

Article

RFID Based Embedded System for Sustainable Food Management in an IoT Network Paradigm

Raúl Parada ^{1,*}, Alfonso Palazón ¹, Carlos Monzo ¹ and Joan Melià-Seguí ²

¹ Estudis de Informàtica, Multimèdia i Telecomunicació Universitat Oberta de Catalunya, Rambla del Poblenou, 156, 08018 Barcelona, Spain

² Internet Interdisciplinary Institute, Universitat Oberta de Catalunya, Avinguda Carl Friedrich Gauss, 5, 08060 Castelldefels, Barcelona, Spain

* Correspondence: rparada@uoc.edu

Received: 31 July 2019; Accepted: 27 August 2019; Published: 1 September 2019



Abstract: A third of the food produced in the world ends up in the rubbish, enough to put an end to world hunger. On the other hand, society is increasingly concerned to bring healthy eating habits. A RFID (radio frequency identification) food management system is designed to palliate the previously described issues in an Internet of Things (IoT) network paradigm. It consists of RFID readers placed on a user's kitchen furniture, which automatically reads food information. There is no need for direct sight between reader and tag, as it occurs through the barcode technology. As a complement, a multi-platform web application is developed, allowing its users to check the date of food expiration and other detailed information. The application notifies the user when a product is about to expire. It also offers recipes that might be prepared with available foods, thus preventing them from being wasted. The recipes are accompanied by their nutritional information, so that the user can exhaustively monitor what he/she eats. This embedded system may provide economic benefits to the manufacturer, since it allows supermarkets to pay for displaying their products advertised through the application. After system deployment, design conclusions are shown, and future improvement points are indicated.

Keywords: internet of things; radio frequency identification; food management; recommendation systems; real-time embedded systems

1. Introduction

One third of the food produced in the world yearly (approximately 1.3 billion tons) ends up in the rubbish [1]. This means that one-third of economic and human resources used in its production are wasted. Moreover, an unnecessary increment of carbon footprint takes place [1]. Besides the waste of food and carbon footprint, another problem related to eating habits is that of health diseases with 39% of adults being overweight, 13% of them being obese, according to data from 2014 [2].

Despite the problem being globally measured, food-related behavior changes may be applied in house environments to obtain a worldwide effect. The kitchen where food is wasted, mainly in the kitchen furniture (fridge, cabinets, and shelves). Consumers are used to making unplanned purchases. They also have no control of available products, its expiration date, nor a list of recipes they can prepare with them. This, added to the purchase of rare ingredients, which are barely used in a single recipe [3], create a loss of money and food for customers. Specifically, customers suffer an average loss of 850 USD/year [3].

Internet of Things (IoT) architectures for smart cities require technologies to sense the environment by connecting intelligent devices to build a sustainable smart society. Within all technologies included in the IoT paradigm, radio frequency identification (RFID) is the most popular technology in retail

scenarios. Specifically, the Electronic Product Code (EPC) [4] is the de facto standard for item-level identification in retail, to monitor stock and detect user-object interactions or hand gestures. Moreover, the information stream provided by RFID technology is a useful source of information to implement intelligent systems, where the electronic labels transform the store products into smart objects, providing rich information such as their type of ingredient, calories, or expiration date.

This paper presents a food management system, RFID Food Management (RFM) to monitor all products available at home, recommending suitable recipes to prepare with them in order to reduce the waste of food in our daily life, which is our overall goal. Furthermore, our real-time embedded system enhances the supply-chain management by offering product promotions to complete recipe recommendations. We also aim to control the amount of consumed calories, so that it can lead to a healthy lifestyle. Specifically, the following objectives have been set out:

- A comparison of real-time recipe recommendation algorithms based on the availability of ingredients in the kitchen;
- A real-time embedded system that monitors the available ingredients, and provides its information for further processing;
- A product promotion system to enhance the supply chain management.

The remainder of this paper is organized as follows: Section 2 introduces the problem motivation and related state of the art. The RFID Food Management system is described in Section 3. Section 4 introduces our approaches of smart recommendation algorithms, which are tested in Section 5. Finally, we conclude the paper in Section 6.

2. Related Work

To date, several authors have developed different approaches to monitor fridge content. Samsung [5] presents a smartfridge equipped with cameras located in its interior. These cameras take photos from the fridge content each time the door is opened, which are sent to our smartphone through a proprietary app afterwards. Furthermore, the smartfridge has a touch screen on the outside that allows the user to buy products, write notes, watch tv, and listen to music. However, photos could not put hidden products into the inventory and information, such as expiration or issued date, can not be provided. LG [6] launched its smartfridge at the end of 2012. This smartfridge is based on a traditional fridge and is equipped with a touch screen and several apps. These apps allow one to manually perform inventories by adding products to a list every time one of them is placed on the fridge, or remove them when they are consumed. These applications also give the option to manually enter the expiration date of products in order to receive notifications on our smartphone when they are about to expire. Other apps provide recipes and the historical consumption of products. Although this smartfridge provides an updated inventory and product information, it requires a manual inventory. Luo et al. [7] introduced a smartfridge prototype where applications focus on providing products nutritional and dietetic information, controlling the user's food-related behavior. Their approach is equipped with a touch screen where one can manually introduce products and their expiration date, so that a warning is issued when products are close to their expiration date. It is possible to search in a database the previously inserted products nutritional information and recipes that might be prepared with them. In addition, each user can create a personal profile with his medical information (i.e., if the user suffers from diabetes or cholesterol), as well as his weight and height, so that he is able to control his body mass. Although Luo et al.'s approach introduced the medical monitorization function, their prototype requires a manual introduction of products and the user has to check the products information on the front of the fridge.

Rouillard [8] presents a smartfridge prototype, Pervasive Fridge, based on a mobile application. It is capable of registering the entrance/exit of products to/from the fridge through barcode and RFID readings, or voice recognition. Once the products have been registered, the user is notified when they are about to expire. Furthermore, the user can check the products nutritional information, recipes,

and dietetic recommendations. This prototype is compatible with all fridges, since the system is based on a mobile application. However, the smartphone requires barcode and RFID reading capabilities. Furthermore, the inventory procedure is performed by reading one product at a time. Hanshen and Dong [9] introduced an RFID-equipped fridge. Their approach provides services to keep control of the user's food-related behavior. For instance, if the user suffers any disease, the fridge interacts with the hospital to gather medical information and recommends the right products. Besides providing product recommendation, the real-time embedded system recommends recipes to lead a healthy life. In addition, their approach creates a list of products to buy based on historical information. As a disadvantage, every user needs a server placed in their house in order to store product information that must be kept updated by the manufacturer. Thus, this approach is poorly scalable because of the need of server pre-configuration, which lacks autonomy.

