



UNIVERSIDADE ATLÂNTICA

LICENCIATURA EM SISTEMAS  
E TECNOLOGIAS DE INFORMAÇÃO

---

**Identifying allergens and  
intolerances in food products  
through the use of a mobile  
application**

---

*Author:*

Luís Filipe de Araújo  
Correia - 20101332

*Supervisor:*

Prof. Dra. Elizabeth  
Carvalho

September 6, 2013



UNIVERSIDADE ATLÂNTICA

LICENCIATURA EM SISTEMAS  
E TECNOLOGIAS DE INFORMAÇÃO

---

**Identifying allergens and  
intolerances in food products  
through the use of a mobile  
application**

---

*Author:*

Luís Filipe de Araújo  
Correia - 20101332

*Supervisor:*

Prof. Dra. Elizabeth  
Carvalho

September 6, 2013



The views expressed in this document are the sole responsibility of the author.



The author, Luís Filipe de Araújo Correia, grants Universidade Atlântica a non-exclusive, non-transferable authorization to archive and make accessible to others this document, free of charge, either partially or in its entirety.

All rights reserved, Luis Correia 2013





# Acknowledgements

- to Marina, my loving companion who kept pushing me further, never giving up and always believing in me
- to my mother Maria Belém and to my father António, who will never read this, may they Rest In Peace
- to my classmates, Paulo, José, Marco and Carlos, I couldn't have made it without you guys
- to all my teachers at Universidade Atlântica, you were the best for the difficult task
- to my employer, Instituto de Informática, I.P., for aiding me throughout these years
- to my friends, António Lopes, João Alves and Sónia Fernandes for helping in proofreading this document
- to Google, for awesome search engine and scholar papers
- to Wikipedia, for providing free access to knowledge even though the academia doesn't like it



# Abstract

Modern society habits have an impact on the way we eat, often leading to the consumption of ready made meals. As more and more ingredients are added to our daily diet some people have developed food allergies. Since there is no cure for food allergies, the best practice is to avoid them by carefully reading the product labels.

Internet enabled smartphones are easily available today and there are already several applications that can help the user to live a healthier lifestyle by providing coaching plans and inviting the user to exercise more.

This work introduces the design, development and implementation of an economic mobile software solution that can give the end-user a good support in identifying allergens in products. During the elaboration of this thesis, a prototype was created to provide a proof of concept mobile application. This prototype included a database, web services and a smartphone mobile application.

Keywords: Food allergies, food intolerance, barcode, smartphone



# Contents

Disclaimer	i
Document rights	iii
Acknowledgements	v
Abstract	vii
Keywords	vii
List of Figures	xi
List of Tables	xiii
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation and Relevancy . . . . .	2
1.2 Goals and objectives . . . . .	3
1.3 Research methodology . . . . .	4
1.4 Document organization . . . . .	4
<b>2 Concepts and Technologies</b>	<b>5</b>
2.1 Concepts . . . . .	5
2.1.1 Food Allergies . . . . .	5
2.1.2 Food labelling . . . . .	6
2.1.3 Barcodes . . . . .	8
2.1.4 Web services . . . . .	8
2.1.5 Privacy and security . . . . .	9
2.2 Technologies . . . . .	10
2.2.1 Barcode identification . . . . .	10
2.2.2 Image recognition . . . . .	10
2.2.3 Database engines . . . . .	11

2.2.4	Smartphone platforms . . . . .	11
2.3	Chapter conclusions . . . . .	12
<b>3</b>	<b>Solution design and architecture</b>	<b>13</b>
3.1	Solution Design . . . . .	13
3.1.1	Database model . . . . .	13
3.1.2	Communications data flow . . . . .	14
3.1.3	Mobile application . . . . .	15
3.1.4	Web Interface . . . . .	16
3.2	Solution architecture . . . . .	17
3.2.1	Database structure . . . . .	17
3.2.2	REST web services . . . . .	18
3.2.3	Android application . . . . .	19
3.2.4	Web Interface . . . . .	20
3.3	Chapter conclusions . . . . .	20
<b>4</b>	<b>Solution Prototype</b>	<b>21</b>
4.0.1	Prototype design and architecture . . . . .	21
4.0.2	User feedback . . . . .	22
4.0.3	Development tools . . . . .	22
<b>5</b>	<b>Conclusions</b>	<b>23</b>
<b>6</b>	<b>Acronyms</b>	<b>25</b>

# List of Figures

2.1	Non-standard labelling in food products (image from Mills et al. (2004)) . . . . .	7
3.1	Database schema (model) . . . . .	14
3.2	Simplified web services request and replies . . . . .	15
3.3	Mobile application main menu screen . . . . .	16
3.4	Database structure with foreign keys . . . . .	18
3.5	Product search information flow . . . . .	18





# List of Tables

1.1	Objectives, Tasks and Expected results . . . . .	3
2.1	REST vs SOAP . . . . .	9
2.2	Android and iOS development compared . . . . .	12

# Chapter 1

## Introduction

The lifestyle we have today has made a profound impact on the way we eat. The long hours at work, the time spent in commuting to and from work and the lack of free time have the inevitable effect of resorting to the consumption of ready made meals. While researchers are still studying the long term effects of using food additives and processed ingredients as part of our daily diet, some people have developed the inability to process (digest) these products.

