صورة تحتوي على طعام

تم إنشاء الوصف تلقائياً

Mobile Application

By

Raghad Salman AlThabeti 43701694

Shrog Turki Almalki 43701801

Raghag Kalaf Althomali 43705998

Atheer Ahmad Altalhi 43702590

Meaad Saad Alzhrani 43709668

Gharam Saud Alharthi 43703404

**Supervised by:**

**Dr. Samah Alajmani**

Department of Computer Science

College of Computer and Information Technology

Taif University, KSA

2020/1442

**Team Members:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Student Name** | **Student ID** | **Date** | **Signature** |
| Raghad Salman AlThabeti | 43701694 |  |  |
| Shrog Turki Almalki | 43701801 |  |  |
| Raghag Kalaf Althomali | 43705998 |  |  |
| Atheer Ahmad Altalhi | 43702590 |  |  |
| Meaad Saad Alzhrani | 43709668 |  |  |
| Gharam Saud Alharthi | 43703404 |  |  |

**Supervised by:** Dr. Samah Alajmani

**Approved by Supervisor Committee:**

|  |  |  |
| --- | --- | --- |
| **Supervisor Name** | **Date** | **Signature** |
|  |  |  |
|  |  |  |
|  |  |  |

Department of Computer Science

Faculty of Computers and Information Technology

Taif University, KSA.

NK OF THINGS

**Document History:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version Number** | **Issued Date** | **Authors** | **Descriptions** | **Signature** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Team Members:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Student Name** | **Student ID** | **Date** | **Signature** |
| **Amal  Mouawad Algethami** | **43203263** |  |  |
| **Ashjan Hassan ALtoirqi** | **43203305** |  |  |
| **Souda Nasser Alsubaie** | **43203187** |  |  |
| **Ghzayel Ali AlHArthi** | **43008445** |  |  |
| **Fatima Tarboosh Almalki** | **43103549** |  |  |

**Supervised by:** <Dr. Wahiba Ben Abdessalem><Date><Signature>

**Approved by Supervisor Committee:**

|  |  |  |
| --- | --- | --- |
| **Supervisor Name** | **Date** | **Signature** |
| Wahiba Ben Abdessalem | 6/12/2015 |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Contact Information

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Authors Information | | | | |
| Name |  | Number |  | Email |
| Raghad Salman AlThabeti |  | 43701694 |  |  |
| Shrog Turki Almalki |  | 43701801 |  |  |
| Raghag Kalaf Althomali |  | 43705998 |  |  |
| Atheer Ahmad Altalhi |  | 43702590 |  |  |
| Meaad Saad Alzhrani |  | 43709668 |  |  |
| Gharam Saud Alharthi |  | 43703404 |  |  |

Supervisor

Dr. Dr. Samah Alajmani

# Students’ Property Right Declaration

I hereby declare that the work in this capstone project at Taif University is my own except for quotations and summaries which have been duly acknowledged. This work with the title **(----------------)** has not been accepted for any degree and is not concurrently submitted for award of other degrees. It is the sole property of Taif University and it is protected under the intellectual property right laws and conventions.

Authors:

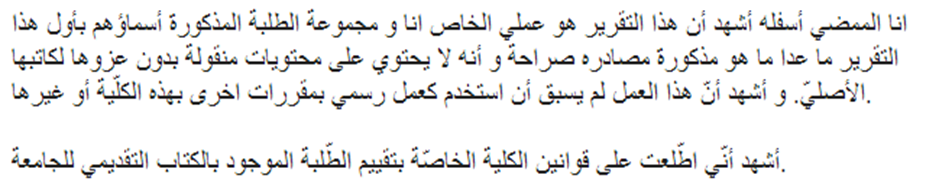
|  |  |  |
| --- | --- | --- |
| **Name** | **Signature** | **Date** |
| Raghad Salman AlThabeti |  |  |
| Shrog Turki Almalki |  |  |
| Raghag Kalaf Althomali |  |  |
| Atheer Ahmad Altalhi |  |  |
| Meaad Saad Alzhrani |  |  |
| Gharam Saud Alharthi |  |  |

|  |  |  |
| --- | --- | --- |
| **Supervisor:** | |  |
| Name:  Dr. Samah Alajmani | Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Date:  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

# Students Anti-Plagiarism Statement

I hereby declare this report is my own work except for properly referenced quotations and contains no plagiarism; it has not been submitted previously for any other assessed unit on this or other degree courses.

I have read and understood the School's rules on assessment offences which are available in the Taif University Handbook



|  |  |  |
| --- | --- | --- |
| **Name** | **Signature** | **Date** |
| Raghad Salman AlThabeti |  |  |
| Shrog Turki Almalki |  |  |
| Raghag Kalaf Althomali |  |  |
| Atheer Ahmad Altalhi |  |  |
| Meaad Saad Alzhrani |  |  |
| Gharam Saud Alharthi |  |  |

# Acknowledgement

First, we thank Allah, the almighty, for giving us the strength to carry on this project, and we would thank **Dr. Samah Alajmani** for allowing us to work under her supervision and helping us whenever we needed help, we would thank our families and our friends for all the support that we get.

# Abstract

Nowadays, dietary guidelines are designed to maintain an adequate amount of nutrition and protect against diet-related diseases, especially cardiovascular disease, and obesity.

In this project, technology will be employed to help the people who follow diet system to get the suitable meals from restaurants, the project will allow the customer to order meals based on the total calories she/he want, so the customer can control the calories she/he takes every day without any extra calories that cause health issues.

This project aims to develop a web application that provides easy access to food information for each meal. Using this application, the user is able to easily search for the right meals for his calories and has been added the trend feature to the meals to find out what the most common and most requested meal, can evaluate and choose the right meal for him easily.

Keywords: Website, Health, Obesity, Diet, Meal, Calories, Development.

# ملخص مقترح البحث

في الوقت الحاضر ، تم تصميم الإرشادات الغذائية للحفاظ على كمية كافية من التغذية والحماية من الأمراض المرتبطة بالنظام الغذائي ، وخاصة أمراض القلب والأوعية الدموية والسمنة.

في هذا المشروع ، سيتم استخدام التكنولوجيا لمساعدة الأشخاص الذين يتبعون نظام الحمية للحصول على الوجبات المناسبة من المطاعم ، وسيسمح المشروع للعميل بطلب وجبات بناءً على إجمالي السعرات الحرارية التي يريدها ، بحيث يمكن للعميل التحكم في السعرات الحرارية يأخذ كل يوم دون أي سعرات حرارية إضافية تسبب مشاكل صحية.

يهدف هذا المشروع إلى تطوير تطبيق ويب يوفر وصولاً سهلاً إلى المعلومات الغذائية لكل وجبة. باستخدام هذا التطبيق ، يمكن للمستخدم البحث بسهولة عن الوجبات المناسبة لسعراته الحرارية ، وقد تمت إضافة ميزة الاتجاه إلى الوجبات لمعرفة الوجبة الأكثر شيوعًا والأكثر طلبًا ، ويمكنه تقييم واختيار الوجبة المناسبة له بسهولة .

الكلمات المفتاحية: الموقع الإلكتروني ، الصحة ، السمنة ، النظام الغذائي ، الوجبة ، السعرات الحرارية ، التنمية.

# Table of Contents

[1 Contact Information iv](#_Toc56176674)

[2 Students’ Property Right Declaration v](#_Toc56176675)

[3 Students Anti-Plagiarism Statement vi](#_Toc56176676)

[4 Acknowledgement vii](#_Toc56176677)

[5 Abstract viii](#_Toc56176678)

[6 ملخص مقترح البحث ix](#_Toc56176679)

[7 Table of Contents x](#_Toc56176680)

[8 Table of Tables xii](#_Toc56176681)

[9 Table of Figures xiii](#_Toc56176682)

[1 CHAPTER 1: INTRODUCTION 1](#_Toc56176683)

[1.1 Project Overview 1](#_Toc56176684)

[1.2 Problem Statement and Motivation 1](#_Toc56176685)

[1.3 Project Objectives 2](#_Toc56176686)

[1.4 Project Scope 2](#_Toc56176687)

[1.5 Significance of the Project 2](#_Toc56176688)

[1.6 Project Plan 3](#_Toc56176689)

[1.7 Summary of The Remaining Chapters 3](#_Toc56176690)

[1.8 Conclusion 3](#_Toc56176691)

[2 CHAPTER 2: LITERATURE REVIEW 4](#_Toc56176692)

[2.1 Introduction 4](#_Toc56176693)

[2.2 Summary of Related Works 4](#_Toc56176694)

[2.3 Comparison of related projects 5](#_Toc56176695)

[2.4 Conclusion 6](#_Toc56176696)

[3 CHAPTER 3: SYSTEM ANALYSIS AND DESIGN 7](#_Toc56176697)

[3.1 Introduction 7](#_Toc56176698)

[3.2 Methodological 7](#_Toc56176699)

[3.3 Scrum team roles 9](#_Toc56176700)

[3.4 Sprint 0 10](#_Toc56176701)

[3.4.1 Requirements analysis and definition 10](#_Toc56176702)

[3.4.2 The product backlogs 14](#_Toc56176703)

[3.4.3 Sprint Planning 15](#_Toc56176704)

[3.5 Sprint (1) 16](#_Toc56176705)

[3.5.4 Sprint Backlog 16](#_Toc56176706)

[3.5.5 Description of the user story “Register” 17](#_Toc56176707)

[3.5.6 Description of the user story “Login” 19](#_Toc56176708)

[3.5.7 Description of the user story “Update profile” 22](#_Toc56176709)

[3.5.8 Description of the user story “Change password” 24](#_Toc56176710)

[3.6 Sprint (2) 27](#_Toc56176711)

[3.6.9 Sprint Backlog 27](#_Toc56176712)

[3.6.10 Description of the user story “Manage meals” 27](#_Toc56176713)

[3.6.11 Description of the user story “Manage orders” 32](#_Toc56176714)

[3.6.12 Description of the user story “Manage users” 34](#_Toc56176715)

[3.7 Sprint (3) 36](#_Toc56176716)

[3.7.13 Sprint Backlog 36](#_Toc56176717)

[3.7.14 Description of the user story “Search meals” 37](#_Toc56176718)

[3.7.15 Description of the user story “View trend meals” 38](#_Toc56176719)

[3.7.16 Description of the user story “Manage cart” 41](#_Toc56176720)

[3.7.17 Description of the user story “Evaluation” 43](#_Toc56176721)

[3.8 Data base design 45](#_Toc56176722)

[3.8.18 The system class diagram 45](#_Toc56176723)

[3.8.19 The Relational Database Schema 45](#_Toc56176724)

[3.9 Conclusion 46](#_Toc56176725)

[4 References 47](#_Toc56176726)

# Table of Tables

[Table ‎2‑1 Comparison of related projects 5](#_Toc56176727)

[Table ‎3‑1 Scrum team roles 9](#_Toc56176728)

[Table ‎3‑2 backlog of the system 15](#_Toc56176729)

[Table ‎3‑3 Sprint planning 15](#_Toc56176730)

[Table ‎3‑4 Sprint (1) backlog 16](#_Toc56176731)

[Table ‎3‑5 3 User story main scenarios: Register 17](#_Toc56176732)

[Table ‎3‑6 User story main scenarios: Login 19](#_Toc56176733)

[Table ‎3‑7 User story main scenarios: Update profile 22](#_Toc56176734)

[Table ‎3‑8 User story main scenarios: Change password 24](#_Toc56176735)

[Table ‎3‑9 Sprint (2) backlog 27](#_Toc56176736)

[Table ‎3‑10 User story main scenarios: Add meal 28](#_Toc56176737)

[Table ‎3‑11 User story main scenarios: Edit meal 28](#_Toc56176738)

[Table ‎3‑12 User story main scenarios: Delete meal 29](#_Toc56176739)

[Table ‎3‑12 User story main scenarios: Delete meal 29](#_Toc56176740)

[Table ‎3‑13 User story main scenarios: View orders 32](#_Toc56176741)

[Table ‎3‑14 User story main scenarios: Change status of order 32](#_Toc56176742)

[Table ‎3‑15 User story main scenarios: View users 34](#_Toc56176743)

[Table ‎3‑16 User story main scenarios: Delete user 34](#_Toc56176744)

[Table ‎3‑17 Sprint (3) backlog 36](#_Toc56176745)

[Table ‎3‑18 User story main scenarios: Search meals 37](#_Toc56176746)

[Table ‎3‑19 User story main scenarios: View trend meals 39](#_Toc56176747)

[Table ‎3‑20 User story main scenarios: Manage cart 41](#_Toc56176748)

[Table ‎3‑21 User story main scenarios: Evaluation 43](#_Toc56176749)

# Table of Figures

[Figure ‎1‑1 Project Plan 3](#_Toc56176750)

[Figure ‎3‑1 Agile methodology 8](#_Toc56176751)