Our work aims to monitor not only those products located inside a fridge but also outside. Most of the previously mentioned approaches do not take into account those products located out of the fridge, even though they are an important part of users' diet and recipes. Our intelligent recommendation approach, RFID Food Management (RFM) is based on real-time RFID readings, where information is located in a centralized server to ease scalability. Our prototype displays product information, inventory, recipes, and supermarket offers through a multi-platform application, which is compatible with any common device. Table 1 shows a comparative capabilities between the different approaches previously described, and our RFM embedded system.

Table 1. The radio frequency identification (RFID) Food Management provides more advantages than other authors' approaches.

Approach	Real-Time Inv.?	Nutri. Inf.?	Recipe Recom.?	Central. Arq.?	Multi-Platform?	Cabinets & Shelves?
Samsung [5]	No	No	No	Yes	Yes	No
LG [6]	No	Yes	Yes	Yes	Yes	No
Luo et al. [7]	No	Yes	No	No	No	No
Rouillard [8]	No	Yes	Yes	Yes	Yes	No
Hanshen and Doug [9]	Yes	Yes	Yes	No	Yes	No
RFID Food Management	Yes	Yes	Yes	Yes	Yes	Yes

3. RFID Food Management System

This section aims to describe all the elements RFID Food Management system is composed of. Figure 1 shows the system scheme with the connection between hardware elements and the software tools setup.

As a brief introduction, the main components are described below:

1. **RFID System:** Is based on hardware elements, allowing the reading of RFID-labeled products to obtain the required item identification;
2. **Database:** Stores product, recipe, offer, and user information;
3. **Server:** Hosts the RFM APP and processes user requests;
4. **RFM App:** Is a software implementation that provides real-time information through a friendly-user interface.

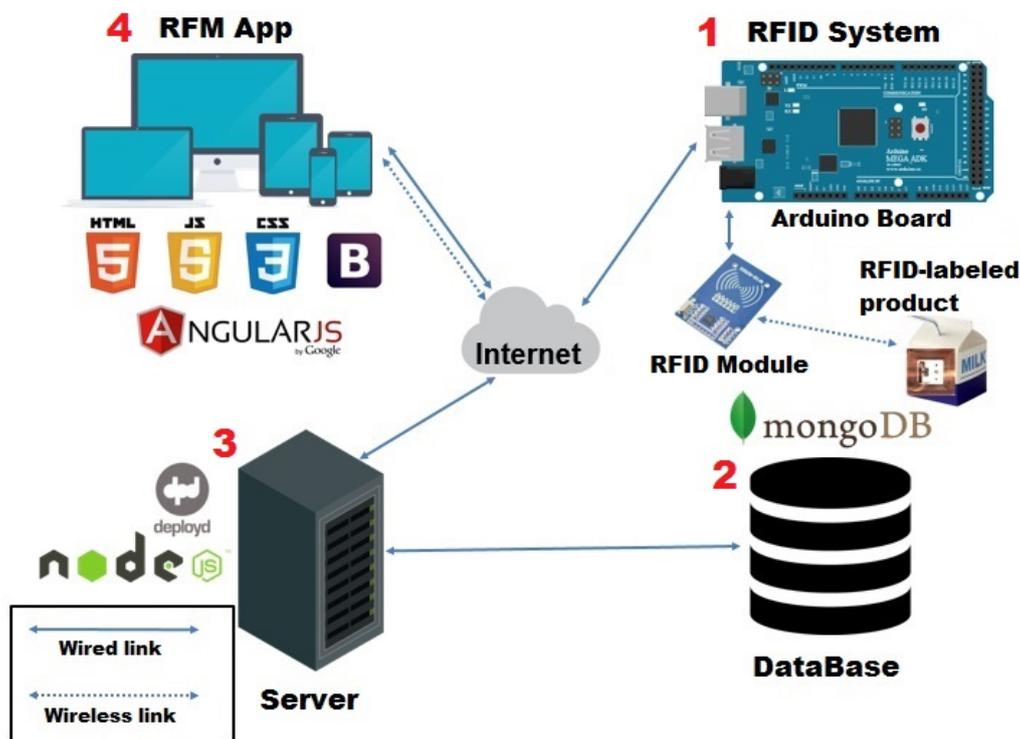


Figure 1. RFM is composed of diverse hardware and software elements.

3.1. RFID System

In a HF EPC [4] RFID communication, RFID antennas are interrogating in a time-multiplexed procedure to RFID passive tags. Those RFID passive tags within the reading range, backscatter the signal back to the RFID reader. The RFID reader can not only inventory RFID tags within its read range, but also high and low-level indicators, which are included in the backscattered signal. High-level indicators, such as the identification code (EPC), timestamp, antenna port, and reader identifier, can also be obtained. One of the many RFID advantages in respect to other technologies is the unique identification of labeled-objects. This means each item is identified in a unique manner. For instance, we can differentiate between two cartons of milk from the same company, which have a different expiration date.

- High-level indicators:
 - Identification code;
 - Timestamp;
 - Antenna port;
 - Reader identifier.
- Low-level indicators:
 - Received signal strength indicator (RSSI);
 - Radio frequency phase (PHASE).

The proposed RFID system is composed of a single-board microcontroller which uses an Arduino Mega [10] due to its easy module connection (i.e., Ethernet and RFID modules). Furthermore, the Arduino Mega provides more memory than other models, which allows the development of more complex programs. The RFID module [11] reads all passive RFID tag labeled-objects [12]. Figure 1 shows how the RFID module is connected to the Arduino board through wired links. Figure 2 represents the

connection between the Arduino board and the RFID Module. Each product is labeled with a passive RFID tag containing an unique identifier, linking the product with its information stored in the database (c.f. Section 3.2). Thus, the RFID Module reads the product passive RFID tag, sending this information back to the Arduino board. The Arduino board sends the information to a server, using an Ethernet module [13] via the SPI (Serial Peripheral Interface) protocol. This information is instantly accessible from the RFM app, as shown in Figure 1.