Since food allergies cannot be cured, studies are being conducted in order to figure out which is the best practice to follow. Most recommend strict avoidance but for children, as Wood (2003) has stated, it can be difficult to achieve. Reading product labels is necessary in order to figure out if there is some kind of ingredient that may affect them. A few of these are nuts, gluten, lactose and dairy products.

In the current day and age, smart-phones with Internet connectivity are easily available and several projects have been created to address the issue of Food Allergies (FA). A search for "allergen" related applications returns a large list of available mobile applications but most of them focus on collecting notes and providing generic information of FA. Until now, the author has not found an application that could fill-in this gap and therefore believes that with this project **it is possible to design, develop and implement a low-cost mobile software solution that can give the end-user a good support in identifying allergens in products.**

## 1.1 Motivation and Relevancy

FA can be fatal, Boyce et al. (2010) reported that "Deaths from food induced anaphylaxis have been reported within 30 minutes to 2 hours of exposure and usually result from cardio-respiratory compromise". The allergic reaction to food ingestion cannot be cured and strict avoidance should be taken. Unfortunately, a large number of the current population is allergic or intolerant to some ingredients present in products.

There are already several software applications that can help the user to live a healthier lifestyle. Exercise workout companion applications, sometimes connected to heart-rate monitors, can in some cases replace a personal trainer. There are also some diet management applications which help to maintain good health. These are only a few examples of an ever increasing list of support applications. However, very little or nothing can help consumers avoid buying the products that could be a problem.

Studies have been conducted to determine what type of labelling and font size should be used for better readability. Wogalter et al. (1993) have conducted an experiment with US common use labelling present in prescription drug packaging. The target groups were the very young and the very old since they represent the extremes. They found that increasing surface area and print size improved readability enormously.

Unfortunately, these days we don't see this reflected in the majority of products. There is still much work to be done in this area and this project can help by showing consumers which (if any) allergens are present in common food products.

## 1.2 Goals and objectives

In order to achieve the intended goal, this project proposes the following objectives, tasks and expected results:

**Problem:** People suffer from FA and have difficulty in identifying allergens in food products.

**Thesis:** It is possible to design, develop and implement a software solution that would help to identify allergens in food products.

**Goal:** Promote a better lifestyle by preventing people from buying food products containing allergens through the use of an economic, easy to use, technological mobile solution.

Objectives	Tasks	Expected results
State-of-the-art in the field	Literature review	Relevant technological, ethical and theoretical aspects related to software development
Conceive and design a mobile solution	Design a prototype and a database to test and evaluate the proposed solution	Solution architecture
Implement and test a mobile solution	Development and implementation of a prototype and a database to test and evaluate the proposed solution	Solution prototype

Table 1.1: Objectives, Tasks and Expected results

The first objective is to research the current state-of-the-art regarding software solutions, which are the trending mobile phone platforms and how food products are being labelled.

The second objective is to conceive and design the project architecture regarding software systems.

The third objective is to implement and test a working, usable prototype that can demonstrate the concept using real products.

### 1.3 Research methodology

Adopting a research methodology is fundamental as it aids design and development as well as provides guidelines to proper execution and research validation. This project bases its premises on that it is possible to design, develop and implement a mobile software solution.

This software solution is a prototype that will be functionally close to what the end product will be. The author will use experiments as the scientific method to evaluate and validate the prototype.

The author is aware that by experimenting and analysing the results he becomes part of the problem, thus jeopardizing the required impartiality. However, the author has chosen to take a pragmatic approach to his research and whichever results that come out of the experiment will be dealt with accordingly. User testing will be important, specially during the early prototype testing phase.

### 1.4 Document organization

This document is organized using the following structure:

- Chapter 2 describes main concepts and related work.
- Chapter 3 describes software solution design and architecture.
- Chapter 4 describes the prototype in detail. Technical difficulties and what ended up implemented will be done here. All user feedback is resumed and presented here as well.
- Chapter 5 presents the conclusions and plans for future work.

# Chapter 2

## Concepts and Technologies

This chapter introduces concepts and technologies in the field of FA prevention and possible choices for mobile platforms. It is important to understand FA from the scientific point of view by searching for relevant articles and current practices. In order to be able to decide which platform to use, the author will analyse leading mobile phone platforms and their corresponding Software Development Kit (SDK).

### 2.1 Concepts

This section introduces key concepts.

#### 2.1.1 Food Allergies

FA are best described as "an abnormal response to a food, triggered by the body's immune system..." - Sampson (2004). It is known that some reactions to allergens can be fatal, mainly due to anaphylactic shock, so anything that could be made available to prevent this is of great value. Due to relevance and importance to the subject, Food Intolerances (FI) will also be covered by this project as proper detection and avoidance also improves the quality of life for those affected.

