. Step 4: Smart Phone User presses "Login" button. Step 6: Smartphone User continues the application as an enrolled User.	Step 5: System Checks username and telephone number is valid.

Alternate Courses	Alt Step 5: System checks username and telephone number if is invalid then show error message "username or telephone is invalid" ,then go to Step 1
Post condition	Smart Phone User logged the system Successfully

Use Case ID	UC-3
Use Case Name	Logout
Primary System Actor	Smart Phone User
Participating Actors	-

Description	This use case describes the event of an Smart Phone User who wants to logout the system
Precondition	Smart Phone User should be logged in the system
Trigger	This use case is initiated when the Smartphone User click "Log Out" Button.
Typical Course Of Events	

Actor Action	System Response
Step 1: Smart Phone User click "Log Out" button	
	Step 2: System save this user as a logged out user.

Alternate Courses	Alternate Step 2: If there is a negative response at step 2, System will send a message with why logout operation cannot completed.
Post condition	Smart Phone User logged out the system Successfully

Use Case ID	UC-4
Use Case Name	Facebook Login
Primary System Actor	Smart Phone User
Participating Actors	-
Description	This use case describes the event of an Smart Phone User who wants to login with Facebook to the system
Precondition	Smart Phone User should be registered in the system
Trigger	This use case is initiated when the Smartphone User wants to logged in to mobile application with Facebook
Typical Course Of Events	

Actor Action	System Response
Step 1: Smart Phone User click "Login with Facebook" button	
	Step 2:System get username, telephone number, profile picture etc.

	Step 3: Smartphone User continues the application as an enrolled User.	
Alternate Courses	Alt Step 2: System checks username and telephone number if is invalid then show error message "username or telephone is invalid" ,then go to Step 1	
Post condition	Smart Phone User logged the system Successfully	

Use Case ID	UC-5	
Use Case Name	Confirm the Table	
Primary System Actor	Smart Phone User	
Participating Actors	Beacon	
Description	This use case describes the control of if system detect correctly where the user is sitting	
Precondition	Smart Phone User should be registered in the system and his/her phone should be at the range of beacon (that located at the table).	
Trigger	This use case is initiated when the Smartphone User wants to use SOS and its features	
Typical Course Of Events		

Actor Action	System Response
Step 2: If Smart Phone User get his/her phone close enough beacon found button become clickable Step 3: Smartphone User click beacon found button	Step 1:After user logged in pop up shown and want user to get closer to beacon
Step 6: Smartphone User Click 'OK'.	Step 4: Application get the business name and table number when user click detect beacon. Step 5:Application ask user If he/she is really sitting there

Alternate Courses	Alt Step 6: Smartphone User Click cancel then go to step 2 to find correct beacon.

Post condition	Smart Phone User can use the SOS.

Use Case ID	UC-6
Use Case Name	Show Menu
Primary System Actor	Smart Phone User
Participating Actors	-
Description	This use case describes the event of an Smart Phone User who wants to see the menu of related business
Precondition	Smart Phone User should be registered in the system, Confirm the Table
Trigger	This use case is initiated when the Smartphone User want to use SOS features

Typical Course Of Events		
	Actor Action	System Response
	Step 1: Smart Phone User click "Show Menu" button	Step 2: System fill the expandable list with the items which is came from server with related Business ID (Get item function do this operation).
	Step 3: Smartphone User can see the menu.	

Alternate Courses	Alt Step 2: If application cannot reach the data of menu from server error message shown to user with possible causes (no internet connection ... etc.), then go to step 1.
Post condition	Smart Phone User saw the menu Successfully

Use Case ID	UC-7
Use Case Name	Look Insight
Primary System Actor	Smart Phone User
Participating Actors	-

Description	This use case describes the event of an Smart Phone User who wants to see menu items' content
Precondition	Smart Phone User should be registered in the system, Confirm the Table, show the menu operation completed successfully
Trigger	This use case is initiated when the Smartphone User want to see menu items' content

Typical Course Of Events		
	Actor Action	System Response
	Step 1: Smart Phone User click words that describe what this item composed of shortly under each item (ex: cheese and tomato sauce.. will be shown under margarita pizza)	Step 2: System show user more detailed content of selected menu item).