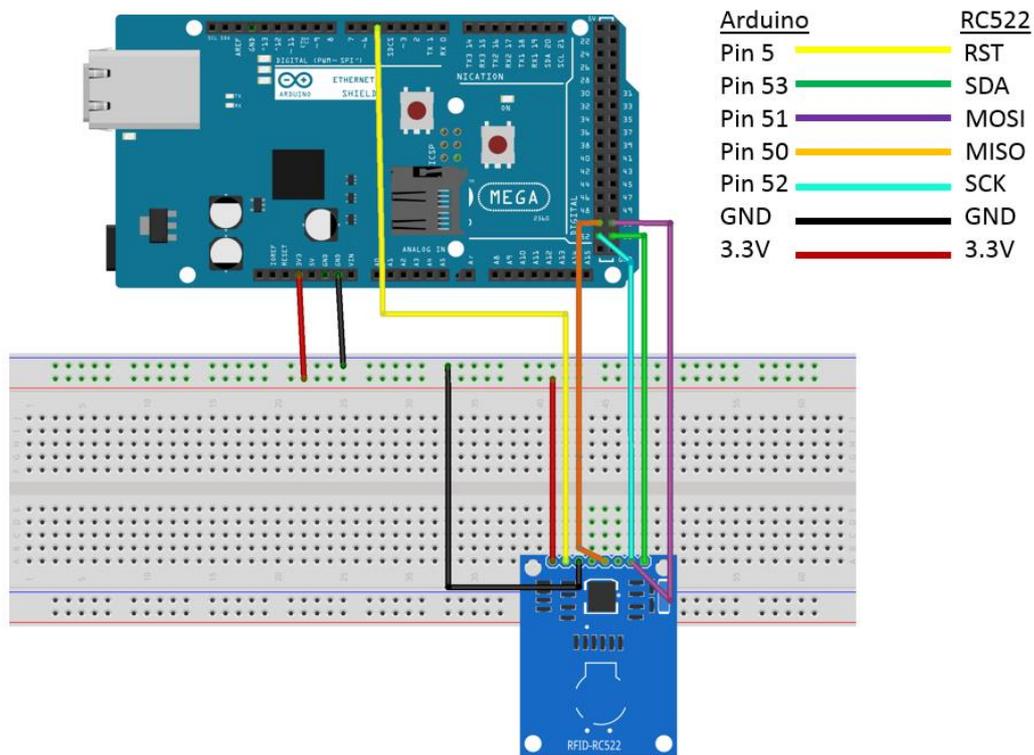


Figure 2. Connection between the RFID module and the Arduino board.

Passive RFID tags are responsible for linking a unique identifier with a particular product. The MIFARE Classic 1K RFID passive tags provide an EEPROM memory of 1KB, organized in 16 sectors with 4 blocks of 16 bytes each. Each sector reserves four blocks, three of them are used to store data and the other one, called a trailer, to keep the secret keys A and B that is required to read the written information. Hence, a total of 768 bytes (16 sectors*3 blocks*16 bytes) are free to save product information. Table 2 tabulates the bytes, per block and sector, used to locate the product information. Although 768 bytes are available to insert product information, only 63 bytes are used in this real-time embedded system version implementation, leaving the rest of bytes for future implementations. Note that the writing procedure is performed manually. However, in real implementation, the next generation of products with RFID tags will already contain this information from the manufacturer [14].

Table 2. Bytes used to locate product information.

Sector	Block	Bytes	Information
15	0–2	0–15	Product name
14	2	0–4	Product identifier
14	2	5–6	Group product belongs to
14	2	7–10	Energy in 100 g (kcal)
14	2	11–15	Quantity of product
14	1	0–2	Fat in 100 g (grams)
14	1	3–5	Carbohydrates in 100 g (grams)
14	1	6–8	Proteins in 100 g (grams)
14	1	9	Contains gluten.
14	1	10	Contains egg
14	1	11	Contains milk
14	1	12	Contains soy
14	1	13	Contains peanuts
14	1	14	Contains nuts
14	1	15	Contains fish
14	0	0	Contains shellfish
14	0	1	Contains celery
14	0	2	Contains sesame
14	0	3	Contains mustard
14	0	4	Is it fresh?
14	0	5–6	Expiration/issue day
14	0	7–8	Expiration/issue month
14	0	9–12	Expiration/issue year
14	0	13–14	Units

3.2. Database

The RFID Food Management system stores rich information related to common ingredients, such as energy (kcal), proteins (g), carbohydrates (g), and fat (g), contained in 100 g. This nutritional information has been directly extracted from an international food composition database [15]. RFM app users request data through an API REST (Representational State Transfer), which is a web service that provides interoperability among computer systems on the Internet. In order to build a simple REST service, we used the *deployd* [16] tool. This open source tool allows one to design, build, and scale REST services for web applications in minutes. It provides a user model with methods to open and close sessions, and a control panel to configure a database. The database implemented in *deployd* is the open source tool *MongoDB* [17], based on the NoSQL paradigm, its architecture relies on collections and documents. One advantage of using a non relational database, is that it optimizes queries that involve huge amounts of registers. Table 3 shows the four collections of data created in the database.

Table 3 presents the list of parameters from the generated collections: Product, Recipe, User, and Offer. Note that each item created from each collection will be identified uniquely.

Table 3. The four collections of non relational data created in MongoDB.