Although the existence of FA is unquestionable, there is still much work to be done in discovering just how many people are affected and how. In his meta-analysis, Rona et al. (2007) suggests that standardized measurement

methods should be used. This meta-analysis researched a total of 934 articles and found that only 51 were appropriate. One of the main flaws is the self-reported FA for which no study can support. It also suggests that there should be more double-blind, placebo-controlled food challenge (DBPCFC), as this is a gold standard.

In a placebo-controlled test, patients ingest foods that include or don't contain the likely allergen. "Double-blind" refers to a method of testing in which neither the allergist nor the patient is certain whether the substance ingested contains the likely allergen.

Rona et al. (2007) then present a careful compilation of results from all 51 articles and conclude that, since there is a great heterogeneity in the measurements, they cannot reassure stakeholders that these differences represent genuine differences between populations. The analysis suggests that a more collaborative study using similar methods will eventually minimize disparate results caused by different methodologies and technologies and also that the current information on FA should not be considered as reliable.

### 2.1.2 Food labelling

Allergic ingredients are widely used by food manufacturers, sometimes included as a sub-processed ingredient. Therefore, proper labelling should be present at all times in all products. In 2003, the European Union passed allergen labelling legislation 2003/89/CE & Directiva (2003) that clearly defined eight major food groups. The following year, similar legislation was passed in the United States of America, Law, 108-282 AUG 2 (2004).

Every product that uses ingredients from these food groups needs to be clearly labelled. Unfortunately, even with proper food labelling, it is not easy to perceive all components or allergens that might be described in the product label. Figure 2.1 shows the different labelling used in various products.

This project will consider the following list of allergens as the baseline for allergen identification:

- Cereals containing gluten
- Crustaceans
- Eggs
- Fish

- Peanuts
- Soy beans
- Milk
- Celery
- Mustard
- Sesame seeds
- Sulphur dioxide and sulphites  
(if at concentrations over 10mg/Kg or 10mg/l)
- . . . and all products thereof

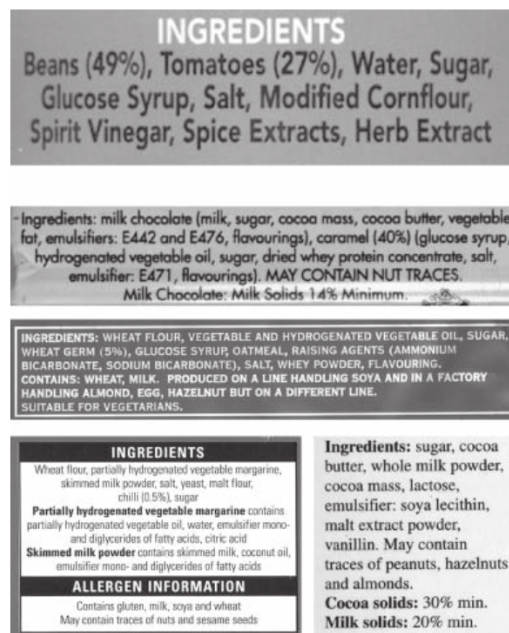


Figure 2.1: Non-standard labelling in food products (image from Mills et al. (2004))

Mills et al. (2004) have researched what is currently being done in order to regulate the actual food allergen labelling as the regulators aren't responding very well to the problem. More specifically, they found out that deciding on which threshold is one of the problem. Even using DBPCFC testing, the results vary wildly as different people have different reactions.



### 2.1.3 Barcodes

A barcode is an optical machine-readable representation of data, usually by varying the widths and spacings of parallel lines. Typical supermarket products have standard ISO/IEC15420 EAN-13 barcode labels. Exceptions are made for products with small packaging which in turn use EAN-8 barcodes. This digital encoding technique enables product identification in a fast and precise way without the need for physical contact.

Most common readers use some sort of infrared light or laser beam to illuminate the material where the barcode is printed. Several techniques for decoding barcodes from images captured by digital cameras are available. One of them is described by Chai & Hock (2005), where they propose a vision based barcode detection algorithm.

### 2.1.4 Web services

Almost all current systems use Web Services (WS) to exchange data and information. From the currently available technologies, REST and SOAP are the most common and usually paired as REST/JSON and SOAP/XML. There is no technical restriction to the combination of technologies, as REST paired with XML is also common. The author has found a very detailed analysis of these technologies done by Pautasso (2008).

Historical, SOAP/XML is being used in enterprise systems as the best method to interoperate with different systems. The main advantages of SOAP/XML are field definitions which are described in the Web Service Description Language (WSDL). WSDL is used to announce the available endpoints and the various field types and constraints used for validation. This is also a disadvantage as all the field descriptions add complexity and size to the payload. It also requires strict type checking at both ends of the conversation.

According to the same analysis, REST/JSON is very lightweight and has low bandwidth usage. JSON was created in 2001 by Douglas Crockford and it is often used for serializing and transmitting simple data. The REST/JSON combination is used in most if not all web applications because JSON is a JavaScript encoded object, the preferred client side web browser scripting language. The format makes it very easy to encode and decode information in a web browser.

Table 2.1 presents a simplified feature comparison between REST and SOAP (compiled from Pautasso (2008)).

