**Requirements Engineering**

**Overview**

This document lists the modules used in the ‘BionicKitchen’ project, as well as their description, function and specifications of each module and device.

**Content**

[**GUI**](#_Toc500258308) 2

[**Read badge**](#_Toc500258309) 3

[**Database**](#_Toc500258310) 3

[**Relate information**](#_Toc500258311) 5

[**Energy button**](#_Toc500258312) 5

[**Web page**](#_Toc500258313) 5

[**Convert spreadsheet to database**](#_Toc500258314) 6

[**Buttons to control program**](#_Toc500258316) 6

[**Export table of DB to spreadsheat**](#_Toc500258317) 7

# **GUI**

**Description:** The GUI (Graphical User Interface) is going to be designed for the chef, therefore he can observe in a monitor which dish request the employee for the current day.

**Acceptance Criteria:**

1. Photo:

* It shall display the photo of employee.
* The size of the photo shall be 300x300 pixels.
* It shall be in format PNG.
* In case of error when try loading photo, it shall show a guest photo.

1. Name:

* It shall display the employee name(first name/last name).
* The name shall respect uppercase and lowercase letters.
* It shall not be longer than 80 characters.

1. Company:

* It shall display the logo of company’s employee.
* The size of the logo shall be 300x72 pixels.
* It shall be in format PNG.
* In case of error when try loading logo, it shall not show a logo.

1. Dish:

* It shall describe the dish and dish number.
* It shall contain the name of the dish shall not be longer than 80 characters and dish number shall not be longer than ‘Guiso <number>’.

1. Interactive background: Depend of dish number, if is VIP or if there is a error.
2. Guiso 1:

* It shall show the color #RRGGBB in a rectangle on screen.
* It shall be in format PNG.

1. Guiso 2:

* It shall show the color #RRGGBB in a rectangle on screen.
* It shall be in format PNG.

1. Guiso 3:

* It shall show the color #RRGGBB in a rectangle on screen.
* It shall be in format PNG.

1. Error or VIP mode:

* It shall show the color #RRGGBB in a rectangle on screen.
* It shall be in format PNG.
* If is error it shall show the text of error in hexadecimal. (Ref. <Error list>)

**Comments (optional):** The GUI is going to be shown in a HD+ monitor. Check the size of the images that are going to been shown. (Ref. <Hardware information>)

# **Read badge**

**Description:** RFID Module reads the employee badge, when it is swipe.

**Acceptance Criteria:**

1. It shall be a external hardware. (Ref. <Hardware information>)
2. The RFID Module shall detect the badge signal
3. The file ‘rfid.py’ shall detect the signal of the RFID Module via USB(Port ttyUSB0). (Ref. Hardware specification)
4. It shall be with radio-frequency identification and a frequency of 125Khz. (Ref. Hardware specification)
5. It shall not need a special configuration.
6. Python script named ‘rfid.py’ shall receive signal and drop noises.
7. Python script named ‘rfid.py’ shall process the signal clean to recollect it to hexadecimal of 10 characters.
8. Python script named ‘rfid.py’ shall send the output to ‘grabInfo.py’.

**Comments (optional):**

N/A

# **Database**

**Description:** The Database shall have three tables that are going to have information of the employees, the companies and other relate information.

**Acceptance Criteria:**

The database shall be named ‘ODTC\_Service’.

1. Table ‘Employees’: it shall contains the following parameters:
2. employee\_id:
   * + It shall be a type ‘int auto increment primary key’.
     + It shall not be null.
3. name:
   * + It shall be a type Varchar (80).
     + It shall contains the employee full name(first name/last name).
     + It shall not be repeat.
     + It shall not be null.
4. badge:
   * + It shall be a type Varchar (10).
     + It shall contains 10 hexadecimal characters of badge.
     + It shall not be repeat.
     + It shall not be null.
5. company:
   * + It shall be a type ‘company\_id’.
     + It shall contains the company number generated in database table Company. (Ref. [company\_id in Company](file:///home/pi/Desktop/BionicKitchen/Documentation/Standardization.xlsx))
     + It shall not be null.
6. photo:
   * + It shall be a type Varchar (13).
     + It shall contain the file name of photo with extension.
     + It shall not be repeat.
     + It shall not be null.
7. is\_active:
   * + It shall be a type Varchar (1).
     + It shall contain a ‘1’ or a ‘0’, the initial value shall be ‘0’. ‘1’ is for active employee and ‘0’ is for inactive employee.
     + It shall not be null.
8. Table ‘Company’, it shall contains the following parameters:
9. company\_id:
   * + It shall be a type ‘int auto increment primary key’.
     + It shall contain the number of company, to relate it to the first table.
     + It shall not be null.
10. name:
    * + It shall be a type Varchar (50).
      + It shall contains the company name.
      + It shall not be repeat.
      + It shall not be null.
11. Table ‘Current\_menu’, it shall contains the following parameters:
12. menu\_id:
    * + It shall be a type ‘int auto increment primary key’.
      + It shall not be null.
13. employee\_id:
    * + It shall be a type ‘employee\_id’.
      + It shall contains the full name generated in database table Employees. (Ref. [employee\_id in Employees](file:///home/pi/Desktop/BionicKitchen/Documentation/Standardization.xlsx))
      + It shall not be null.
14. dish:
    * + It shall be a type Varchar (80).
      + It shall contain the dish that the employee selected.
      + It shall not be null.
15. company\_id:
    * + it shall be a type ‘company\_id’.
      + it shall contains the company number generated in database table Company. (Ref. [company\_id in Company](file:///home/pi/Desktop/BionicKitchen/Documentation/Standardization.xlsx))
      + It shall not be null.
16. date:
    * + It shall be a type date.
      + It shall contain date in format yyyy-mm-ddthh-mm-ss.
      + It shall not be null.
17. served:
    * + It shall be a type Varchar (1).
      + It shall contain if the employee was served or not.
      + It shall not be null.

