

# SOFTWARE ENGINEERING

## TERM PROJECT: **FOODIES'NEST**

**Submitted by:**

Mishaal Hajiani 13050

Sabika Nasir 13053

Areeba Shamsi 13028

**Instructed By:** Naveen Zehra Quazilbash

## Contents

ABSTRACT .....	3
PROCESS MODEL .....	4
BUSINESS REQUIREMENT DOCUMENT .....	6
Document control .....	7
Project Summary .....	8
Purpose .....	8
Objectives .....	8
Background .....	8
Business Drivers .....	9
Project Scope .....	10
In Scope Functionality .....	10
Out of Scope Functionality .....	10
Product Perspective .....	10
Constraints .....	10
Issues .....	10
Business Process Overview .....	11
Business Requirements .....	12
FUNCTIONAL REQUIREMENTS .....	13
NON-FUNCTIONAL REQUIREMENTS .....	14
Performance Requirements .....	14
Safety Requirements .....	14
Security Requirements .....	14
Software Quality Attributes .....	14
WIREFRAME .....	15
SOFTWARE REQUIREMENT SPECIFICATION .....	16
Revision History .....	17
Introduction .....	18
Document Purpose .....	18
Product Scope .....	18
Intended Audience and Document Overview .....	18
Overall Description .....	19
Product perspective .....	19
Product Functionality .....	19
Use Case .....	20
Operating Environment .....	21
Design and Implementation Constraints .....	21
User Documentation .....	21
Assumptions and Dependencies .....	21
External interfaces Requirements .....	22
User Interfaces .....	22
Hardware Interfaces .....	22
Software Interfaces .....	22
Communications interface .....	22
USE CASE .....	23
CLASS DIAGRAM .....	24
SEQUENCE DIAGRAM .....	25
FOODIES' NEST ESTIMATION METRICS .....	31
TEST CASES .....	35
SCREENSHOTS OF SOFTWARE .....	55
GANTT CHART .....	60
BIBLIOGRAPHY .....	62

## ABSTRACT

In the present period of fast food, numerous canteens center around snappy planning and rapid conveyance of requests as opposed to offering a rich feasting dining. Until as of late, all these conveyance orders were put to the servers or via telephone subsequently, there are various downsides to this system, including the inconvenience of the client needing a physical duplicate of the menu, nonattendance of a visual affirmation that the request was set effectively, and a need of a worker picking up the telephone and taking requests. Along these lines, we propose a robotized framework, a method of requesting sustenance's online appropriate for college/school students. The fundamental favorable position of this framework is that it enormously disentangles the requesting procedure for both the client and the cafeteria's workers. At the point when the client visits the requesting website page, they are given an intuitive and up-to-date menu, with every single accessible alternative and costs in view of the selected choices. After choice, the item is then added to their cart and the client can audit the items chosen before finalizing. This gives an instant visual affirmation of what was chosen and ensures that the restaurant in the demand is what was proposed.

The food items are prepared by house-spouses or youthful grown-ups who wish to procure earning this way. This will be an interface to make a positive effect on nation's economy by furnishing ladies with a simple method to maintain their own business in their area. For that reason, The Food Hub group has just directed many studies identified with nourishment utilization in Pakistan, fundamentally focusing on the house-spouses and youthful people.

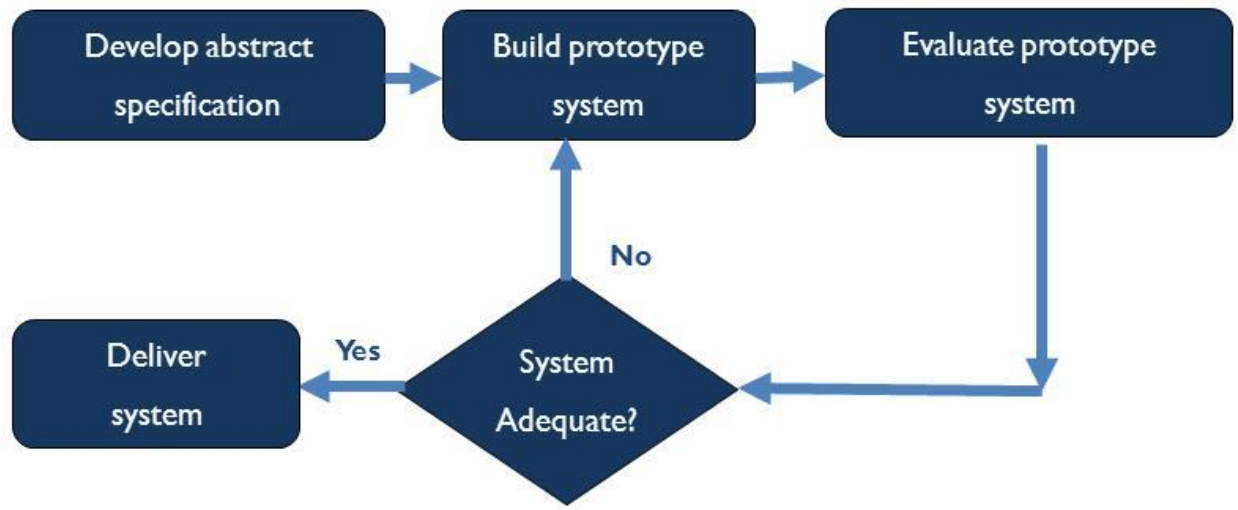
The objective here is to not simply facilitate the work of colleges/school's canteens, yet additionally students and laborers who don't approach home-made nourishment, and unnecessarily expend eatery suppers. What's more, there are moms and spouses who are searching for approaches to offer the sustenance they make.

Once a request is put on the website page, it is recovered, continuously, by a work area application on the canteen's end. This enables canteen's workers to rapidly experience the set request and deliver the important things with negligible postponement and disarray.

The Foodies' Nest is an application-based startup whose point is to give a platform to ladies to offer nutritious and sterile home-made nourishment. They'll be alluding these ladies as home-culinary specialists. This application has offered chance to every one of a ton of home chefs. The home chefs post the items they are offering on the application, stating the planning time and pre-arrange conditions. Clients can peruse the application and scan for the food item, they have a craving for eating and can put in their request likewise.

## Process Model

The Foodies' Nest implementation and design starts with gathering the requirements and examining the environment in which it has to operate. After considering the purpose of this system that is to give a platform to the underprivileged ladies to offer nutritious and sterile home-made nourishment with a wish to procure earning through this way and to provide an ease to the user to order the food they wish to eat on-line through our system, we have realized that the system should be built with minimum errors possible. Therefore, for the development of our system, Foodies' Nest, we have decided to use PROTOTYPE model to satisfy the major requirement which is a frequent interaction of the end users with our system. Hence, the system is decided to be developed in prototypes and ranges of increments for having an advantage for modifying and refining the system readily and easily without problems as per the end-user and customer's feedback. This model helps in identifying the errors and missing functionality at an early phase. A prototype of a system is developed with interpretative and fundamental attributes so that the user can evaluate it and suggest any improvements that are important to be catered in the system. This user/customer evaluation phase of the system continues until a prototype is made which satisfies the customers need. Developing a prototype basically provides with a system with full functionality for the user to understand how and what functions will the system perform.



#### Prototype Model

The advantage of using the Prototype model is that the users are actively involved in the development phase with the functional requirement being the priority in this phase. The foodies' nest prototype can be distributed among all the stake holders to ensure the correct functionality of the system and in the end finalize the prototype with all the requirements tested so that the development of the real software can begin and produce a refined and a perfect system at the end.

# BUSINESS REQUIREMENT DOCUMENT

## Document control

### Version Control

All revisions made to this document are listed below in chronological order.

Version	Author(s)	Notes
1.0	Mishaal Hajiani, Areeba Shamsi and Sabika Nasir	Original Draft

### Purpose

A business requirements document (BRD) details the business solution for a project including the documentation of customer needs and expectations. This BRD describes the software functional and nonfunctional requirements for release 1.0 of the Foodies' Nest. The document is intended to be used by the members of the project team that will implement and verify the correct functioning of the system. Unless otherwise noted, all requirements specified here are high priority and committed for release 1.0.

### Objectives

This system helps to manage and run the canteen business systematically. In this management system, we will provide a website that can be used by the customers to order food. In addition to this, it should also maintain the daily expenses incurred by the staff. It's an automated system, a technique of ordering foods online applicable for university students. Several inquiry facilities should also be provided to view the expenses incurred/ planned menus/cash payment etc. When the customer visits the ordering webpage, they are presented with an interactive and up-to-date menu, with all available options and prices based on the selected options. After selection, the item is then added to their cart, the customer can review the details before checking out. This provides an instant visual confirmation of what was selected and guarantees that the eatery in the request is what was proposed. This provides an instant visual confirmation of what was selected and guarantees that the eatery in the request is what was proposed.

### Background

In today's age of fast food and take-out, many canteens have chosen to focus on quick preparation and speedy delivery of orders rather than offering a rich dining experience. Until very recently, all of these delivery orders were placed to the waiters or over the phone, but there are many disadvantages to this system, including the inconvenience of the customer needing to have a physical copy of the menu, lack of a visual confirmation that the order was placed correctly, and the necessity for the canteen to have an employee answering the phone and taking orders. The food items are prepared by house-wives or young adults who wish to earn this way. This will be a platform to create a positive impact on country's economy by providing women with an easy way to run their own business in their domain. For that purpose, The Foodies Nest team has already conducted many surveys related to food consumption in Pakistan, primarily targeting the house-wives and young individuals. Our target here is to not just ease the work of university's canteens, but also students and workers who don't have access to home-made food, and excessively consume restaurant meals. In addition, there are mothers and wives who are looking for ways to sell the food they make.

**Name Proposal** Considering the health risks associated with the restaurant food, and as an attempt to open opportunities for women and individuals selling food, we came up with a project: **Foodies' Nest-happiness is homemade**



## Business Drivers

Stakeholder	Function
Customer	Place an order and pay for the order
Home Chef	Upload and create new Food product
Canteen Owner	Maintains Customer's order and canteen's inventory management.
Admin	Deals with Canteen's and Chef's Payment. Maintains café, chef and student details providing them the access.

### In Scope Functionality

Overall scope summarized as:

- Cash Management
- Crowd Management
- Materials Management
- Cost Management
- Automation

### Out of Scope Functionality

- ❑ Excel export has not been developed for food items due to some criticality
- ❑ Payment through banking system not supported yet.

### Product Perspective

This System helps the individual's food to be managed more effectively and efficiently by digitizing meal ordering, billing and inventory control. The system processes transaction and stores the resulting data. Commission is applied on the food item and profit can be calculated helping the admin/manager to make appropriate business decisions.

### Constraints

- Impending changes to privacy regulations may impact data dictionary design.
- Timeline for enterprise platform updates will impact execution of testing plan.

### Issues

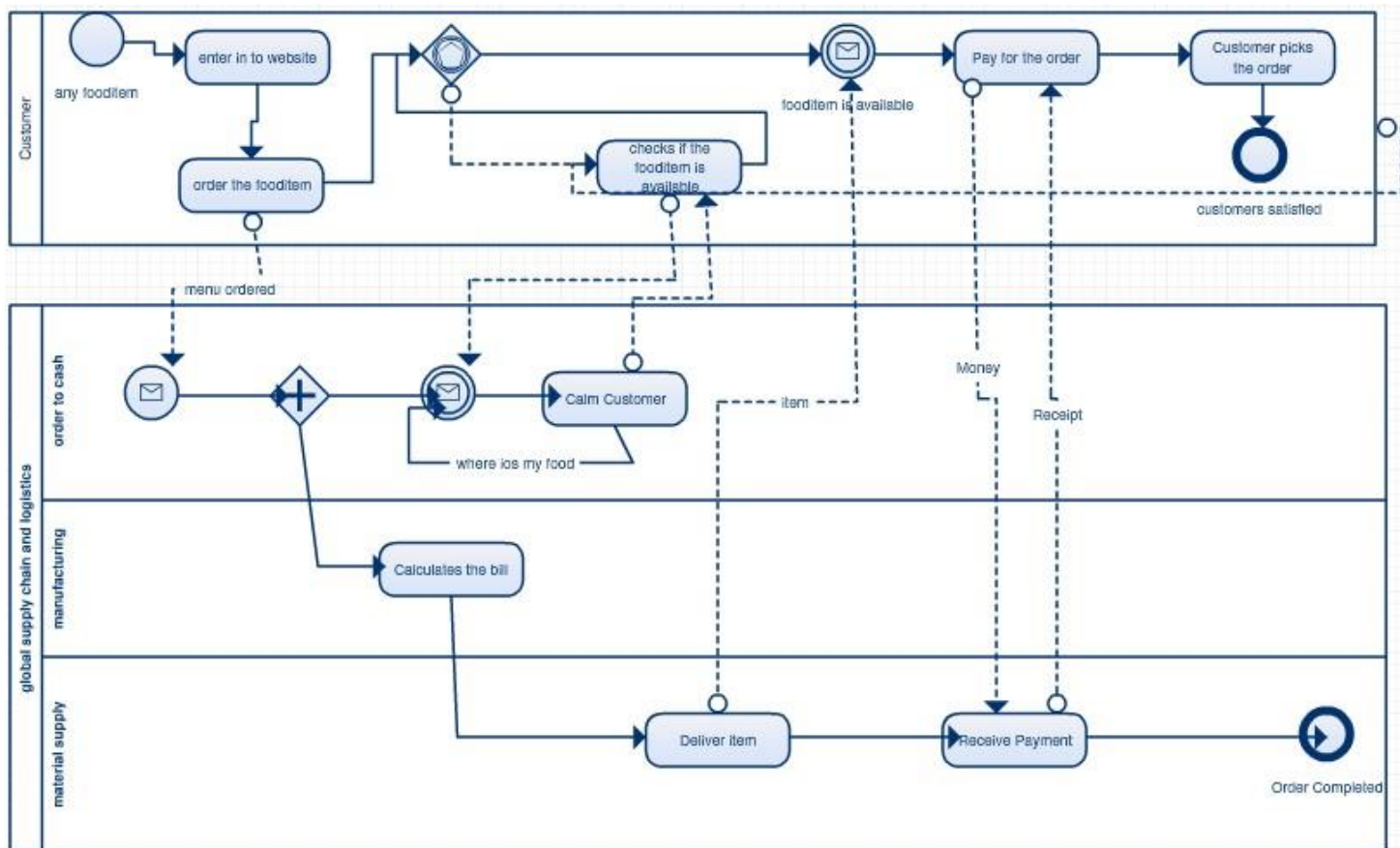
- Requires an active internet connection

### Challenges

- Maintaining multiple locations of canteen consolidated utilization is extremely tedious
- Re-ordering of things to be managed to stock at correct time.
- Subsidized food pursuit

## Business Process Overview

### Current business Process



## Business Requirements

The requirements in this document are prioritized as follows:

Value	Rating	Description
1	Critical	This requirement is critical to the success of the project. The project will not be possible without this requirement.
2	High	This requirement is high priority, but the project can be implemented at a bare minimum without this requirement.
3	Medium	This requirement is somewhat important, as it provides some value, but the project can proceed without it.
4	Low	This is a low priority requirement, or a “nice to have” feature, if time and cost allow it.

