**Easy Menu**

Project members

|  |  |  |
| --- | --- | --- |
| Reem hamood | 341203834 | s.mu.edu.sa@341203834 |
| Sarah Alsalman | 341204563 | s.mu.edu.sa@341204563 |
| Ghadi Aloqaily | 341205118 | s.mu.edu.sa@341205118 |

Deadline for submission : 3rd May 2017.

**Chapter1: Introduction**

* **The Idea of the App**

We have a bunch of apps on how to cook, but this one will use sort of search technology to match the ingredients you have (so you don't have to rush to find things at the shop) and menu's list which can be served with such ingredients in your kitchen.

We can also combine with real-time suggestions by other people who is online as well

* **Problem definition**
* This application will help to find an appropriate dish.
* This application will help you to find a dish according to what ingredients you already have.

**Chapter 2: General Description**

* **Overview of all possible scenarios including their functionalities**

The easy menu app is to prepare a list of foods and drinks in a series of dishes to satisfy customers and prepare them for public and private events.

The main function of this application make a good dish based on what ingredients that you have. Also, it can calculate the calories in each dish.







User

**Admin**

- View menu: view existing menu to eat.

- Select category: categories available or available to eat.

- Type of food: determine the quality of eating.

- Sign in: the application should allow users to access.

- Add note: add comments.

- Keep adding

- Add recipes: calculate calories and protein.

* **Product perspective and characteristics.**

this app will use the sort of search

technology to match the ingredients you have (so you don't have to rush to find things at the shop) and menu's list which can be served with such ingredients in your kitchen.

We can also combine with real-time suggestions by other people who is online, as this application can help people choose the right dish in certain quantities.

The main features are a window to display

the ingredients so that the user can choose or see all options.

* **Product Functions**
* Product functions which your product will do

The app first will

- display the log in or enter as guest page

- then it will display what type of food and ingredients

- after the user will choose the ingredients they app will display the dishes

- after the user choose the dish it will display the ingredients, instructions and it will calculate the calories and protein.

- the app allow user can put private note they public the note with the app community

- there will be menu based on type of the food sweet, main course

, breakfast, drinks, sea food, Indian food etc .

* **Assumptions and Dependencies.**

means what have you assumed for example there was no such app or there were apps available with limited functions etc and why you decided to go for it. Dependencies should reflect the needs that your product a\must acquire to function such as some specific hardware or software or networks.

There is no such application now, a new opportunity for users

there sometimes people have some amounts but do not know the appropriate dish for it

**Chapter 3: Requirements**

1. **Login Requirements**

|  |  |
| --- | --- |
| Requirement  number | Requirement statement |
| R1 | The system shall provide facility to login access the system |

1. **Forgot Password Requirements**

|  |  |
| --- | --- |
| Requirement  number | Requirement statement |
| R2 | The system shall allow the user reset password |

1. **Logoff Requirements**

|  |  |
| --- | --- |
| Requirement  number | Requirement statement |
| R3 | The system shall provide for user logoff from the application |

4. **Display components requirements**

|  |  |
| --- | --- |
| Requirement  number | Requirement statement |
| R4 | The system must provide components and basic components |
| R5 | The system must provide the general level of details that should be mentioned as the name of the dish, and from which country they belong |
| R6 | Display the total calorie of the user in the dish |

**5- View, edit, and save the dish**

|  |  |
| --- | --- |
| Requirement  number | Requirement statement |
| R8 | Mention some flavors suitable for the dish |
| R9 | Add a picture of the dish after applying the ingredients |
| R10 | The possibility to save the dish and delete the image |
| R11 | Share the dish with friends |
| R12 | Ability to evaluate and comment |

**6-Send notifications**

|  |  |
| --- | --- |
| Requirement  number | Requirement statement |
| R13 | Ability to send notifications to the user for reminders |

1. **Sound selection Requirement**

|  |  |
| --- | --- |
| Requirement  number | Requirement statement |
| R14 | The system shall provide for user the ability to choose the sound for the notifications and alerts. |

1. **Control Requirement**

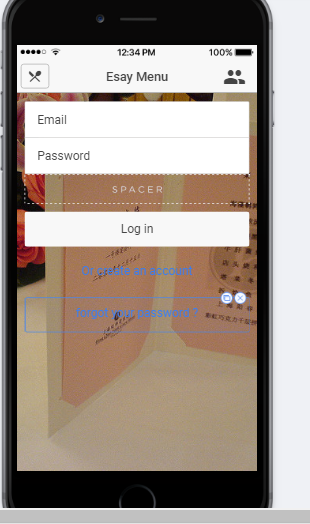
|  |  |
| --- | --- |
| Requirement  number | Requirement statement |
| R14  5 | The system should provide the ability to modify, delete, and save the dish |