Product			Recipe			User		
Name	Type	Description	Name	Type	Description	Name	Type	Description
id	str.	Unique MongoDB id.	id	str.	Unique MongoDB id.	Id	str.	Unique MongoDB id.
rfm	str.	RFM identifier	name	str.	Recipe name	username	str.	Username to access the app.
antenna	str.	Antenna identifier	type	str.	Recipe type	password	str.	Password to access the app.
name	str.	Recipe name	instructions	str.	Preparation instructions	firstName	str.	User name
UID	str.	Unique tag id.	photo	str.	Image of the recipe	secondName	str.	User surname
foodId	str.	Food identifier	cGluten	bool.	Contains gluten	rfm	str.	RFM identifier
group	str.	Group product belongs to	cEgg	bool.	Contains eggs	cGluten	bool.	Allergic to gluten
dateOfExpiry	str.	Expiration date	cMilk	bool.	Contains milk	cEgg	bool.	Allergic to eggs
fresh	bool.	Is the product fresh?	cSoy	bool.	Contains soy	cMilk	bool.	Allergic to milk
quantity	str.	Quantity	cPeanuts	bool.	Contains peanuts	cSoy	bool.	Allergic to soy
unit	str.	Units	cNut	bool.	Contains nuts	cPeanuts	bool.	Allergic to peanuts
cGluten	bool.	Contains gluten	cFish	bool.	Contains fish	cNut	bool.	Allergic to nuts
cEgg	bool.	Contains eggs	cShellFish	bool.	Contains shellfish	cFish	bool.	Allergic to fish
cMilk	bool.	Contains milk	cCelery	bool.	Contains celery	cShellFish	bool.	Allergic to shellfish
cSoy	bool.	Contains soy	cSesame	bool.	Contains sesame	cCelery	bool.	Allergic to celery
cPeanuts	bool.	Contains peanuts	cMustard	bool.	Contains mustard	cSesame	bool.	Allergic to sesame
cNut	bool.	Contains nuts	energy	int.	Kcals per portion	cMustard	bool.	Allergic to mustard
cFish	bool.	Contains fish	fat	int.	Fat per portion	Offer		
cShellFish	bool.	Contains shellfish	carbo	int.	Carbohydrates per portion	Name	Type	Description
cCelery	bool.	Contains celery	protein	int.	Proteins per portion	id	str.	Unique MongoDB id.
cSesame	bool.	Contains sesame	people	int.	Number of portions	foodId	str.	Food identifier
cMustard	bool.	Contains mustard	ingredients	array	Array of ingredients.	url	str.	Link to offer
energy	int.	Kcals in 100 g of product				text	str.	Text describing offer
fat	int.	Fat in 100 g of product				price	int.	Price of the offer
carbo	int.	Carbohydrates in 100 g				unit	str.	Unit of the product
protein	int.	Proteins in 100 g of product						

3.3. Server

The server is the element in charge of processing user requests to the database, returning back the requested information. These input and output operations are written in Javascript using the open-source cross-platform NodeJS [18]. The communication between the Arduino board and the server, works as follows: Every time the fridge or pantry is closed, the Arduino board reads all the products and sends their information to the server through a POST order. The server processes the received data, if that data does not contain a previously available product, the product is deleted, the non-previously existing products are added to the database, and the rest remain the same.

3.4. RFM App

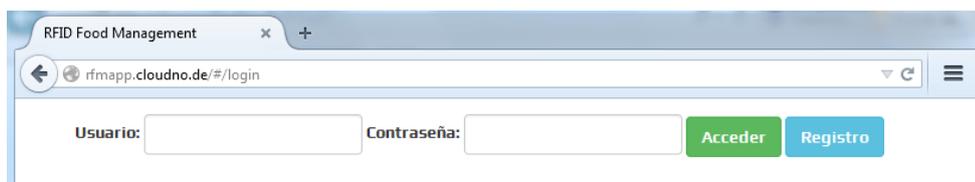
The RFM app front-end has been implemented using HTML5, the latest version of the web page and web application developing standard de facto. It has a user-friendly interface. Our approach relies on single page application (SPA) implementation, in which the information is dynamically loaded, depending on the user needs. Conversely, in the multi page web applications (MPA), the server sends all the page information every time the user requests something. Thus, the single page application offers a faster page loading, better user-experience, no need to write server side code to render the web page, and splits both data representation and processing [19]. In addition, other frameworks and libraries have been used to ease the developing task. Thus, the AngularJS framework [20] allows one to write web applications from the client side in a structured manner, easing the communication with the server. This framework follows a software design pattern known as Model View Controller (MVC). This pattern allows one to separate user interface from the application's logical part. Lastly, in order to provide a more visually attractive application that might adapt to any device (a.k.a responsive design), we used Bootstrap [21], an open source framework.

Next, the main RFM app interfaces are displayed, describing their functions. Figure 3 presents the RFM app login and registration pages. Figure 3a shows the main page. If the user has already been registered, he only needs to login with his credentials. Otherwise, he has to be registered by clicking on the blue button. Figure 3b presents the registration form, which a new user is required to fill up, in order to obtain his personal information, as well as the allergies he might have. This information will be used to recommend recipes later. Thereby, the user is required to insert a username (1) and password (2) as credentials to access the application. His name (3) and surname (4) are also required. Since the user owns the kitchen furniture, we insert a RFM identifier (5) to link the kitchen furniture content to the owner(s). Since the percentage of children with food allergies has increased by 50% from 1997 to 2011, and this number is still increasing [22], we introduced allergens information in the registration form. The user with food allergies can activate the checkbox (6), and multiple allergens will be shown (7). Otherwise, the system will assume the user has no food allergies. Finally, the user finishes the registration process and creates his account by pressing the blue button (8).

