**NOTE: This template is shareware downloaded from** [**www.processimpact.com**](http://www.processimpact.com)**. All shareware payments are donated to the Norm Kerth Benefit Fund to help a consultant who is disabled with a brain injury. Please visit** [**http://www.processimpact.com/norm\_kerth.html**](http://www.processimpact.com/norm_kerth.html) **to make a shareware payment ($10 suggested). Thank you!**

Software Requirements Specification

for

Turkish Legends

Version 2.0 approved

Prepared by Antoinet Dorian, Delenclos Victor, Davias Aymeric

Villejuif Kebab Brigade

22/05/2018

Table of Contents

Table of Contents ii

Revision History iii

1. Introduction 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Project Scope 1

1.5 References 1

2. Overall Description 1

2.1 Product Perspective 1

2.2 Product Features 1

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 2

2.6 User Documentation 3

2.7 Assumptions and Dependencies 3

3. System Features 4

3.1 System Feature 1 6

3.2 System Feature 2 (and so on) 6

4. External Interface Requirements 8

4.1 User Interfaces 8

4.2 Hardware Interfaces 11

4.3 Software Interfaces 11

4.4 Communications Interfaces 11

5. Other Nonfunctional Requirements 12

5.1 Performance Requirements 12

5.2 Safety Requirements 12

5.3 Security Requirements 12

5.4 Software Quality Attributes 12

6. Other Requirements 13

Appendix A: Glossary 13

Appendix B: Analysis Models 13

Appendix C: Issues List 16

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason for Changes** | **Version** |
| Initial Software | 11/05 | None | 1.0 |
| Final Version | 18/05 | Payment added. Hash Added. | 2.0 |

# Introduction

## Purpose

The product we are specifying here is the Turkish Legends v2.0. This PC-exclusive software aims to help customers to order customizable kebabs by choosing their ingredients, customers being registered in a database with logins, so they can save their kebabs easily. The software also allows staff members to consult the orders, and validate it.

## Document Conventions

The typographical conventions are the following:

* Arch-Title: Calibri 18, Bold, Underlined
* Titles: Calibri 14, Bold
* Others: Arial 11

## Intended Audience and Reading Suggestions

This document is for everybody interested in this project, since the project is finished, it can be used by both developers and project managers; and as computer science skills are not mandatory, it can be read by any user and marketing staff member.

## Project Scope

This software allows hungry customers to buy kebabs online, and on the other hand allows staff members to deal with previous orders. The project could evolve by adding a PayPal functional checkout function, for example.

## References

<https://stackoverflow.com> – Used for Hashing

<https://docs.oracle.com/javase/7/docs/api/java> – Used for GUI parts

# Overall Description

## Product Perspective

Turkish Legends is inspired from Subway fastfood, because you can customize your Kebab like you customize your sandwich at Subway. The deliver aspect is also interesting in the market with the growth of Uber Eat, and Deliveroo.

## Product Features

As a customer:

You can login or register if you haven’t an account yet.

You can create a Kebab, that means you can select a bread, one or multiple vegetables, none, or multiple sauce, and one meat.

You can save you Kebab in order to be able to command to command it faster.

You can order it.  
When you order a Kebab, you can select how you pay between cash or credit card.

As an admin:

You can see and modify orders’ status.