**6- External Interface Requirements**

|  |  |
| --- | --- |
| Requirement  number | External application requirements are hardware, software, or database elements that a system or component must interact with it ... |
| R15 | Design the application's exterior |

* + 1. **User Interfaces**

|  |  |
| --- | --- |
| **ID/Number** | **Requirement Text** |
|  | The public interface of the application should help attract and use the user |



## 1.2 non-Functional Requirements

* + 1. **Usability**

|  |  |
| --- | --- |
| **ID/Number** | **Requirement Text** |
|  | The application and its interfaces must be easy to useful and manipulate |

* + 1. **Database**

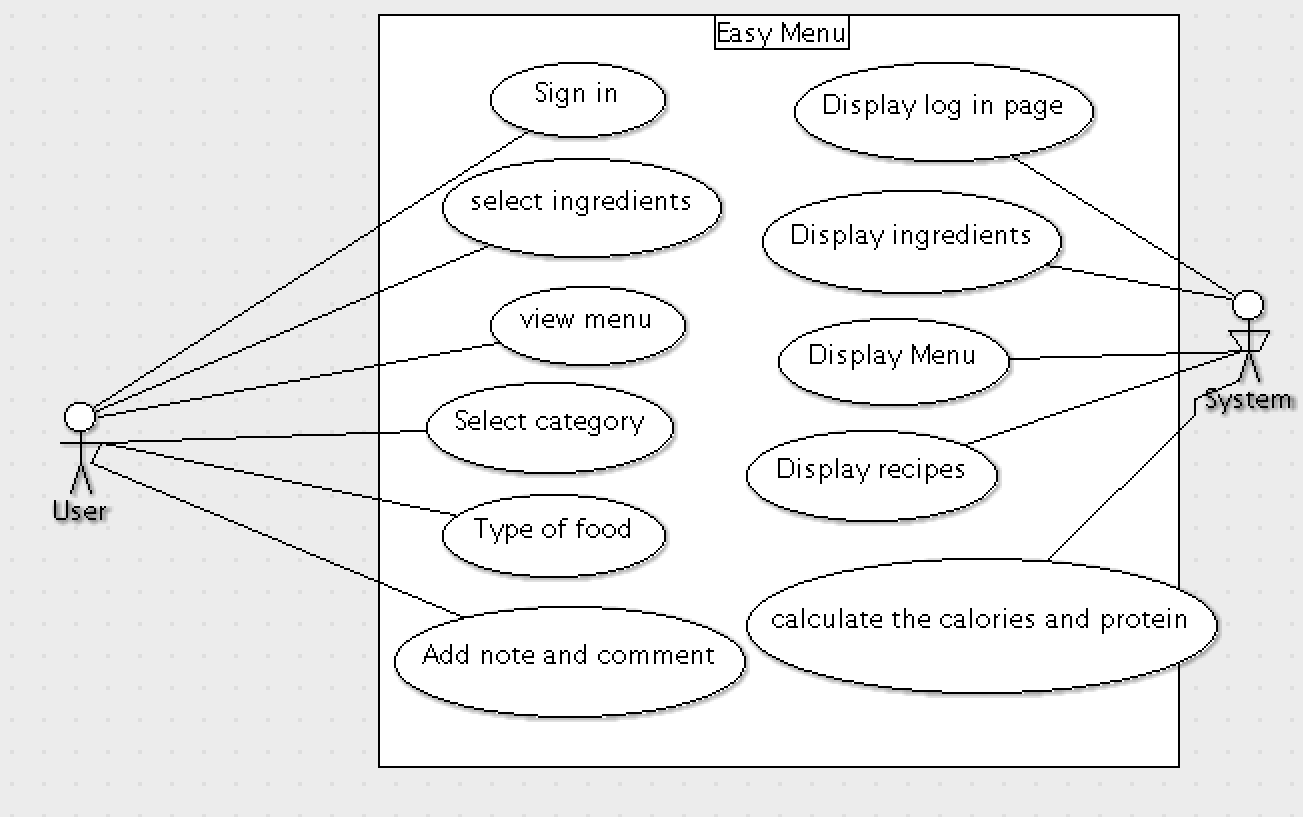
|  |  |
| --- | --- |
| **ID/Number** | **Requirement Text** |
|  | The system should allow the provision of a data security base to protect existing or stored data from those trying to access. |