[Figure ‎3‑2 Use case diagram 13](#_Toc56176752)

[Figure ‎3‑3 User interface prototype: Register 17](#_Toc56176753)

[Figure ‎3‑4 Class diagram: Register 18](#_Toc56176754)

[Figure ‎3‑5 Sequence diagram: Register 18](#_Toc56176755)

[Figure ‎3‑6 User interface prototype: Login 19](#_Toc56176756)

[Figure ‎3‑7 Class diagram: Login 20](#_Toc56176757)

[Figure ‎3‑8 Sequence diagram: Login 21](#_Toc56176758)

[Figure ‎3‑9 User interface prototype: Update profile 22](#_Toc56176759)

[Figure ‎3‑10 Class diagram: Update profile 23](#_Toc56176760)

[Figure ‎3‑11 Sequence diagram: Update profile 23](#_Toc56176761)

[Figure ‎3‑12 User interface prototype: Change password 24](#_Toc56176762)

[Figure ‎3‑13 Class diagram: Change password 26](#_Toc56176763)

[Figure ‎3‑14 Sequence diagram: Change password 26](#_Toc56176764)

[Figure ‎3‑15 User interface prototype: Manage meals 27](#_Toc56176765)

[Figure ‎3‑16 Class diagram: Manage meals 30](#_Toc56176766)

[Figure ‎3‑17 Sequence diagram: Add meal 30](#_Toc56176767)

[Figure ‎3‑18 Sequence diagram: Edit meal 31](#_Toc56176768)

[Figure ‎3‑19 Sequence diagram: Delete meal 31](#_Toc56176769)

[Figure ‎3‑20 User interface prototype: Manage orders 32](#_Toc56176770)

[Figure ‎3‑21 Class diagram: Manage orders 33](#_Toc56176771)

[Figure ‎3‑22 Sequence diagram: Manage orders 33](#_Toc56176772)

[Figure ‎3‑23 User interface prototype: Manage users 34](#_Toc56176773)

[Figure ‎3‑24 Class diagram: Manage users 35](#_Toc56176774)

[Figure ‎3‑25 Sequence diagram: Manage users 36](#_Toc56176775)

[Figure ‎3‑26 User interface prototype: Search meals 37](#_Toc56176776)

[Figure ‎3‑27 Class diagram: Search meals 38](#_Toc56176777)

[Figure ‎3‑28 Sequence diagram: Search meals 38](#_Toc56176778)

[Figure ‎3‑29 User interface prototype: View trend meals 39](#_Toc56176779)

[Figure ‎3‑30 Class diagram: View trend meals 40](#_Toc56176780)

[Figure ‎3‑31 Sequence diagram: View trend meals 40](#_Toc56176781)

[Figure ‎3‑32 User interface prototype: Manage cart 41](#_Toc56176782)

[Figure ‎3‑33 Class diagram: Manage cart 42](#_Toc56176783)

[Figure ‎3‑34 Sequence diagram: Manage cart 42](#_Toc56176784)

[Figure ‎3‑35 User interface prototype: Evaluation 43](#_Toc56176785)

[Figure ‎3‑36 Class diagram: Evaluation 44](#_Toc56176786)

[Figure ‎3‑37 Sequence diagram: Manage cart 44](#_Toc56176787)

[Figure ‎3‑38 System class diagram 45](#_Toc56176788)

# CHAPTER 1: INTRODUCTION

## Project Overview

Technology has entered all kinds of life in all sectors of life, and under the ever-increasing use and rapid development and provision of needs and facilities to humanity and the growing presence of the spirit of the community by identifying common topics in social media and with the entry of applications into the restaurant sector and delivery, which is one of the most advanced sectors in application stores, technology has become a major part of the business processes.

Statistics from Insider Intelligence state that m-commerce will reach $284 billion in USA by the end of 2020, that means 45% of the total e-commerce will be throw mobiles [1].

In recent years, healthy thinking about eating has increased, many applications have been launched to calculate calories, and some use food image recognition to estimate calories relatively.

Currently there are no applications that allows customers to order meals depending on calories, we believe that showing the calories in meals helps to choose enough of an urgent need, without any extra calories, so we aim in this project a develop a different application, that facilitates the meals ordering for users who follow a healthy diet, where the customer will be able to order meals by searching for the required calories, and displaying meals based on his calories target, in addition, the application will show (Trend meals) to help customers in finding out what is the most popular meal.

## Problem Statement and Motivation

Food and nutrition play a hugely important role in causing and preventing many diseases, overweight and obesity, lack of physical activity, low fruit and vegetable intake, high cholesterol and blood glucose is a combination of interlinked problems, which competes on the highest-ranking position among public health threats.

Hence, choosing the right meal become a vital issue to follow a healthy eating style. Although it is not an easy due to the lake of the nutritional information of the meal.

## Project Objectives

This project aims to achieve the following objectives:

1. Spread awareness of the importance of healthy nutrition.
2. Helping people maintain a proper diet and choosing the suitable meal.
3. To implement mobile application that provide the nutritional information of the meal.
4. This project aims to add a trend list for the best meal in it.

## Project Scope

This system will have implemented as an Android mobile application and will cover Saudi Arabia cities.

This system will be done by the end of this academic year 1442/2020 and will be implemented in English language.

## **Significance of the Project**

The expected benefits of our project are:

1. Facilitate the getting of healthy meals.
2. Increasing the health level of users by reducing obesity.
3. Promote healthy food restaurants by offering their meals to users.
4. Increasing the quality of restaurant services by allowing the user to rate them.

## Project Plan

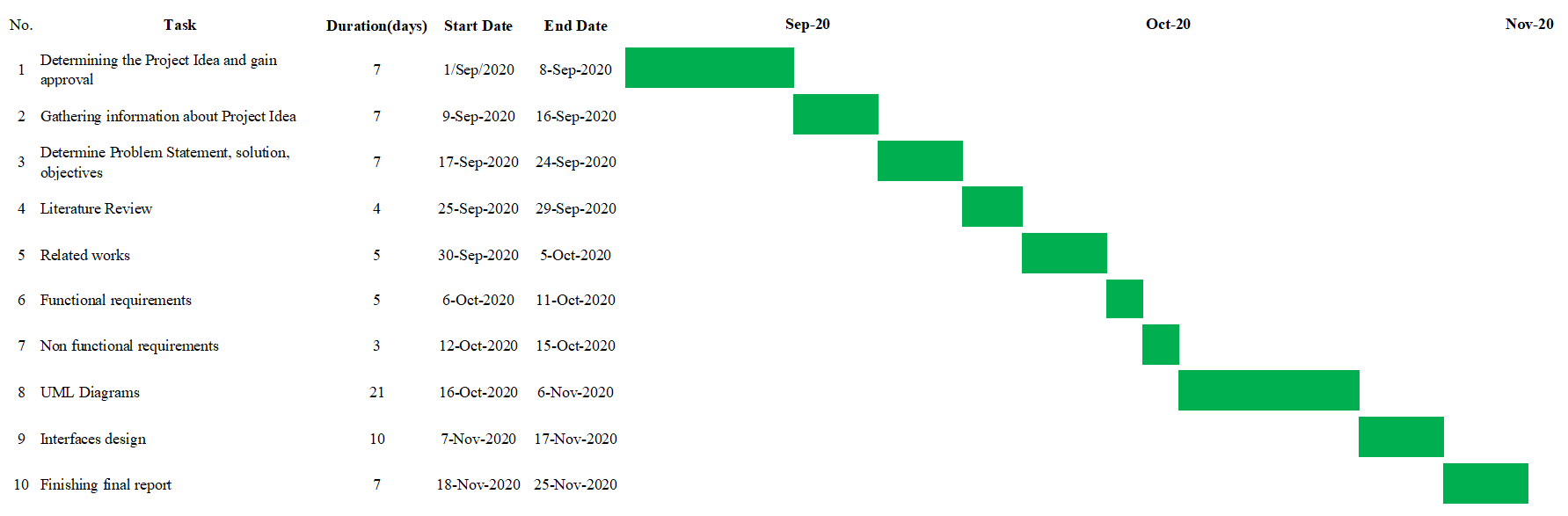


Figure ‑ Project Plan

## Summary of The Remaining Chapters

This report is composed of four chapters

The next two chapters will deal with a literature review and compares existing similar application to the proposed project. Also, it will define the methodological choice and development tools.

Chapter three will focus on analysis and design of all gathered requirements using the Agile methodology.

Finally, a conclusion and future work will be presented at the end of this report.

## Conclusion

In this chapter, we provided a general description about project idea and we defined the problem statement, project scope and objectives of the proposed application, next chapter we will talk about literature review include related works and methodological choices.

# CHAPTER 2: LITERATURE REVIEW

## Introduction

In this chapter, we will discuss some similar projects and compare them to our proposed project, and we compare these applications with the proposed system.

## Summary of Related Works

A survey of several related works has been done. These works include Talabat App which is an online food website which presents restaurants search and food delivery services. It gives the customer a choice for a great food to eat, Talabat has set foot on 7 countries including Kuwait, the business also provides more than 4,300 different cuisines [2]. UberEats is another application that be studied, it is an online food ordering app .Initially, Uber Eats started with delivering meals only during the lunch time but with time they decided to change it as an app that delivers fresh food both during lunch and dinner times, UberEats does not support Saudi Arabia or Arabic language [3]. MyFitnessPal is one of the most popular calorie counters right now. It tracks your weight and calculates a recommended daily calorie intake. It also contains a well-designed food diary and an exercise log [4], it does not support ordering food from restaurants, and some of its features can only be accessed in the premium version.

Hungerstation is one of the most used application is Saudi Arabia for ordering foods from restaurants, Hungerstation support cash payment and credit card payment, it supports widely number of restaurants and it fast in delivery. Jahez is application that support only Saudi Arabia that provide ordering food from restaurants, Jahez Allow users to schedule the time of delivery and provides home made meals form productive families. Wssel is another application that support ordering foods from restaurants and ordering items from markets, its simple to use application, but it not fast enough compared to other applications.

## Comparison of related projects

Table ‑ Comparison of related projects

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Talabat** | **Uber Eats** | **MyFitnessPal** | **Hungerstation** | **Jahez** | **Wssel** | **Proposed system** |
| Simple to use | Yes | Yes | No | Yes | Yes | Yes | Yes |
| Fast | Yes | Yes | Yes | Yes | No | No | No |
| Platform | Android, IOS | Android, IOS | Web | Android, IOS | Android, IOS | Android, IOS | Android |
| Support Arabic language | Yes | No | No | Yes | Yes | Yes | Yes |
| Support Saudi Arabia | Yes | No | No | Yes | Yes | Yes | Yes |
| Support searching depending on calories | No | No | No | No | No | No | Yes |
| Free | No | No | No | No | No | No | Yes |
| Support ordering meals | Yes | Yes | No | Yes | Yes | Yes | Yes |
| Trending meal | Yes | Yes | No | Yes | No | No | Yes |
| Taste evaluation | No | No | No | No | No | No | Yes |
| Restaurant Quality | Yes | Yes | No | Yes | Yes | Yes | Yes |
| Current place | Yes | No | No | Yes | Yes | Yes | Yes |

## Conclusion

In this we discussed several applications than related to the proposed system, we found out the advantages and disadvantages of each, we will cover the disadvantages of the similar application in the proposed system.

# CHAPTER 3: SYSTEM ANALYSIS AND DESIGN

## Introduction

In this chapter the analysis and design of the proposed system will be done, functional and nonfunctional requirements will be defined and the UML diagrams of the system will be drawn

## Methodological

Software Development Life Cycle (SCLD) is a process used by the software industry to design, develop and test high quality software. The SDLC aims to produce a high-quality software that meets or exceeds customer expectations, reaches completion within times and cost estimates [5].

The SDLC has several phases start from planning, then requirement gathering, analysis, development and testing.

These phases can accomplish in many methodologies, like waterfall, agile and scrum, in this project we will use the agile methodologies because will be discussed next sections.

Agile model is a combination of iterative and incremental process models; it focusses on process adaptability and customer needs by rapid delivery of working software product. Agile method breaks the product into small builds, these builds are provided in iterations, iterative approach is taken and working software build is delivered after each iteration. Each build is incremental in terms of features; the final build holds all the features required by the customer. [6].

Agile methodology selected for this project because it focusses on customer, it involves the customer regularly. Agile can deal with changes in requirements more flexibly, this methodology suitable where is available time is short, by Agile methodology, more than one version of the application is released each version containing updates and improvements of the previous version and it includes code testing, as in agile methodology this project will be developed in incremental and iterative manner.