## User Classes and Characteristics

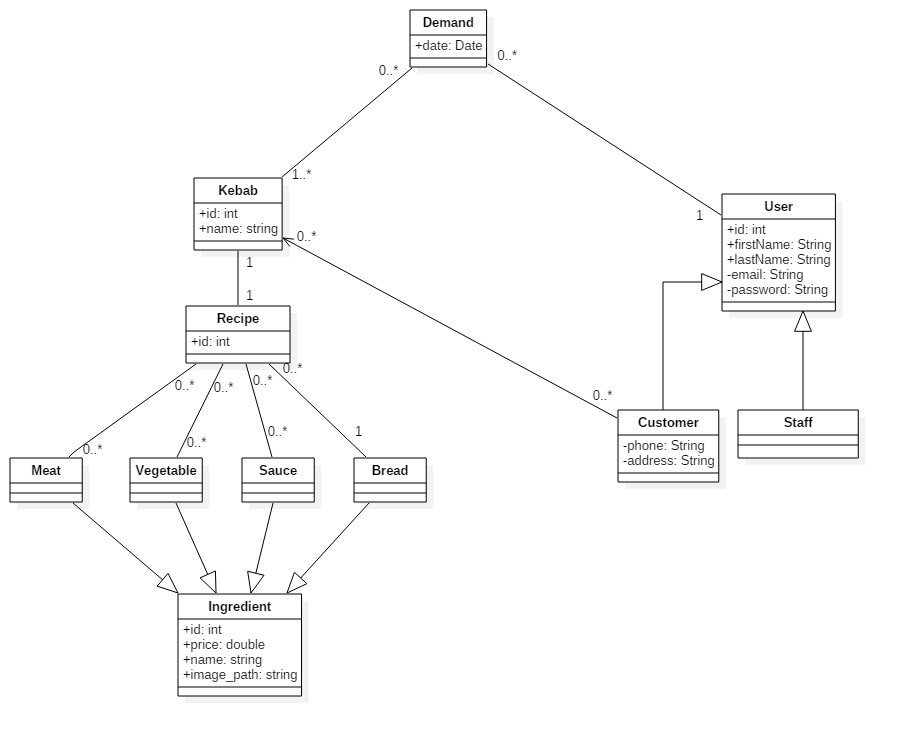
The most important user is the customer. He has access to his saved Kebab, and his current orders. His account password is hashed and his informations, such as name or address, are stored in the database.

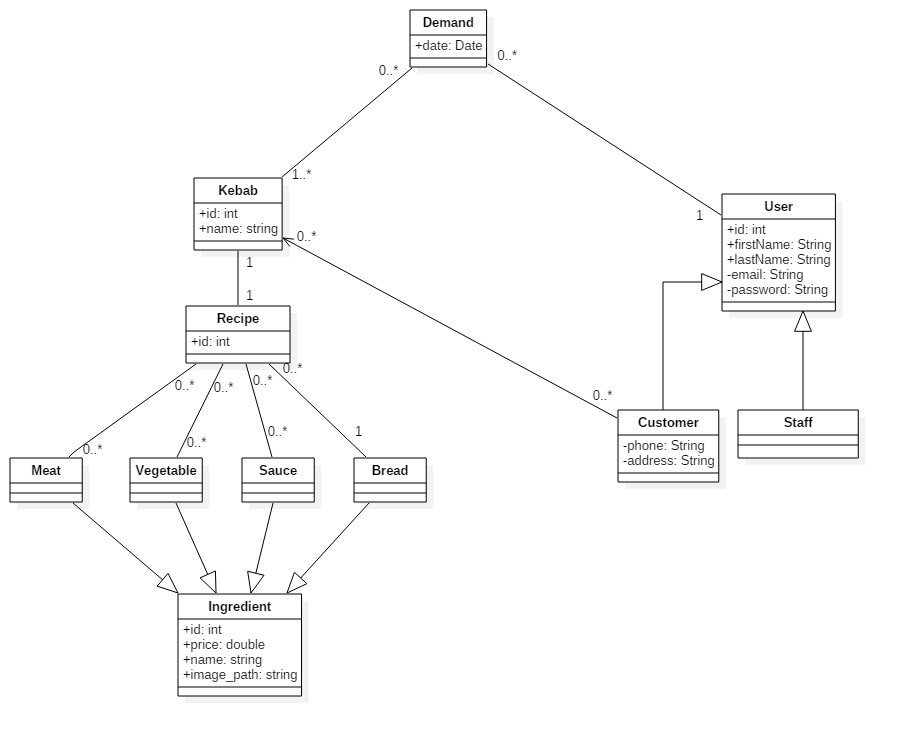
The second user is the staff. He has access to all orders and all users’ informations. He can modify the status of whatever order he wants.

## Operating Environment

The software is made for PC, Windows exclusively. MySQL is needed. The latest version of Java is also needed.