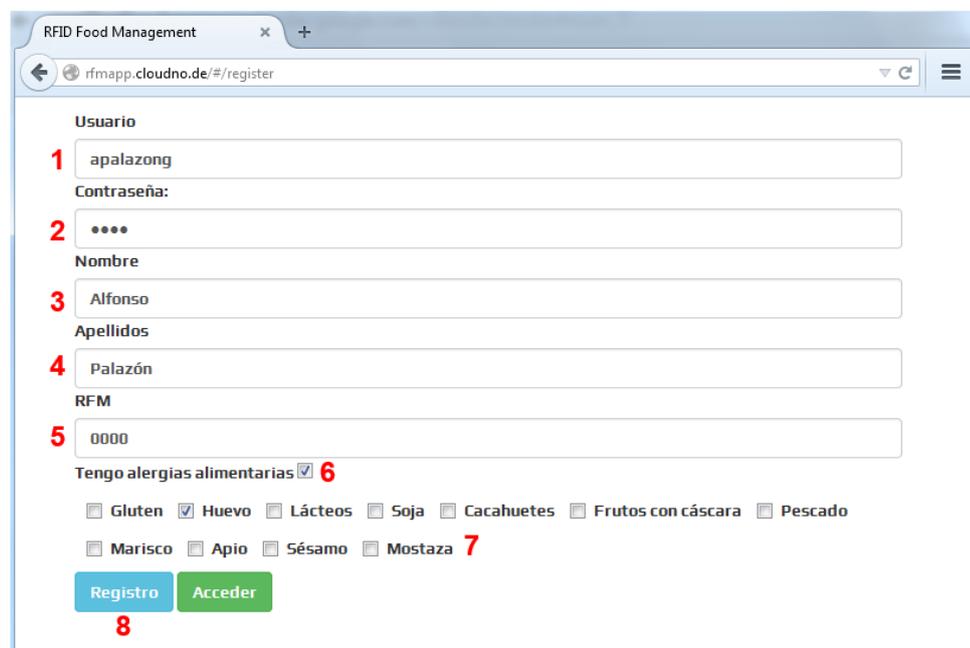
Once the user logs into the application, he has many available options. Figure 4 shows two RFM app images. Figure 4a shows the recipes menu, with a set of filters to get recipe recommendations. All the functions are accessible from the left side bar. The different functions are: A list of the available products in the kitchen furniture (1), list of all available recipes (2), product offers at nearby supermarkets (3), help section to guide the user (4), and log out (5). The RFM app aims to ease the user's search of recipes, based on his preferences, available products, and allergies. Thus, we designed filters for that matter. The user can search for a given recipe manually (6), filtering by categories such as fish, vegetables, etc. (7), based on available products (8) and considering food allergies (9). Figure 4b describes one user's available product. In this example with the whole milk, the application shows a picture of it (1), its allergens (2), its name (3), its location (4), its quantity (5), its expiration or issued date (6), and supermarket offers of the product, if they exist (7). Since the expiration and issued date are linked to the fact of throwing away food, the background color changes depending on whether the product is fresh (blue), if it is not fresh and will expire, or is about to do so (red), and if it is not fresh

and it is in a good condition (black) (6). In addition, nutritional information such as energy, proteins, carbohydrates, and fat, is included, it refers to 100 g of product (8).

Besides product information, recipe and offer menus (c.f. Figure 4) are described. Figure 5 presents screenshots from the recipe menu, in Figure 5a–c and the offer menu in Figure 5d. Figure 5a shows the main section from the recipe menu, the presentation Table (1). The recipe option is also composed of three other tabs: Ingredients (2), recipe preparation text (3), and information (4). The presentation tab also displays a picture of the recipe (5). Figure 5b introduces the ingredients tab, providing information about the needed products to prepare the recipe. The ingredients tab displays the quantity of ingredients that is needed, depending on the number of people (1). Ingredients are shown as a list with different background colors: Red for non-available ingredients (2), red with a yellow border for non-available ingredients with available supermarket offers (3), and green for available ingredients (4). Note, that the ingredient can also change its availability (from green to red) depending on the number of portions we want to prepare (1). For instance, if we want to prepare stuffed eggplant with 500 g of chopped meat for four people, the app will color the background in red, because 960 g are needed. Figure 5c shows the information tab which explains the nutritional information per portion (1) and allergens (2). The offer menu is shown in Figure 5d, in which the product name is displayed (2), together with the price per unit (1). Moreover, the user can access to the offer by clicking on the product name (2).



(a) Main page.

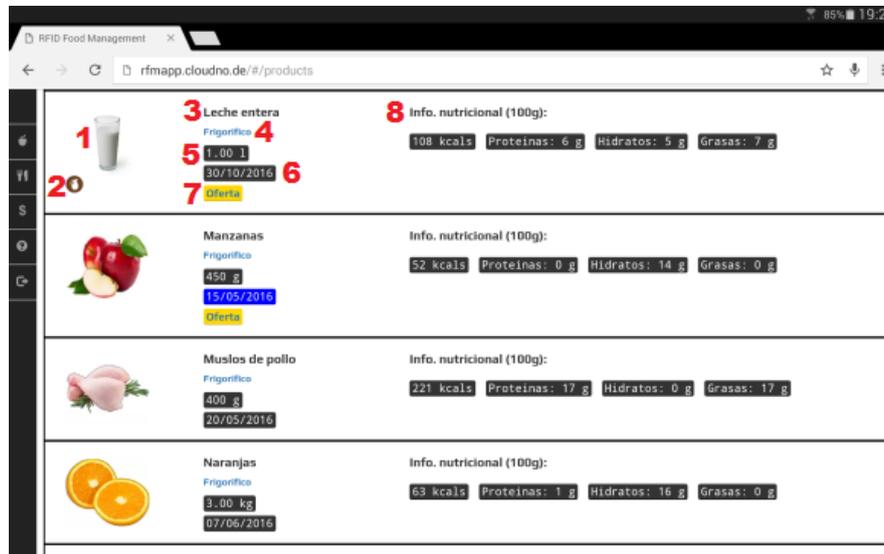


(b) Registration procedure.

Figure 3. Main page and registration form.



(a) Main menu.

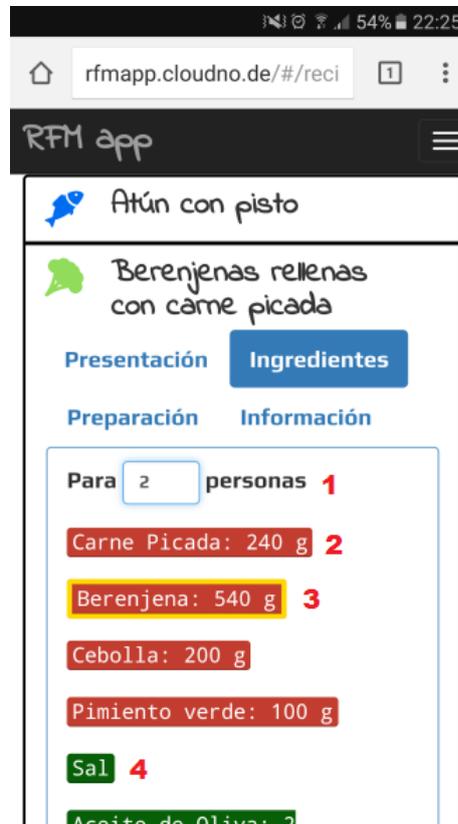


(b) Product information.

Figure 4. The user can access many options from the main menu.



(a) Presentation tab



(b) Ingredients tab.

Figure 5. Cont.