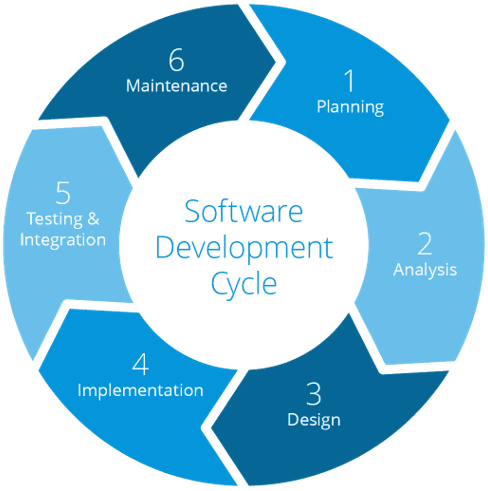


Figure ‑ Agile methodology

Agile methodology consists of six phases:

1. Planning: planning is the first phase for any project. It establishes a high-level view of the intended project, determines its scope and goals and find business requirements of the project, in this phase the proposal of the project was created, the objectives were defined, also the scope of the project is determined.
2. Analysis: analysis it is a structured process for studying the system and its parts, in this phase the business requirements of the proposed system will be translated from high level to functional requirements, nonfunctional requirements will be found, also, questionnaire will be designed, distributed, and its results will be analyzed. The use case of the proposed system will be created in this phase.
3. Design: design phase determines how to meet the requirements identified in the analysis phase, the system design helps in specifying hardware and software requirements to complete the proposed project, and it helps in defining the overall system architecture. In the proposed project, this phase includes designing: activity diagrams, class diagram, and data modeling.
4. Implementation: in this phase the real application will be created first, the database must be implemented, and the application interfaces will be created the proposed application database will be implemented using MySQL server management system, and Android studio is the IDE that will be used to create android application.
5. Testing: the propose of the testing is to ensure that the developed system meet and required business requirements and achieve the project objective, in this phase all buttons and menus of the system will be checking to ensure that it work fine, and the all functions of the application must be tried to check if its works as expected.
6. Maintenance: Maintenance is necessary to eliminate errors in the system during its working, in this phase the errors and bugs that appear in testing phase must be fixed, so the application become ready to be used.

## Scrum team roles

The following table shows the member SCRUM and their roles.

|  |  |
| --- | --- |
| **Role** | **Name team member** |
| **Product owner** | Users: Students and teachers |
| **SCRUM Master** | Raghad Salman AlThabeti |
| **SCRUM Team** | Shrog Turki Almalki  Raghag Kalaf Althomali  Atheer Ahmad Altalhi  Meaad Saad Alzhrani  Gharam Saud Alharthi |

Table ‑ Scrum team roles

## Sprint 0

### Requirements analysis and definition

The requirements are that the customer requires from a system and the constrains under which it operates and is developed, there are two kind of requirements, functional and nonfunctional requirements.

#### Functional requirements

Functional requirements describe what the system should do, each actor of the proposed system has its functional requirements like the following.

Restaurant functional requirement:

1. Register: the owner of the restaurant on this page records the restaurant information to create account for the restaurant this information includes: name, email, phone number, photo, location and password.
2. Login: the owner of the restaurant on this page logged in to the site through the email and password that he registered on the registration page to enter his home page on the site provided that the email and the password that entered them are identical with the email and password entered on the registration page.
3. Manage meals: the owner of the restaurant on this page modifies the meals list, which includes: meal name, calories, price and photo.
4. Manage orders: the owner of the restaurant on this page can view all orders and can view the details, such as the meals in the order and the customer contact information, and change the status of the order in case the order out for delivery or canceled.
5. Update profile: the owner of the restaurant on this page can change the account information, including name, email, phone number, photo, location.
6. Security: the owner of the restaurant can change the password of account.
7. Logout: the owner of the restaurant can logout from his account.

Customer functional requirement

1. Register: in this process, the customer will enter all information on the registration page to create an account, this information including name, email, phone number, and password.
2. Login: in this process, the customer will enter the email and password that he entered on the registration page to enter the home page, provided that the email and password are correct as entered on the registration page and the browser save the user in case of entering the site again.
3. Search meals: on this page, the customer browses the menus presented by the restaurant, from food and drinks, the customer will able to search meals according to the number of calories that the meal contain.
4. Manage cart: The customer add meal to cart, delete meal from cart, place order, place order will include payment, payment is an operation done on the banking system not the proposed system.
5. View trend meal, in this page the customer can view the meals that the other customers order is frequently, including the calories and number of customers thar order the meal.
6. Evaluate: on this page, the customer to evaluate the taste of meal and restaurant, the evaluation will include the quality of restaurant services, taste of the meal, the speed of delivery and packaging quality.
7. Update profile: the customer on this page can change the account information, information including name, email, phone number.
8. Security: the customer can change the password of account.
9. Logout: the customer of the restaurant can logout from his account.

Administrator functional requirements

1. Login: administrator on this page login to the system by writing the email and password.
2. Manage users: on this page, the administrator will be logged on to the users list page. The administrator can review all users either they are customers or restaurants information and delete any of them, the administrator will be able to review the menu of restaurants before accept them.
3. Manage meals: the owner of the restaurant on this page modifies the meals list, which includes: meal name, calories, price and photo.
4. Update profile: the admin on this page can change the account information, information including name, email, phone number.
5. Security: the admin can change the password of account.
6. Logout: the admin can logout from his account.

Deliveryman functional requirements

1. Register: the delivery man on this page records the restaurant information he wants to add on the site to display the items and lists he owns through the site such as: email, password, restaurant name, restaurant location, restaurant contact information, etc.
2. Login: the delivery man on this page logged in to the site through the email and password that he registered on the registration page to enter his home page on the site provided that the email and the password that entered them are identical with the email and password entered on the registration page.
3. Manage orders: the delivery man on this page can view all orders and can view the details of orders and change the status of the order to accepted and delivered.
4. Update profile: the deliveryman on this page can change the account information, information including name, email, phone number.
5. Security: the deliveryman can change the password of account.
6. Logout: the delivery man can logout from his account.

#### Use Case Diagrams

Is a representation of a user's interaction with the system that shows the relationship between the user and the different [use cases](https://en.wikipedia.org/wiki/Use_case) in which the user is involved [7].

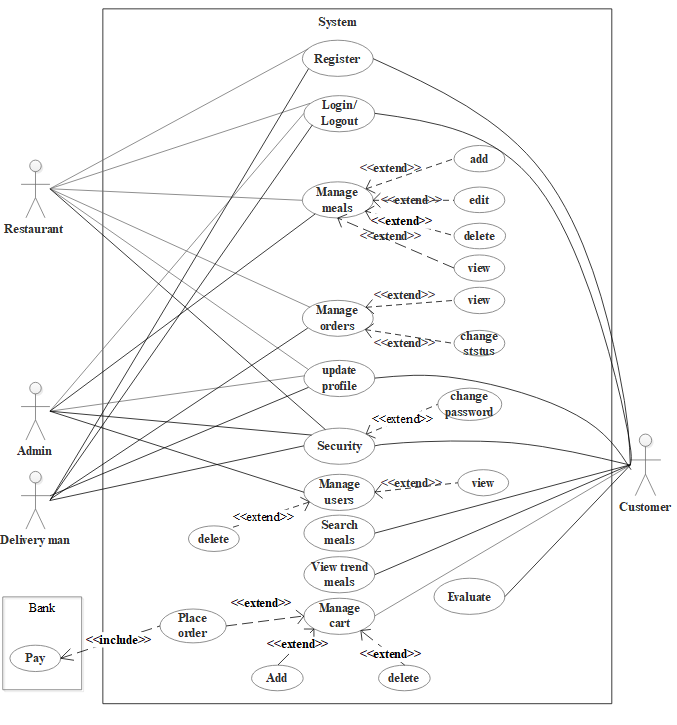


Figure ‑ Use case diagram

#### Non-functional requirements

A non-functional requirement is a statement of how a system must behave; it is a constraint upon the systems behavior. [8].

1. Usability: usability is the degree to which a software can be used by specified consumers to achieve quantified objectives with effectiveness, efficiency, and satisfaction in a quantified context of use, the proposed system will be easy to use and simple interface, so any user will be use it without needing to any training.
2. Flexibility: the process of development of the system to modify or add-on functions after it is published is easy and did not need much effort, the proposed system will be implemented using Java programming language, which is wide used programming language, so the proposed application will be easy to expanded and improved.
3. Performance: It is the amount of work accomplished by a computer system. Depending on the context and high computer performance, the proposed them will work speedy and the will response to the users request very quickly.
4. Reliability: the software will meet all the functional requirements without any unexpected behavior, the proposed system will be tested and all bugs will be fixed, so the user can use the system without any non-expected outcome.
5. Security: as the proposed system support online payment, it will be secure so all transactions will be done in safe way.
6. Maintainability: the software should be written clearly and concisely. The code will be well documented.

### The product backlogs

The Product Backlog is an ordered list of everything that might be needed in the product and is the single source of requirements for any changes to be made to the product [9].

The effort determines the work needed to complete the task and the priority determines the importance of this task for the user. The effort is computed as follows:

Table ‑ backlog of the system

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **User Story** | **Effort** | **Priority** | **Description** | **Risk** |
| 1 | Register | 3 | 95% | Create account for returant, delivery man and customer | Mid |
| 2 | Login | 3 | 95% | Login to the system using username and password | Mid |
| 3 | Update profile | 3 | 95% | Update account information and password of account | Low |
| 4 | Manage meal | 4 | 95% | Add, edit, and delete meals | Mid |
| 5 | Manage orders | 4 | 95% | View and update status of orders | Mid |
| 6 | Manage users | 4 | 90% | View and delete users | Mid |
| 7 | Search meals | 3 | 85% | Search for meals depends of calories | Mid |
| 8 | View trend meal | 3 | 85% | View meals that ordered from other customers. | Mid |
| 9 | Manage cart | 3 | 85% | Add meals to cart, delete meals from cart and place order. | Low |
| 10 | Evaluate | 4 | 90% | Evaluate meal taste and restaurant. | Mid |

### Sprint Planning

Table ‑ Sprint planning

|  |  |  |
| --- | --- | --- |
| **Sprint** | **User Story** | **Time** |
| Sprint 1 | Register  Login  Update profile  Change password |  |
| Sprint 2 | Manage meal  Manage orders  Manage users |  |
| Sprint 3 | Search meals  View trend meal  Manage cart  Evaluate |  |

## Sprint (1)

### Sprint Backlog

Table ‑ Sprint (1) backlog

|  |  |  |
| --- | --- | --- |
| **ID** | **User Story** | **Tasks** |
| 1 | Register | Create new account |
| 2 | Login | Login to the system |
| 3 | Update profile | Update user account data |
| 4 | Change password | Change password |

### Description of the user story “Register”

#### Prototype of the user interface “Register”

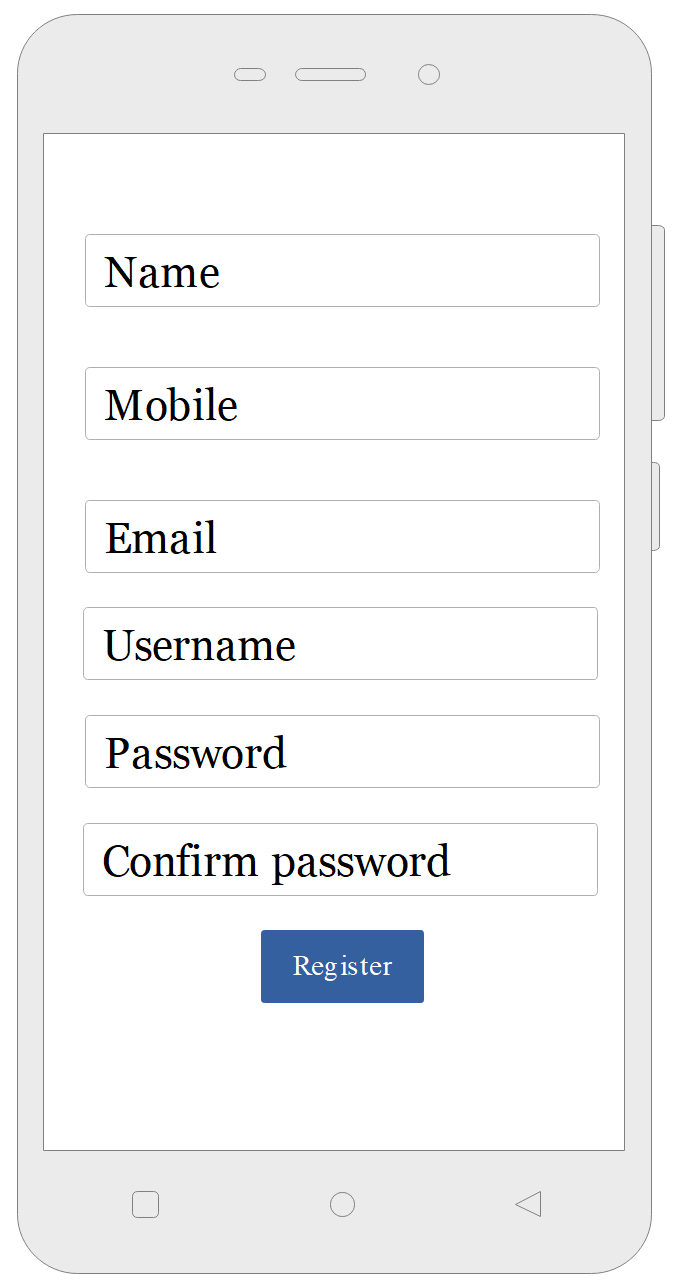


Figure ‑ User interface prototype: Register

#### User story main scenarios “Register”

Table ‑ 3 User story main scenarios: Register

|  |  |
| --- | --- |
| **User story** | Register |
| **Abstract** | Name, mobile, email, username, and password needed to create new account. |
| **Actor** | Deliveryman, restaurant, customer. |
| **Precondition** | Installing the application |
| **Post condition** | The account is created |
| **Main scenario** | **[Begin]**  1. The actor enter information  2. The actor clicks on the "register" button  3. The system saves all the data in the database.  **[End]** |