## Design and Implementation Constraints

The application is made in Java. The database used is an SQL database. The database must be named “turkishlegends” and there must be the following tables:



## User Documentation

For the customer:

You have to login in to use the app. If you don’t have an account, you have to register. Then you have to enter all informations asked. Then you can log in.

To create your Kebab, you have to select a bread. You can select a meat, one to 3 vegetables, one sauce. To save your Kebab, press “Create a Legend”. Then you’ll see your Kebab in the bottom left of the window. You can select the Kebab you want. Then you’ll see what’s in it, and the price. You can press “Order” to order it, or “Delete” to delete it.

For the staff:

You have to login. You can see the different orders. You can modify the status of an order by selecting the order and pressing “Next Step” to go to the next step, or “Previous Step” to go to the previous one. You have to refresh to see new orders by pressing “Refresh”

## Assumptions and Dependencies

Please make sure you have the database running; before launching the software, or the link with database may not work.

## Budget

Each member is paid 22$ per hour. We worked for 10h per person (30h total approx.) , so we need 220$ per person. We are 3 so the bill is 660$.

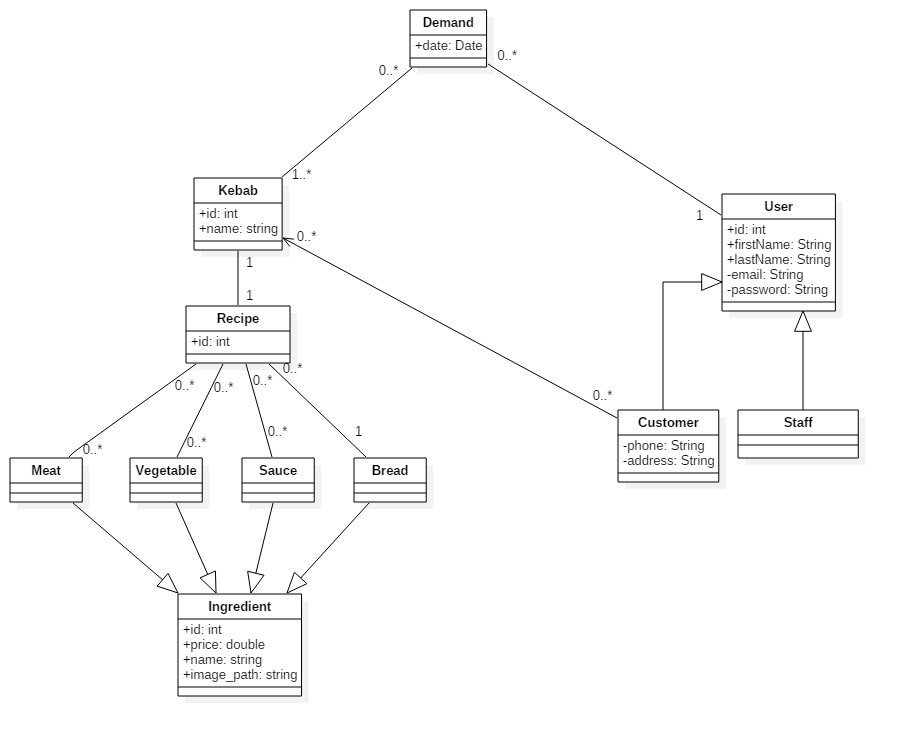
# System Features

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

## Project Features

For the moment, the customer can customize his Kebab, save and order it. Select how he pays. The staff can modify the status of the order.

We can add a feature you say to the customer if there is a problem with his order, for example if an ingredient is missing, or if there is a problem with his address.