Alternate Courses	Alternate Step 2: If there is a negative response at step 2, System will send a message with why Look Insight operation cannot completed.
Post condition	Smart Phone User saw the detailed of selected menu item successfully

Use Case ID	UC-8
Use Case Name	Select Item
Primary System Actor	Smart Phone User
Participating Actors	-
Description	This use case describes the event of an Smart Phone User who wants to order an item
Precondition	Smart Phone User should be registered in the system, Confirm the Table, show the menu operation completed successfully
Trigger	This use case is initiated when the Smartphone User select one item of the menu

Typical Course Of Events		
	Actor Action	System Response

Step 1: Smart Phone User click the menu item	Step 2: System marked this Item as selected item and give toast message like "X Siparislere Eklendi"

Alternate Courses	Alternate Step 2: If there is a negative response at step 2, System will send a message with why select item operation cannot completed.
Post condition	Smart Phone User saw the detailed of selected menu item successfully

Use Case ID	UC-9
Use Case Name	Attach Note
Primary System Actor	Smart Phone User
Participating Actors	-
Description	This use case describes the event of an Smart Phone User who wants to attach note to selected items
Precondition	Smart Phone User should be registered in the system, Confirm the Table, show the menu operation completed successfully, At least one item should be selected
Trigger	This use case is initiated when the Smartphone User click attach note button
Typical Course Of Events	

Actor Action	System Response
Step 1: Smart Phone User click attach note button Step 2:Smart Phone User write whatever want to attach as an order note	Step 3: System attach written note to related order and toast message will show to user "Siparislere Not Eklendi"

Alternate Courses	Alternate Step 3: If there is a negative response at step 2, System will send a message with why Attach Note operation cannot completed.

Post condition	Smart Phone User add the order note successfully

Use Case ID	UC-10
Use Case Name	Give Order
Primary System Actor	Smart Phone User
Participating Actors	-
Description	This use case describes the event of an Smart Phone User who wants to attach note to selected items
Precondition	Smart Phone User should be registered in the system, Confirm the Table, show the menu operation completed successfully, At least one item should be selected
Trigger	This use case is initiated when the Smartphone User click give order button

Typical Course Of Events	Actor Action	System Response
	Step 1: Smart Phone User click give order button	Step 2:System Show a pop-up that includes selected items
	Step 3: User Click "Confirm Order" button	Step 4: System send selected items to server
		Step 5: System show toast message like "Siparisiniz basarıyla verildi"

Alternate Courses	Alternate Step 3: User Click "Reject Order" turn step 1 Alternate Step 5:System Show error message with definition if selected items cannot send to server for any reason, turn step 1
Post condition	Smart Phone User give order successfully

Use Case ID	UC-11
Use Case Name	Call Garson

Primary System Actor	Smart Phone User
Participating Actors	-
Description	This use case describes the event of an Smart Phone User who wants to interact with garson
Precondition	Smart Phone User should be registered in the system, Confirm the Table
Trigger	This use case is initiated when the Smartphone User click "Call Garson Button"

Typical Course Of Events	Actor Action	System Response
	Step 1: Smart Phone User click "Call Garson Button"	Step 2:System send a notification to Desktop application

Alternate Courses	Alternate Step 2:System Show error message with definition if call garson order cannot send to server for any reason, turn step 1
Post condition	Smart Phone User give order successfully

Use Case ID	UC-12
Use Case Name	Pay the Bill
Primary System Actor	Smart Phone User
Participating Actors	-
Description	This use case describes the event of an Smart Phone User who wants pay his/her Bill via Application
Precondition	Smart Phone User should be registered in the system, Confirm the Table, show the menu operation completed successfully, At least one item should be selected and Give Order function should be completed successfully
Trigger	This use case is initiated when the Smartphone User click "Pay the Bill Button"

Typical Course Of Events		
	Actor Action	System Response
	Step 1: Smart Phone User click "Pay the Bill Button"	
		Step 2:System check if user register his/her credit card to SOS Step 3:If user register his/her credit card to system, SOS check if credit card has a enough limit to pay bill

Alternate Courses	Alternate Step 2,3: If there is a negative response at step 2 or step 3, System will send a message with why pay bill operation cannot completed
Post condition	Smart Phone User pay the bill successfully

Use Case ID	UC-13
Use Case Name	Check the Table
Primary System Actor	Smart Phone User
Participating Actors	-
Description	This use case describes the event of an Smart Phone User who wants to check table situation like order situation or total cost, User can also check the table history that he/she sit before.
Precondition	Smart Phone User should be registered in the system, Confirm the Table
Trigger	This use case is initiated when the Smartphone User click "Check Table Button"

Typical Course Of Events		
	Actor Action	System Response
	Step 1: Smart Phone User click "Check Table " button	
		Step 2:System check the table content from server and show to user

Alternate Courses	Alternate Step 1: If User wants to check the table which he/she sit before click "Check Old Tables" button Alternate Step 2: If there is a negative response at step 2, System will send a message with why check table operation cannot completed.