#### Design of the user story “Register”

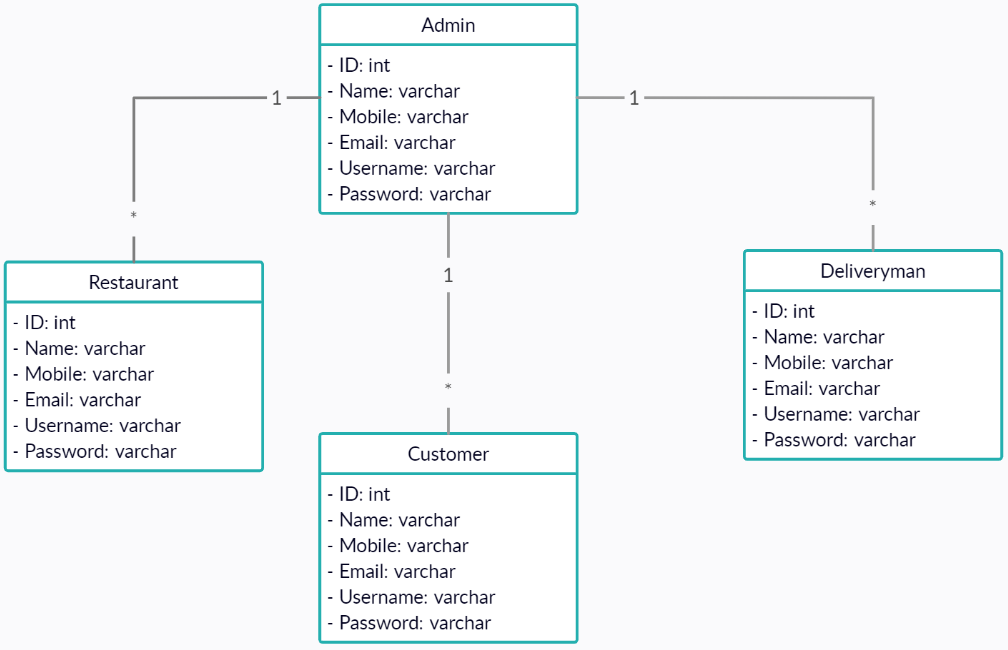


Figure ‑ Class diagram: Register

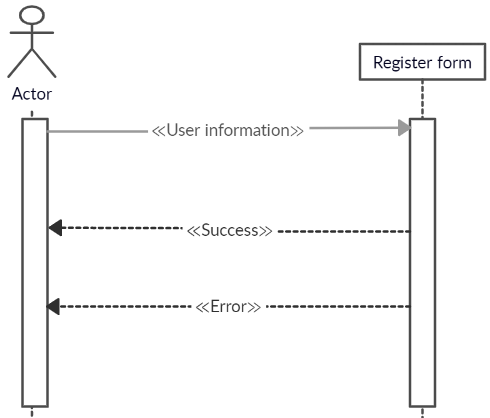


Figure ‑ Sequence diagram: Register

### Description of the user story “Login”

#### Prototype of the user interface “Login”

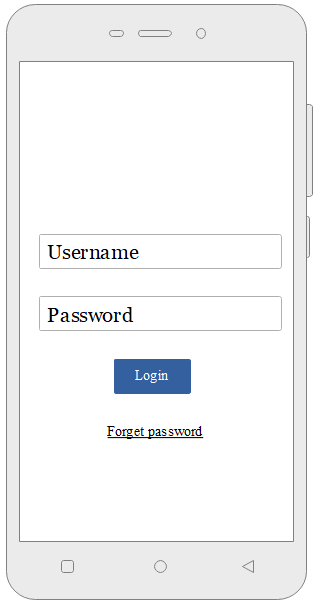


Figure ‑ User interface prototype: Login

#### User story main scenarios “Login”

Table ‑ User story main scenarios: Login

|  |  |
| --- | --- |
| **User story** | Login |
| **Abstract** | Username and password needed to lignin in the account. |
| **Actor** | Administrator, deliveryman, restaurant, customer. |
| **Precondition** | Register in the application |
| **Post condition** | Actor enters account |
| **Main scenario** | **[Begin]**  1. The actor enter username and password  2. The actor clicks on the "login" button  3. The actor enters account.  **[End]** |

#### Design of the user story “Login”

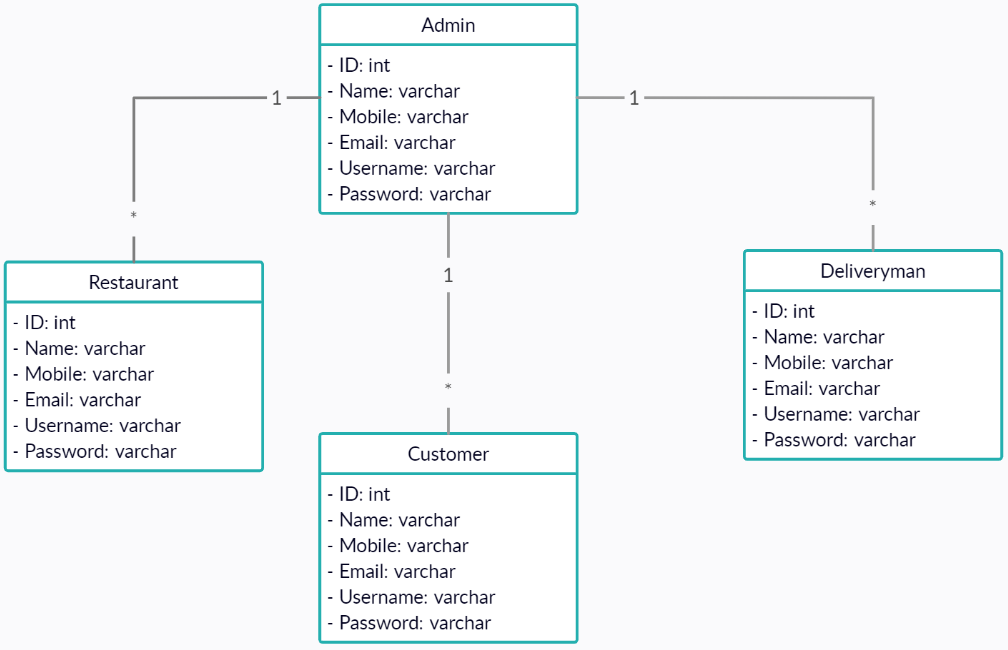


Figure ‑ Class diagram: Login

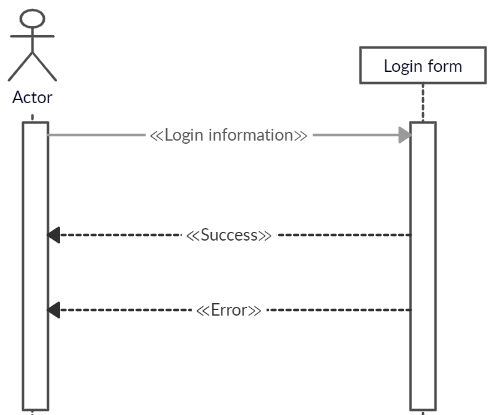


Figure ‑ Sequence diagram: Login

### Description of the user story “Update profile”

#### Prototype of the user interface “Update profile”

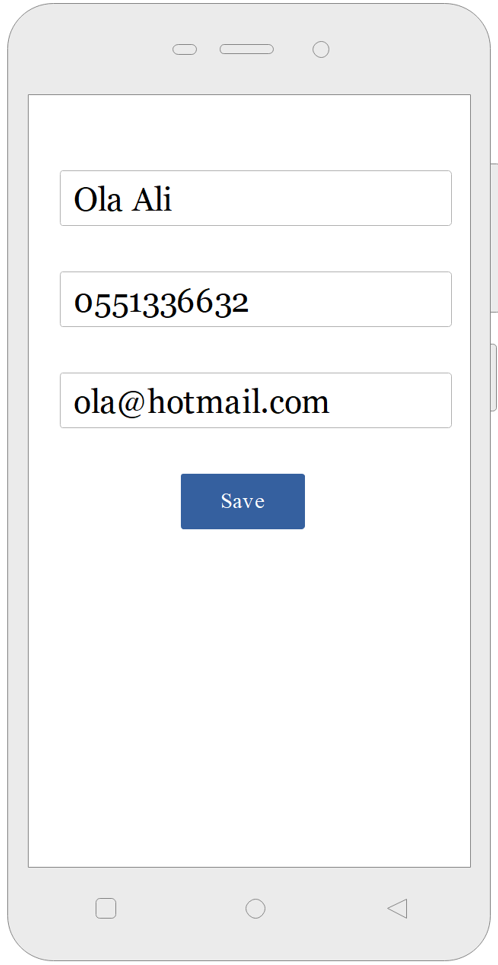


Figure ‑ User interface prototype: Update profile

#### User story main scenarios “Update profile”

Table ‑ User story main scenarios: Update profile

|  |  |
| --- | --- |
| **User story** | Update profile |
| **Abstract** | Name, mobile, email needed to update account. |
| **Actor** | Admin, deliveryman, restaurant, customer. |
| **Precondition** | Login in the application |
| **Post condition** | The account is updated |
| **Main scenario** | **[Begin]**  1. The actor enter new information  2. The actor clicks on the "save" button  3. The system saves all the data in the database.  **[End]** |