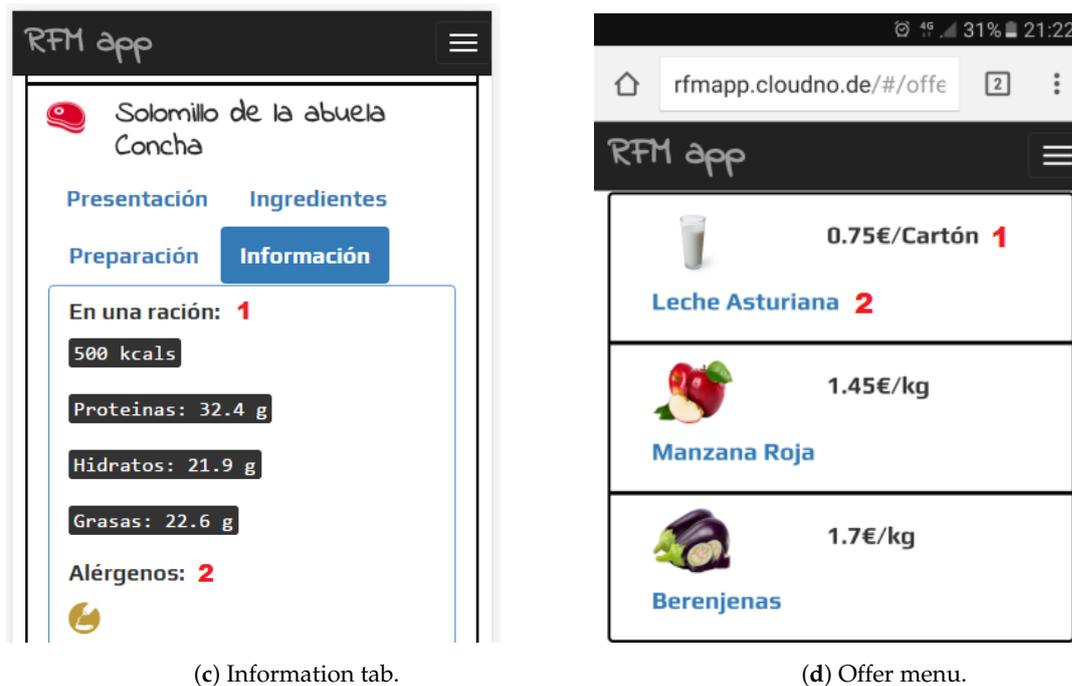


Figure 5. Recipe and Offer menu complete the RFM App.

Note, that all images shown above were captured from different devices. Thus, Figures 3a,b and 4a were captured from a personal computer. Figure 4b captured from a tablet with Android. Figure 5a,c from a smartphone with iOS, and Figure 5b,d from a smartphone with Android. We decided to insert captures from different devices to show how RFM App adapts to any type of device and platform.

4. Food Manager Algorithms

This section aims to present our approaches to recipe recommendations based on product availability in the kitchen furniture and its allergen content.

The first approach, Ingredient-Percentage based algorithm, only considers the number of available ingredients. Thus, the system will recommend a recipe if the user has a half or more of its required ingredients. Hence, if the recipe R is composed of n ingredients, and we have $n/2$ or more of these ingredients, this recipe will be recommended. Our second approach, Ingredient-Weight based algorithm, performs its recommendation depending on the importance of the ingredients, by assigning them a numerical weight in each recipe. We assume a weight of importance W for a given recipe, expressed as $W = W_Y + W_N$, where W_Y is the sum of importance weights from available ingredients, and W_N as the sum of importance weights from those non-available ingredients. Thereby, if W_Y is higher than W_N , in recipe R , this recipe will be displayed. Figure 6 presents an example of the described algorithms, deciding to recommend or not to recommend a stuffed eggplants recipe. The user has in his kitchen furniture: Minced meat, grated cheese, salt, oil, missing eggplants, onions, and pepper. The ingredient-percentage based algorithm will recommend the recipe since there are more available ingredients than missing ones, being four out of seven ingredients (more than 50% of available ingredients). The ingredient-weight based algorithm, will not recommend the stuffed eggplant recipe, since one of the main ingredients, eggplants, is not available, being W_N higher than W_Y , with sum of 7 and 8, respectively.

In Figure 6, we observed how both Ingredient-Percentage and Ingredient-Weight based algorithms work. The following section presents a real-time test of recipe recommendations.

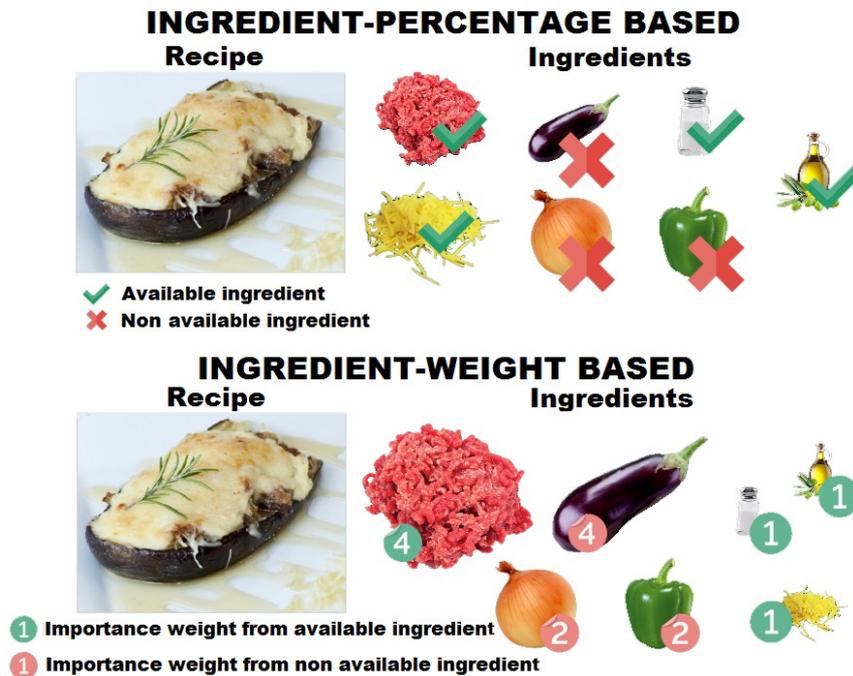


Figure 6. Example of a recipe recommendation using our proposed algorithms.

5. Demonstration

This section evaluates our recipe recommendation algorithm’s approaches.