Post condition	Smart Phone User check the table2 successfully

Use Case ID	UC-14
Use Case Name	Login
Primary System Actor	Desktop Application User
Participating Actors	WMI
Description	This use case describes the event of an Desktop Application user Login
Precondition	Temporary Password should be taken from WMI after shaken hand with them.
Trigger	This use case is initiated when the user wants to start use SOS Desktop Application

Typical Course Of Events	Actor Action	System Response
	Step 1:Desktop Application User enters username Step 2: Desktop Application User enters temporary password taken from WMI Step 3: Desktop Application User presses "submit" button. Step 6: Desktop Application User fill the "new password" field and click submit	 Step 4: System Checks the Validations and if permanent password is true. Step 5:New screen opens and wants user to determine permanent password Step 7:System save the profile of user to database

Alternate Courses	Alt Step 7: System checks validations and if is invalid then show error message according to missing part , go to Step 6 Alt Step 4: System checks validations and if is invalid then show error message according to missing part , go to Step 1
Post condition	Desktop Application User registered the system Successfully

Use Case ID	UC-15
Use Case Name	Approve Order
Primary System Actor	Desktop Application User
Participating Actors	-
Description	This use case describes the event of a Desktop Application user approve order that comes from smartphone users (clients).
Precondition	Desktop Application User should be logged in
Trigger	This use case is initiated when the user click approve order

Typical Course Of Events		
	Actor Action	System Response
	Step 1:Desktop Application User click approve button near the order came from clients	Step 2: Server save this information to database. And send this information to related client via push notification.

Alternate Courses	Alternate Step 2: If there is a negative response at step 2, System will send a message with why "approve order" operation cannot completed.
Post condition	Desktop Application User approved order Successfully

Use Case ID	UC-16
Use Case Name	Cancel Order
Primary System Actor	Desktop Application User
Participating Actors	-
Description	This use case describes the event of a Desktop Application user cancel order that comes from smartphone users (clients).
Precondition	Desktop Application User should be logged in
Trigger	This use case is initiated when the user click cancel order

Typical Course Of Events		
	Actor Action	**System Response**
	Step 1:Desktop Application User click cancel button near the order came from clients Step 2: Desktop application user attach a note that explain why did order cancel.	Step 3: Server save this information to database. And send this information to related client via push notification.

Alternate Courses	Alternate Step 3: If there is a negative response at step 3, System will send a message with why "Cancel Order" operation cannot completed.
Post condition	Desktop Application User cancel order Successfully

Use Case ID	UC-17
Use Case Name	Charge Off
Primary System Actor	Desktop Application User
Participating Actors	-
Description	This use case describes the event of an Desktop Application user charge off the selected table if client pay his/her bill without using mobile application
Precondition	Desktop Application User should be logged in
Trigger	This use case is initiated when the user click "Charge Off "button

Typical Course Of Events		
	Actor Action	**System Response**
	Step 1:Desktop Application User select a table Step 2: Desktop application user click "Charge Off" button.	Step 3: Server save this information to database. And send this information to related client via push notification.

Alternate Courses	Alternate Step 3: If there is a negative response at step 3, System will send a message with why "Charge Off" operation cannot completed.
Post condition	Desktop Application User Charged Off selected table Successfully

Use Case ID	UC-18
Use Case Name	Send Notification to Users Instantly
Primary System Actor	Desktop Application User
Participating Actors	-
Description	This use case describes the event of a Desktop Application user, user can send a notification to clients about campaigns and sales.
Precondition	Desktop Application User should be logged in
Trigger	This use case is initiated when the user click "Send Notification "button

Typical Course Of Events		
	Actor Action	System Response
	Step 1:Desktop Application User click a "Send Notification" button Step 2: Desktop application user write at opened textbox whatever they want to send all clients sitting that business at that moment.	Step 3: Server save this information to database. And send this information to related clients via push notification.