#### Design of the user story “Update profile”

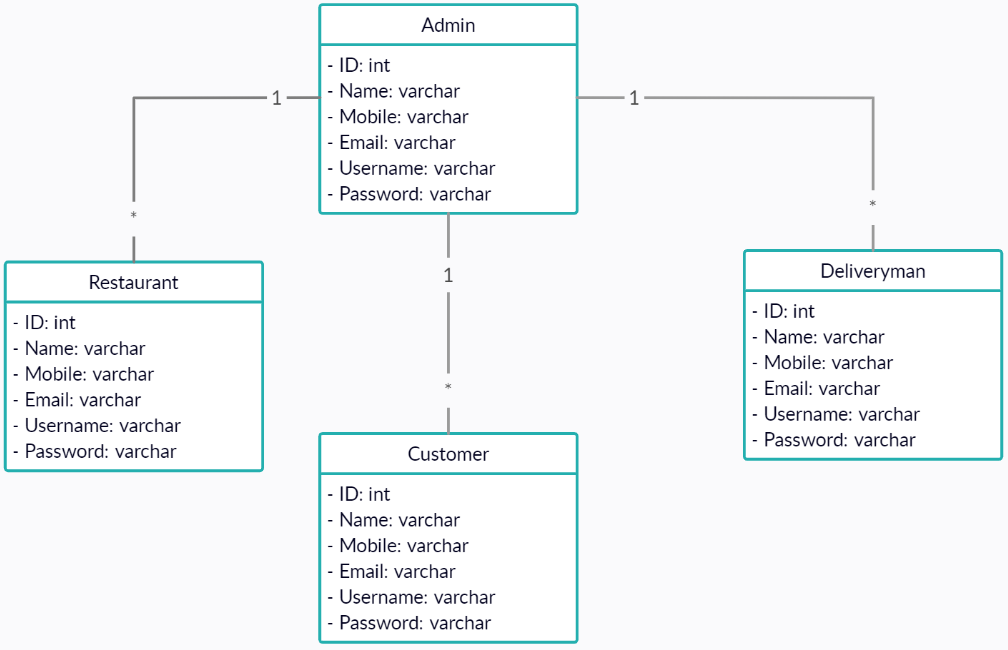


Figure ‑ Class diagram: Update profile

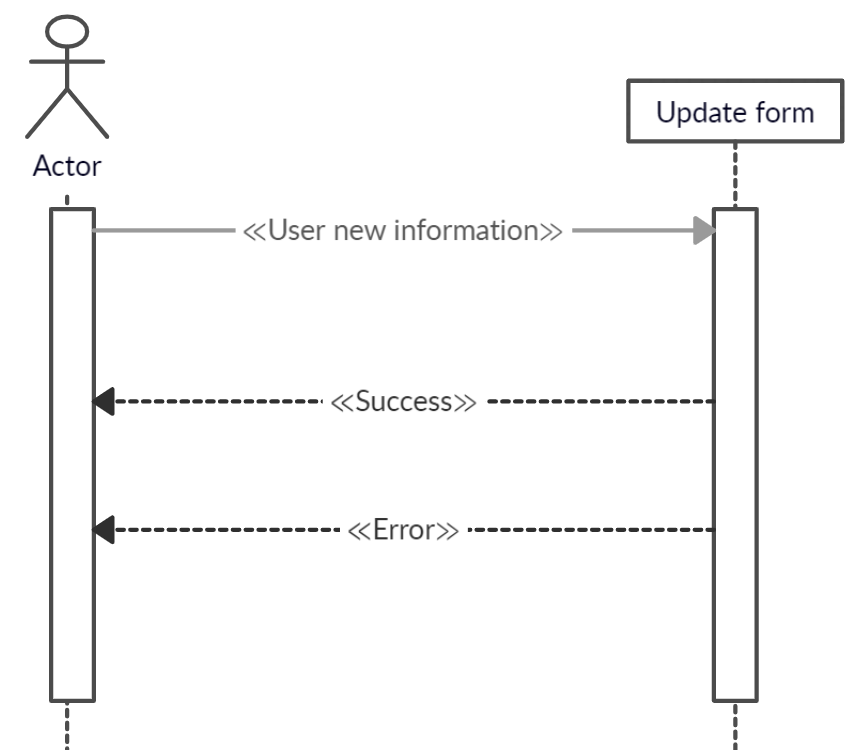


Figure ‑ Sequence diagram: Update profile

### Description of the user story “Change password”

#### Prototype of the user interface “Change password”

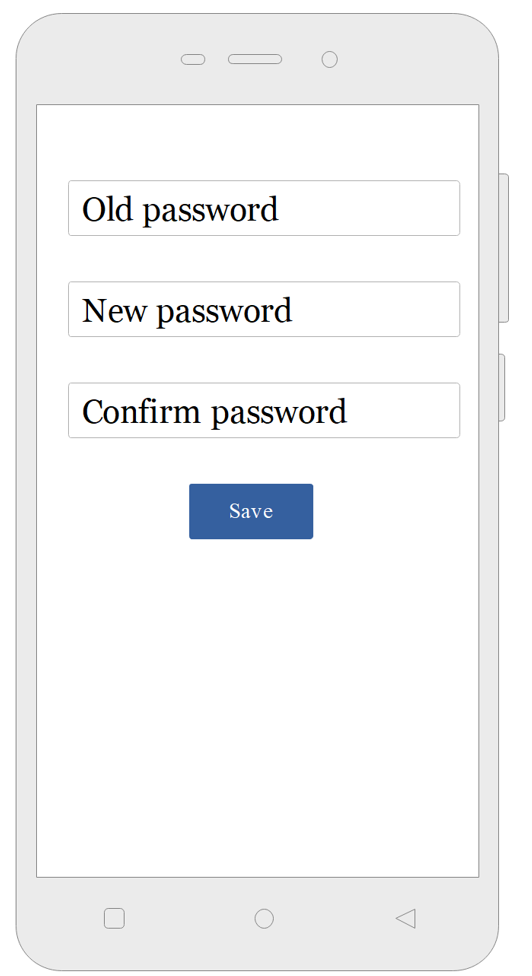


Figure ‑ User interface prototype: Change password

#### User story main scenarios “Change password”

Table ‑ User story main scenarios: Change password

|  |  |
| --- | --- |
| **User story** | Change password |
| **Abstract** | Old password, new password and password confirmation are needed to change the account password |
| **Actor** | Admin, deliveryman, restaurant, customer. |
| **Precondition** | Login in the application |
| **Post condition** | The account is updated |
| **Main scenario** | **[Begin]**  1. The actor enter old, new and confirm password  2. The actor clicks on the "save" button  3. The system updates account password.  **[End]** |

#### Design of the user story “Change password”

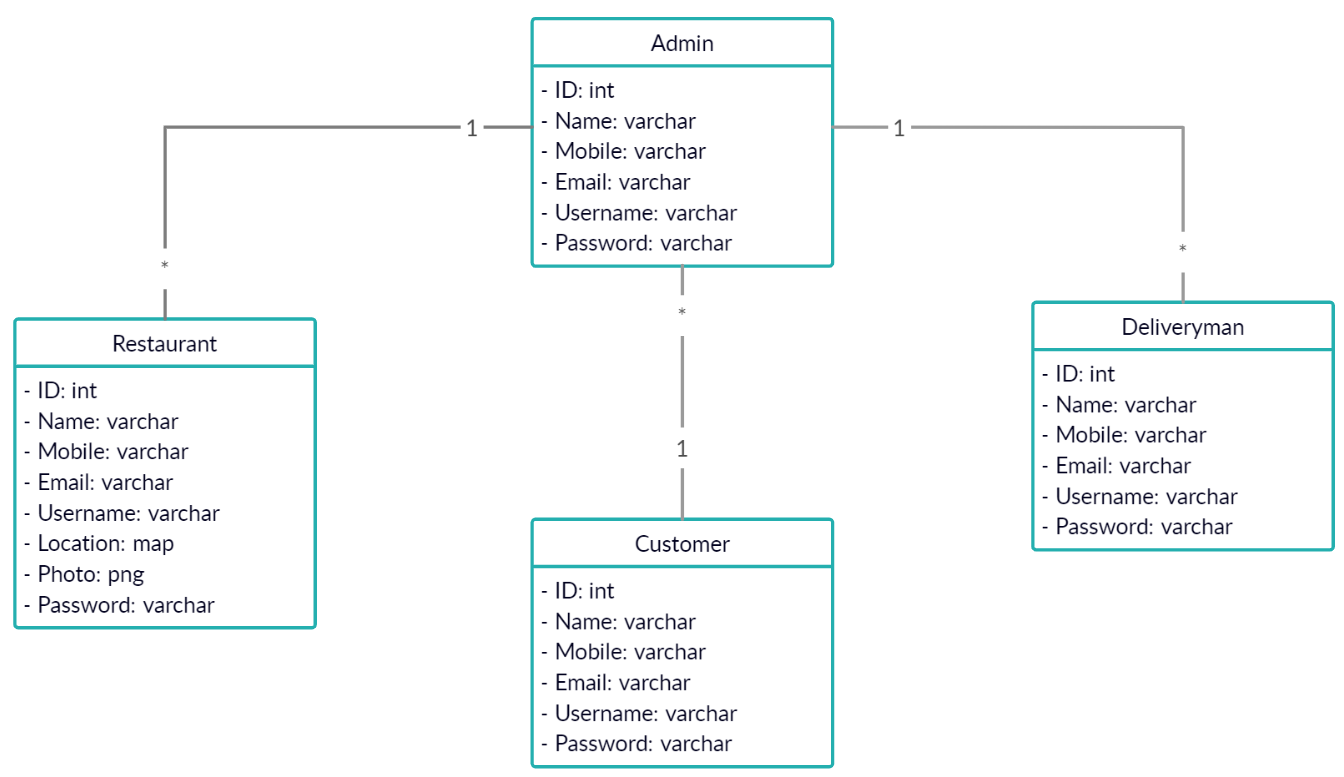


Figure ‑ Class diagram: Change password

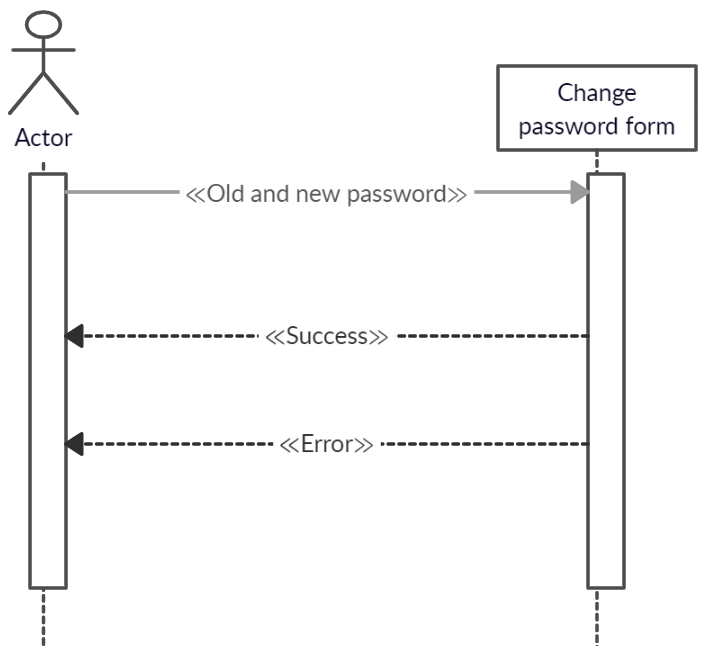


Figure ‑ Sequence diagram: Change password

## Sprint (2)

### Sprint Backlog

Table ‑ Sprint (2) backlog

|  |  |  |
| --- | --- | --- |
| **ID** | **User Story** | **Tasks** |
| 1 | Manage meals | Add meal  Edit meal  Delete meal  View meal |
| 2 | Manage orders | View orders  Change status |
| 3 | Manage users | View users  Delete users |

### Description of the user story “Manage meals”

#### Prototype of the user interface “Manage meals”

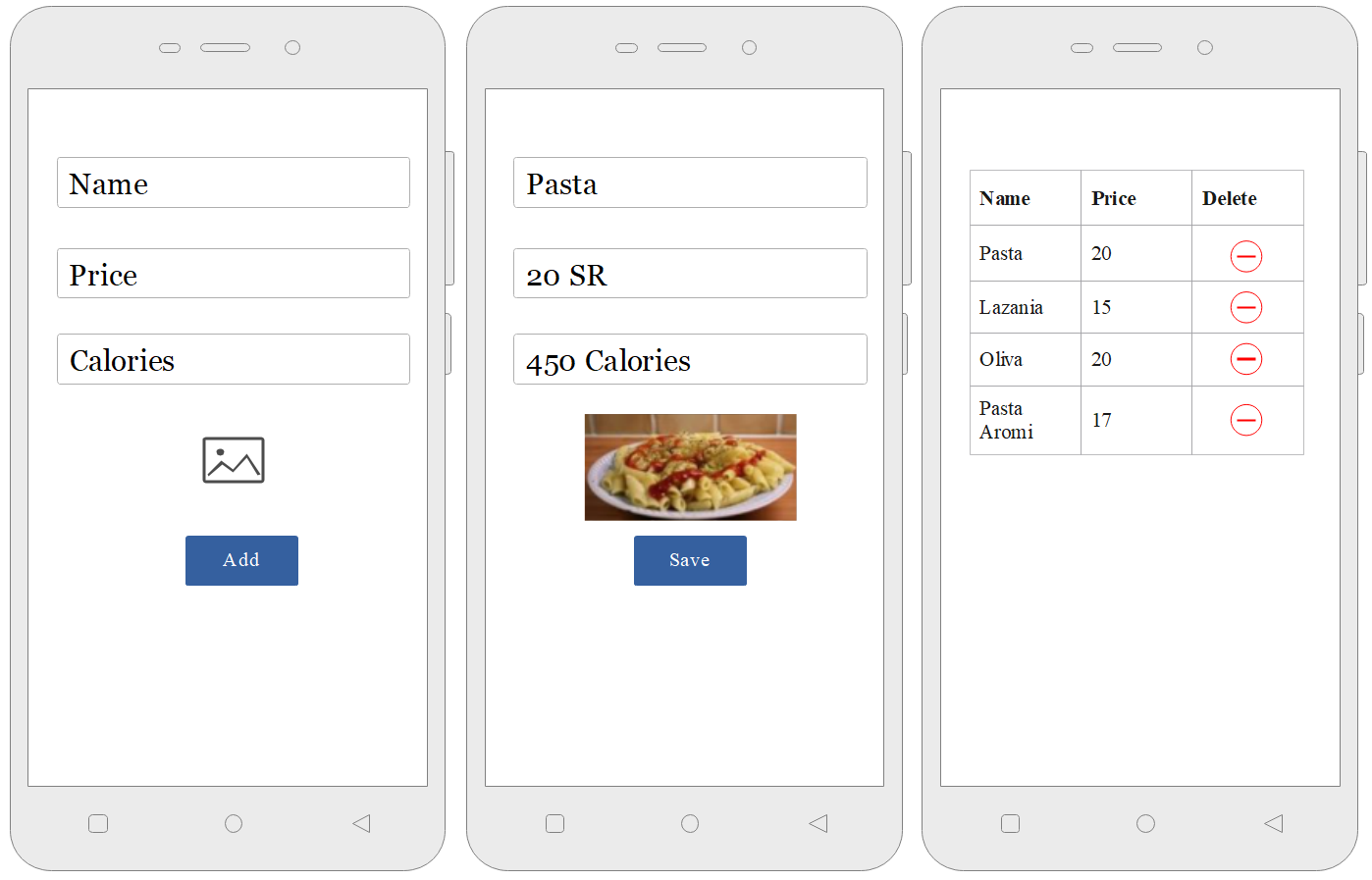


Figure ‑ User interface prototype: Manage meals

#### User story main scenarios “Manage meals”

Table ‑ User story main scenarios: Add meal

|  |  |
| --- | --- |
| **User story** | Add meal |
| **Abstract** | Name, price, calories, and meals photos |
| **Actor** | Restaurant |
| **Precondition** | Login in the application |
| **Post condition** | The meal is saved |
| **Main scenario** | **[Begin]**  1. The actor enter meal information  2. The actor clicks on the "add" button  3. The system saves all the data in the database.  **[End]** |