* + 1. **Hardware Interfaces**

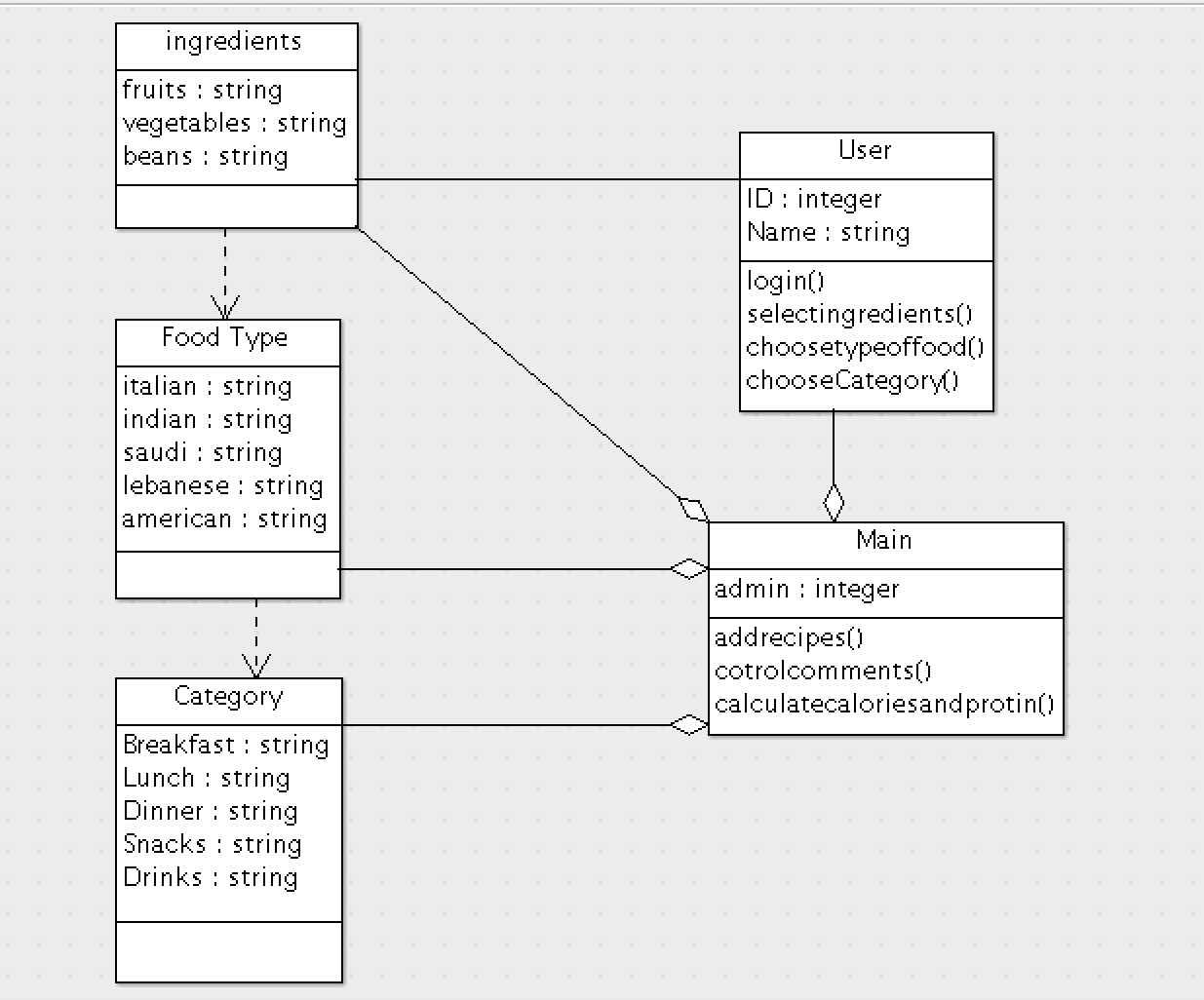
|  |  |
| --- | --- |
| **ID/Number** | **Requirement Text** |
|  | The main features that must be possessed by the user and the programmer:  PC:  Memory  Time\date  Control Panel  The language  And communication over the Internet |