## Functional Requirements

Req#	Priority	Description
F-01	1	User shall create an account
F-02	1	User shall login
F-03	1	The system shall let a user who is logged into the Foodies' nest System place an order.
F-04	1	The system shall confirm that the User is registered
F-05	1	If the User is not registered, the system shall give the user options to sign up now
F-06	1	The system shall show user today's menu
F-07	1	The system shall display the remaining available food items in the menu.
F-08	2	The system shall allow the user to indicate the number of units of each menu item that he/she wishes to order.
F-09	2	If the User does not confirm the meal order, the User may either edit or cancel the order.
F-10	1	A chef shall sign in to upload or create a new food product
F-11	1	The system shall confirm if the chef is registered or not

F-12	1	If the Chef is not registered, the system shall give the user options to sign up now
F-13	3	The shall tell the system how many items they are going to commit to sell
F-14	1	A canteen owner shall login to view the total sales done today
F-15	1	The system shall let the canteen owner who logged into the Foodies' nest System to accept the user's order.

## Non-Functional Requirements

### Performance Requirements

- PE-1: The system shall accommodate all users during the peak usage time window of 9.00am to 5.15 pm local time
- PE-2: All Web pages generated by the system shall be fully downloadable in no more than 10 seconds over a 40KBps modem connection.
- PE-3: Responses to queries shall take no longer than 7 seconds to load onto the screen after the user submits the query.

### Safety Requirements

No safety requirements have been identified.

### Security Requirements

- SE-1: Users shall be required to log in to the Cafeteria Ordering System for all operations except viewing a menu.
- SE-2: The system shall permit cafeteria staff members to view the sales inventory report or menu of their respective café only. Rest information to be hided.
- SE-3: The system shall permit Users to view only their own placed orders, not orders placed by other Users.

### Software Quality Attributes

- Availability-1: The Cafeteria Ordering System shall be available to users on the corporate Intranet and to dial-in users 99.9% of the time.
- Robustness-1: If the connection between the user and the system is broken prior to an order being either confirmed or canceled, the system is responsible to enable the user to recover an incomplete order.

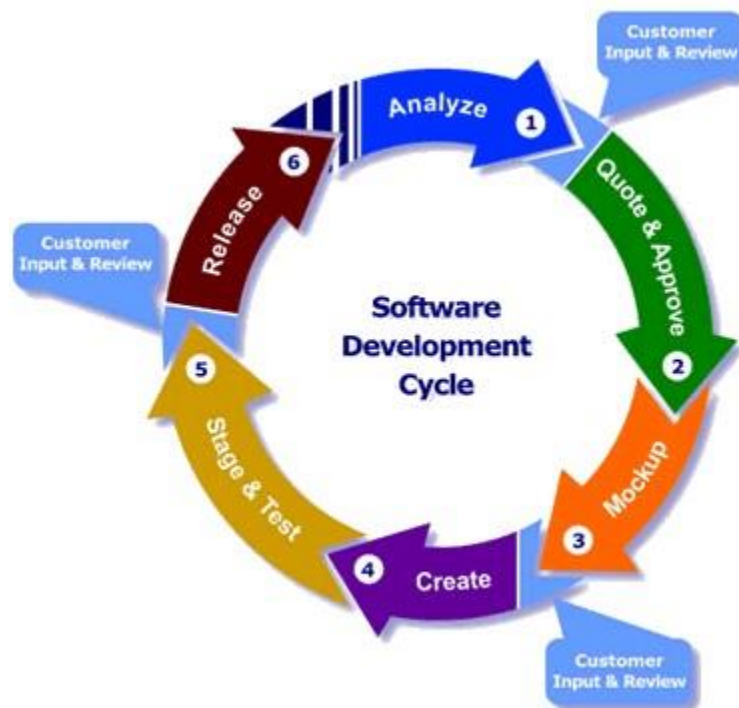
## Technology

Envisaging mode of deliverance to the audience: Web browsing

## Budget and Timescale

Timescale:

To develop a software, a software is expected to undergo the following process



*Typical Software Development Process*

Rough budget for this application:

The software is still under development phase, to compute the estimated cost along with other parameters, we require parameter called Lines of code (LOC). Since, the coding is under process; LOC cannot be determined.

## Wireframe

Refer to the separate document. We have created a prototype using the software 'Justinmin

# Software Requirement Specification



## Revision History

Version	Author(s)	Notes
1.0	Mishaal Hajiani, Areeba Shamsi and Sabika Nasir	Original Draft

## Introduction

### Document Purpose

This document represents a detailed explanation of the features, objectives, user interface, and describes the functional and non-functional requirements of the application Foodie's Nest. Moreover, this document will also provide the description of how the system will perform. This, document is intended to be used by the stakeholders as well as the developers of the system to verify the correct functioning of the whole framework.

### Product Scope

This system will allow the user to order the food from the cafeteria on-line on campus and also permits the process of selling the food of home chefs who want to get empowered by offering nutritious and sterile home-made nourishment on the campus. This system will provide an easy access of food to the students as well as the staff of the IBA. The system will be accessed through an app, through which user on the campus can scan and access the food list, and place an order online on the cafes. The detailed project description is available in the abstract of this project submitted.

### Intended Audience and Document Overview

The SRS is intended to be used by all the team members. This document provides sequential overview of the whole project like what is the system about and how is it implemented.

## Overall Description

### Product perspective

The Foodie's Nest is a new system that replaces the manual ordering of food from the cafes of the IBA. This system, provides a platform to the underprivileged ladies to offer their nutritious and sterile home-made food on the campus and procure earning this way. Moreover, this system helps the cafe owners to manage their café more effectively and efficiently by computerized meal ordering, billing and inventory control.

The system will store the data gathered by processing the sale of the items, through which reports can be generated to analyze which chef is providing more sale and hence, helps to make appropriate business decisions.

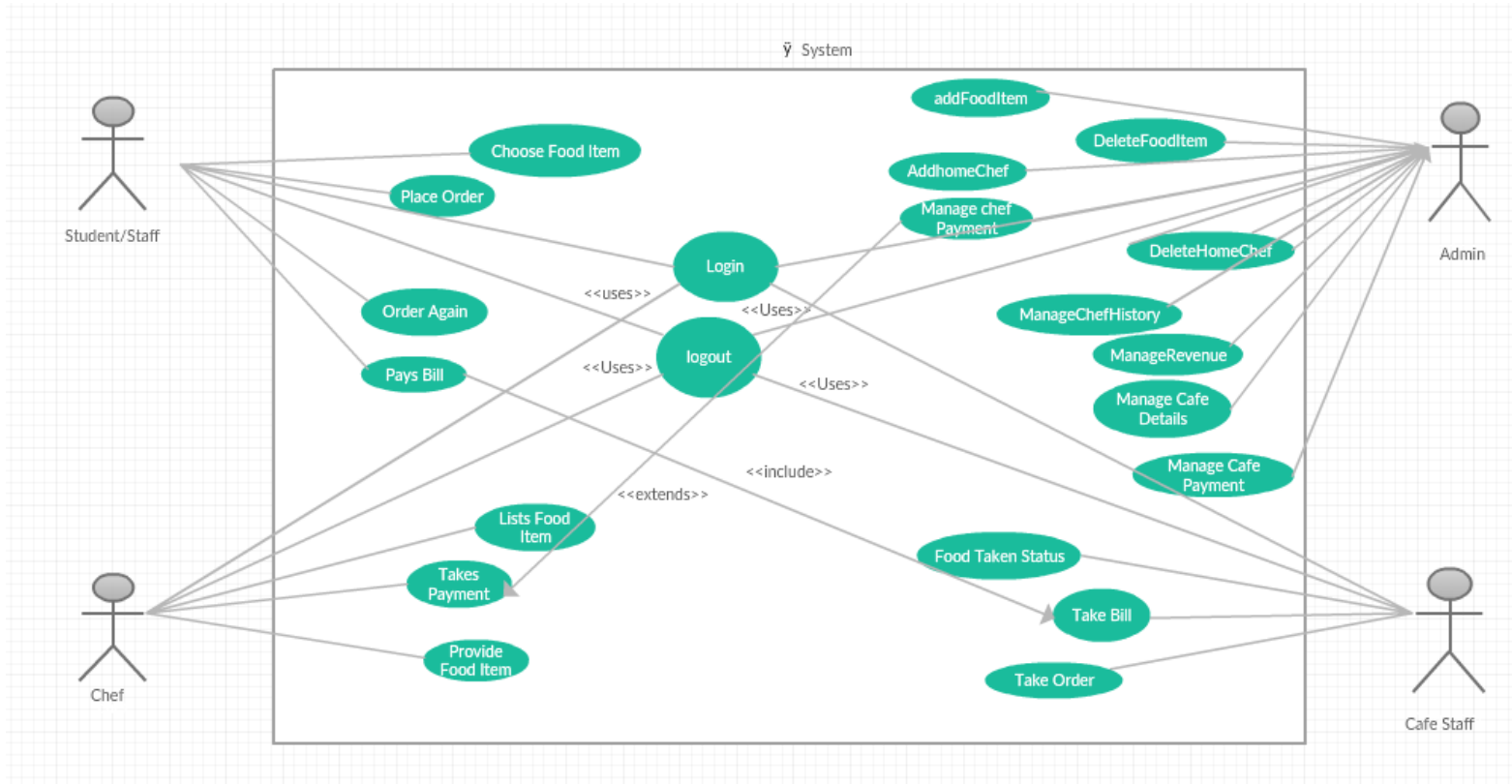
This system creates an ease for the students and staff of the IBA, as they are only required to request the website page, they are given an intuitive and up-to-date menu, with every single accessible alternative and costs in view of the selected choices.

### Product Functionality

The major functions of the system:

- access via Website
- Login/logout
- Available food in the menu
- Order placement online
- chef details
- Account management
- Café details

## Use Case



## Operating Environment

- Operating system (windows)
- Language (c# ,css3, html5,SQL)
- Database (SQL developer)
- Browser (any of opera, chrome, Mozilla etc.)
- Web server(IIS express)
- Software development kit(JDK)

## Design and Implementation Constraints

- Impending changes to privacy regulations may impact data dictionary design.
- Timeline for enterprise platform updates will impact execution of testing plan.

## User Documentation

The system will provide the user who will access the system for the first time with the template of steps describing on how to access the menu, order the meals and make changes. The user shall not be able to place order through this template. Furthermore, the system shall provide an online video too for a better understanding.

## Assumptions and Dependencies

- AS-1: The cafe is open for breakfast and lunch every day in which user is expected to be on the campus.
- DE-1: The operation of the Foodie's Nest depends on changes being made in the Cafe Inventory System to update the availability of food items.

## External interfaces Requirements

### User Interfaces

The system shall provide a help link from each displayed HTML page to explain how to use that system. The user interface will be implemented using any web browser. The interface should be user friendly so that user shall be able to access the menu and place an order.