We performed the recipe recommendation algorithms by using the passive RFID tags as labeled-products, and the RFID system, composed of an Arduino board and a RFID module, as the reading system installed in the kitchen furniture. Figure 7 shows the elements mentioned above, while Figure 7a represents the passive RFID tags, which store product information, Figure 7b shows the RFID system ready to read and process the information contained in the passive RFID tags.

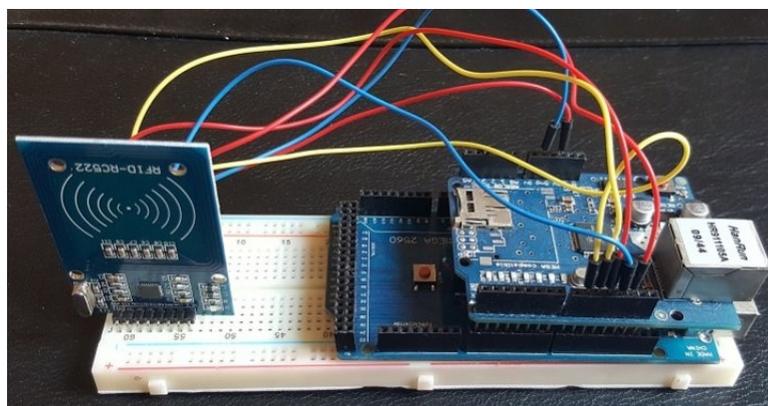
The evaluation of the recipe recommendation algorithms was done by adding, step by step, ingredients to the RFM App, and reading the passive RFID tags associated to the different products. Initially, we assumed the kitchen furniture was empty. Later, we checked the recommended recipes according to the products introduced in each step. Table 4 tabulates the recipe recommendations returned when ingredients were introduced, with both Ingredient-Percentage and Ingredient-Weight based algorithms. The column “Inserted products” shows the ingredients in each “Step” returning the new recipes recommended by the the system in the recommended recipes columns. At Step 1, no ingredients are introduced in the embedded system so that no recipe is shown with neither the Ingredient-Percentage nor with the Ingredient-Weight based Algorithm. In Step 2, the Ingredient-Percentage based Algorithm returns a mashed carrots recipe, despite the fact that no carrots were introduced. Since milk represents 50% of the mashed carrots recipe ingredients and is available, the recipe is recommended. The Ingredient-Weight based algorithm performs a better recommendation as it recommends a mashed potatoes recipe, in which potatoes are the main ingredient. In step 5, the Ingredient-Percentage algorithm recommends stuffed eggplant recipe, despite neither meat nor eggplants being present. However, the Ingredient-Weight based algorithm recommends the stuffed eggplants when the eggplant is introduced in step 7. Note that, from these recommended recipes shown at each step, we also have to consider also previous ones. For instance, at Step 3 with the Ingredient-Weight based algorithm, no new recipes are recommended, however, the mashed potatoes recipe will be displayed.

From Table 4, we observed how both Ingredient-Percentage and Ingredient-Weight based algorithms return recipes when ingredients are introduced, and more importantly, the recommendations adapt in progression with the new ingredients. Overall, the Ingredient-Weight

based Algorithm returns better matches, according to the ingredients introduced, and the recommended recipes, as it considers the main ingredient to be key for recipe preparation.



(a) RFID tags linked to a given product.



(b) RFID system composed of an Arduino board and a RFID module.

Figure 7. Elements used to evaluate our approaches.

Table 4. Algorithm evaluation.

Step	Inserted Products	Recommended Recipes	
		Ingredient-Percentage Based	Ingredient-Weight Based
1	No ingredients	- No recipes	- No recipes
2	Potatoes Milk	- Mashed carrots	- Mashed potatoes
3	Salt Black pepper	- Mashed potatoes	- No new recipes
4	Eggs	- Breaded pork tenderloins Egg custard	- Egg custard
5	Oil Onions	- Stuffed eggplants - Breaded pork tenderloins - Murciana salad - Spaghetti carbonara - Fried milk	- Fried milk
6	Rice	- Stewed lentils with vegetables and rice	- No new recipes
7	Eggplant Tomatoes	- Vegetables salad - Grilled vegetables - Steamed mussels with vinaigrette - Greek salad - Bolognese pasta with eggplant	- Vegetables salad - Stuffed eggplants
8	Chicken thighs	- No recipes	- Chicken paella
9	Tuna	- Tuna with ratatouille	- Tuna with ratatouille
10	Octopus Feta cheese	- No new recipes	- Octopus salad with cilantro and mint - Greek salad

6. Conclusions

Roughly, developed countries waste as much food (222 million tonnes) as the net food production in sub-Saharan Africa (230 million tonnes) on an annual basis. This fact, being unfair to sub developing countries with low resources for food production, also affects the carbon footprint by emitting 3.3 billion tonnes of CO₂ to the atmosphere. Moreover, the number of people with health issues, such as being overweight and obesity, is increasing worldwide, provoking an increment of medical expenses due to doctor assistance, medication, and surgeries.

The Smart City paradigm aims to enhance existing services to improve people's lives. IoT provides the tools for cities to become smart. Within the IoT technologies involved in the creation of services in the Smart City paradigm, the radio frequency identification technology allows interaction between sensor devices. Since RFID is capable of identifying labeled-item uniquely and is widely used for inventory procedures.

This paper presented a food management system, RFID Food Management. RFM is composed of a RFID system capable of reading labeled-items, and a multi-platform application, which processes the RFID information in real time. Two approaches of intelligent recommendation algorithms were implemented and tested: Ingredient-Percentage based algorithm, which recommends, according to the number of ingredients available, and the Ingredient-Weight based algorithm which performs its recommendations based on the importance of the available ingredients within the recipe. Demonstration results comparing these algorithms conclude that a more precise recommendation is obtained when using the Ingredient-Weight based algorithm, because of the importance it gives to the main ingredients of each recipe. The contributions of this work include a multi-platform and centralized application, in which users can obtain recipe recommendations according to the content of the kitchen furniture, in real time, and anywhere. Furthermore, this work enhances the supply chain value in an IoT context by offering to the consumer daily product promotions, benefiting both user and retailer by saving and earning money, respectively. Since RFID is becoming popular to label products in supermarkets for fast checkout processes [23], we believe that the combination of RFID [24] and intelligent recommendation algorithms, will reduce the waste of food worldwide.