**Chapter 4: UML design**

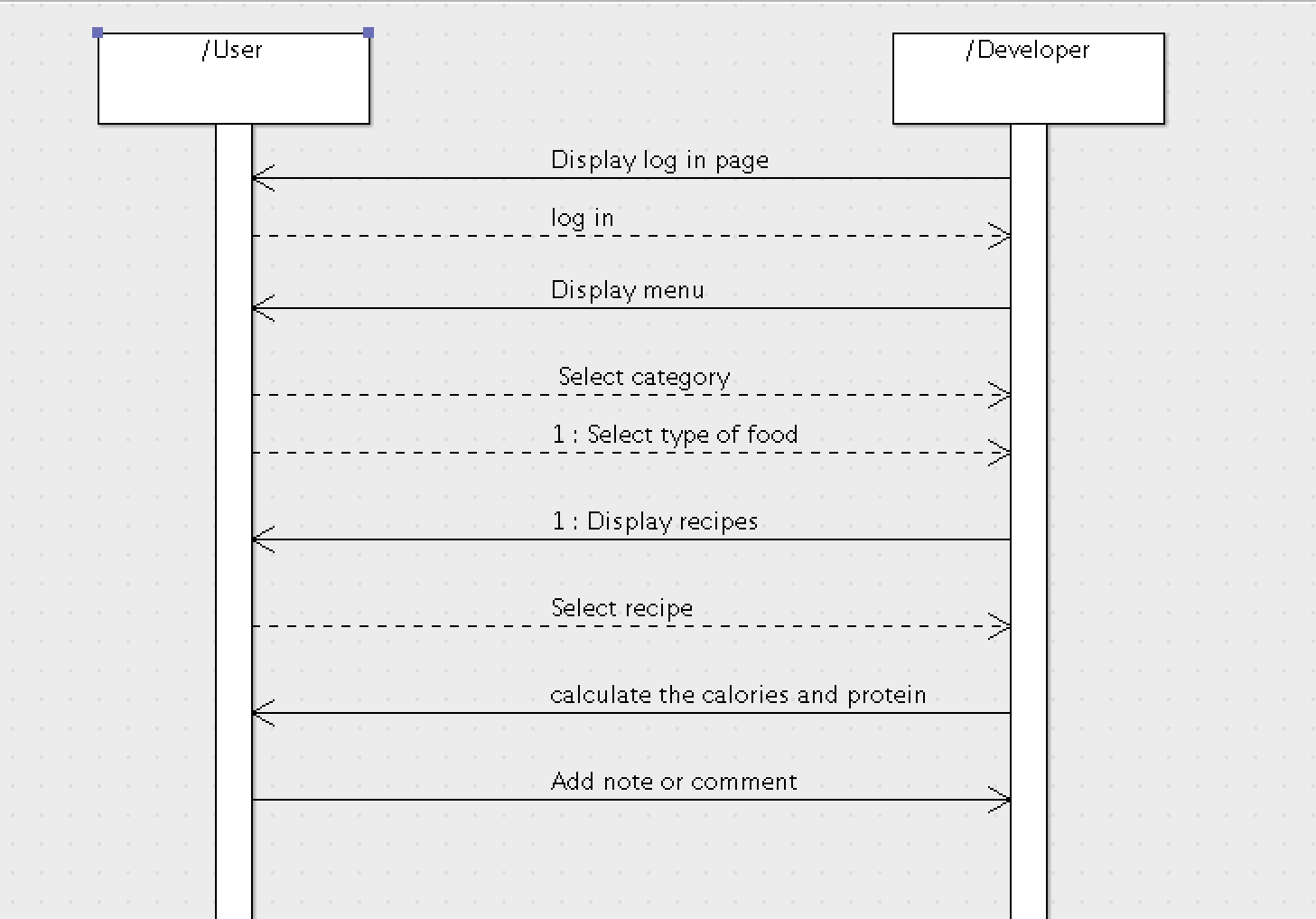
* **Use-case Diagram**



* **Class diagram**



* **Sequence Diagram**



* **State chart diagram**
* **Activity Diagram**

**Chapter 5: Questions**

* **which software development process you used? (waterfall or Agile). why?**

We have chosen Agile Methodology. Because the Agile methodology allows you to make changes, it’s easier to add features that will keep you up to date with the latest developments in your industry.

* **Explain in details your process of development. Provide examples.**

We did a research about applications and programming languages to create an app and we decided to create app for ios by swift programming language. because its most popular.

Also, in this application, we used ARGOUML to make the UML diagrams and we designed the application interface by using [proto.io](http://proto.io/)

There are two ways to test

1. The emergent characteristics of the task are tested by the "system test"

2. To meet customer needs, "Customer Data Test"

* Document all meetings, emails, phone call or other communication tools used for the project. show print screen if possible.
* If you have written a requirement, then you changed it through the development process. Show the initial requirement and the change has been made.
* example of a data processing system is the process of maintaining a check register.