Table ‑ User story main scenarios: Edit meal

|  |  |
| --- | --- |
| **User story** | Edit meal |
| **Abstract** | Name, price, calories, and meals photos |
| **Actor** | Restaurant |
| **Precondition** | Login in the application |
| **Post condition** | The meal is saved |
| **Main scenario** | **[Begin]**  1. The actor enter meal new information  2. The actor clicks on the "save" button  3. The system saves all the data in the database.  **[End]** |

Table ‑ User story main scenarios: View meal

|  |  |
| --- | --- |
| **User story** | View meal |
| **Abstract** | View meal from restaurant menu |
| **Actor** | Restaurant, Admin |
| **Precondition** | Login in the application |
| **Post condition** | The meal is deleted |
| **Main scenario** | **[Begin]**  1. The actor enters meals screen  2. The system shows meals list.  **[End]** |

Table ‑ User story main scenarios: Delete meal

|  |  |
| --- | --- |
| **User story** | Delete meal |
| **Abstract** | Delete meal from restaurant menu |
| **Actor** | Restaurant |
| **Precondition** | Login in the application |
| **Post condition** | The meal is deleted |
| **Main scenario** | **[Begin]**  1. The actor clicks on the "delete" button  2. The system delete meal from the database.  **[End]** |

#### Design of the user story “Manage meals”

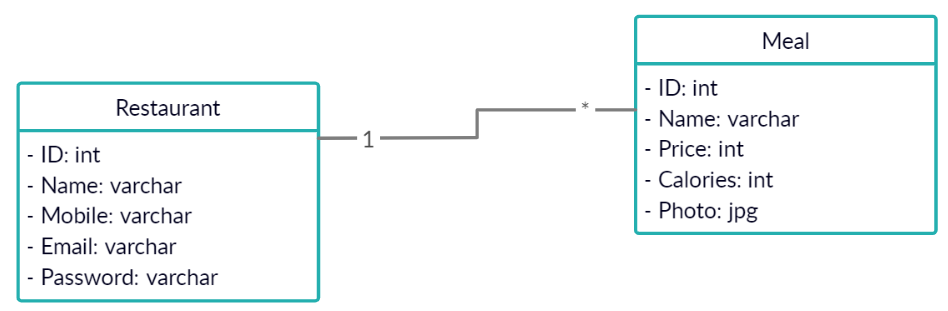


Figure ‑ Class diagram: Manage meals

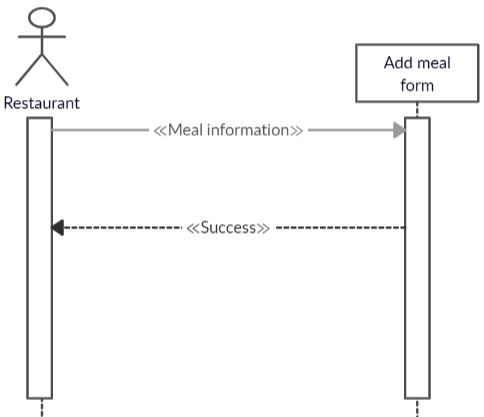


Figure ‑ Sequence diagram: Add meal

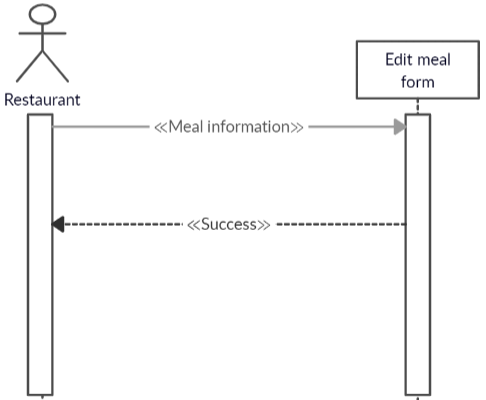


Figure ‑ Sequence diagram: Edit meal

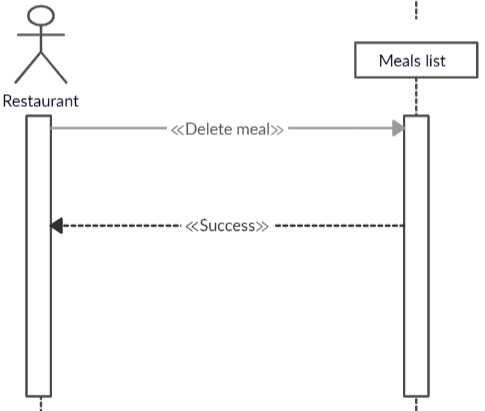


Figure ‑ Sequence diagram: Delete meal

### Description of the user story “Manage orders”

#### Prototype of the user interface “Manage orders

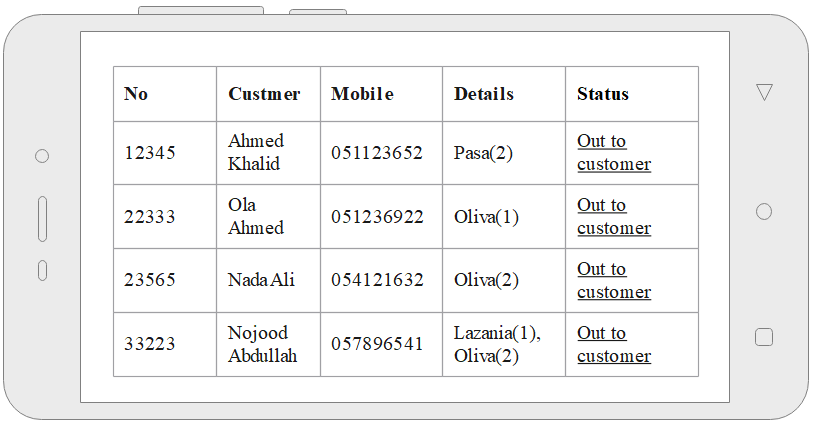


Figure ‑ User interface prototype: Manage orders

#### User story main scenarios “Manage orders

Table ‑ User story main scenarios: View orders

|  |  |
| --- | --- |
| **User story** | View order |
| **Abstract** | View the orders and its details |
| **Actor** | Restaurant, Deliveryman |
| **Precondition** | Login in the application |
| **Post condition** | The meal orders and its details are viewed |
| **Main scenario** | **[Begin]**  1. The actor enter orders page  2. The system shows all order details.  **[End]** |

Table ‑ User story main scenarios: Change status of order

|  |  |
| --- | --- |
| **User story** | Change status of order |
| **Abstract** | Change the status of the order to “out to customer” or delivered |
| **Actor** | Restaurant, deliveryman |
| **Precondition** | Login in the application |
| **Post condition** | The status of order is saved |
| **Main scenario** | **[Begin]**  1. The actor press “out to customer” or “delivered” button  2. The system saves order status in database.  **[End]** |

#### Design of the user story “Manage orders”

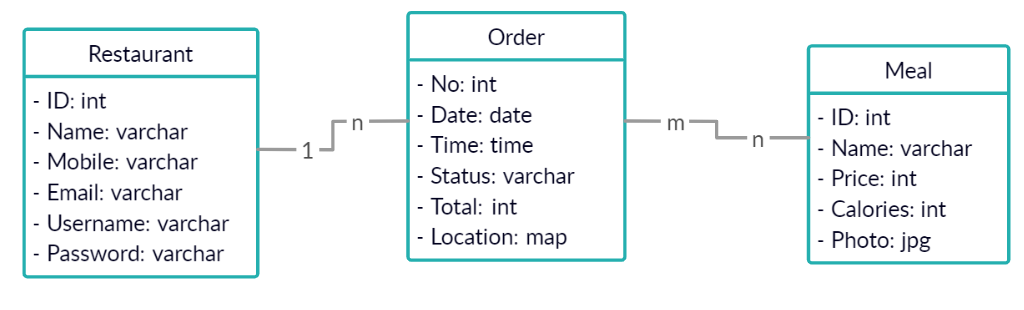


Figure ‑ Class diagram: Manage orders

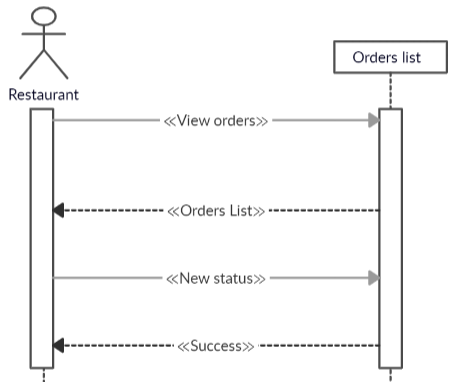


Figure ‑ Sequence diagram: Manage orders

### Description of the user story “Manage users”

#### Prototype of the user interface “Manage users”

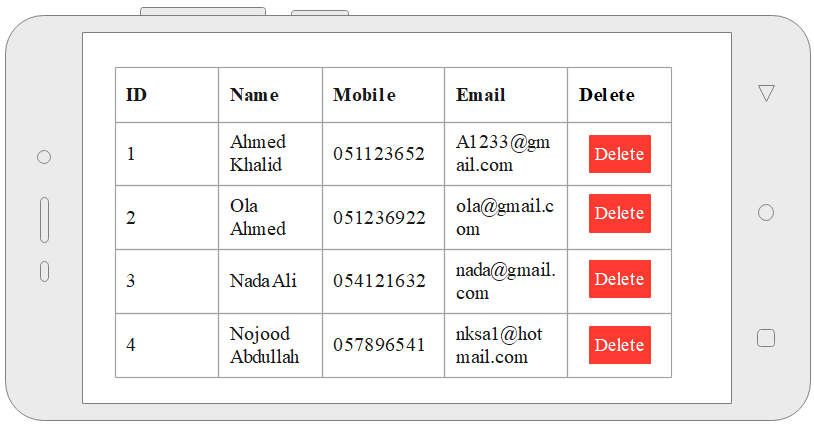


Figure ‑ User interface prototype: Manage users

#### User story main scenarios “Manage users”

Table ‑ User story main scenarios: View users

|  |  |
| --- | --- |
| **User story** | View users |
| **Abstract** | View the users list |
| **Actor** | Admin |
| **Precondition** | Login in the application |
| **Post condition** | The list of users is shown |
| **Main scenario** | **[Begin]**  1. The actor enter users’ page  2. The system shows all users details.  **[End]** |

Table ‑ User story main scenarios: Delete user

|  |  |
| --- | --- |
| **User story** | Delete user |
| **Abstract** | Delete user from the system |
| **Actor** | Admin |
| **Precondition** | Login in the application |
| **Post condition** | The user is deleted from the system |
| **Main scenario** | **[Begin]**  1. The actor press “delete” button  2. The system delete the user from database.  **[End]** |