We envision a world where the waste of food is reduced since users with traditional kitchen furniture together with a low-cost RFID system and a simple application can check the content of their kitchen furniture and obtain real-time recipe recommendations based on the ingredient availability. As future work, we plan to enhance our RFM real-time embedded system by:

- Introducing international ingredients and recipes to adapt to all kind of kitchens;
- Testing our approach in traditional kitchen furniture with UHF RFID system;
- Implementing intelligent functions, as Fuzzy Logic to receive notifications about food consumption, and gamification principals to motivate users for reducing food waste;
- Designing a cloud-based architecture to keep all the information per user and home "online". Furthermore, this approach may help with the algorithm running process.

Author Contributions: Conceptualization, R.P.; methodology, R.P., A.P. and C.M.; software, R.P. and A.P.; validation, R.P. and A.P.; formal analysis, R.P., A.P. and C.M.; investigation, A.P.; resources, A.P.; data curation, R.P. and A.P.; writing–original draft preparation, R.P.; writing–review and editing, R.P., A.P., C.M. and J.M.-S.; visualization, R.P. and A.P.; supervision, R.P. and C.M.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Food and Agriculture Organization of the United Nations. *Food Wastage Footprint: Impacts on Natural Resources*; Food and Agriculture Organization of the United Nations: Rome, Italy, 2013; ISBN 978-92-5-107752-8. Available online: <http://www.fao.org/docrep/018/i3347e/i3347e.pdf> (accessed on 31 July 2019).
2. World Health Organization. Global Health Observatory (GHO) Data. 2014. Available online: http://www.who.int/gho/ncd/risk_factors/overweight/en/ (accessed on 31 July 2019).
3. Populus. The Rotten Truth about Household Food Waste. 2016. Available online: <http://www.populus.co.uk/2016/09/rotten-truth-household-food-waste/> (accessed on 31 July 2019).
4. EPCglobal. *EPC Radio-Frequency Identity Protocols EPC Class-1 HF RFID Air Interface Protocol for Communications at 13.56 MHz*; Version 2.0.3; GS1: Brussels, Belgium, 2011.
5. Samsung. Samsung Introduces an Entirely New Category in Refrigeration as Part of Kitchen Appliance Lineup at CES 2016. 2016. Available online: <https://news.samsung.com/global/> (accessed on 31 July 2019).
6. CBS Interactive Inc. A Close Look at LG's Smart ThinQ LFX31995ST Refrigerator (Hands-on). 2013. Available online: <https://www.cnet.com/products/lg-smart-thinq-lfx31995st-refrigerator/preview/> (accessed on 31 July 2019).
7. Luo, S.; Jin, J.; Li, J. A smart fridge with an ability to enhance health and enable better nutrition. *Int. J. Multimedia Ubiquitous Eng.* **2009**, *4*, 66–80.
8. Rouillard, J. The Pervasive Fridge. A smart computer system against uneaten food loss. In Proceedings of the Seventh International Conference on Systems (ICONS2012), Saint Gilles, Belgium, 29 February–5 March 2012; pp. 135–140.
9. Hanshen, G.; Dong, W. A Content-aware Fridge based on RFID in smart home for home-healthcare. In Proceedings of the 11th International Conference on Advanced Communication Technology (ICACT), Gangwon-Do, Korea, 15–18 February 2009; pp. 987–990.
10. Banzi, M.; Cuartielles, D.; Igoe, T.; Martino, G.; Mellis, D. Arduino Mega. 2016. Available online: <https://www.arduino.cc/en/Main/ArduinoBoardMega2560> (accessed on 31 July 2019).
11. NXP Semiconductors N.V. MIFARE RC522 Datasheet. 2014. Available online: http://www.nxp.com/documents/data_sheet/MFRC522.pdf (accessed on 31 July 2019).
12. NXP Semiconductors N.V. MIFARE Classic 1K Datasheet. 2011. Available online: http://www.nxp.com/products/identification-and-security/mifare-ics/mifare-classic:MC_41863 (accessed on 31 July 2019).
13. Banzi, M.; Cuartielles, D.; Igoe, T.; Martino, G.; Mellis, D. Ethernet Shield W5100. 2016. Available online: <https://www.arduino.cc/en/Main/ArduinoEthernetShield> (accessed on 31 July 2019).
14. Checkpoint Systems, Inc. DECATHLON Futher Extends Its RFID Source Tagging Program for Millions of Products In All its Stores Around the World. 2016. Available online: <http://us.checkpointsystems.com/news-events/news-item/> (accessed on 31 July 2019).
15. United States Department of Agriculture. USDA Food Composition Databases. 2016. Available online: <https://ndb.nal.usda.gov/ndb/> (accessed on 31 July 2019).
16. Deployd. The Simplest Way to Build an API. 2012. Available online: <http://deployd.com/> (accessed on 31 July 2019).
17. MongoDB Inc. For Giant Ideas. 2009. Available online: <https://www.mongodb.com/> (accessed on 31 July 2019).
18. Dahl, R. NodeJS. 2009. Available online: <https://nodejs.org/en/> (accessed on 31 July 2019).
19. Eikos Partners. Multi Page Web Applications vs. Single Page Web Applications. 2015. Available online: <http://www.eikospartners.com/blog/multi-page-web-applications-vs.-single-page-web-applications> (accessed on 31 July 2019).
20. Google, Inc. AngularJS. 2010. Available online: <https://angularjs.org/> (accessed on 31 July 2019).
21. Twitter, Inc. Bootstrap. 2010. Available online: <http://getbootstrap.com/> (accessed on 31 July 2019).
22. Food Allergy Research & Education, Inc. Food Allergy Facts and Statistics for the U.S. 2016. Available online: <http://www.foodallergy.org/file/facts-stats.pdf> (accessed on 31 July 2019).

23. Advanced Mobile Group. Scanning Away Food Waste? How RFID Tags Could Minimize Retail Food Waste by 20 Percent. 2015. Available online: <http://foodtank.com/news/2015/09/> (accessed on 31 July 2019).
24. Bloom, J. The Secret Behind Amazon Go: RFID. 2015. Available online: <http://www.advancedmobilegroup.com/blog/the-secret-behind-amazon-go-rfid> (accessed on 31 July 2019).



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).