**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**

[Content 2](#_Toc500839507)

[1. GUI 3](#_Toc500839508)

[2. Read badge 4](#_Toc500839509)

[3. Database 4](#_Toc500839510)

[4. Relate information 6](#_Toc500839511)

[5. Energy button 6](#_Toc500839512)

[6. Web page to upload menu 6](#_Toc500839513)

[7. Web page to manage employees 7](#_Toc500839514)

[8 Convert spreadsheet to database 9](#_Toc500839515)

[9. Service buttons 9](#_Toc500839516)

[10. Export table of DB to spreadsheet 9](#_Toc500839517)

# **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:
     1. It shall display the photo of employee.
     2. The size of the photo shall be 300x300 pixels.
     3. The name shall not be longer than 9 characters.
     4. The name shall be five numbers, hyphen, one number, hyphen, one number (without spaces). Eg ‘12345-6-7’
     5. It shall be in format PNG.
     6. In case of error when try loading photo, it shall show a guest photo.
     7. It shall be in the folder ‘Photos’ of the BionicKitchen project.
  2. Name:
     1. It shall display the employee name (first name/last name).
     2. The name shall respect uppercase and lowercase letters.
     3. It shall not be longer than 80 characters.
  3. Company:
     1. It shall display the logo of company’s employee.
     2. The size of the logo shall be 300x72 pixels.
     3. The name should be the company acronym in uppercase.
     4. It shall be in format PNG.
     5. It shall be stored in the folder ‘Logos’ of the BionicKitchen project.
     6. In case of error when try loading logo, it shall not show a logo.
  4. Dish:
     1. It shall contain the dish name and dish number.
     2. Dish name shall not be longer than 80 characters.
     3. Dish number shall not be longer than 6 characters. Eg. ‘Guiso#’.
  5. Interactive background: Depend of dish number, if it’s VIP or if there is an error.
     1. Guiso 1:
        1. It shall show the color: #444141 in a rectangle on screen.
        2. Shall named ‘Interactive\_background\_1’
        3. It shall be in format PNG.
        4. It shall be in the folder ‘Texture’ of the BionicKitchen project.
     2. Guiso 2:
        1. It shall show the color #931A21in a rectangle on screen.
        2. Shall named ‘Interactive\_background\_2’
        3. It shall be in format PNG.
        4. It shall be in the folder ‘Texture’ of the BionicKitchen project.
     3. Guiso 3:
        1. It shall show the color: #2B2A2A in a rectangle on screen.
        2. Shall named ‘Interactive\_background\_3’
        3. It shall be in format PNG.
        4. It shall be in the folder ‘Texture’ of the BionicKitchen project.
  6. Error or VIP mode:
     1. It shall show the color: #501B29 in a rectangle on screen.
     2. Shall named ‘Interactive\_background\_4’
     3. It shall be in format PNG.
     4. It shall be in the folder ‘Texture’ of the BionicKitchen project.
     5. 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 an external hardware. (Ref. <Hardware information>)
  2. The RFID Module shall detect the RFID tags of 125 kHz (Employee Badge).
  3. The file ‘rfid.py’ shall receive the Tag number of RFID Module via USB(Port ttyUSB0). (Ref. Hardware specification)
  4. The RFID module shall be a RFID technology of low frequency of 125 KHz. (Ref. Hardware specification)
  5. Python script named ‘rfid.py’ shall process signal of the tag and drop noises.
  6. The output of the python script mentioned shall be an hexadecimal of 10 characters (Shall be the tag signal after it processed).
  7. 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:**

* 1. The database shall be named ‘ODTC\_Service’.
     1. Table ‘Employees’: it shall contain the following parameters:
        1. employee\_id:

1. It shall be a type ‘int auto increment primary key’.
2. It shall not be null.
   * + 1. name:
3. It shall be a type Varchar (80).
4. It shall contains the employee full name(first name/last name).
5. It shall not be repeatable.
6. It shall not be null.
   * + 1. badge:
7. It shall be a type Varchar (10).
8. It shall contain 10 hexadecimal characters of badge.
9. It shall not be repeatable.
10. It shall not be null.
    * + 1. company:
11. It shall be a type ‘company\_id’.
12. It shall contain the company number generated in database table Company. (Ref. [company\_id in Company](file:///home/pi/Desktop/BionicKitchen/Documentation/Standardization.xlsx))
13. It shall not be null.
    * + 1. photo:
14. It shall be a type Varchar (13).
15. It shall contain the file name of photo with extension.
16. It shall not be repeatable.
17. It shall not be null.
    * + 1. is\_active:
18. It shall be a type Varchar (1).
19. 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.
20. It shall not be null.
    * 1. Table ‘Company’, it shall contains the following parameters:
         1. company\_id:
21. It shall be a type ‘int auto increment primary key’.
22. It shall contain the number of company, to relate it to the first table.
23. It shall not be null.
    * + 1. name:
24. It shall be a type Varchar (50).
25. It shall contains the company name.
26. It shall not be repeatable.
27. It shall not be null.
    * 1. Table ‘Current\_menu’, it shall contains the following parameters:
         1. menu\_id:
28. It shall be a type ‘int auto increment primary key’.
29. It shall not be null.
    * + 1. employee\_id:
30. It shall be a type ‘employee\_id’.
31. It shall contains the full name generated in database table Employees. (Ref. [employee\_id in Employees](file:///home/pi/Desktop/BionicKitchen/Documentation/Standardization.xlsx))
32. It shall not be null.
    * + 1. dish:
33. It shall be a type Varchar (80).
34. It shall contain the dish that the employee selected.
35. It shall not be null.
    * + 1. company\_id:
36. it shall be a type ‘company\_id’.
37. it shall contains the company number generated in database table Company. (Ref. [company\_id in Company](file:///home/pi/Desktop/BionicKitchen/Documentation/Standardization.xlsx))
38. It shall not be null.
    * + 1. date:
39. It shall be a type date.
40. It shall contain date in format yyyy-mm-ddthh-mm-ss.
41. It shall not be null.
    * + 1. served:
42. It shall be a type Varchar (1).
43. It shall contain if the employee was served or not.
44. It shall not be null.