## Use Cases

## 

Scenarios :

1-Refresh

|  |  |  |
| --- | --- | --- |
| **Number** | 1 | |
| **Name** | Refresh | |
| **Summary** | Refresh the list of orders | |
| **Priority** | Normal | |
| **Preconditions** | REQ-1: being logged in as a staff member | |
| **Postconditions** | None | |
| **Primary actor(s)** | Staff member | |
| **Secondary actor(s)** | Customer | |
| **Trigger** | Clicking the “refresh” button | |
| **Main scenario** | **Step** | **Action** |
|  | 1 | The orders are retrieved from the database |
|  | 2 | The list of orders in the interface is cleared |
|  | 3 | The list of orders is updated in the interface |
| **Extensions** | **Step** | **Branching action** |
|  |  |  |
| **Open issues** | Issue #1 | The current trigger is not necessarily adapted to the use of the application. An observer design pattern would be more efficient. |

2-Create

|  |  |  |
| --- | --- | --- |
| **Number** | 2 | |
| **Name** | Create and Save | |
| **Summary** | Create and Save Kebabs | |
| **Priority** | Normal | |
| **Preconditions** | REQ-1: being logged in as a customer | |
| **Postconditions** | None | |
| **Primary actor(s)** | Customer | |
| **Secondary actor(s)** | None | |
| **Trigger** | Clicking the “Create Legend” button | |
| **Main scenario** | **Step** | **Action** |
|  | 1 | The Name is checked |
|  | 2 | The Bread, Vegetables, Meat and Sauce are checked |
|  | 3 | The Kebab is created in Database |
|  | 4 | The Kebab is displayed in Interface |
| **Extensions** | **Step** | **Branching action** |
|  | 1 | Name is invalid -> Back to step 1 |
| **Open issues** |  |  |

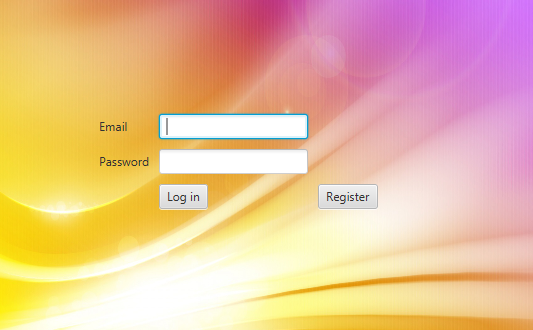
3- Modify

|  |  |  |
| --- | --- | --- |
| **Number** | 3 | |
| **Name** | Modify | |
| **Summary** | Modify Order | |
| **Priority** | Normal | |
| **Preconditions** | REQ-1: being logged in as a staff member | |
| **Postconditions** | None | |
| **Primary actor(s)** | Staff member | |
| **Secondary actor(s)** | Customer | |
| **Trigger** | Clicking the “Previous Step” or “Next Step” button | |
| **Main scenario** | **Step** | **Action** |
|  | 1 | The software checks if the move is valid (you cant Previous Step a step 1) |
|  | 2 | The selected order is updated in the database |
|  | 3 | The selected order is updated in the interface |
| **Extensions** | **Step** | **Branching action** |
|  |  |  |
| **Open issues** |  |  |

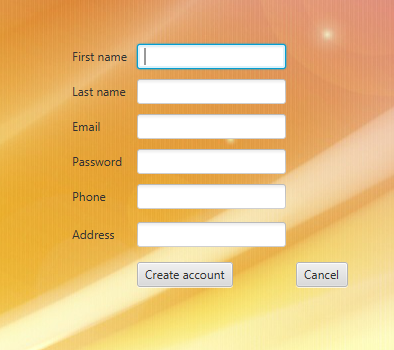
# External Interface Requirements

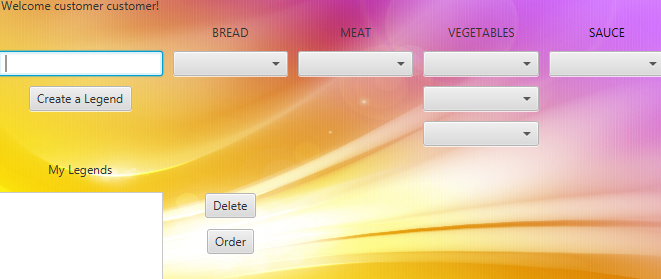
## User Interfaces

Both types of users:

