Smart Fridge - G02

Ambient Intelligence - MEIC - Instituto Superior Técnico

João Daniel Silva - 86445

Ricardo Velhinho - 86505

Contents

List of Figures

Abstract

In this day and age, technology keeps on growing exponentially and, thus, so does the many areas it is applied to. Mainly, ambient intelligence is becoming more and more a focus point of our lives, especially because they are starting to surround us everywhere, we go and in everything we do. Our approach to ambient intelligence was to create a simulation in which the user could do most of the boring and time-wasting tasks related to food being kept in the fridge. For this, we applied many methodologies related to ambient intelligence in order to keep the main principles intact in what concerns being able to integrate in the surrounding environment, interoperating with other devices, interact with the user in many ways, learning from the users' choices and so on. From this simulation. we were able to successfully develop a smart fridge with the main functionalities required to be properly designated as an solid member of the ambient intelligence family and we concluded that, even with the overflowing exploration and development of new ideas in the area, there is always a way to keep on creating and implementing new innovative ideas if the right team with the right idea is behind it.

Keywords

Ambient; Intelligence; Interoperability; Environment; Integrated; Smart; Fridge

1 Introduction

1.1 Motivation

A great deal of our lives revolves around our food habits and it's inevitable since it a source of nutrients and substances required for our organism, health and well-being. However, with the constant evolution and diversification of ingredients and recipes, a great amount of effort comes with being able to choose the right recipes and food items for our goals, whether it's being healthy while having a diversified diet, following a strict diet to lose weight or even for more specific purposes, such as gaining muscle or toning the body. Whatever the purpose may be, it takes a great toll on our mind always having to keep track of what we are eating, what we should eat, where to get the ingredients necessary and then on top of that, learning to cook the recipe. All of this can easily become avoidable using something we've grown accustomed to implement in most areas of our lives, technology. Specifically, together with Ambient Intelligence and it's principles, we can come up with a smart fridge that can solve all these issues and even provide more solutions for problems we weren't aware of and, ultimately, make our lives easier when it comes to food and health.

1.2 Scope and Objectives

With the implementation of the smart fridge, we intend not only to get rid of common issues such as keeping track of what ingredients are in the fridge and in what amount through cameras and sensors which will detect whenever an ingredient in added or removed from the fridge, select personalized recipes recommended by the smart fridge which will complement our goal, having the system to be able to uphold many user profiles, allowing for the many users of a household to be able to have their own personalized interface according to their preferences and all of this with room for improvement and interoperability with other devices.

Having all this in mind, we propose a possible interface for a smart fridge that could help people manage food more efficiently, that allows to manage the normal functionalities of a fridge as well as many more and keep track of food availability, all personalized to the users' needs.

2 Related work

2.1 React

To build this interface we decided to stick with the main 3 pillars of front end. Specifically, HTML5, CSS3 and JavaScript. However, since there are libraries and frameworks that can further improve our work, we decided to use React because it is a popular, lightweight and easy to use JavaScript library which allows to improve the performance of web apps through the use of a tree of components, which adds a hierarchical structure to the app and allows for principles such as component isolation, abstraction and reusability and, thus, gives our app a better performance. Besides all of this, React was also something we were somewhat experienced with which influenced our selection as well. It's also an easier and minimalistic way of developing our app as a whole and component by component. The main reason we decided to go for this approach is because the smart fridge we want to develop is very behavior driven and requires a lot of interaction with the users. Therefore, even though there could be some other alternatives when it comes to developing a simulation for a smart fridge such as choosing to do a desktop app or a mobile app, our go to solution was the creation of a web app, which is usually the most common approach people interact with on a daily basis and are used to. Therefore, the React library seemed like the best way of developing a web app which could handle all the requirements with ease and make the app have as smooth interactivity as possible.

2.2 Auxiliary Libraries

Together with React, we also used some other helper libraries, such as react-router-dom in order to allow different urls for different screens of the smart fridge as well as improving the performance by giving each screen their own specific component. We also made use of lodash, which is a library intended to ease the manipulation of arrays, which we needed quite a bit in order to display the information on each screen. Last but not least we also used a User Interface framework known as Bootstrap which handled the styling and structure of our app through class names keywords implemented in the jsx elements, also improving the performance by making the code simple and free of noise/polluting code.

2.3 Hardware

We did not develop the project to run in specific hardware such as Arduino or Raspberry Pi since we did not have prior experience with such technologies and due time constraints, we kept our solution as simulation based, which was be a very trustworthy approach as well. So, this simulation is intended to be executed on the PC with auxiliary of a localhost server and installation of NPM (Node Package Manager). NPM allows to install many third party packages such as React itself and all the other helper libraries we used in order to develop the smart fridge simulation.