### Hardware Interfaces

- Intel® Core™ i7-6500U CPU @ 2.50GHz, 2592 MHz, 2 Core(s), 4 Logical Processor(s)
- RAM (4GB)
- Hard disk (20GB)
- Keyboard (122 keys)

### Software Interfaces

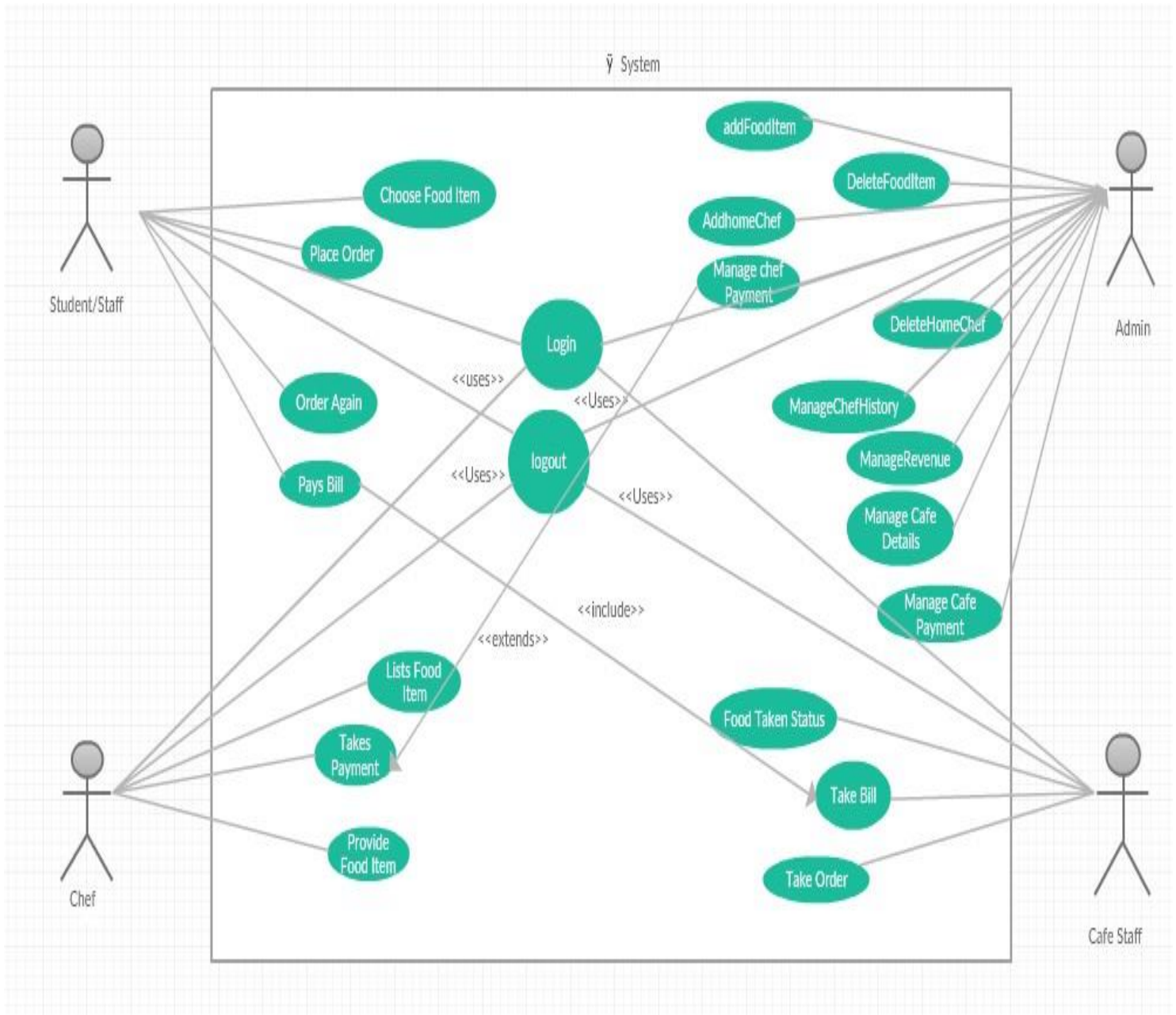
To store the information necessary for the Foodie's Nest to operate, the Foodie's Nest will interface with the DBMS (Database Management System). This DBMS interface should be able to provide, with minimum latency and on request, all the data related to the day Menu, users (login details, chef details, café details). Additionally, the DBMS must take all the data provided to it by the Foodie's Nest and archive it. This data includes the records of all the transactions and orders executed by the system. All the data must be stored by the DBMS such that it can be used for accountability and accounting.

### Communications interface

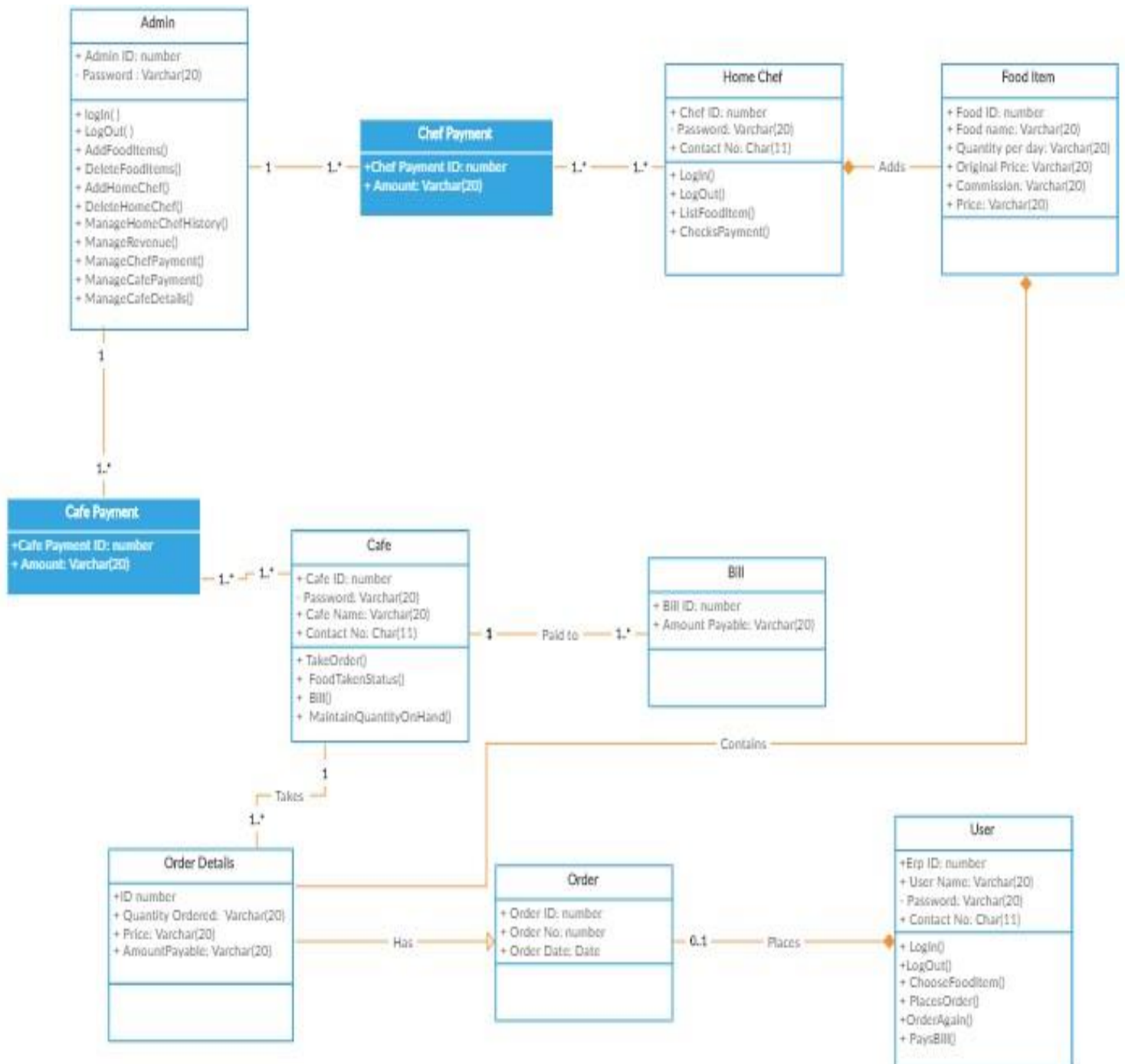
CI-1: The Foodie's Nest System shall generate an online invoice to the user (Student/staff) to confirm the acceptance of an order, price, and delivery instructions.

CI-2: The Foodie's Nest System shall generate a message to the user (student/staff) to report Any problems with the meal order or delivery after the order is accepted.