	REST	SOAP
Characteristics	Operations are defined in the messages	Operations are defined as WSDL ports
	Unique address for every process	Unique address for every operation
Self-declared advantages	Late binding is possible	Debugging is possible
	Process instances are created explicitly	Complex operations can be hidden behind facade
Possible disadvantages	Large number of objects	Previous information on operations and semantics needed
	Managing the URI namespace can become cumbersome	Process instances are created implicitly

Table 2.1: REST vs SOAP

### 2.1.5 Privacy and security

The best practices in system security define that each password should be unique between different systems. This practice makes it very hard for users to memorize unique passwords. To overcome this, the project will rely only on secure user account management services available natively on the mobile phone and as such they will be used as the key to access the system.

The communications to and from the server will be encrypted using a Secure Socket Layer (SSL) connection, thus preventing eavesdropping. While it is known that using SSL degrades performance, as Kant et al. (1999) have shown, it is best to accept the potential loss of performance on behalf of the added security. The collected data during the usage of the system will be kept private and only used for statistical analysis. No user identifiable information will be provided (if ever) to third parties.

## 2.2 Technologies

Projects have been created to address the issue of FA or FI. As an example, a quick search on Google Play (an application download service from google) with the term "allergen" returns a large list of available applications. By reading the application's details, it is possible to conclude that most of them focus on note collection and generic information of FA.

### 2.2.1 Barcode identification

Broadening the search outside specific application stores, the "Mobile Expert and Networking System for Systematical Analysis of Nutrition-based Allergies" (MENSSANA) project Arens et al. (2008) was found. The main goal of this project was to provide a bridge between the patient and the doctor and to improve the data quality of the consumption habits of the patient. MENSSANA dates back to 2008, an epoch where smartphone processing capabilities were weak and Internet connectivity was not ubiquitous. The MENSSANA project's main objectives were "Develop a barcode reading Personal Allergy Assistant (PAA), which helps to distinguish permitted and prohibited food", "Definition of a standardised Electronic Patient Record for Allergies (EPRA) with coherent patient diaries, acquisition of patient anamnesis, representation of allergy tests", "Provide food based allergy specific information via Internet" and "Conducting a controlled clinical study to evaluate health economic effects and quality of life".

### 2.2.2 Image recognition

Image recognition usage in mobile phones is still a difficult subject but as more powerful handsets are developed, more can be done. There is some interesting work being done by Julier (2012), in which image markers are used to make representation of 3D virtual objects as seen from the mobile phone's on-board camera. Usually this is only used as proof of concept or in games. Unfortunately, the libraries and software developer kits that are currently available need excessive amounts of data to store object information.

### 2.2.3 Database engines

Today, most if not all projects use some form of database to store information. There are different database types designed to specific needs but overall we could classify them as RDBMS or NoSQL. RDBMS are relational databases, such as Oracle DB, Microsoft SQL Server and MySQL. On the other hand, NoSQL stands for a key-value database store where there is no relation between the different tables.

As Cattell (2010) has researched, for large distributed systems there are advantages and disadvantages for the two approaches. While it is difficult to maintain consistency for distributed RDBMS, usual NoSQL implementation opt for a "mostly consistent state".

### 2.2.4 Smartphone platforms

According to a recent Gartner report Goasduff & Pettey (2012), Google's Android market share was 50.0% in 4Q11, whereas iOS from Apple scored only 23.8%. At the moment, the Windows Mobile platform from Microsoft is almost irrelevant due to its 2% share. The other platforms are RIM's BlackBerryOS, which targets the enterprise segment, Microsoft's Windows Mobile and Windows Phone that don't have a large adoption and finally there is also Nokia's Symbian, which is usually used in mid to low-end phones. The remaining platforms are insignificant and therefore were not considered.

There are some considerations to be made when choosing the best platform:

- SDK availability and complexity
- Learning curve of the programming language
- Cost of deploying the application

When programming for Apple iOS devices some prior considerations need to be taken. For example, the SDK is called XCODE and requires an Apple computer running a recent version of the OS/X Operating System. While using and developing with XCODE is free, there are yearly fees for submitting applications to the App Store. The fees start at \$99USD/year for individual and small companies and go up to \$299USD/year for enterprises. The programming language is Objective-C and although it's not very dissimilar to C++, there are some differences.

The requirements for programming for Android are different. SDK is available for Windows, MAC OS/X and Linux and covers the majority of available operating systems. Traditionally the supported Integrated Development Environment is Eclipse but since May of 2013, the Android team suggests that developers should use Android Studio. The cost for deploying applications to Google Play as an individual has a one time fee of 25EUR. The programming language has the same syntax as Java although it is not standard Java.

The author is aware of "hybrid" software platforms, such as Phonegap and Appcellerator but due to the uniqueness of barcode reading, decided that only native SDK's should be considered for this project as not all hybrid platforms provide a unique method for handling the smartphone's hardware.

Comparison between Android and iOS development (information obtained from the project's websites).