Alternate Courses	Alternate Step 3: If there is a negative response at step 3, System will send a message with why "Send Notification" operation cannot completed.
Post condition	Desktop Application User Send Notification Successfully

Use Case ID	UC-19
Use Case Name	Edit Menu Item
Primary System Actor	Desktop Application User

Participating Actors	-
Description	This use case describes the event of an Desktop Application user, user can edit the menu items like remove add and change the variable of related item (like price or definition)
Precondition	Desktop Application User should be logged in
Trigger	This use case is initiated when the user click "Edit Menu "button

Typical Course Of Events	Actor Action	System Response
	Step 1:Desktop Application User click a "Edit Menu " button	Step 2:New pop up opens which include the list of menu items
	Step 3:Desktop Application User click on the item that wants to edit	Step 4: Selected Item's properties are show to User.
	Step 5:Desktop Application User selects a function like remove or change property and do the related operation by clicking remove or change property button	
	Step 6: If Desktop Application User wants to add a new menu item, click add item button and fill the validated areas of new item.	
	.	Step 7: System save the changes to database.

Alternate Courses	Alternate Step 7: If there is a negative response at step 7, System will send a message with why "Edit Menu Item" operation cannot completed.
Post condition	Desktop Application User Edit a Menu Item Successfully

Use Case ID	UC-20
Use Case Name	Login
Primary System Actor	Chef

Participating Actors	WMI
Description	This use case describes the event of an Informative Panel Application User Login
Precondition	Temporary Password should be taken from WMI after shaken hand with them.
Trigger	This use case is initiated when the user wants to start use SOS Desktop Application

Typical Course Of Events		
	Actor Action	System Response
	Step 1: Desktop Application User enters temporary password taken from WMI Step 2: Informative Panel Application User presses "submit" button.	Step 3: System Checks the Validations and if permanent password is true. Step 4:New screen opens and wants user to determine permanent password
	Step 5: Informative Panel Application User fill the "new password" field and click submit	Step 6:System save the profile of user to database

Alternate Courses	Alt Step 7: System checks validations and if is invalid then show error message according to missing part , go to Step 5 Alt Step 4: System checks validations and if is invalid then show error message according to missing part , go to Step 1
Post condition	Informative Panel Application User registered the system Successfully

Use Case ID	UC-21
Use Case Name	Approve Order
Primary System Actor	Chefs
Participating Actors	-

Description	This use case describes the event of an Information Panel Application User approve order that comes from smartphone users (clients).
Precondition	Informative Panel Application User should be logged in
Trigger	This use case is initiated when the user click approve order

Typical Course Of Events		
	Actor Action	System Response
	Step 1: Informative Panel Application User click approve button near the order came from clients	
		Step 2: Server save this information to database. And send this information to related client via push notification.

Alternate Courses	Alternate Step 2: If there is a negative response at step 2, System will send a message with why "Approve Order" operation cannot completed.
Post condition	Informative Panel Application User approved order Successfully

Use Case ID	UC-22
Use Case Name	Cancel Order
Primary System Actor	Informative Panel Application User
Participating Actors	-
Description	This use case describes the event of an Informative Panel Application User cancel order that comes from smartphone users (clients).
Precondition	Informative Panel Application User should be logged in
Trigger	This use case is initiated when the user click cancel order

Typical Course Of Events		
	Actor Action	System Response
	Step 1: Informative Panel Application User click cancel button near the order came from clients	

	Step 2: Informative Panel Application User attach a note that explain why did order cancel.	
		Step 3: Server save this information to database. And send this information to related client via push notification.
Alternate Courses	Alternate Step 3: If there is a negative response at step 3, System will send a message with why "Cancel Order" operation cannot completed.	
Post condition	Informative Panel Application User cancel order Successfully	

2.1.2 Actor survey

Customer (*Mobile Application User*)	Contains the major part of our users. All the persons who are going to expect service from the cafe or restaurant may be involved in this actor type. They are going to be involved our system via mobile application. They will be able to communicate with cafe and restaurants with mobile application. Checking menu, creating order, tracking his/her order, checking and paying the bill of the table are the main functionalities which this actor type will have.
Chef (*Information Panel Application*)	Chefs working on the business may be involved in this actor type. They will be informed about the customer actions and will be able to control the status of those actions. Checking order status and controlling them, checking table status are the main functionalities which this actor type will have.
Waiter (*Mobile Application User*)	Waiter working on the business involved in this actor type. They can use mobile application as a customer with master login. Only difference between customer and this role is server will know if mobile application user is customer or Waiter/Waitress. This type of actor is necessary to demand the orders of customer who do not have a smartphone. With this type of actor SOS become a compact system that every business can use without considering their client profile.