**Comments (optional):** Check the Database template in the folder documentation of the BionicKitchen project.

# **Relate information**

**Description:** The file ‘grabInfo.py’ locate in ‘BionicKitchen/Resources’ shall connect to the database to match inputs of employee and generated outputs for ‘arrangedInfo.py’.

**Acceptance Criteria:**

1. Badge to Employee: First, know who’s the employee that is swiping his badge. Is going to be match the number of badge with the information on the first table.
2. The badge shall contain a hexadecimal of 10 characters.
3. Employee to Menu: After detecting the employee, know which dish selected for the current day. The information of the table, is going to be match with an Excel document, where is the information of the Menu.
4. Excel document: Only with one sheet, with a column for the names of the employees, and a column for each day of the next two weeks that the kitchen service is going to be offer. (Ref. <Excel template>)

**Comments (optional):**

N/A

# **Energy button**

**Description:** Push button for the power supply of Raspberry.

**Acceptance Criteria:**

1. Python Script shall Turn on and Turn off raspberry, when button is pressed.
2. It shall close all the programs before Raspberry is turn off to avoid damages.
3. It shall automatically run main program (GUI) of the BionicKitchen project, when Raspberry is turn on.
4. Type of connection of the button shall be Pull Down.

**Comments (optional):** For more information, check the Hardware specifications in the folder documentation of the BionicKitchen project.

# **Web page**

**Description:** Web page to upload excel file to the Raspberry.

**Acceptance Criteria:**

1. The web page will be mounted on raspberry, and can be accessed from another computer on the local network.
2. The name of the web page shall be ‘Sonora Offshore’s Service’.
3. Between of name shall show the logo of ‘Obregon Technology Development Center’.
4. Written in HTML5 for text.
5. Written in CSS3 for design.

* The page shall have a similar program design to ‘BionicKitchen’.

1. Written in PHP7 to upload the file. The script ‘upload.php’ to upload spreadsheet of the menu, with follow verifications:

* It shall save in ‘uploads’.
* The spreadsheet only can be upload each 14 days, if try upload a file early or after time this refused the file and show a warning in the web.
* It shall only accepts XLSX format, if try upload another type of file this refused the file and show a warning in the web.
* It shall only accepts spreadsheet with this name: “Encuesta\_de\_comedor”, if try upload a file with another file name this refused the file and show a warning in the web.
* It shall not be greater than 20MB.

1. Written in JavaScript for AJAX.
2. Written in Python2.7 for execute validations.
3. Only accepts XLSX format.
4. Only can upload one document each two weeks.

**Comments (optional):**

N/A

# **Convert spreadsheet to database**

**Description:** The script ‘convertProcess.py’ to enter values from excel document ‘Encuesta\_de\_Comedor.xlsx’ into the database ‘OTDC\_Service’ in table ‘Current\_menu’.

**Acceptance Criteria:**

1. Excel document ‘Encuesta\_de\_Comedor.xlsx’ in ‘uploads’.
2. The script iterate in undefined columns and 11 rows of ‘Encuesta\_de\_Comedor.xlsx’, to extract name of employee, dish number and dish name according to rows of date. (Ref. [EExcel template](file:///home/pi/Desktop/BionicKitchen/Documentation/Template_Encuesta_de_Comedor.xlsx))
3. Information selected will get into a tuple. The turple shall contain ‘employee\_id, dish, company\_id, date, served’. (Ref. [Database](#_Database))
4. Connect to ‘OTDC\_Service’ in table ‘Current\_menu’ and add tuple with information. (Ref. [Database](#_Database))

**Comments (optional):**

N/A

# **Service buttons**

**Description:** The buttons are for the user, needs three push buttons of plastic for good handle of control the GUI: Served button, Cancel button & Close program Button. (Ref. <Hardware documentation>)

**Acceptance Criteria:**

1. Served button:
   * Shall indicate that the employee was served.
   * Shall permit the RFID Module to read another badge.
   * Shall update the served status in the database. (Ref. [Database](#_Database))
2. Canceled button:
   * Shall indicate that the employee wasn’t served.
   * Shall permit the RFID Module to read another badge.
3. Close program button:
   * Shall close the main program (GUI).
   * Shall save information, for avoid loss of information.

**Comments (optional):** The program shall wait an answer from the served o canceled buttons, either doesn´t continue reading badges. For more information, check the Hardware specifications in the folder documentation of the BionicKitchen project.

# **Export table of DB to spreadsheat**

**Description:** The actions ejecuted by the badge and buttons are reflected on ‘OTDC\_Service’ in table ‘Current\_menu’, therefore these results are exported to a Excel file named ‘Current\_menu.xlsx’, where a user (Administrator) can check the logistic of the BionicKitchen.

**Acceptance Criteria:**

The spreadsheet ‘Current\_menu.xlsx’ shall export daily and have the following columns:

1. Employee name, it shall be only full name.
2. Employee company, it shall be only name of company.
3. Dish, it shall be the id of dish and name of dish in parenthesis.
4. Date, it shall only contain date in format yyyy-mm-ddthh-mm-ss.
5. Status, it shall only contain a ‘1’ for served and ‘0’ for not served.

**Comments (optional):**

(Ref. Template exportExcel)