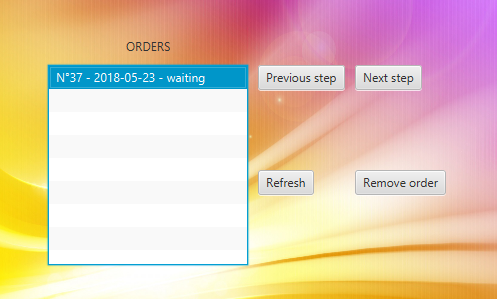
Application Start with Login:

Customer Only:

When Register is pressed from login page:

When logged (still as customer):

You can choose, from left to right, Bread/Meat/Vegetables and Sauce, then press Create a Legend Button to have your Kebab created in the database. Created kebabs will show up on bottom left corner. You can select one from this list, then press either delete or order depending on what you want to do with these kebabs. When Order is picked, you’ll be ask to choose your mean of payment to complete the order.

On Logged as Staff Member:

As staff member, you must select an order, then click either Previous Step or Next Step buttons to validate (or not) customers’ orders. You can also hit Refresh to get the latest orders, or remove the selected order with the Remove button.

## Hardware Interfaces

As the software is for now only supported on Windows, there is not much to tell about this.

## Software Interfaces

When an action is performed on our database via the Create a Legend/Delete Kebab/Previous Step/Next Step/Remove Order buttons, our software uses MySQL to perform the operations on our database. We also use JavaFX for all the GUI.

## Communications Interfaces

None used, sorry.

# Other Nonfunctional Requirements

## Performance Requirements

As the application is pretty easy to run for a computer, it should be okay for every windows-booting PC.

## Safety Requirements

None is needed since we already got a security on possible empty fields the user can leave on register screen.

## Security Requirements

This software is not designed for vegan customers; you can custom your kebab to make a vegan one, but if you can’t the possibility of having meat in a kebab, this software is not recommended.

## Software Quality Attributes

It is allowed to not fill attributes fields, the software will just ask to reenter values that are mandatory, such as email on register for example.

# Other Requirements

Appendix A: Ids

You can connect to a admin account with following identifiers:

Mail : admin

PW : admin

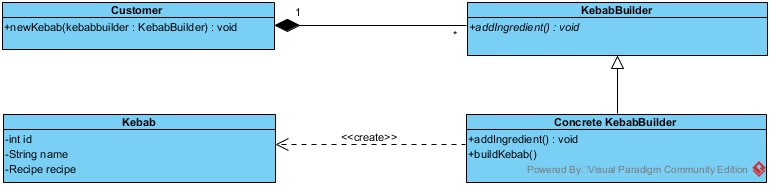
You can connect to a customer test account with following identifiers :