## USE CASE

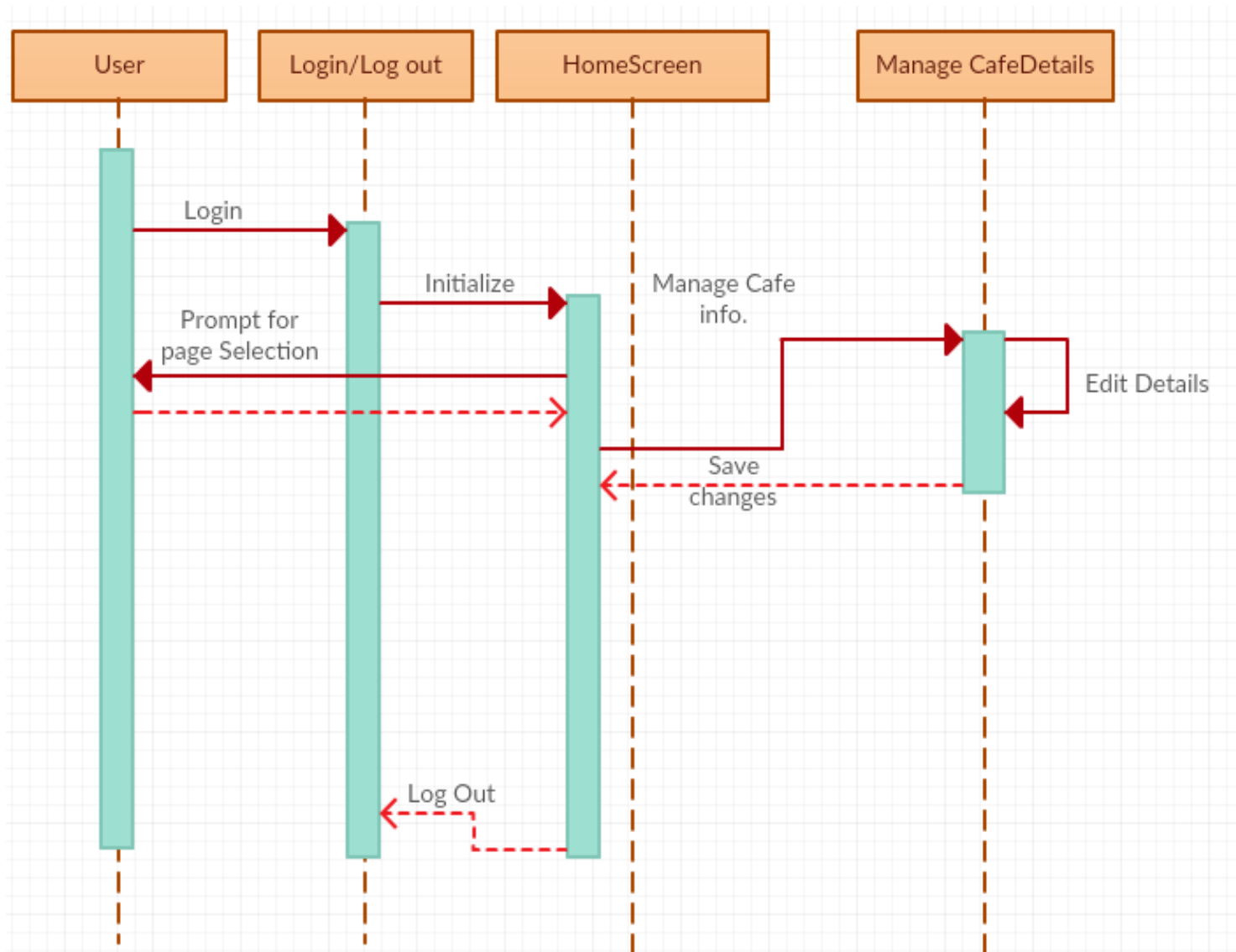


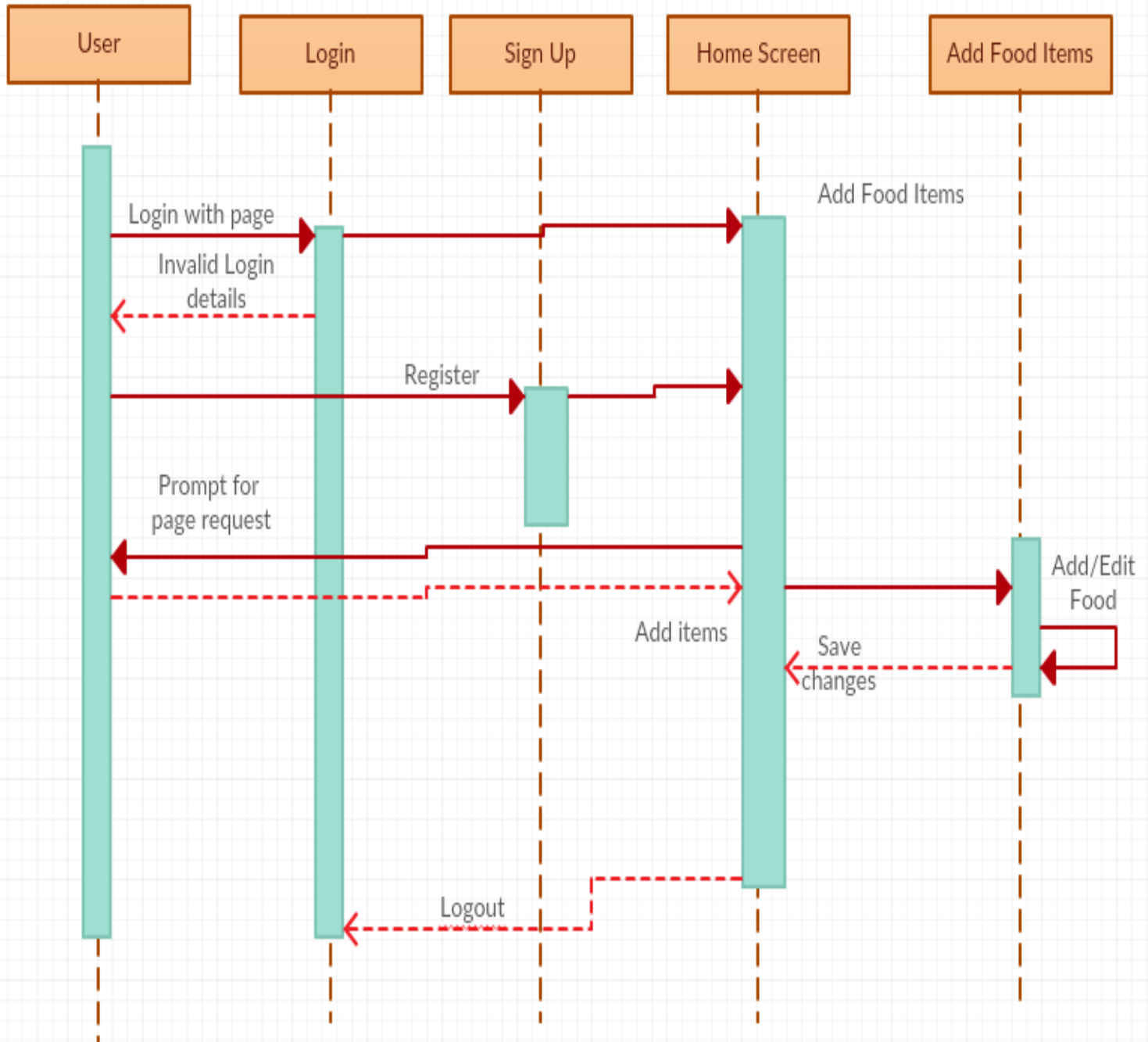
## CLASS DIAGRAM

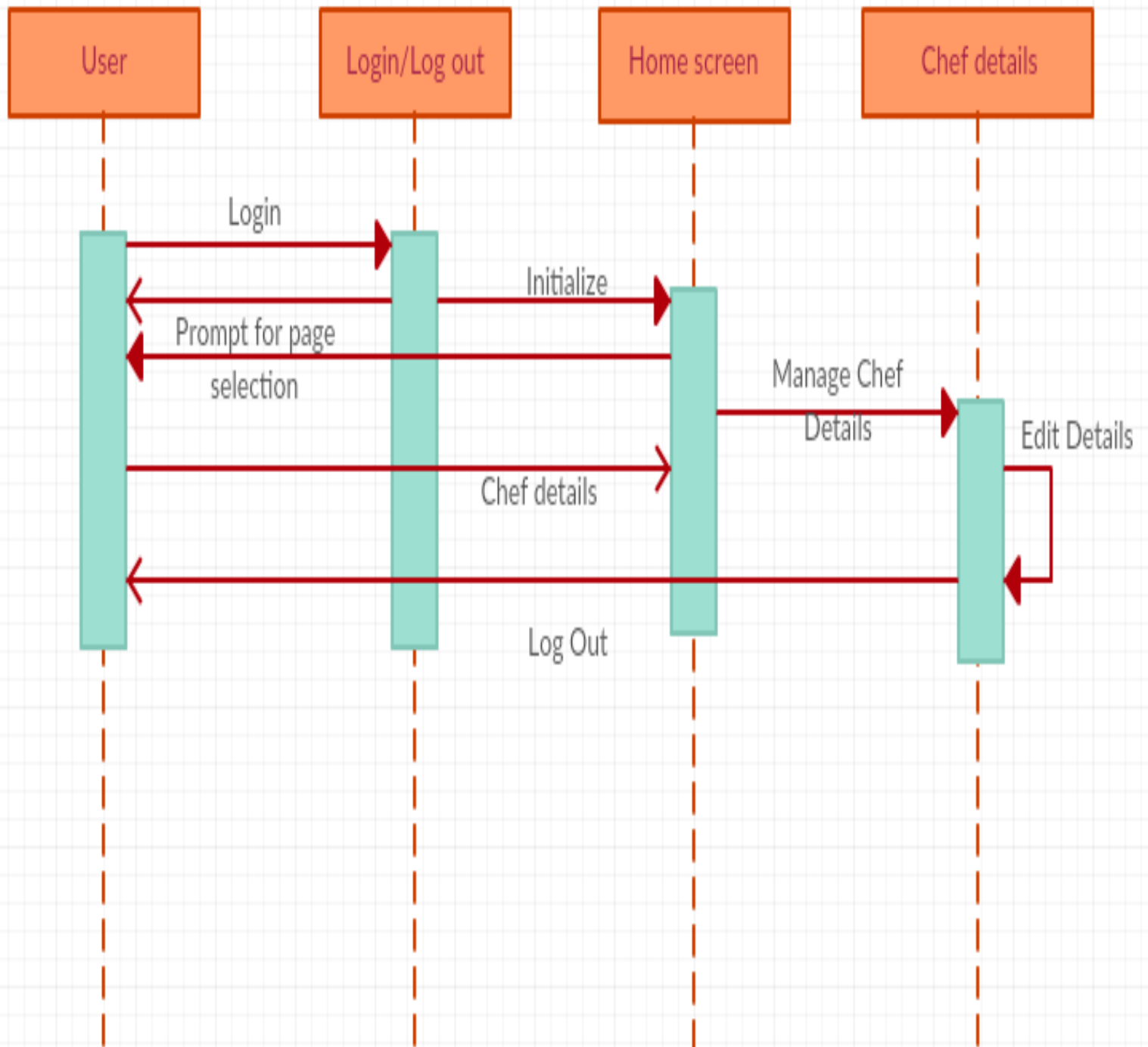


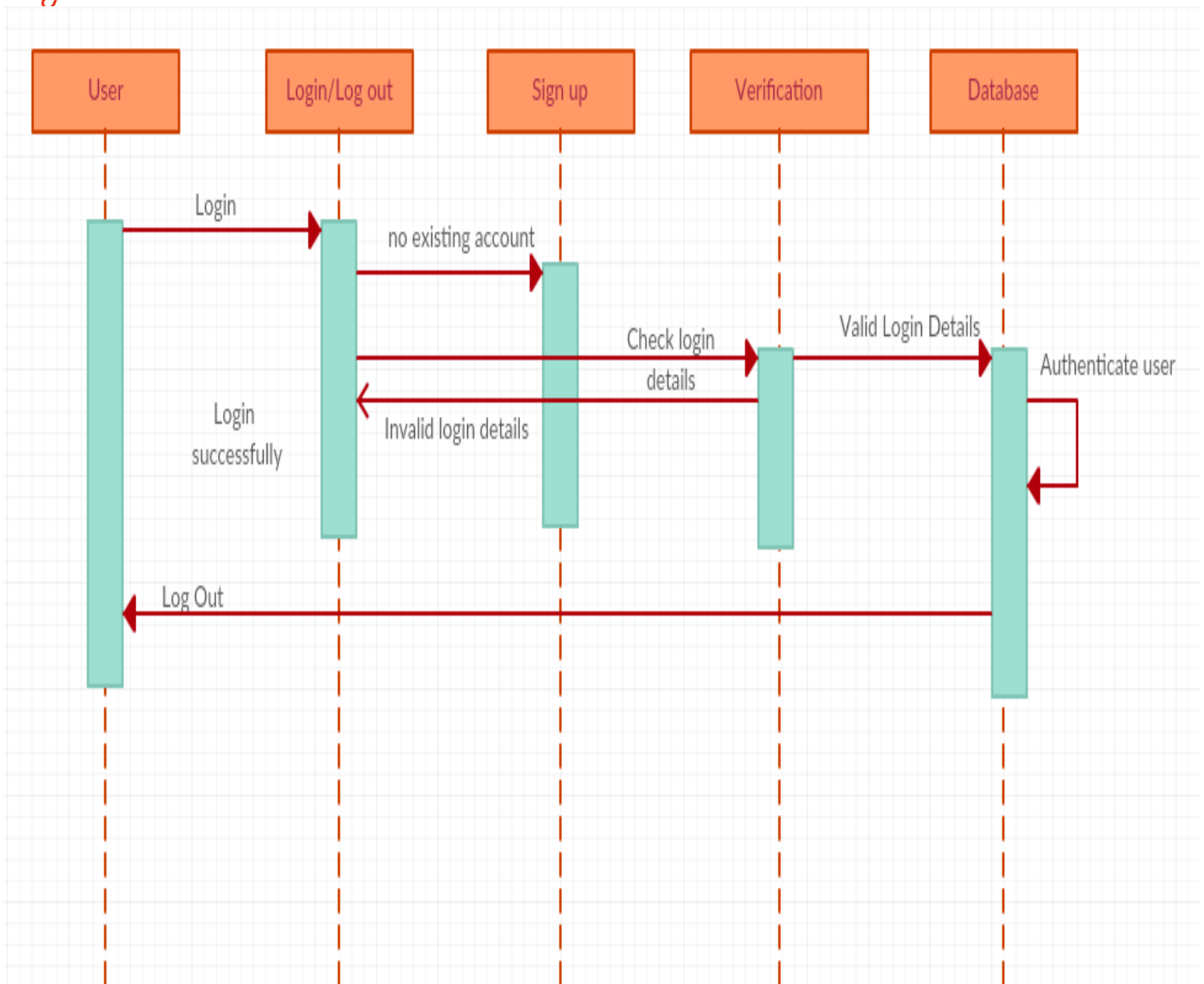


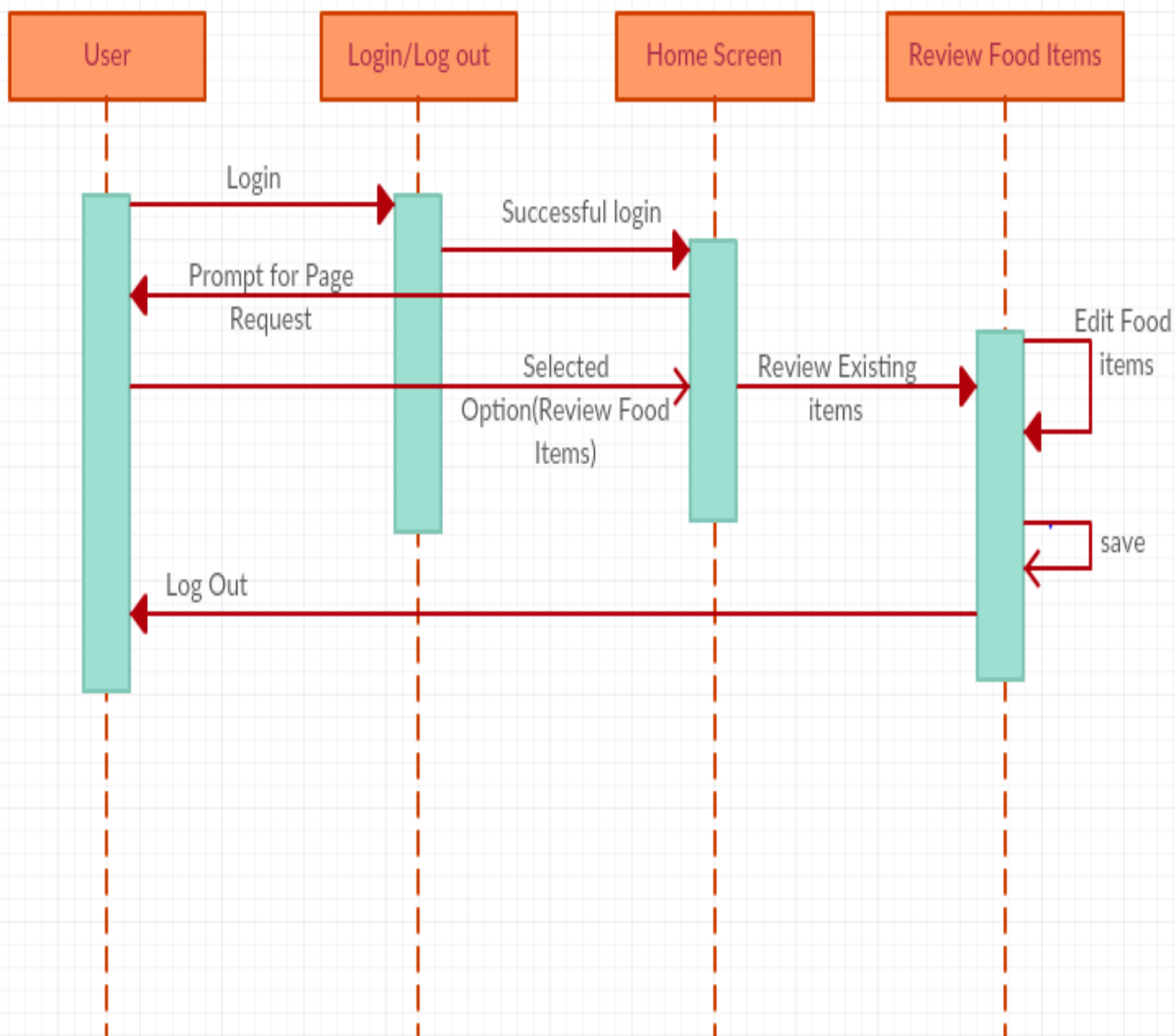
## Sequence Diagram

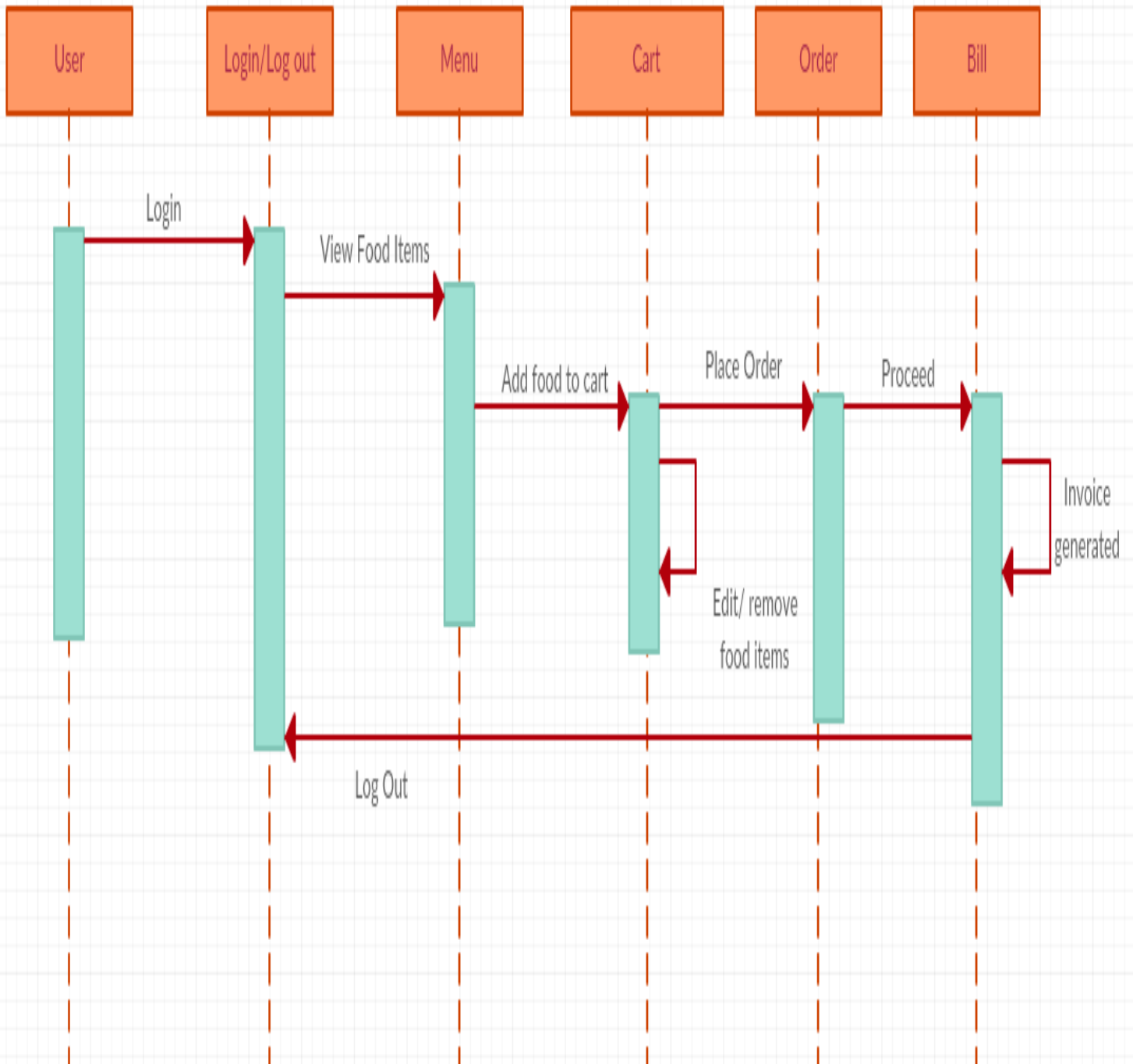












## FOODIES' NEST ESTIMATION METRICS

### 1. Calculation of the unadjusted function points(UFP)

The table below shows the function types and the weighting factors for the varying complexities.

Function type	Simple	Average	Complex
Internal Logical File	7	10	15
External Interface File	5	7	10
External Input	3	4	6
External Output	4	5	7
External Inquiry	3	4	6

Using these definitions above, the files types in my project can be counted as follows:

		Weighting Factor			Count
		Simple	Average	Complex	
Inputs	Admin Login	3			64
	User Login	3			
	Chef Login	3			
	Canteen Login	3			
	User Sign Up		4		
	Welcome Admin		4		
	Chef Sign Up		4		
	Edit Order		4		
	Food item Details		4		
	Menu Access	3			
	Place Order	3			
	Dish listing form		4		
	Review existing dish			6	
	Admin Café details	3			
	Welcome Canteen	3			
	Welcome Chef	3			
	Menu		4		
	View Chef Payments		4		
Outputs	Invoice		5		10
	Sales Report		5		
Internal Logical File	Account Management		10		10
External Inquiry	Cart		4		33
	Chef Details			6	
	Contact Us	3			
	Home Page		4		
	Review Existing Dish		4		
	Chef Logout	3			
	Café Logout	3			
	User Logout	3			
	Admin Logout	3			
External Interface File					0
Total UFP					117



## 2. Calculate Adjusted Function Point

Number	Complexity Weighting Factor	Value
1	Will the application use data communications?	4
2	Are data or functions distributed?	3
3	Are there specific performance objectives that must be met?	4
4	Will the application run on a heavily used configuration requiring special design considerations?	1
5	Will the transaction rate of the application be high?	4
6	On-line data entry	5
7	Will the application be designed for end-user efficiency?	5
8	Will there be on-line updates?	5
9	Is complex processing logic involved?	2
10	Is there an intent to provide usability for other applications?	1
11	How important are installation ease and conversion?	0
12	How important is operational ease?	4
13	Will the application be accessed from multiple sites?	0
14	Is there an intent with the design to facilitate change?	3
	<b>Total complexity adjustment value</b>	<b>41</b>

Total Unadjusted Function Points (UFP) = 117

Product Complexity Adjustment (PC) =  $0.65 + (0.01 * 41) = 1.05$

Total Adjusted Function Points (FP) = UFP \* PC = 122.85

Language Factor (LF) for SQL, visual Basic, HTML (assuming average) = 40 ,47 and 53 respectively.

Lines of Code (LOC) =  
SQL=  $122.85 \times 40 = 4914$  LOC  
Visual Basic=  $122.85 \times 47 = 5773.95$  LOC  
HTML=  $122.85 \times 53 = 6511.05$  LOC  
Total= 17199 LOC= 17.2 KLOC

## Analysis

Assuming the mode to be organic

- a) Effort=  $a \times (KLOC)^b = 2.4 \times (17.2)^{1.05} = 47.59 \sim 47$  person-month
- b) Duration=  $a \times (Effort)^b = 2.5 \times (47.59)^{0.38} = 10.849$  months
- c) Average Staffing= Effort/Duration =  $47.59/10.849 = 4.407$  person

Average Productivity= LOC/Effort=  $17199/47.59 = 361$

## TEST CASES

<b>Test Case ID</b>	01	<b>Test Case Description</b>	Test the Login Functionality of Admin		
<b>Created By</b>	Mishaal	<b>Reviewed By</b>	Sabika	<b>Version</b>	1