	Android	iOS
Operating system for SDK	Windows, MAC OS/X and Linux	Requires recent MAC OS/X
Programming Language	Java	Objective-C
Application store fees	25EUR (one time)	\$99USD/year

Table 2.2: Android and iOS development compared

## 2.3 Chapter conclusions

After analysing all potential technologies, the author is considering choosing Android as the mobile software platform and as such, the programming language will be Java. External code will be used from available open-source libraries as they will be responsible for barcode scanning. For the server, the database engine candidate will probably be MySQL. From the available server scripting languages, PHP5 is a likely candidate. The web interface will make use of current technologies. HTML5, Javascript and CSS3 standards are being widely used today, as they provide browser independent, fluid and responsive user experience. No decision has been made so far as to which web service type will be used.

# Chapter 3

## Solution design and architecture

This chapter describes the design and architecture for the software solution. The following sections will describe the design for the database model, the web services and communication data flow, the mobile application and the web interface.

### 3.1 Solution Design

#### 3.1.1 Database model

The proposed database in Figure 3.1 follows the relational model with few entities and relationships. The entities are Users, Products and Allergens. Tables are functionally related and, for example, products may have one or more allergens and each allergen will belong to a specific category.

Users may have one or more allergies that will be classified into the categories (*e.g.* hardshell nuts). The system keeps a history of all products scanned even if the product is not present in the database. The full database structure is detailed in the database dictionary annex.

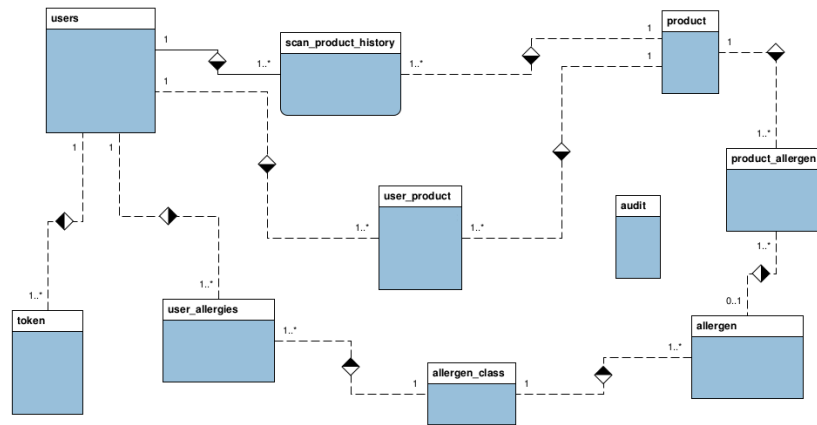


Figure 3.1: Database schema (model)

### 3.1.2 Communications data flow

This project adopts a well known client-server model in which the mobile application is the client and the web service is the server. The requests will always be initiated by the client and can be simple text strings or arrays of information. This ensures that data size will be kept to the minimum necessary for the task.

The system is designed to work in every possible network condition and data size matters specially when broadband is being used and taxed by the megabyte. The author is well aware that free wireless access is not yet available everywhere.

The web services provide an interface for the database tables. Sending read or write requests for the specific web services will read and write into the specific tables. Requesting a read operation from a table is mostly straightforward whilst writing requires additional checks to keep data consistent. Figure 3.2 shows a simple HTTP POST operation, returning one of three possible results.

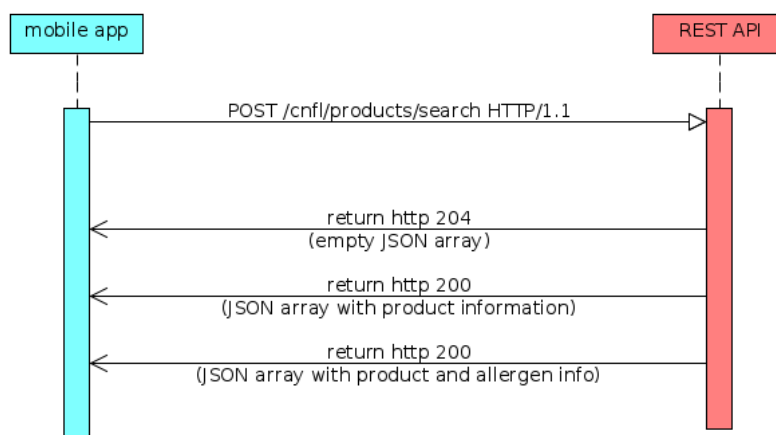


Figure 3.2: Simplified web services request and replies

### 3.1.3 Mobile application

The mobile application is the most visible part of the system. It is responsible for processing user requests, makes use of the smart-phone's camera to scan and decode the barcodes and to invoke remote WS.

Users have a preferences screen where they define the allergy categories and also a scan history where all previously scanned products can be seen. Figure 3.3 shows a mockup of the main menu.