Overall, we opted for the development of mainly frontend while simulating a backend since we wanted to showcase the functionalities we can offer through our app.

3 Description of the solution

3.1 Architecture of the solution

Since we are developing an interface through the React library, our architecture is based on a hierarchy, which means that there is a tree of components, allowing to solidify a well defined and organized structure for our app and ease the development and maintenance of the code involved.

Being a tree of components, we can then divide these into specific components on the lower levels such as reusable bubbles containing information, middle level components such as the parts of the screen into which we want to split a whole screen such as for example having the main screen after unlocking the smart fridge consist of the left side, which is all about what the smart fridge intends to show and interact with the user, and the right side which, in our implementation, is considered a simulation area where we can accomplish to show the user how the smart fridge would behave in case of behavior not currently possible due to technology constraints such as having sensors and cameras available to use. Furthermore, there are components belonging to the higher levels, and these are considered to be the screens themselves as a whole, which wrap and contain all the other components of the lower levels.

3.2 Components overview

The main component of our app corresponds to the main screen containing all of the fridge information as well as bubbles which allow to execute other functionalities such as swapping the user profile, being able to favorite a specific recommended recipe and change the preferences/settings of the app itself. The interactivity between components comes from having a single source of truth where we initially, when starting the simulation, as the component is being created, retrieve all the required information about the users, their recipes, what preferences should be available to change and what ingredients are currently in the fridge from a fake backend which we created specifically intended with the purpose of simulating our application without needing to access a database or handle specific tasks such as http requests to the web app. Therefore, after starting the simulation, we all the information required and, so, all we need in order to be able to communicate between the different screens and components is have a single source of truth at the top of the tree of components, which allows us to pass the required information from parent components to child components successively. Because of this, we can also interact and manipulate the behavior through handlers which also occur at the top of the tree but are triggered wherever the user clicks, allowing us to dynamically interact with the information from any component we wish too and the changes will propagate to the rest of the app.

4 Details of the solution

4.1 Implementation details

4.1.1 Main Page

The MainPage.jsx component consists of the main screen being used whenever the smart fridge is unlocked by a user and, therefore, contains both the Fridge.jsx component as well as the Simulation.jsx component. This component acts both as a wrapper as well as a medium to transmit information from and to the children components.

4.1.2 Fridge

The Fridge.jsx component can be considered the one with the most code implementation since it is where most of the functionality will occur. This component will take the information provided by the parent components and, through array manipulation and use of lower level components, display such information in a way that allows the user to interact with it. It starts off by splitting the component into two.

The first part consists of a layout similar to a fridge skeleton, which shows the most common properties users are usually interested when checking the fridge, such as the fridge and freezer temperatures as well as what ingredients are available in the fridge and in what quantity. The way it is implemented allows for the user to manually change the temperatures up or down, according to what they may prefer, and also allows, if there are more than a certain amount of different ingredients on the fridge, to scroll through the inventory through the use of arrow icons. Besides this, there is also a pagination icon which is also attached to how many pages of the inventory is required. All of the required calculations in order to display such information is done before rendering the display, allowing for the screens to change whenever there is behavior.

The second part of the screen consists of three main rows. The first one allows to check the current user and swap the user profile for a different one, since we assume there are different people in the house which want to be able to have their own personalized app. The second row allows to check the recommended recipes by the app and also allows the user to interact with it by favoriting (clicking the star icon).

The third and final row allows the user to change the preferences/settings of the system and personalize however they may desire. For example, one can change if they prefer the system to automatically send a notification to their smartphone whenever a certain ingredient is running low.

4.1.3 Simulator

The Simulator.jsx component consists of the abstract side of the representation which the user usually doesn’t see but is a part of our app in order to showcase our functionalities to the audience through a simulation. It contains lower level components embedded in the icons represented in order to allow for manipulation of information through handlers by simulating common tasks of ambient intelligence such as voice commands and motion.

4.2 User Interface

As of the time writing the report, the full interface is not yet concluded. However, the main page where most of the interaction occurs is finished and, so, we can check how the user will be able to see and interact with the interface while using the main functionalities.