Administrator (*Desktop Application User*)	The administrator of the business will be involved in this type of actor. Every functionalities that Employee role has are also provided to this type of actor. Moreover, they will be able to administrate the menu of the business and get the statistical or financial data, which is required for the domestic purposes of the business, from the desktop application.

2.2 Interfaces

This section briefly describes the interfaces that must be supported by the application.

2.2.1 User Interfaces

- Mobile Application Interfaces

First interface of our mobile application will be the authentication interface. When the user has started the mobile application, the application will try to authenticate the user, if the user has not done it before. If the user do not have an account, account might be created with Register button. If the user has an account, account might be logged in with Login button. Besides all these, Login with Facebook feature will be provided as well for the users that choose to authenticate themselves with their Facebook account. If the user successfully completes the authentication once, this interface will not be shown to customer for the further usages of the mobile application for simplicity and practicality purposes (until the user chooses Logout option). *<This interface might be associated with the above use cases UC-1, UC-2, UC-3, and UC-4>*

Second interface of our mobile application will be the connection interface. When the user successfully Login to our system, the system needs to clarify where the customer is to present further Self Order System functionalities. The interface will have an explanation text and visual to encourage user for bringing the smartphone to near of the Beacon device. When the smartphone receives the Beacon signal, it will come out the signal and the message will be delivered to the server with the expurgated numeric values. *<This interface might be associated with the above use cases UC-5>*

Next interface of our mobile application will be ordering interface. When the user successfully communicate with the Beacon and the server, the system will be aware of the identity and the location of the user. From now on, the user might have access to major Self

Order System functionalities. This interface will come with the menu of the business and the user will be able to check it. Also, the user will be able to create order from this menu, track the orders and pay the bill of the table with the simple graphical user interface tools which are smoothly placed on this interface. This is the interface which the user will spend most of his/her time on our mobile application. So, this is the most crucial interface in terms of design and functionality. *<This interface might be associated with the above use cases UC-6, UC-7, UC-8, UC-9, UC-10, UC-11, UC-12, and UC-13>*

The last interface of our mobile application will be features interface. The user will be able to reach this interface from the ordering interface anytime s/he desires. This interface will provide some extra features to the user like Logout, view previous orders, change the user information. *<This interface might be associated with the above use cases UC-3 and UC-13>*

- Desktop Application Interfaces

First interface of our desktop application will be Login interface. Since desktop application will be used by the users which are restaurant employees, the users will be registered by us.

After login, users will be in second interface which has mainly two topics. They are show table contents and edit menu. The show table content button will direct user to page that allows user to see whole contents of tables and manipulate tables. By saying manipulate, the user will be able to cancel order, approve order, charge off and send notification to tables. In the edit menu part of the interface, user will see the items, their information and their default pictures. By the help of the buttons edit-add-remove, user will be able to change that information. *<This interface might be associated with the above use cases UC-15, UC-16, UC-17, UC-18, and UC-19>*

- Information Panel Interfaces

The information panel will have one interface and will be the simplest among all of the user interfaces. This interface will provide an information screen to the user. The user will be able to see the status of orders and will be able to control them. When the order reached to the information panel, user might change the status of the order as 'Order is Approved, or 'Order is Cancelled'. So, this will also lead customers of the business to track the status of their order. *<This interface might be associated with the above use cases UC-21 and UC-22>*

2.2.2 Hardware Interfaces

Since the system will be based on two platform which are mobile and desktop, Android Smartphone and Personal Computer are the main hardware interfaces of the system. Also, the device called 'Beacon' which is going to help us to establish the connection between user and business is another hardware interface of the system.

2.2.3 Software Interfaces

Android Studio: IDE for developing our Android application.

NetBeans: IDE for developing our Desktop application.

Java Development Kit 1.8: The main software development kit for both of our applications.

Android SDK API 18+: Comprehensive set of development tools for Android programming.

LAMP (abbreviation of Linux-Apache-MySQL-PhP quadruplet): Common archetypal model of web service development. Server-side is going to base on LAMP.

2.2.4 Communications Interfaces

There will be two types of communications in the system.

- Between the Beacon and the mobile application is going to be provided with BLE signals.
- Between Server and mobile or desktop application is going to be provided with HTTP POST and GET protocols.

2.3 Constraints

Because of the communication between smartphone and Beacon requires BLE technology, some outdated smartphones will not support it. Android supports BLE technology as from API level 18 (Android 4.3 Jelly Bean). For this reason, the customer needs to have an Android based smartphone which has API level 18+.