<b>Tester's Name</b>	Areeba	<b>Date Tested</b>	December 12, 2018	<b>Test Case (Pass/Fail/Not Executed)</b>	Pass
----------------------	--------	--------------------	-------------------	---	------

S #	Prerequisites:
1	Access to Browser
2	
3	
4	

S #	Test Data
1	Admin id= 111
2	Password=aaaa
3	
4	

<b>Test Scenario</b>	Verify on entering valid userid and password, the admin can login
----------------------	---

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to http://localhost:53528/adminlogin.aspx	Site should open	As Expected	Pass
2	Enter Adminid & Password	Credential can be entered	As Expected	Pass
3	Click Login	Admin is logged in	As Expected	Pass
4				

Admin ID

Password

Login

Test Case ID	02	Test Case Description	Test the dish listing form		
Created By	Mishaal	Reviewed By	Sabika	Version	1

Tester's Name	Areeba	Date Tested	December 12, 2018	Test Case (Pass/Fail/Not Executed)	Pass
---------------	--------	-------------	-------------------	------------------------------------	------

S #	Prerequisites:
1	Access to Browser
2	
3	
4	

S #	Test Data
1	Chef Id= 1
2	
3	
4	
5	
6	

**Test Scenario**

Asks homechef to fill the dish listing form and inserts the food item details in database

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to http://localhost:53528/dishlistingform.aspx	Site should open	As Expected	Pass
	Food ID	Auto generate	As Expected	
2	Enter food name		As Expected	Pass
3	Enter quantity		As Expected	Pass
4	Enter Price		As Expected	
5	Upload image		As Expected	
6	Enter date		As Expected	
7	Chef id	Retrieved from login page	As Expected	
8	Click submit	submitted	As Expected	

***DISH LISTING FORM***

Food ID

Food name

Quantity

Price

Image

Choose File No file chosen

Date

Chef ID

Submit

Test Case ID	03	Test Case Description	Test the Review existing dishes		
Created By	Mishaal	Reviewed By	Sabika	Version	1

Tester's Name	Areeba	Date Tested	December 12, 2018	Test Case (Pass/Fail/Not Executed)	Pass
---------------	--------	-------------	-------------------	------------------------------------	------

S #	Prerequisites:
1	Access to Browser
2	
3	
4	

S #	Test Data
1	Food Id= 1
2	Food name= Biryani
3	Quantity= 15
4	Price=200
5	Date=09/12/2018
6	Chef id=1

**Test Scenario** This screen is for a particular chef who selects the end date for reviewing the food items of the respective end date month

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to http://localhost:53528/reviewdish listing.aspx	Site should open	As Expected	Pass
2	Chef ID	Retrieved from login form	As Expected	Pass
3	View food id	Food id of Retrieved chef id	As Expected	Pass
4	View Food id	Food id of Retrieved chef id	As Expected	Pass
5	Edit Price	Edited	As Expected	Pass
6	Edit quantity	Edited	As Expected	Pass
7	Select End date	End date is selected from the drop down	As Expected	Pass



Chef ID  End Date

Food_id	Food_name	Quantity_per_day	Original_price	Date	line_total	End_date	
1	Biryani	15	200	2018-12-09	3000	12/31/2018 12:00:00 AM	<a href="#">Edit</a>
Payment= Rs. 66000							

Test Case ID	04	Test Case Description	Test the Chef details		
Created By	Mishaal	Reviewed By	Sabika	Version	1

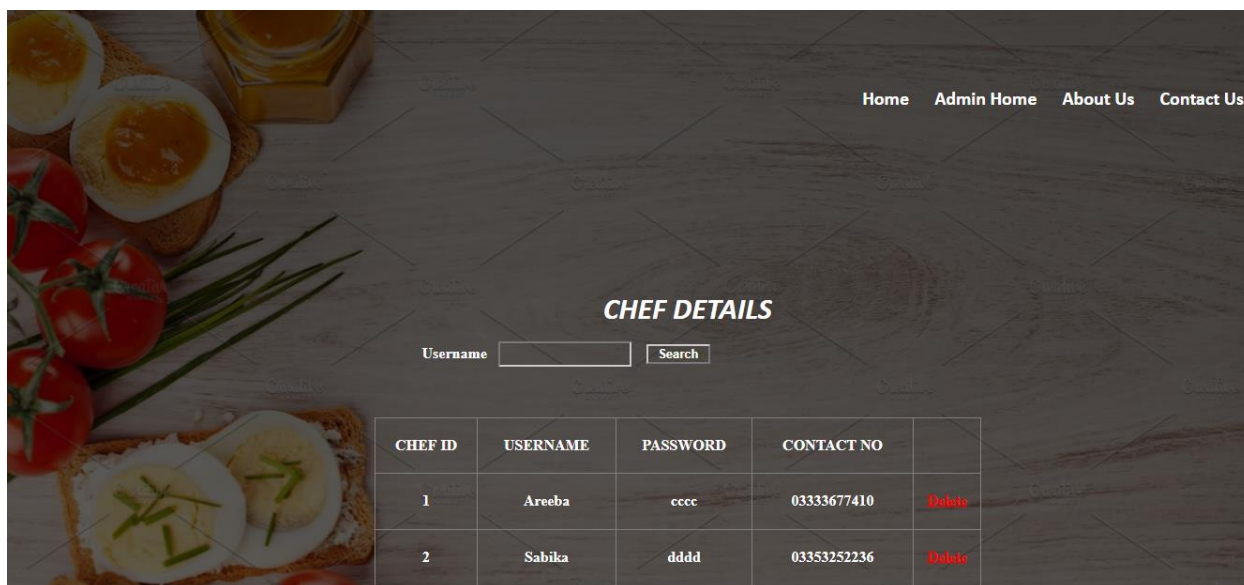
Tester's Name	Areeba	Date Tested	December 12, 2018	Test Case (Pass/Fail/Not Executed)	Pass
---------------	--------	-------------	-------------------	------------------------------------	------