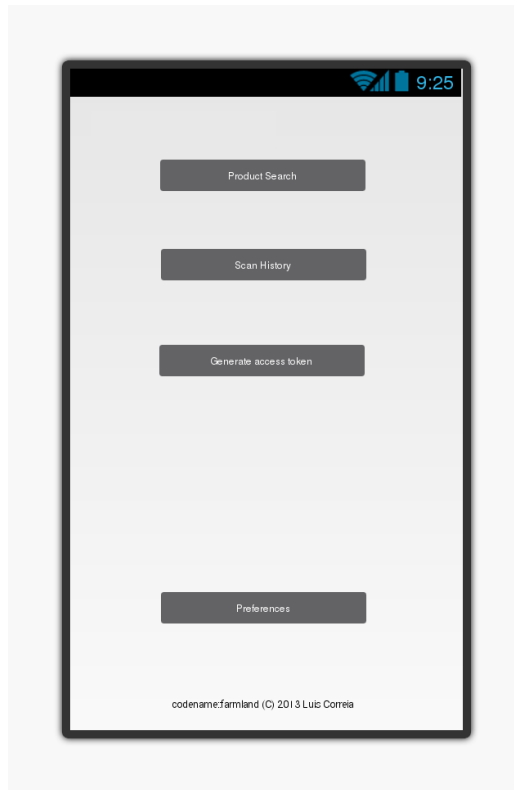


Figure 3.3: Mobile application main menu screen

### 3.1.4 Web Interface

The web interface will perform most of the tasks that are possible with the mobile application. The relevant limitation is scanning for barcodes, as we cannot assume every device is able to do this. However, the user can input the barcode numbers and get the same information. There are no additional functionalities provided by this web interface regarding the system.

The author decided to make this interface available for the convenience of using the system on a larger screen. The system will work without ever using the web interface. This web interface will have a responsive design in order to accommodate for the differences in screen sizes and browser versions and provide the best user experience possible.

## 3.2 Solution architecture

The technical architecture chosen for this project only uses Open Source technologies. Based on the possible choices provided by the research, the author chose Android as the preferred mobile platform. The main reasons were the multiple supported operating systems and the lower cost for deploying the application. All server side code will use PHP5 as the scripting language and MySQL as the database engine. All WS will be RESTful, using JSON for serializing and transferring data.

### 3.2.1 Database structure

The database structure displayed in Figure 3.4 follows the original design planned for this project very closely. There will be three major tables for products, users and allergens. The "glue" tables connect logically between them. The *user\_product* contains the user preferences for either positive or negative products and the *scan\_product\_history* is mostly self explanatory. It also stores all products scanned by the user.

The *product\_allergen* table will establish the relation between products and allergens, if any. In order to simplify choosing the users allergies, the *allergen\_class* table was created. Since the allergens can be grouped by class or category, it is easier to relate the user to the allergies by the use of the *user\_allergies* class. Finally, the *allergen* table only has just the known name and the reference to the allergen category.

The token table will only be used to manage access to the web interface and it may not actually be used at all. The audit table will have a log of all the actions taken by the user in the system. It will be used for gathering statistics on system use.

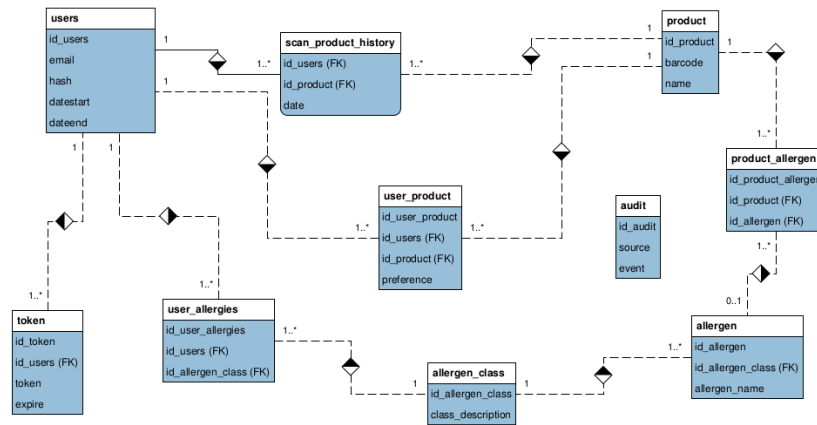


Figure 3.4: Database structure with foreign keys

### 3.2.2 REST web services

The web services endpoint implementation makes use of an open source PHP class. This class Sekar (2012) is very easy to understand and use. The actual implementation chosen was to use just one base RESTapi class and make three separate uses for each of the web services endpoints. The database helper functions are also common and provide direct access to the tables.

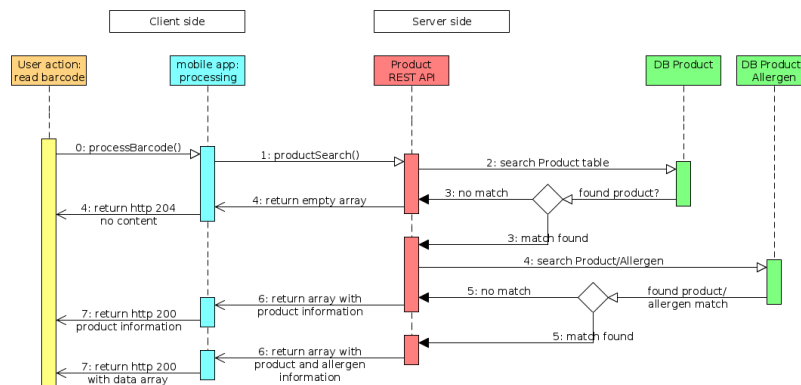


Figure 3.5: Product search information flow

Figure 3.5 describes how the "search product" data flows from the client application to the server and the possible information returned. After the

user has read a valid EAN13 barcode, the application calls the "processBarcode()" method, passing the string that represents the barcode. The REST client invokes the corresponding remote endpoint for "product search". The REST WS queries the database to search for the product and for any information about allergens present and returns it as an array. Provisions are made in case the product does not exist in the database or if the product is present but has no known allergens.