3. Specific requirements

This section contains all of the functional and nonfunctional requirements of the system in detail for the following stages of development process.

3.1 Functional Requirements

Below are the Major Functional Requirements of the Server Side

- Server should save registered clients.
- Server should save logged in clients.
- Server should save the orders.
- Server should save the total cost of each map.
- Server should return a correct response according to opcode of request.

Below are the Major Functional Requirements of Mobile Application Side

- User will be able to login and register.
- User will be able to see menu.
- User will be able to give order and call garson.
- User will be able to pay the bill.
- User will be able to check his/her tables' dynamic status.

Below are the Major Functional Requirements of Desktop Application Side

- User will be able to login and register.
- User will be able to both approve and cancel the orders.
- User will be able to Charge Off the wanted table.
- User will be able to send a notification to clients.

Below are the Major Functional Requirements of Information Panel Application Side

- User will be able to login
- User will be able to approve or cancel the orders.

3.2 Nonfunctional Requirements

This section will handle the subject nonfunctional requirements in mainly 4 headlines.

3.2.1 Usability
Since the project consists of three different application and all of them will be designed to reduce complexity as much as possible, all type of users will start to use applications

efficiently in pretty short time. However, mobile application will appeal to larger scale of users beside desktop application and information panel application. Thanks to the 'First Time Introduction' feature of the Android development, we will also have a chance to give tutorial to users when they are started the application for the first time. If we fail on supplying simple usability, the potential users may abandon the application. So, this is a very critical bullet that we definitely need to accomplish.

3.2.2 Reliability

Since our project aims to collaborate with cafe and restaurant businesses, we have to supply absolute reliability for them. These businesses more generally work between 8:00 and 24:00. This means, our system should be available at least 66% of the full day. However, this does not mean we may shutdown the system for the rest of the day. Our system will be constructed as 24 hours working system but maintenance processes for the main servers might be completed between 00:00 - 08:00 which is off-peak hours. However, we are also aware of that building the big systems on a single server might cause significant failures, if the server is exposed to some unexpected factors like cyber-attack, overload, technical problems, etc. So, our systems will not depend on a single server and all the security measures will be provided before releasing the product. Since our project aims to make grand improvements for the infrastructure of the businesses, reliability is one of the most important requirement we have to supply. Losing the trust of the businesses will affect the future of our project and operation of the business very badly.

3.2.3 Performance

Since our project does not depend on complex calculations, the expected performance will be satisfactory for the users. Because the system is going to be built on well-prepared servers, we are expecting good performance like most of the Android applications that has taken place in our lives. JSON objects will be send through between servers and clients. Thanks to the small-sized structure of the JSON objects, our system will not require high-speed network connection from the users. All the functionalities that our system involves are planned to be completed under 100 milliseconds on average conditions. This time is suitable for all type of users in our system. However, we will be able to track user information and access data of the users, thanks to Google Analytics system, day by day. We will have chance to improve our system performance when the user count approaches to our maximum capacity.

3.2.4 Supportability

As mentioned in the 2.3 Constraints subsection, our project has a limitation about the minimum API level we may choose. When the minimum API level is chosen high, software development kit supplies more functionality and extra features which means life becomes easier for the developer. However, this situation decreases the potential user number of the application because every Android device does not support higher level API's. So, if you increase the API level, you decrease the potential customer percentage. For this reason this is a very critical decision for Android developers. In this project, we are going to choose API level 18 to target maximum percentage of the Android users as we can.

4. Data Model and Description

In this part of the SRS you may observe our class diagrams. In this project there are 3 major classes includes the main data that our system use these classes are Urunler, BeaconData and UserData. BeaconData can be considered as a Related_Business Class since BeaconData class includes these information. Major variable can be considered as a Business_ID and Minor variable can be considered as a Branc_ID. With these information our system can make all server requests and can do all functionalities mentioned before. Some widget and template variables are not included at the class diagram since they do not describe our system such as Button, TextBox objects and some temporary variables like *<cntr>* or *<i,j>* etc.

Figure 2 – Class Diagram

5. References

[1] IEEE Guide for Software Requirements Specifications," in IEEE Std 830-1984, vol., no., pp.1-26, Feb. 10 1984, doi: 10.1109/IEEESTD.1984.119205, URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=278253&isnumber=6883

[2] Appendix C of Don Widrig, Dean Leffingwell, "Managing Software Requirements: A Unified Approach," Addison-Wesley Professional, Release Date: October 1999, ISBN: 0201615932.