S #	Prerequisites:
1	Access to Browser
2	
3	
4	

S #	Test Data
1	Username = Areeba
2	
3	
4	
5	
6	

**Test Scenario** Verifying whether the search is retrieving the results and displaying the result of the key word inserted

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to http://localhost:53528/chefdetails.aspx	Site should open	As Expected	Pass
2	Search chef though username	Search successful	As Expected	
3	Chef id	Reteived from database by using username	As Expected	Pass



Test Case ID	05	Test Case Description	Test the Chef details		
Created By	Mishaal	Reviewed By	Sabika	Version	1

Tester's Name	Areeba	Date Tested	December 12, 2018	Test Case (Pass/Fail/Not Executed)	Pass
---------------	--------	-------------	-------------------	------------------------------------	------

S #	Prerequisites:
1	Access to Browser
2	
3	
4	

S #	Test Data
1	Chef id =1
2	End date=09-12-2018
3	
4	
5	
6	

<b>Test Scenario</b>	Verifying whether the search is retrieving the results and displaying the result of the key word inserted
----------------------	---

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to http://localhost:53528/chefdetails.aspx	Site should open	As Expected	Pass
2	Chef id	Retrieved from above	As Expected	Pass
3	Select end date	End date is selected from the drop down	As Expected	Pass
4	Search for food items	Retrieves food items of the specified chef id	As expected	Pass

Chef ID		End Date		mm/dd/yyyy		Search	
FOOD ID	FOOD NAME	QUANTITY PER DAY	ORIGNAL PRICE	CHEF ID	LINE TOTAL	DATE	END DATE
1	Biryani	15	200	1	3000	2018-12-09	12/31/2018 12:00:00 AM
2	Burger	10	200	2	2000	2018-12-12	12/31/2018 12:00:00 AM
3	mutton biryani	10	200	1	2000	2018-12-13	12/31/2018 12:00:00 AM
					Payment=Rs.	140000	

Test Case ID	06	Test Case Description	Test the Chef details		
Created By	Mishaal	Reviewed By	Sabika	Version	1

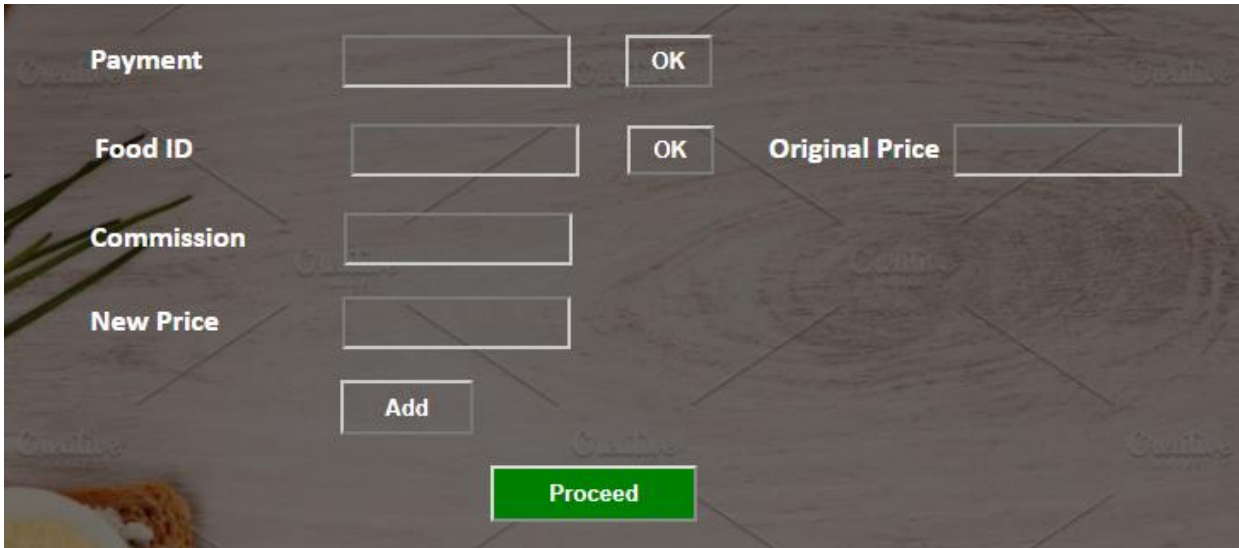
Tester's Name	Areeba	Date Tested	December 12, 2018	Test Case (Pass/Fail/Not Executed)	Pass
---------------	--------	-------------	-------------------	------------------------------------	------

S #	Prerequisites:
1	Access to Browser
2	
3	
4	

S #	Test Data
1	Food_id=1
2	Original price =200
3	
4	
5	
6	

**Test Scenario** Implementing insert functionality

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to http://localhost:53528/chefdetails.aspx	Site should open	As Expected	Pass
2	Food id	Retrieved from above	As Expected	
3	Original Price	Retrieved from database by using Food id	As Expected	Pass
4	Inserts commission	Inserts successfully	As Expected	Pass
	Calculates new price using original price and commission.	Calculation successful	As Expected	Pass



Payment  OK

Food ID  OK Original Price

Commission

New Price

Add

Proceed



Test Case ID	07	Test Case Description	Test the Chef details		
Created By	Mishaal	Reviewed By	Sabika	Version	1

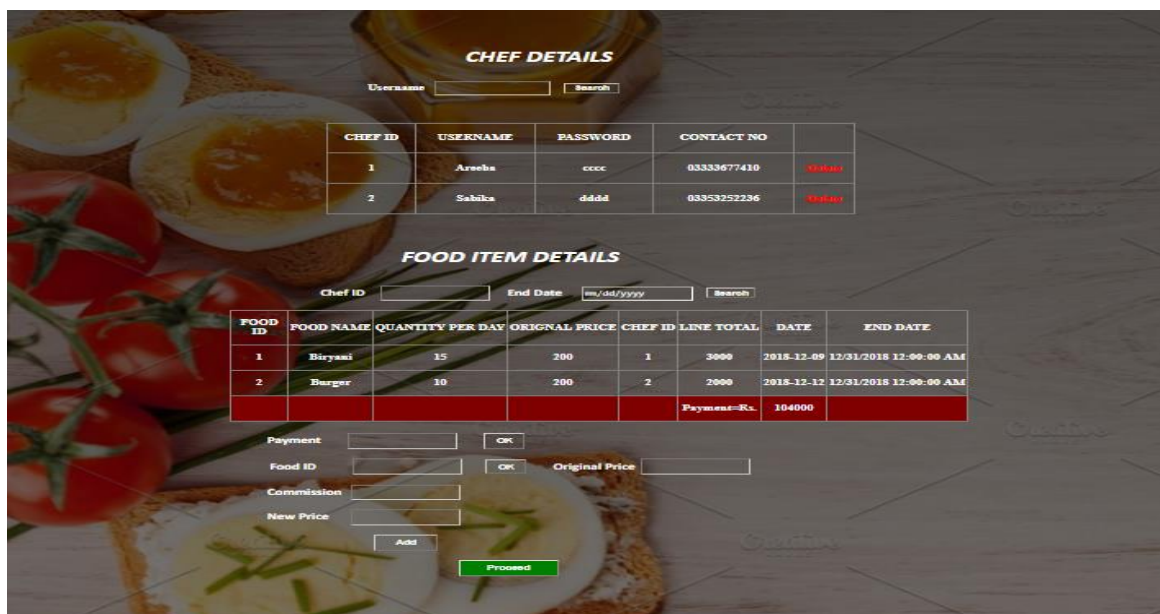
Tester's Name	Areeba	Date Tested	December 12, 2018	Test Case (Pass/Fail/Not Executed)	Pass
---------------	--------	-------------	-------------------	------------------------------------	------

S #	Prerequisites:
1	Access to Browser
2	
3	
4	

S #	Test Data
1	Payment =104000
2	Chef id=1
3	
4	
5	
6	

**Test Scenario** Verifying whether the search is retrieving the results and displaying the result of the key word inserted

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to http://localhost:53528/chefdetails.aspx	Site should open	As Expected	Pass
2	Chef id	Retrieved from above	As Expected	Pass
3	Total payment	Retrieved from a calculated field	As Expected	Pass
4	Click ok	Records added to the database	As Expected	Pass



The screenshot shows two web pages. The top page, titled 'CHEF DETAILS', has a search bar for 'Username' and a 'Search' button. Below it is a table with columns: CHEF ID, USERNAME, PASSWORD, CONTACT NO, and a status column. The table contains two records: one for Areeba (id 1, password cccc) and one for Sabika (id 2, password dddd). The bottom page, titled 'FOOD ITEM DETAILS', has a search bar for 'Chef ID' and 'End Date' with a 'Search' button. Below it is a table with columns: FOOD ID, FOOD NAME, QUANTITY PER DAY, ORIGINAL PRICE, CHEF ID, LINE TOTAL, DATE, and END DATE. The table contains two records: Biryani (id 1, quantity 15, price 200) and Burger (id 2, quantity 10, price 200). At the bottom of the page, there is a 'Payment' field with a value of 104000, and buttons for 'Add' and 'Proceed'.

Test Case ID	08	Test Case Description	Test the Food Item details		
Created By	Mishaal	Reviewed By	Sabika	Version	1

Tester's Name	Areeba	Date Tested	December 12, 2018	Test Case (Pass/Fail/Not Executed)	Pass
---------------	--------	-------------	-------------------	------------------------------------	------

S #	Prerequisites:
1	Access to Browser
2	
3	
4	

S #	Test Data
1	Food id= 1
2	Commission=0.25
3	Newprice=250
4	
5	
6	

<b>Test Scenario</b>	Implements edit functionality
----------------------	-------------------------------

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to <a href="http://localhost:53528/fooditemdetails.asp">http://localhost:53528/fooditemdetails.asp</a>	Site should open	As Expected	Pass
2	food id		As Expected	Pass
3	Edits commission	Updates successfully	As Expected	Pass
4	Edits new price	Updates successfully	As Expected	Pass
5	Click proceed	Databsse is updated successfully	As Expected	Pass

[Admin Home](#)

### COMMISSION

Food_id	Commission (%)	Price_per_dish(Rs)	
1	0.25	250	<a href="#">Edit</a>

3 0.35 270 [Edit](#)

[Proceed](#)

Test Case ID	09	Test Case Description	Test the Login Functionality of Cafe		
Created By	Mishaal	Reviewed By	Sabika	Version	1

Tester's Name	Areeba	Date Tested	December 12, 2018	Test Case (Pass/Fail/Not Executed)	Pass
---------------	--------	-------------	-------------------	------------------------------------	------

S #	Prerequisites:
1	Access to Browser
2	
3	
4	

S #	Test Data
1	Cafe id= 222
2	Password=bbbb
3	
4	

<b>Test Scenario</b>	Verify on entering valid userid and password, the Café can login
----------------------	--

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to http://localhost:53528/cafe/login.aspx	Site should open	As Expected	Pass
2	Enter Cafeid & Password	Credential can be entered	As Expected	Pass
3	Click Login	Cafe is logged in	As Expected	Pass

### CAFE LOGIN

Cafe ID	<input type="text"/>
Password	<input type="password"/>
<input type="button" value="Login"/>	

Test Case ID	10	Test Case Description	Test the Order details		
Created By	Mishaal	Reviewed By	Sabika	Version	1

Tester's Name	Areeba	Date Tested	December 12, 2018	Test Case (Pass/Fail/Not Executed)	Pass
---------------	--------	-------------	-------------------	------------------------------------	------

S #	Prerequisites:
1	Access to Browser
2	
3	
4	

S #	Test Data
1	Order id = 45
2	
3	
4	
5	
6	

**Test Scenario** Verifying whether the search is retrieving the results and displaying the result of the order id selected

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to http://localhost:53528/orderdetails.aspx	Site should open	As Expected	Pass
2	Search food order details though order id	Search successful	As Expected	Pass

Food_id	Food_name	Price_per_dish	Quantity_per_day	Total Price
1	Biryani	250	15	250
3	mutton biryani	270	10	270
				520

### CAFE ORDER DETAILS

	Order_id	Order_date	Erp_id
<a href="#">Select</a>	45	1/4/2019 12:00:00 AM	13050
<a href="#">Select</a>	44	12/13/2018 12:00:00 AM	13050
<a href="#">Select</a>	43	12/13/2018 12:00:00 AM	13050
<a href="#">Select</a>	42	12/13/2018 12:00:00 AM	13050
<a href="#">Select</a>	41	12/13/2018 12:00:00 AM	13050
<a href="#">Select</a>	40	12/13/2018 12:00:00 AM	13050
<a href="#">Select</a>	39	12/13/2018 12:00:00 AM	13050
<a href="#">Select</a>	38	12/13/2018 12:00:00 AM	13050