### 3.2.3 Android application

The mobile applications for Android will be written in a language very similar to Java. Each screen in an Android application is called an Activity and it is just a combination of a Java class and one layout file for element positioning. The main menu serves as the gateway for the various functions of the application.

When tapping a button, the corresponding Activity is started. Additionally, each time the application is started, a background service probes for the server's availability and if available, obtains a session token.

The specific implementation of this project relies on an external library for scanning and decoding barcodes, Zxing (2013) "zebra crossing". For the purposes of rapid prototyping the author chose to use the "Intent Integration" which relies on the "Barcode Scanner" application (available on Google Play) to be installed. Intent integration is a way to programmatically exchange information between different Android applications.

If the product search button is tapped, the *ProductSearch* Activity is invoked, which in turn will request scanning via Intents from the "Barcode Scanner" application. After the barcode string is obtained, the application then goes through the process already described in Figure 3.5.

The access token is the author's interpretation of how the password paradigm should be in the future. Since the whole system requires a smartphone, it makes no sense to force the user to memorize yet another password just to use this service.

Instead, every time the user wants to access the web interface, he/she should create an access token through the mobile interface. The application requests a specific WS that creates an access token for that specific user. The token is just a six character string of alphanumeric characters.

### **3.2.4 Web Interface**

The web interface uses technology from Ink (2012) - a fluid, responsive and simple framework for creating websites. It takes away all the complexity of creating websites prepared for all kinds of screen sizes and browser quirks.

## **3.3 Chapter conclusions**

Based on what was shown in this chapter, the prototype should be developed considering all the defined requisites. The database should also follow the proposed model, linking the tables with the WS and also the web interface. The Android application screens may be very similar to what was presented in the mockup or adapted for better usability.

# Chapter 4

## Solution Prototype

This chapter describes the technical details of prototype implementation.

### 4.0.1 Prototype design and architecture

The prototype architecture followed most all the steps that were planned. During database implementation and after analysing some of the most complex queries, it was noted that there were considerable improvements to the query by adding indexes.

The database was manually populated with real data from six products and all allergens properly documented. The *allergen\_class* table was loaded with the ten allergy categories and the *allergen* table was loaded with the allergens present in the selected products. All other tables were properly loaded with the relevant information. Although only six products are available at the this time, all the data is consistent.

The web services classes were implemented server side without any noticeable problems. The RESTapi class does validation proper input and processes the data accordingly. Proper return codes are returned in any case, either success or failure. Client side, there were some minor problems on figuring out how to properly decode JSON objects but it was just a matter of searching for examples on how to solve it.

The barcode scanning integration was implemented using the example code provided by the Zxing (2013) project. There were no changes or further need to adapt the integration routines.

As for the Android application not all planned modules were implemented. For now the *preferences* and the *scan history* screens are unavailable. The user account is created automatically when starting the application and is recorded both on the smartphone and in the database. User account deletion was not implemented. However, the user can remove the application or delete the application data.

### 4.0.2 User feedback

The comments received from the users that tested the prototype application were related to the poor variety of products available in the database. The implemented modules worked without problems requiring Internet connectivity on the smartphone.

The users have reported testing the application with the following smartphones:

- Huawei X5 (U8800)
- HTC One
- bq Aquaris

### 4.0.3 Development tools

The prototype was developed using Fedora Linux 19 x86\_64 on a Apple Macbook and a self-built AMD desktop computer. Eclipse Kepler was the Integrated Development Environment used with Android SDK updated to the currently (September of 2013) available API level 18. The devices used during development and testing were a Samsung Galaxy S Duos smartphone and a bq Edison 10" tablet.

The database was created with MySQL Workbench and the WS deployed on a shared web hosting server running Linux. Additional database queries and tests were done via the hosting server's phpMyAdmin web application. File transfer to the hosting server was done through a SSL WebDAV tunnel. WS testing was done via the Android application and also using a web based service provided by <http://hurl.it> .

# Chapter 5

## Conclusions

The work conducted while writing this thesis has contributed to a better knowledge of FA and also to understanding the current state of affairs concerning food product labelling and legislation. As main result, a prototype was successfully implemented that can demonstrate the proposed concept, achieving all three objectives.

Prototype implementation followed the proposed model with very few changes. Some of the functionalities in the Android application were not implemented and error checking and trapping was not implemented in depth. The database benefited from creating additional indexes for some tables as it improved performance in data access.