**Comments (optional):** Check the DB Template Database templatein 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 the employee that is swiping his badge is. It shall match the number of badge with the information on the table “Employees” of the DB project.
     1. The badge data shall contain a hexadecimal of 10 characters.
  2. Employee to Menu: After detecting the employee, know which dish selected for the current day. The information of the table shall match with Encuesta\_de\_Comedor.xlsx (in the foler “uploads”), where is the information of the Menu.
     1. Encuesta\_de\_Comedor.: 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.
     1. It shall close all the programs before Raspberry is turn off to avoid damages in the files.
     2. It shall automatically run main program (GUI) of the BionicKitchen project, when Raspberry is turn on.
  2. 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 to upload menu**

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

**Acceptance Criteria:**

* 1. The web page shall be mounted on raspberry
  2. Shall be accessed of another computer on the local network.
  3. The name of the web page shall be ‘BK Service’.
  4. Between of name shall show the logo of ‘Obregon Technology Development Center’.
  5. Shall be written in HTML5 for text.
  6. Shall be written in CSS3 for design.

1. The page shall have a similar program design to ‘BionicKitchen’ GUI.
   1. Shall be written in PHP7 to upload the file. The script ‘upload.php’ to upload spreadsheet of the menu, with follow verifications:
2. It shall save in folder ‘uploads’.
3. The spreadsheet shall 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.
4. It shall only accepts XLSX format, if try upload another type of file this refused the file and show a warning in the web.
5. 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.
6. It shall not be greater than 20MB.
   1. Shall be written in JavaScript for AJAX.
   2. Shall be written in Python2.7 for execute validations.
   3. Shall only accepts XLSX format.
   4. Shall only can upload one document each two weeks.

**Comments (optional):**

N/A

# **Web page to manage employees**

**Description:** Web page to manage the employees list in the Database.

**Acceptance Criteria:**

* 1. The web page shall be mounted on raspberry
  2. Shall be accessed of another computer on the local network.
  3. The name of the web page shall be ‘BK Management’.
  4. Between of name shall show the logo of ‘Obregón Technology Development Center’.
  5. Shall be written in HTML5 for text.
  6. Shall be written in CSS3 for design.

1. The page shall be a modern design.
   1. Shall be written in PHP7
      1. It shall connecting to database.
      2. It shall update information of employee of the database.
      3. It shall eliminate employee of the database.
      4. It shall register employee of the database.
      5. It shall search employee of the database.
      6. It shall show information in page of employee.
      7. It shall verification all fields of placeholders. (Ref. Dot 10 of this section)
   2. Shall be written in JavaScript for AJAX.
   3. Shall be written in python for execute validations
   4. Shall be able to register employees to the DB. To register an employee shall request this information:
      1. Full name:
2. Shall be written correctly
3. Shall respect upper case and lower case
   * 1. Company:
4. Shall be written correctly
5. Shall respect upper case and lower case
6. Shall verify that the company is in the DB, either shall send warning.
   * 1. Badge (Employee badge):
7. Shall not be repeated, either shall send warning.
8. Shall be provided by the RFID reader
   * 1. Photo:
9. The photo shall be grab of database.
   1. Shall be able to modify employee information of DB. To modify an employee shall search employee and change the necessary information:
      1. Full name:
10. Shall be written correctly
11. Shall respect upper case and lower case
    * 1. Company:
12. Shall be written correctly
13. Shall respect upper case and lower case
14. Shall verify that the company is in the DB, either shall send warning.
    * 1. Badge (Employee badge):
15. Shall not be repeated, either shall send warning.
16. Shall be provided by the RFID reader
    * 1. Photo:
17. The photo shall be grab of the database.
    1. Shall be able to eliminate employee of DB
    2. Employee need to be searched (Ref. Dot 13 of this section)
    3. If employee information is correct, a button shall eliminate the employee of DB
    4. Shall send a warning confirmation before employee is deleted.
    5. Shall be able to search employees of the DB
       1. Search shall be by the employees name.
       2. If search is successful, shall display employees information.
       3. If search is failed, shall display that the employee isn’t in the DB.
    6. Shall send a warning to confirm that the information is correct, before is uploaded.

**Comments (optional):**

N/A

# **Convert spreadsheet to database**

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

**Acceptance Criteria:**

* 1. Excel document ‘Encuesta\_de\_Comedor.xlsx’ shall be in the folder ‘uploads’.
  2. The script shall iterates 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 shall get into a tuple. The tuple shall contain ‘employee\_id, dish, company\_id, date, served’. (Ref. [Database](#_Database))
  4. Shall 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:
  2. Shall indicate that the employee was served.
  3. Shall permit the RFID Module to read another badge.
  4. Shall update the served status in the database. (Ref. [Database](#_Database))
  5. Shall be color green
  6. Canceled button:
  7. Shall indicate that the employee wasn’t served.
  8. Shall permit the RFID Module to read another badge.
  9. Shall be color yellow (TBD)
  10. Close program button:
  11. Shall close the main program (GUI).
  12. Shall save information, for avoid loss of information.
  13. Shall be color red (TBD)

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

# **Export table of DB to spreadsheet**

**Description:** The actions executed 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)