Test Case ID	11	Test Case Description	Test the Login Functionality of Chef		
Created By	Mishaal	Reviewed By	Sabika	Version	1

Tester's Name	Areeba	Date Tested	December 12, 2018	Test Case (Pass/Fail/Not Executed)	Pass
---------------	--------	-------------	-------------------	------------------------------------	------

S #	Prerequisites:
1	Access to Browser
2	
3	
4	

S #	Test Data
1	Chef id= 1
2	Password=cccc
3	
4	

<b>Test Scenario</b>	Verify on entering valid chefid and password, the Chef can login
----------------------	--

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to <a href="http://localhost:53528/cf.aspx">http://localhost:53528/cf.aspx</a>	Site should open	As Expected	Pass
2	Enter Chefid & Password	Credential can be entered	As Expected	Pass
3	Click Login	Chef is logged in	As Expected	Pass

[Home](#) [About Us](#) [Contact Us](#)

### CHEF LOGIN

Chef_id	<input type="text"/>
Password	<input type="password"/>
<a href="#">Login</a>	<a href="#">No account? Register here!</a>

Test Case ID	12	Test Case Description	Test the Login Functionality of a user		
Created By	Mishaal	Reviewed By	Sabika	Version	1

Tester's Name	Areeba	Date Tested	December 12, 2018	Test Case (Pass/Fail/Not Executed)	Pass
---------------	--------	-------------	-------------------	------------------------------------	------

S #	Prerequisites:
1	Access to Browser
2	
3	
4	

S #	Test Data
1	user id= 13050
2	Password=dddd
3	
4	

<b>Test Scenario</b>	Verify on entering valid userid and password, the user can login
----------------------	--

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to http://localhost:53528/studentlogin.aspx	Site should open	As Expected	Pass
2	Enter userid & Password	Credential can be entered	As Expected	Pass
3	Click Login	User is logged in	As Expected	Pass



User ID

Password

Login

[No account? Register here!](#)

Test Case ID	13	Test Case Description	Testing the add functionality of the cart		
Created By	Mishaal	Reviewed By	Sabika	Version	1

Tester's Name	Areeba	Date Tested	December 31, 2018	Test Case (Pass/Fail/Not Executed)	Pass
---------------	--------	-------------	-------------------	------------------------------------	------



S #	Prerequisites:
1	Access to Browser
2	
3	
4	

S #	Test Data
1	
2	
3	
4	

<b>Test Scenario</b>	Selecting the quantity and adding item to the cart
----------------------	--

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to http://localhost:53528/menu.aspx	Site should open	As Expected	Pass
2	Select Quantity of the desired food item	Selected quantity added	As Expected	Pass
3	Click Add to cart	Item added to the cart	As Expected	Pass

## MENU

Food_id: 1	Food_id: 3
Food_name: Biryani	Food_name: mutton biryani
Price_per_dish: 250	Price_per_dish: 270
Quantity: 15 <input type="text"/>	Quantity: 10 <input type="text"/>
	
<input type="button" value="Add to Cart"/>	<input type="button" value="Add to Cart"/>

No of Food Items in your cart: 1 [Show Cart](#)

Test Case ID	14	Test Case Description	Testing the remove functionality of items from the cart		
Created By	Mishaal	Reviewed By	Sabika	Version	1

Tester's Name	Areeba	Date Tested	December 31, 2018	Test Case (Pass/Fail/Not Executed)	Pass
---------------	--------	-------------	-------------------	------------------------------------	------

S #	Prerequisites:
1	Access to Browser
2	
3	
4	

S #	Test Data
1	Food Id= 1
2	Food name= Biryani
3	Quantity= 1
4	Price=250

**Test Scenario**

Removing the selected food item from the cart on the selection of remove button

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to http://localhost:53528/cart.aspx	Site should open	As Expected	Pass
2	Select remove button	Item deleted from the cart	As Expected	Pass

## CART

S.No	Food ID	Food Name	Price	Quantity	Total Price	
1	1	Biryani	250	1	250	<a href="#">Remove</a> <a href="#">Modify</a>
2	1	Biryani	250	1	250	<a href="#">Remove</a> <a href="#">Modify</a>
Total Amount					500	

No of Food Items in your cart: 2

[Checkout](#)



Test Case ID	15	Test Case Description	Testing the edit functionality of items from the cart		
Created By	Mishaal	Reviewed By	Sabika	Version	1

Tester's Name	Areeba	Date Tested	December 31, 2018	Test Case (Pass/Fail/Not Executed)	Pass
---------------	--------	-------------	-------------------	------------------------------------	------

S #	Prerequisites:
1	Access to Browser
2	
3	
4	

S #	Test Data
1	Food Id= 1
2	Food name= Biryani
3	Quantity= 1
4	Price=250

<b>Test Scenario</b>	Modifying the quantity of the selected food item from the cart
----------------------	--

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to http://localhost:53528/cart.aspx	Site should open	As Expected	Pass
2	Select modify button	Text field appears to input the modified quantity.	As Expected	Pass
3	Select save	Quantity modified	As Expected	Pass

### EDIT CART

S.No	Label
Food ID	Label
Food Name	Label
td.auto-style11ce	Label
Quantity	1
Total Price	Label
Save	

Test Case ID	16	Test Case Description	Testing the checkout functionality from the cart		
Created By	Mishaal	Reviewed By	Sabika	Version	1

Tester's Name	Areeba	Date Tested	December 31, 2018	Test Case (Pass/Fail/Not Executed)	Pass
---------------	--------	-------------	-------------------	------------------------------------	------

S #	Prerequisites:
1	Access to Browser
2	
3	
4	

S #	Test Data
1	Food Id= 1
2	Food name= Biryani
3	Quantity= 1
4	Price=250

<b>Test Scenario</b>	Checking out from the cart
----------------------	----------------------------

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to http://localhost:53528/cart.aspx	Site should open	As Expected	Pass
2	Select Checkout button	Directed to placeorder page	As Expected	Pass

## CART

S.No	Food ID	Food Name	Price	Quantity	Total Price		
1	1	Biryani	250	1	250	<a href="#">Remove</a>	<a href="#">Modify</a>
2	1	Biryani	250	1	250	<a href="#">Remove</a>	<a href="#">Modify</a>
Total Amount					500		

No of Food Items in your cart: 2

[Checkout](#)

Test Case ID	17	Test Case Description	Testing the order placement functionality		
Created By	Mishaal	Reviewed By	Sabika	Version	1

Tester's Name	Areeba	Date Tested	December 31, 2018	Test Case (Pass/Fail/Not Executed)	Pass
---------------	--------	-------------	-------------------	------------------------------------	------

S #	Prerequisites:
1	Access to Browser
2	
3	
4	

S #	Test Data

<b>Test Scenario</b>	Placing the order
----------------------	-------------------

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to <a href="http://localhost:53528/placeorder.aspx">http://localhost:53528/placeorder.aspx</a>	Site should open	As Expected	Pass
2	Select Place order button	Directed to invoice page	As Expected	Pass

## USER ORDER DETAILS

Erp ID	13050
Order ID	46
Order Data	1/4/2019

S.No	Food ID	Food Name	Price	Quantity	Total Price
1	1	Biryani	250	1	250
2	1	Biryani	250	1	250
Total Amount					500

Place Order

Test Case ID	18	Test Case Description	Testing the payment status functionality		
--------------	----	-----------------------	--	--	--

Created By	Mishaal	Reviewed By	Sabika	Version	1
------------	---------	-------------	--------	---------	---

Tester's Name	Areeba	Date Tested	December 12, 2018	Test Case (Pass/Fail/Not Executed)	Pass
---------------	--------	-------------	-------------------	------------------------------------	------

S #	Prerequisites:
1	Access to Browser
2	
3	
4	

S #	Test Data
1	
2	
3	
4	
5	
6	

<b>Test Scenario</b>	Updating the payments status to true of the chef who has collected the payment.
----------------------	---

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed /Suspended
1	Navigate to http://localhost:53528/viewchefpayment.aspx	Site should open	As Expected	Pass
2	Tick the checkbox	Search successful	As Expected	Pass
3	Click update	Updated successfully in database	As Expected	Pass

## CHEF PAYMENTS

Chef_id	Username	Contact_no	End_date	Chef_payment	Payment_status
1	Areeba	03333677410	12/31/2018 12:00:00 AM	6000	<input type="checkbox"/>
1	Areeba	03333677410	12/31/2018 12:00:00 AM	6000	<input type="checkbox"/>
<input type="button" value="Update"/>					

\

Test Case ID	19	Test Case Description	Test the Sign Up Functionality of User		
Created By	Mishaal	Reviewed By	Sabika	Version	1

Tester's Name	Areeba	Date Tested	Jan 5, 2019	Test Case (Pass/Fail/Not Executed)	Pass
---------------	--------	-------------	-------------	------------------------------------	------

S #	Prerequisites:
1	Access to Browser
2	
3	
4	

S #	Test Data
1	
2	
3	
4	

<b>Test Scenario</b>	Adding the user details to the database.
----------------------	--

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to http://localhost:53528/studentsignup.aspx	Site should open	As Expected	Pass
2	Enter ERP	Credential added to the database	As Expected	Pass
3	Enter User Name	Credential added to the database	As Expected	Pass
4	Enter Password	Credentials added to the database	As Expected	Pass
5	Enter contact number	Credentials added to the database	As Expected	Pass

Erp\_id

Username

Password

Contact\_no

**success**

**SignUp**

Test Case ID	20	Test Case Description	Test the Sign Up Functionality of Chef		
Created By	Mishaal	Reviewed By	Sabika	Version	1

Tester's Name	Areeba	Date Tested	Jan 5, 2019	Test Case (Pass/Fail/Not Executed)	Pass
---------------	--------	-------------	-------------	------------------------------------	------

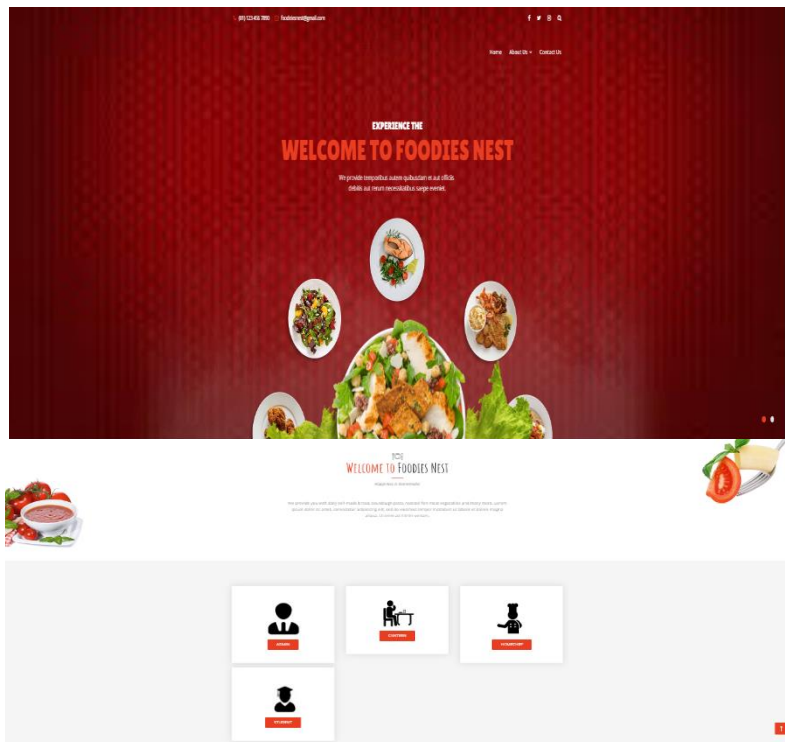
S #	Prerequisites:
1	Access to Browser
2	
3	
4	

S #	Test Data
1	
2	
3	
4	

**Test Scenario** Adding the chef details to the database.

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to <a href="http://localhost:53528/chefsignup.aspx">http://localhost:53528/chefsignup.aspx</a>	Site should open	As Expected	Pass
2	Enter User Name	Credential added to the database	As Expected	Pass
3	Enter Password	Credentials added to the database	As Expected	Pass
4	Enter contact number	Credentials added to the database	As Expected	Pass