Future work is planned in all aspects of the system. The Android application will have all functionalities completed, the user interface be localized to Portuguese and polished for better usability. The database performance will be monitored so as to gather usage patterns and make the necessary improvements.

The author plans to implement some or all of the ideas on this list:

- Deploy application through Google Play for further testing
- The project's source code for all components will be available publicly during October of 2013. (GPLv2 license)
- Rethink the whole system to accommodate large scale usage.
- Perhaps change the database from the relational model to a key-value paradigm.



- Provide the same technology for iOS and Windows Phone devices.
- Maybe use hybrid mobile application solutions.
- Provide a local database on the smartphone for caching product and allergen combinations already scanned.

Final thoughts:

Starting such a big project like this without a properly curated database has risks. Perhaps a mixed model, starting from an existing product database and expanding it through the introduction of new fields with a human-driven classification and selection of possible allergens.

Later on, when critical mass has been achieved, the new paradigm of crowd-sourced database construction should be considered. Crowd-sourced database construction relies on data input from the end user. This input data is kept separate until data validation from a curator is done. After validation this data is merged into the database.

# Chapter 6

## Acronyms

<b>FA</b>	Food Allergies
<b>FI</b>	Food Intolerances
<b>SDK</b>	Software Development Kit
<b>SSL</b>	Secure Socket Layer
<b>WS</b>	Web Services
<b>WSDL</b>	Web Service Description Language



# Bibliography

- 2003/89/CE & Directiva (2003), ‘(Texto relevante para efeitos do EEE)’, (12), 15–18.
- Arens, A., Rösch, N., Feidert, F., Harpes, P., Herbst, R., Mösges, R. & Santec, H. T. (2008), ‘Mobile electronic patient diaries with barcode based food identification for the treatment of food allergies Barcode based electronic patient’, **4**(3), 1–5.
- Boyce, J. a., Assa’ad, A., Burks, a. W., Jones, S. M., Sampson, H. a., Wood, R. a., Plaut, M., Cooper, S. F., Fenton, M. J., Arshad, S. H., Bahna, S. L., Beck, L. a., Byrd-Bredbenner, C., Camargo, C. a., Eichenfield, L., Furuta, G. T., Hanifin, J. M., Jones, C., Kraft, M., Levy, B. D., Lieberman, P., Luccioli, S., McCall, K. M., Schneider, L. C., Simon, R. a., Simons, F. E. R., Teach, S. J., Yawn, B. P. & Schwaninger, J. M. (2010), *Guidelines for the diagnosis and management of food allergy in the United States: report of the NIAID-sponsored expert panel.*, Vol. 126, Elsevier Ltd.
- Cattell, R. (2010), ‘Scalable SQL and NoSQL Data Stores’, **39**(4).
- Chai, D. & Hock, F. (2005), ‘Locating and Decoding EAN-13 Barcodes from Images Captured by Digital Cameras’, (01010), 1556–1560.
- Goasduff, L. & Pettey, C. (2012), ‘Press Release’, **11**, 1–3.
- Ink (2012), ‘INK INterface Kit’.  
**URL:** *ink.sapo.pt*
- Julier, S. (2012), ‘Projects in VR’.
- Kant, K., Iyer, R., Corporation, I., Mohapatra, P. & Lansing, E. (1999), ‘Architectural Impact of Secure Socket Layer on Internet Servers’.
- Law, 108-282 AUG 2, . (2004), ‘Title ii—food allergen labeling and consumer protection’, pp. 905–911.

- Mills, E. N. C., Valovirta, E., Madsen, C., Taylor, S. L., Vieths, S., Anklam, E., Baumgartner, S., Koch, P., Crevel, R. W. R. & Frewer, L. (2004), 'Information provision for allergic consumers—where are we going with food allergen labelling?', *Allergy* **59**(12), 1262–8.
- Pautasso, C. (2008), 'REST vs. SOAP: Making the Right Architectural Decision'.
- Rona, R. J., Keil, T., Summers, C., Gislason, D., Zuidmeer, L., Sodergren, E., Sigurdardottir, S. T., Lindner, T., Goldhahn, K., Dahlstrom, J., McBride, D. & Madsen, C. (2007), 'The prevalence of food allergy: a meta-analysis.', *The Journal of allergy and clinical immunology* **120**(3), 638–46.
- Sampson, H. A. (2004), 'Update on food allergy', *Journal of Allergy and Clinical Immunology* **113**(5), 805—819.
- Sekar, A. K. (2012), 'Create a RESTful Services API in PHP'.  
**URL:** <http://www.9lessons.info/2012/05/create-restful-services-api-in-php.html>
- Wogalter, M. S., Forbes, R. M. & Barlow, T. (1993), 'Alternative Product Label Designs: Increasing the Surface Area and Print Size'.
- Wood, R. A. (2003), 'The Natural History of Food Allergy'.
- Zxing (2013), 'Zebra Crossing. Multi-format 1d/2d barcode image processing library with clients for android, java. Website'.  
**URL:** <http://code.google.com/p/zxing/>