#### Design of the user story “Manage users”

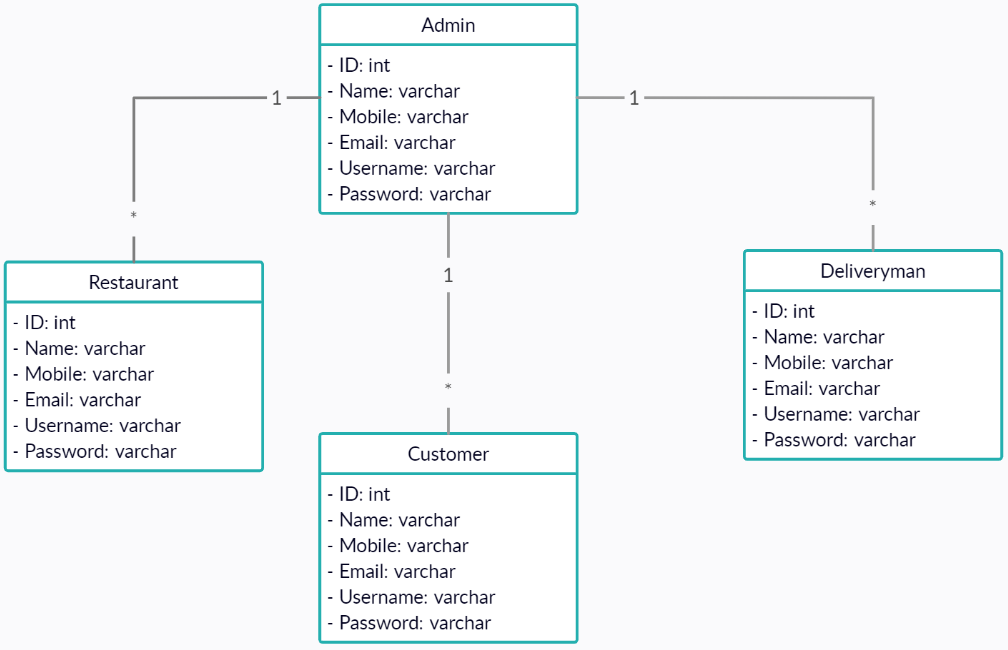


Figure ‑ Class diagram: Manage users

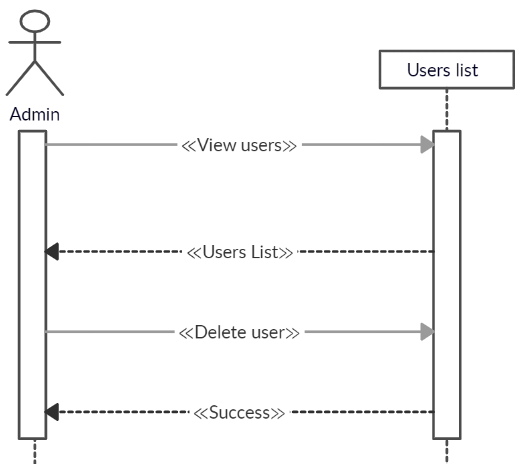


Figure ‑ Sequence diagram: Manage users

## Sprint (3)

### Sprint Backlog

Table ‑ Sprint (3) backlog

|  |  |  |
| --- | --- | --- |
| **ID** | **User Story** | **Tasks** |
| 1 | Search meals | Search meals  Add to cart |
| 2 | View trend meal | View trend meal |
| 3 | Manage cart | Delete meal  Place order |
| 4 | Evaluate | Evaluate meal taste and restaurant |

### Description of the user story “Search meals”

#### Prototype of the user interface “Search meals”

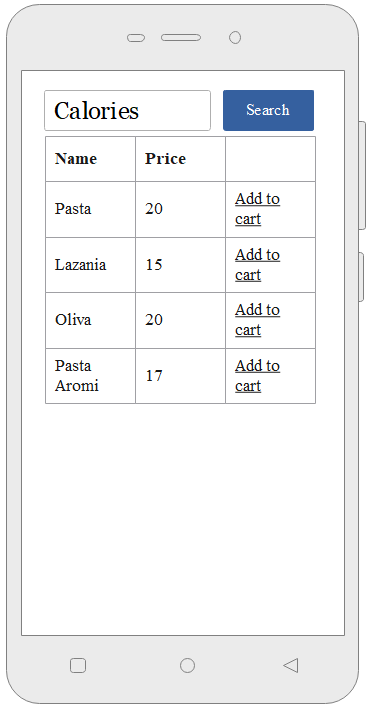


Figure ‑ User interface prototype: Search meals

#### User story main scenarios “Search meals”

Table ‑ User story main scenarios: Search meals

|  |  |
| --- | --- |
| **User story** | Search meals |
| **Abstract** | Search for meals depends on calories |
| **Actor** | Customer |
| **Precondition** | Login in the application |
| **Post condition** | The meals list is shown |
| **Main scenario** | **[Begin]**  1. The actor enter target calories  2. The actor clicks on the "search" button  3. The system show the list of meals.  4. Customer click add to cart  **[End]** |

#### Design of the user story “Search meals”

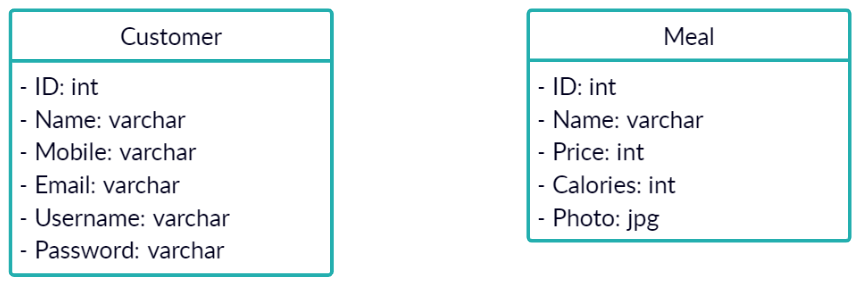


Figure ‑ Class diagram: Search meals

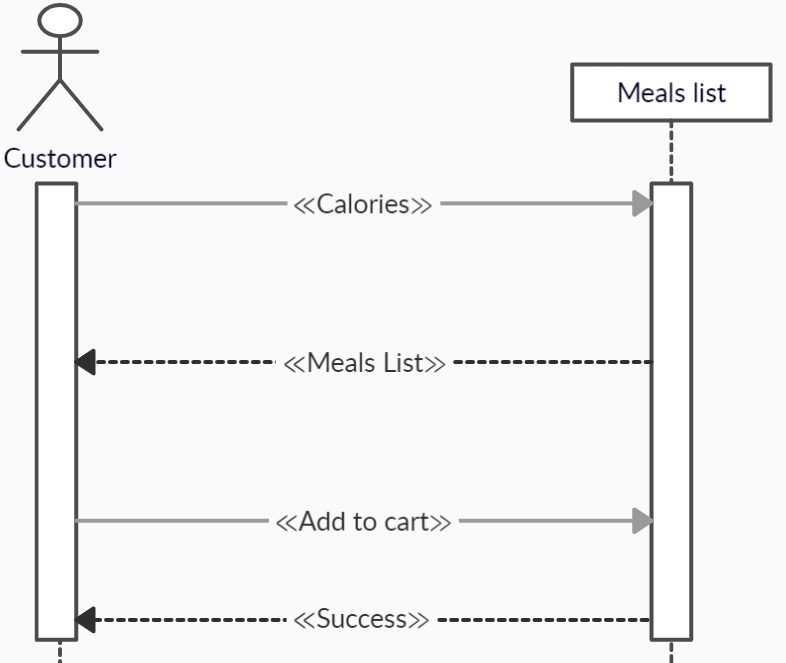


Figure ‑ Sequence diagram: Search meals

### Description of the user story “View trend meals”

#### Prototype of the user interface “View trend meals”

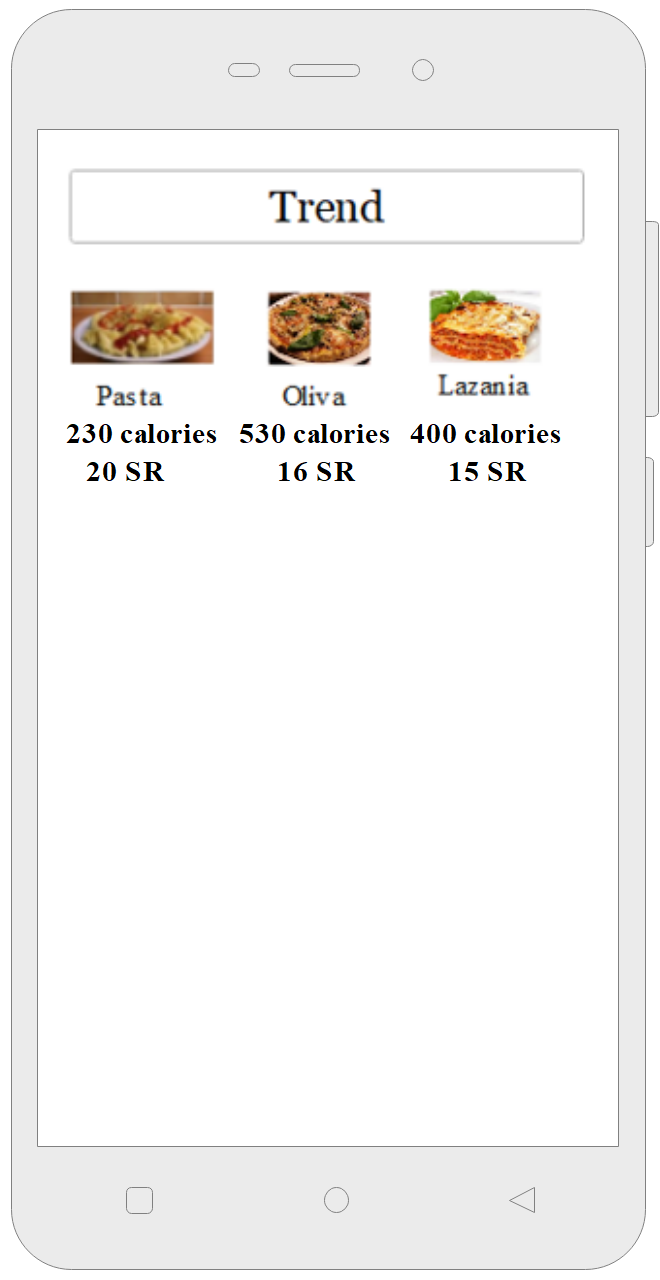


Figure ‑ User interface prototype: View trend meals

#### User story main scenarios: View trend meals

Table ‑ User story main scenarios: View trend meals

|  |  |
| --- | --- |
| **User story** | View trend meals |
| **Abstract** | View trend meals |
| **Actor** | Customer |
| **Precondition** | Login in the application |
| **Post condition** | The trend meals list is shown |
| **Main scenario** | **[Begin]**  1. The actor enter target calories  2. The actor clicks on the "search" button  3. The system show the list of meals.  **[End]** |

#### Design of the user story “View trend meals”

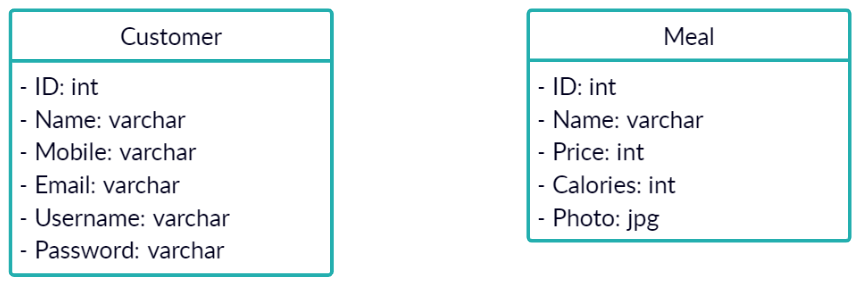


Figure ‑ Class diagram: View trend meals

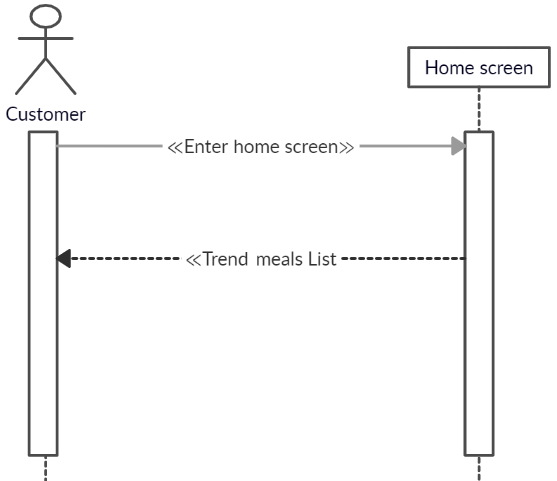


Figure ‑ Sequence diagram: View trend meals

### Description of the user story “Manage cart”

#### Prototype of the user interface “Manage cart”

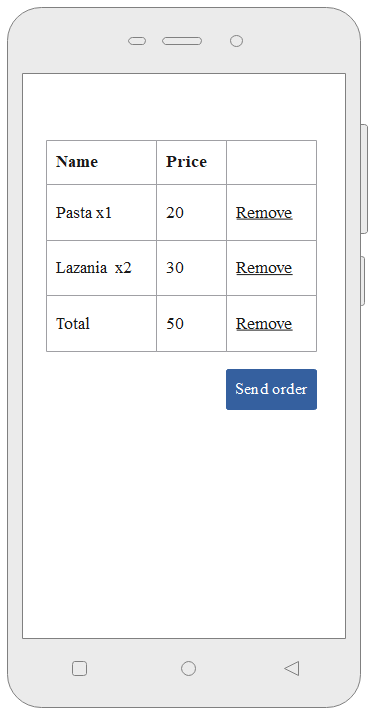


Figure ‑ User interface prototype: Manage cart

#### User story main scenarios: Manage cart

Table ‑ User story main scenarios: Manage cart

|  |  |
| --- | --- |
| **User story** | Manage cart |
| **Abstract** | Remove item from cart or send order |
| **Actor** | Customer |
| **Precondition** | Login in the application |
| **Post condition** | The order is sent |
| **Main scenario** | **[Begin]**  1. The actor enters cart screen  2. The actor clicks on the "Send order" button  3. the actor pay the custom of the order either on delivery payment or online payment.  3. The system saves order in the database.  **[End]** |