## SCREENSHOTS OF SOFTWARE



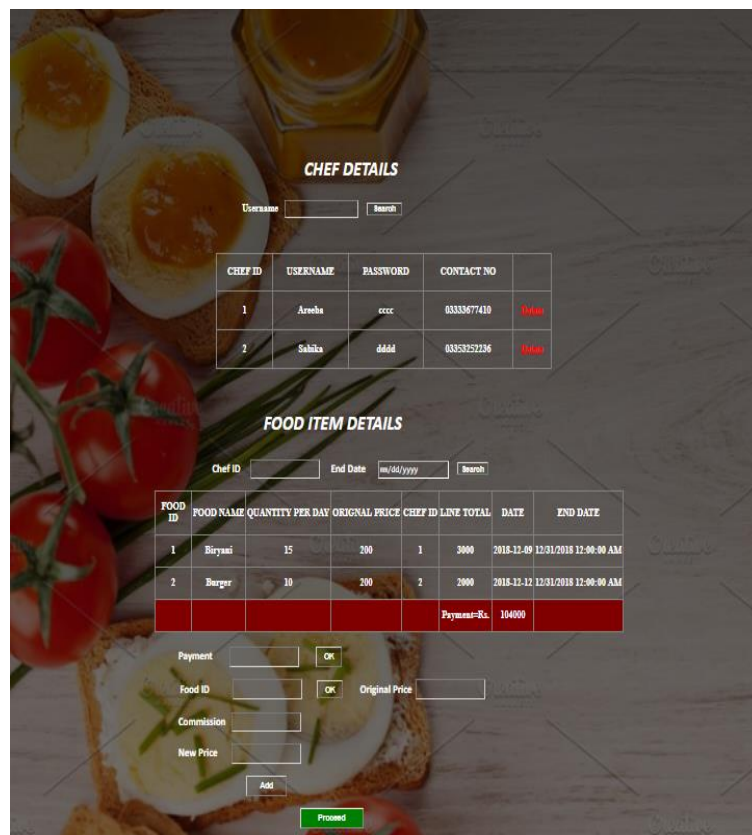
Admin ID

Password

Login



Welcome Admin



## CAFE DETAILS


Cafe Id	Cafe name	Contact No
111	Aman Cafe	03232219900

## COMMISSION

Food_id	Commission (%)	Price_per_dish(Rs)	
1	0.25	250	<a href="#">Edit</a>
3	0.35	270	<a href="#">Edit</a>

[Proceed](#)

## MENU LIST

Food_id	Food_name	Quantity_per_day	Price_per_dish (Rs)	Image
1	Biryani	15	250	
3	mutton biryani	10	270	

## CAFE LOGIN

Cafe ID	<input type="text"/>
Password	<input type="password"/>
<input type="button" value="Login"/>	



## Welcome Staff

[ORDER DETAILS](#)

### CHEF LOGIN

Chef_id	<input type="text"/>
Password	<input type="password"/>
<a href="#">Login</a> <a href="#">No account? Register here!</a>	

### CAFE ORDER DETAILS

	Order_id	Order_date	Exp_id
<a href="#">Select</a>	45	1/4/2019 12:00:00 AM	13050
<a href="#">Select</a>	44	12/13/2018 12:00:00 AM	13050
<a href="#">Select</a>	43	12/13/2018 12:00:00 AM	13050
<a href="#">Select</a>	42	12/13/2018 12:00:00 AM	13050
<a href="#">Select</a>	41	12/13/2018 12:00:00 AM	13050
<a href="#">Select</a>	40	12/13/2018 12:00:00 AM	13050
<a href="#">Select</a>	39	12/13/2018 12:00:00 AM	13050
<a href="#">Select</a>	38	12/13/2018 12:00:00 AM	13050

### Welcome Chef

<a href="#">ADD NEW DISH</a>	<a href="#">REVIEW EXISTING DISH</a>
------------------------------	--------------------------------------

Food_id	Food_name	Price_per_dish	Quantity_per_day	Total Price
1	Biryani	250	15	250
3	mutton biryani	270	10	270
				520

### DISH LISTING FORM

Food ID	<input type="text" value="4"/>
Food name	<input type="text"/>
Quantity	<input type="text"/>
Price	<input type="text"/>
Image	<input type="button" value="Choose File"/> No file chosen
Date	<input type="text" value="mm/dd/yyyy"/>
Chef ID	<input type="text"/>
<input type="button" value="Submit"/>	

Chef ID  End Date

Food_id	Food_name	Quantity_per_day	Original_price (Rs)	Date	line_total	End_date	
1	Biryani	15	200	2018-12-09	3000	12/31/2018 12:00:00 AM	<a href="#">Edit</a>

3	mutton biryani	10	200	2018-12-13	2000	12/31/2018 12:00:00 AM	<a href="#">Edit</a>
---	----------------	----	-----	------------	------	------------------------	----------------------

Payment= Rs. 102000

[Home](#) [About Us](#) [Contact](#)

### MENU

Food_id: 1	Food_id: 3
Food_name: Biryani	Food_name: mutton biryani
Price_per_dish: 250	Price_per_dish: 270
Quantity: 15 <input type="button" value="1"/>	Quantity: 10 <input type="button" value="1"/>
	
<input type="button" value="Add to Cart"/>	<input type="button" value="Add to Cart"/>

User ID	<input type="text"/>
Password	<input type="text"/>
<input type="button" value="Login"/>	<a href="#">No account? Register here!</a>

No of Food Items in your cart: 1 [Show Cart](#)

## CART

S.No	Food ID	Food Name	Price	Quantity	Total Price	
1	1	Biryani	250	1	250	<a href="#">Remove</a> <a href="#">Modify</a>
2	1	Biryani	250	1	250	<a href="#">Remove</a> <a href="#">Modify</a>
Total Amount					500	

No of Food Items in your cart: 2

[Checkout](#)

## USER ORDER DETAILS

Erp ID	13050
Order ID	46
Order Date	1/4/2019

S.No	Food ID	Food Name	Price	Quantity	Total Price
1	1	Biryani	250	1	250
2	1	Biryani	250	1	250
Total Amount					500

[Place Order](#)

## INVOICE

ERP ID	13050
ORDER ID	46
ORDER DATE	1/4/2019

S.No	Food ID	Food Name	Quantity	Price	Total Price
1	1	Biryani	1	250	250
2	1	Biryani	1	250	250

GRAND TOTAL 500

Declaration: We declare that this invoice shows actual price of the food items described

THIS IS A COMPUTER GENERATED INVOICE AND DOES NOT REQUIRE SIGNATURE

## GANTT CHART

ID	Task Name	Duration	Start	Finish	Predecessors	% Complete
1	<b>Foodies Nest-Software Development</b>	<b>88 days?</b>	<b>9/12</b>	<b>1/5</b>		<b>98%</b>
2						
3						
4	<b>Project approval</b>	<b>7 days?</b>	<b>9/12</b>	<b>9/20</b>		<b>100%</b>
5	Abstract	7 days?	9/12	9/20		100%
6	<b>Entity Relationship diagram</b>	<b>5 days?</b>	<b>9/12</b>	<b>9/18</b>	<b>4</b>	<b>100%</b>
7	Drafting ERD	5 days?	9/12	9/18		100%
8	Schema generation	2 days?	9/17	9/18	7	100%
9	<b>Prototype</b>	<b>14 days?</b>	<b>9/12</b>	<b>10/1</b>	<b>4</b>	<b>100%</b>
10	Wireframe	7 days?	9/12	9/20		100%
11	Team Members Review	1 day?	9/21	9/21	10	100%
12	Finalising Wireframes	6 days?	9/24	10/1	11	100%
13	<b>System requirement Specification</b>	<b>1 day?</b>	<b>9/13</b>	<b>9/13</b>	<b>5,7</b>	<b>100%</b>
14	Functional Requirements/Non-Functional Requirements	1 day?	9/13	9/13		100%
15	Use case	1 day?	9/13	9/13	14	100%
16	<b>Business Requirement Document</b>	<b>3 days?</b>	<b>9/12</b>	<b>9/14</b>	<b>5</b>	<b>100%</b>
17	Audience Survey	2 days?	9/12	9/13		100%
18	Business Process Overview Diagram	1 day?	9/14	9/14	17	100%
19	<b>Class Diagram</b>	1 day?	10/2	10/2	9,7	100%
20	<b>Sequence Diagram</b>	1 day?	10/2	10/2	9	100%
21	<b>Design</b>	<b>71 days?</b>	<b>9/12</b>	<b>12/14</b>	<b>9</b>	<b>95%</b>
22	Logo Design	1 day?	9/12	9/12		100%
23	Theme Design	1 day?	9/13	9/13	22	100%
24	Web designing	35 days?	10/30	12/14	23	95%
25	<b>Product development</b>	<b>26.45 day</b>	<b>11/14</b>	<b>12/17</b>	<b>21</b>	<b>97%</b>
26	Algorithm	2.9 days?	11/14	12/17	8	100%
27	Database	4 days?	11/15	11/19	26	100%
28	SQL developer-connections	3 days?	11/17	12/15	27	90%
29	<b>Testing</b>	<b>22 days?</b>	<b>12/10</b>	<b>1/5</b>	<b>25</b>	<b>100%</b>
30						
31	Testing	22 days?	12/10	1/5		100%
32	Testing 1	3 days?	12/10	12/12		100%
33	Testing 2	4 days?	1/1	1/5	32	100%
34	<b>Windows version software Release</b>	1 day	1/5	1/5		100%



## BIBLIOGRAPHY

- SRS template: <https://www.uccs.edu/Documents/tboults/srs.doc>
- Use Case created on: <https://creately.com/app/?diagID=johi00y4#>
- en.wikipedia.org
- Microsoft Developer Network (MSDN): <http://msdn2.microsoft.com/en-us/default.aspx>: This is a valuable online resource, and is a must for any developer using Microsoft tools
- BRD template reference: <https://www.pandadoc.com/business-requirements-documenthttps://www.pandadoc.com/business-requirements-document-template/template/>
- Test case : <https://www.guru99.com/test-case.html>
- Cost Estimation : <https://cs.uwaterloo.ca/~apidduck/CS846/Seminars/abbas.pdf>
- en.wikipedia.org
- Microsoft Developer Network (MSDN): <http://msdn2.microsoft.com/en-us/default.aspx>: This is a valuable online resource, and is a must for any developer using Microsoft tools
- BRD template reference: <https://www.pandadoc.com/business-requirements-documenthttps://www.pandadoc.com/business-requirements-document-template/template/>
- SRS template: <https://www.uccs.edu/Documents/tboults/srs.doc>
- Use Case created on: <https://creately.com/app/?diagID=johi00y4#>
- Homepage template: <https://colorlib.com/wp/free-restaurant-website-template/>
- Pictures used in our software are all taken from google:  
Admin icon:  
[https://www.google.com/search?tbm=isch&sa=1&ei=NpkzXPmREoKo1fAP07-dyAY&q=admin+icon&oq=admin+icon&gs\\_l=img.3..35i39j0l9.41375.42536..42751...0.0..0.287.1172.0j2j3.....1....1..gws-wiz-img.....0i67j0i7i30.1lBJ4oB7yL8#imgsrc=umo2EpOS9mvogM:](https://www.google.com/search?tbm=isch&sa=1&ei=NpkzXPmREoKo1fAP07-dyAY&q=admin+icon&oq=admin+icon&gs_l=img.3..35i39j0l9.41375.42536..42751...0.0..0.287.1172.0j2j3.....1....1..gws-wiz-img.....0i67j0i7i30.1lBJ4oB7yL8#imgsrc=umo2EpOS9mvogM:)  
Canteen icon:  
[https://www.google.com/search?tbm=isch&sa=1&ei=F5kzXPL9G-WH1fAPzZOJgAY&q=canteen+icon&oq=canteen+icon&gs\\_l=img.3..0i7i30l2.22065.28133..28339...5.0..1.319.3716.0j1j12j2.....1....1..gws-wiz-img.....0j35i39j0i30j0i7i10i30j0i8i7i30.lpS4V71Uq-w#imgdii=HN-AIQ0y2XdJDM:&imgsrc=KqRhI5leChnZwM:](https://www.google.com/search?tbm=isch&sa=1&ei=F5kzXPL9G-WH1fAPzZOJgAY&q=canteen+icon&oq=canteen+icon&gs_l=img.3..0i7i30l2.22065.28133..28339...5.0..1.319.3716.0j1j12j2.....1....1..gws-wiz-img.....0j35i39j0i30j0i7i10i30j0i8i7i30.lpS4V71Uq-w#imgdii=HN-AIQ0y2XdJDM:&imgsrc=KqRhI5leChnZwM:)  
Student icon:  
[https://www.google.com/search?tbm=isch&sa=1&ei=AJgzXlnhMvac1fAPhL25iA4&q=student+icon&oq=student+icon&gs\\_l=img.3..35i39j0l9.79745.80934..81311...0.0..0.314.1622.0j1j5j1.....1..1..gws-wiz-img.....0i67j0i7i30.c3lEf7oyr6g#imgsrc=-WxoT9FDkU3QLM](https://www.google.com/search?tbm=isch&sa=1&ei=AJgzXlnhMvac1fAPhL25iA4&q=student+icon&oq=student+icon&gs_l=img.3..35i39j0l9.79745.80934..81311...0.0..0.314.1622.0j1j5j1.....1..1..gws-wiz-img.....0i67j0i7i30.c3lEf7oyr6g#imgsrc=-WxoT9FDkU3QLM)  
Chef icon:  
[https://www.google.com/search?tbm=isch&sa=1&ei=oZgzXKLXCdmV1fAP856hsA0&q=chef+icon&oq=chef+icon&gs\\_l=img.3..0i7i30l10.2225.2810..2921...0.0..0.189.550.0j3.....1....1..gws-wiz-img.....0..01k3\\_61\\_Hc#imgsrc=vHSnRMmsuU9qOM](https://www.google.com/search?tbm=isch&sa=1&ei=oZgzXKLXCdmV1fAP856hsA0&q=chef+icon&oq=chef+icon&gs_l=img.3..0i7i30l10.2225.2810..2921...0.0..0.189.550.0j3.....1....1..gws-wiz-img.....0..01k3_61_Hc#imgsrc=vHSnRMmsuU9qOM)