Mail : custom@er

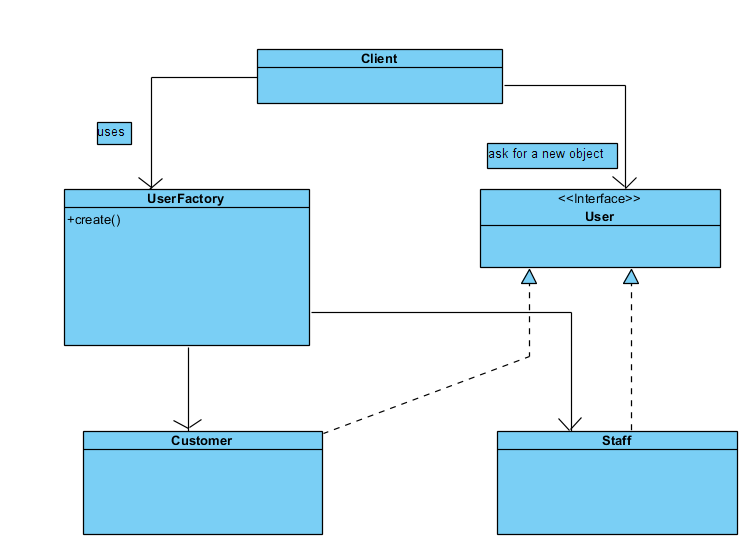
PW : customer

Appendix B: Analysis Models

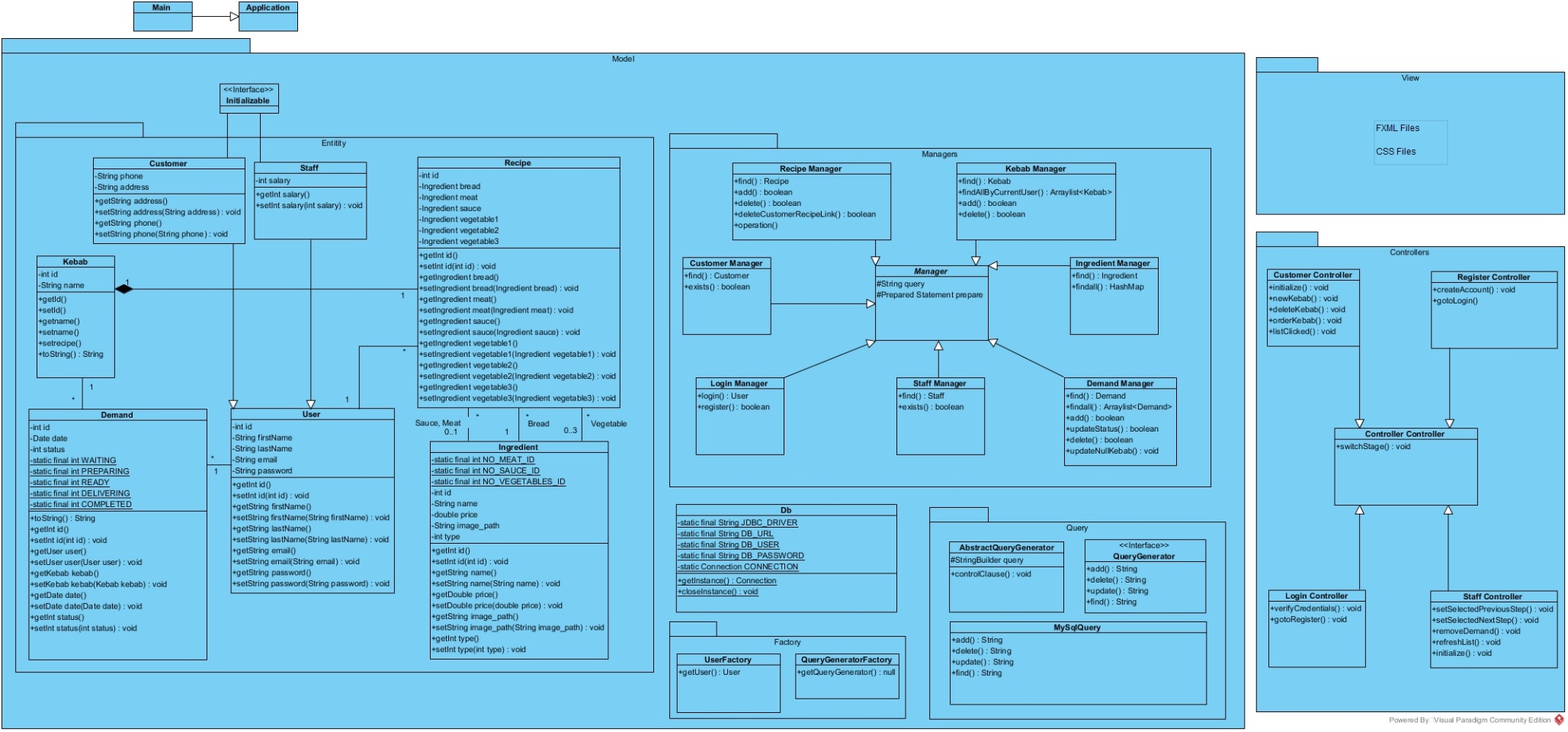
Builder Pattern:



Factory Pattern :



System Class Diagram on next page:



Appendix C: Issues List

Not really an issue, but Credit Card Number is useless and not stored anywhere; the payment isn’t made by the way, it’s a school project, not a one-armed bandit.