#### Design of the user story “Manage cart”

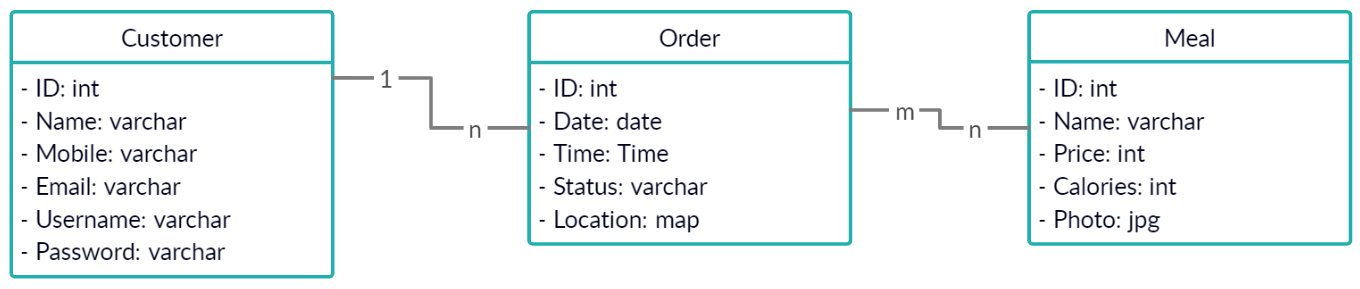


Figure ‑ Class diagram: Manage cart

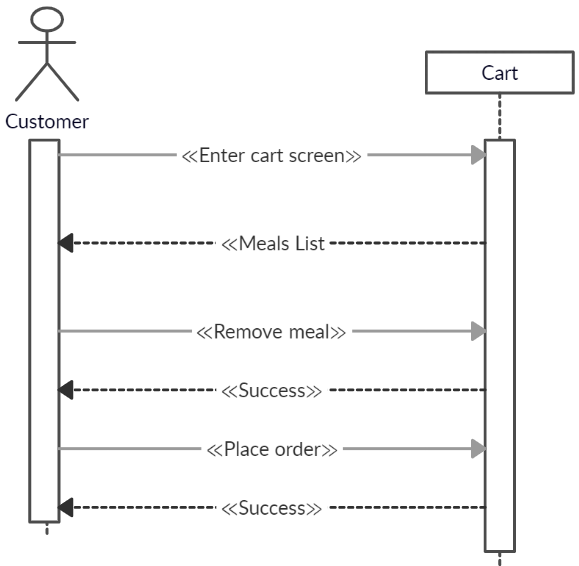


Figure ‑ Sequence diagram: Manage cart

### Description of the user story “Evaluation”

#### Prototype of the user interface “Evaluation”

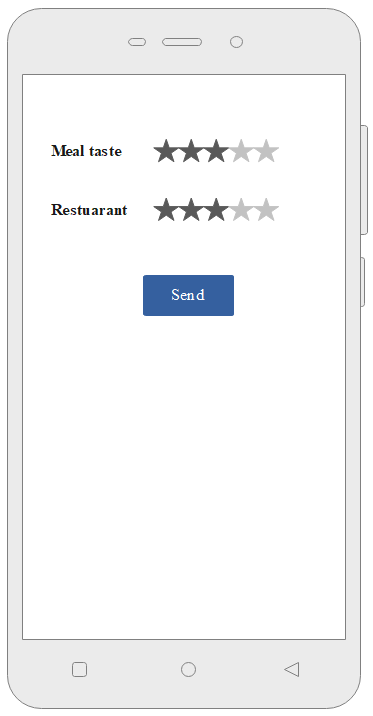


Figure ‑ User interface prototype: Evaluation

#### User story main scenarios: Evaluation

Table ‑ User story main scenarios: Evaluation

|  |  |
| --- | --- |
| **User story** | Evaluation |
| **Abstract** | Evaluate meal taste and restaurant |
| **Actor** | Customer |
| **Precondition** | Login in the application |
| **Post condition** | The evaluation is saved |
| **Main scenario** | **[Begin]**  1. The actor select evaluation  2. The actor clicks on the "send " button  3. The system saves evaluation in the database.  **[End]** |

#### Design of the user story “Evaluation”

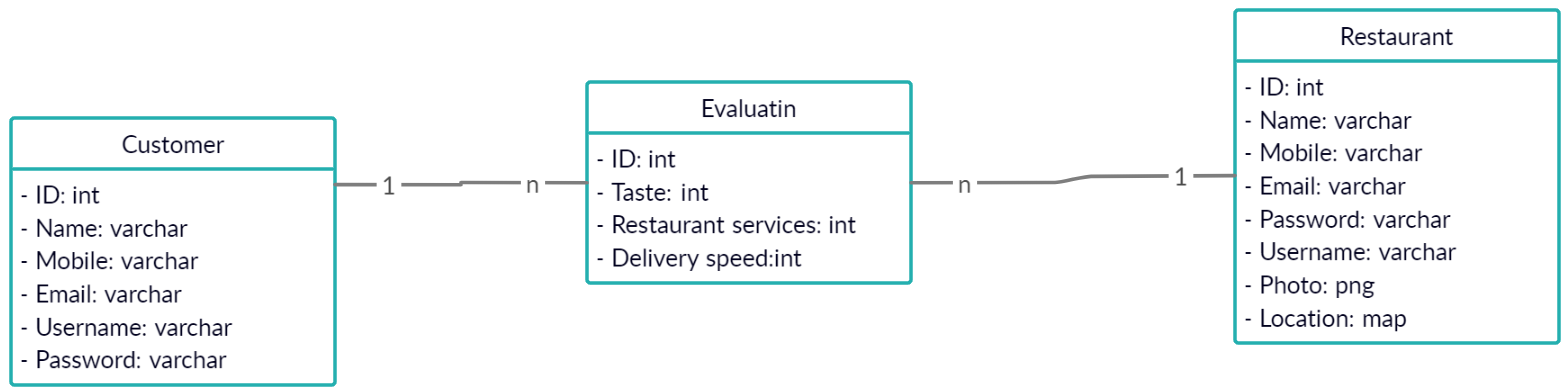


Figure ‑ Class diagram: Evaluation

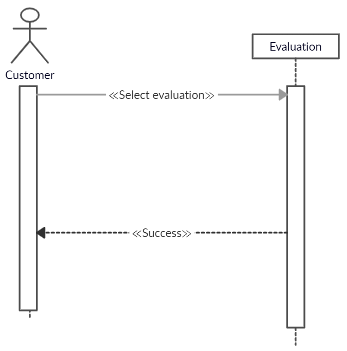
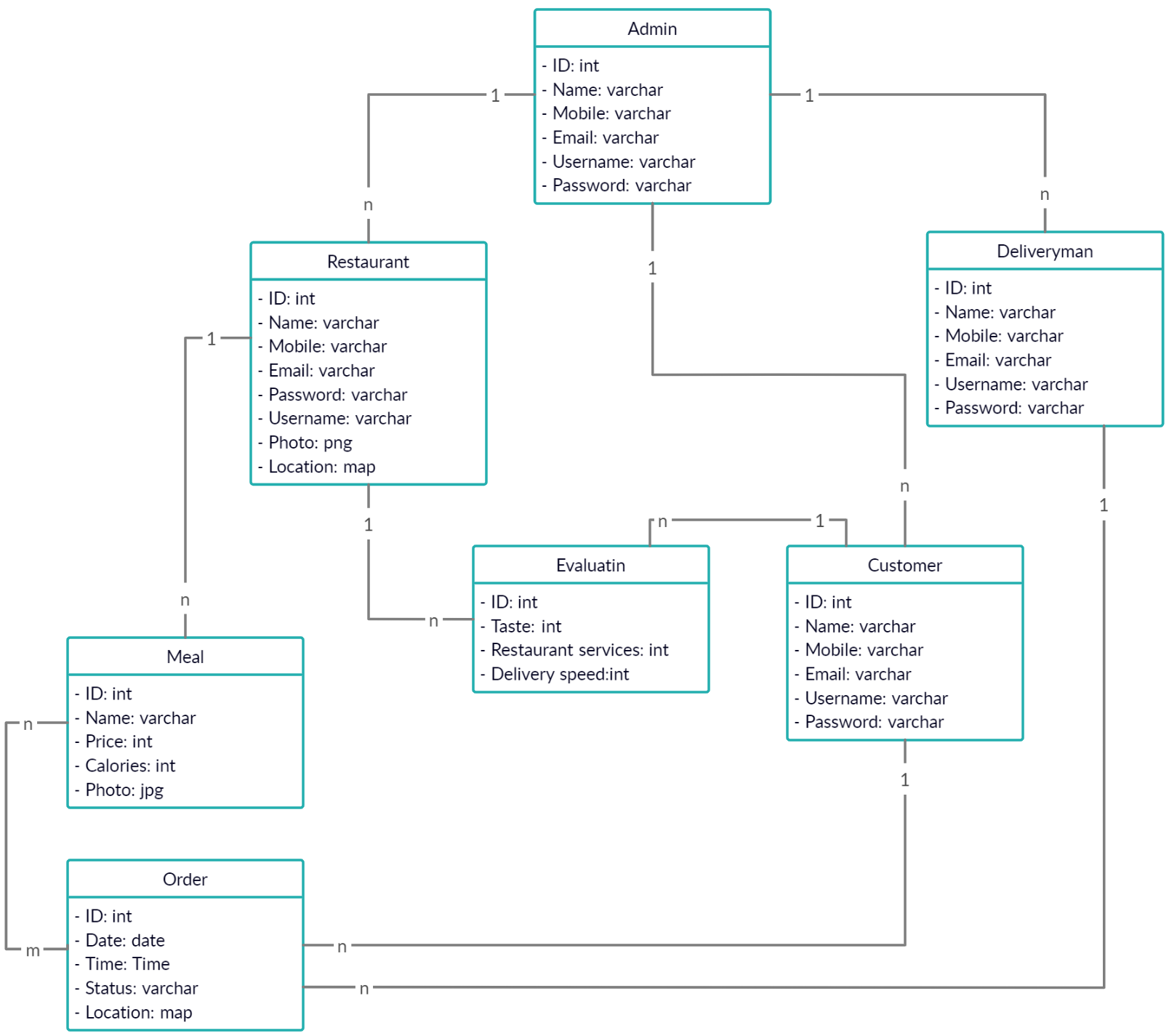


Figure ‑ Sequence diagram: Manage cart

## Data base design

### The system class diagram



Figur e ‑ System class diagram

### The Relational Database Schema

**Admin** (**AID**, Name, Mobile, Email, Username, Password)

**Deliveryman** (**DID**, Name, Mobile, Email, Username, Password)

**Customer** (**CID**, Name, Mobile, Email, Username, Password)

**Restaurant** (**RID**, Name, Mobile, Email, Location, Photo, Username, Password)

**Meal** (**MID**, Name, Price, Calories, Photo, **#RID**)

**Order** (**OID**, Date, Time, Location, Status, **#CID, #DID**)

**Evaluation** (**EID**, Taste, RestaurantQuality, Deliveryspeed, **#CID, #RID**)

**OrderMeals** (#**OID, #MID,** qty)

## Conclusion

By the end of this chapter the analysis and designed phases are finished, this chapter included finding the requirements of the system, draw UML diagrams, and design database.

# References

|  |  |
| --- | --- |
| [1] | "mobile commerce," [Online]. Available: https://www.businessinsider.com/mobile-commerce-shopping-trends-stats. |
| [2] | "talabat," [Online]. Available: https://www.talabat.com. |
| [3] | "ubereats," [Online]. Available: https://www.ubereats.com. |
| [4] | "myfitnesspal," [Online]. Available: https://www.myfitnesspal.com/. |
| [5] | "sdlc overview," [Online]. Available: https://www.tutorialspoint.com/sdlc/sdlc\_overview.htm. [Accessed 25 10 2018]. |
| [6] | "Agile methodology," [Online]. Available: http://www.tutorialspoint.com/sdlc/sdlc\_agile\_model.htm. [Accessed 1 11 2018]. |
| [7] | "Use Case," [Online]. Available: https://www.researchgate.net/publication/220373667\_Use\_Case\_Diagrams\_in\_Support\_of\_Use\_Case\_Modeling\_Deriving\_Understanding\_from\_the\_Picture. |
| [8] | "non functional requirements," [Online]. Available: https://reqtest.com/requirements-blog/functional-vs-non-functional-requirements/. |
| [9] | "product backlog," [Online]. Available: https://www.wibas.com/scrum/product-backlog/en. [Accessed 1 11 2018]. |