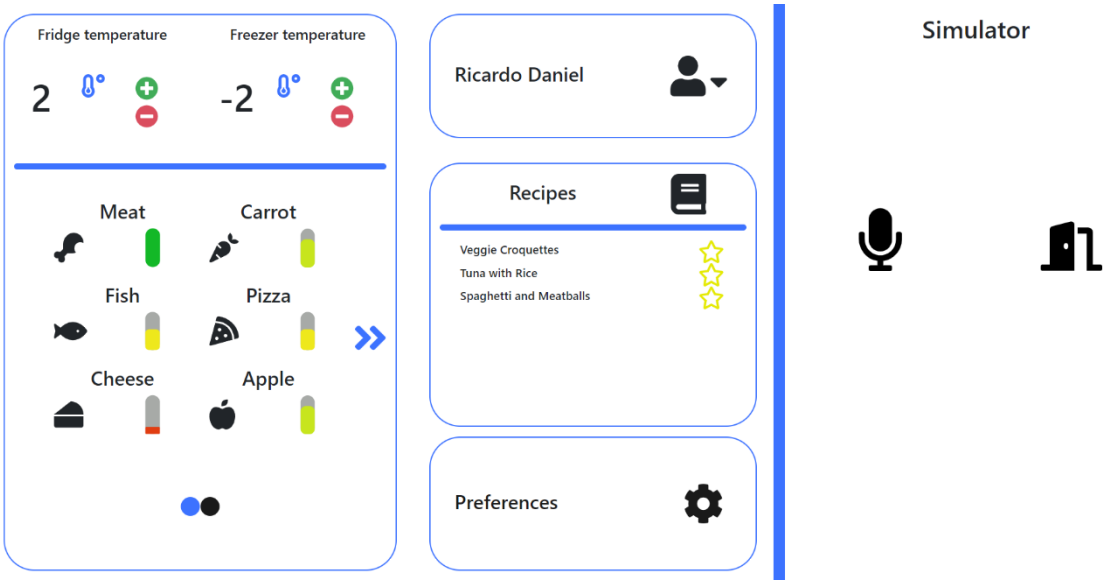

Figure 1 – Main Screen of the Smart Fridge App

In this main screen, we can interact with it by changing the temperatures, sliding the inventory tab to the sides to see the full range of ingredients, select the preferred recommended recipes and be able to swap user profiles as seen in the figure below.

Figure 2 – User Profile selection

Besides this, there is also a way for the user to change his preferences/settings by clicking the preferences bubble in the lower-right corner.

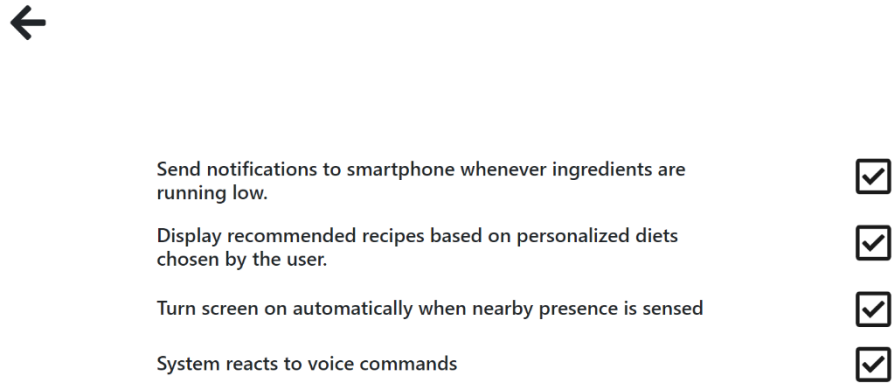

Figure 3 – User Preferences screen

4.3 Examples of use

4.3.1 Changing temperatures

Starting from the most basic functionalities offered by the app in order to allow the user to manipulate and control however they feel most comfortable with, we allow the user to manipulate the temperatures set for both the fridge and the freezer. First of all, a user only needs to unlock the app and, in the main screen, there is a large fridge resembling bubble on the left which was designed with the purpose of making the user quickly relate that part of the app to the fridge itself and it's contents. Inside this bubble, on the top, a user can check the temperatures and manipulate them by pressing either the plus or minus icon. Having a particular example in mind, a user can change the fridge temperature from 2 to -2 degrees and the freezer temperature from -2 to 0 degrees if he wants to by simply pressing the icons according to their desires.

4.3.2 Checking, inserting and removing ingredients

Since this is a fridge app, the most important and most valued aspect comes from being able to quickly tell what ingredients are in the fridge and in what quantity. Besides, the fridge should also be able to quickly update whenever a new ingredient is inserted or removed. So, to check the inventory, all a user needs to do is unlock the screen and the first thing that pops ups in great effect is the fridge inventory. In order to insert or remove ingredients, the user opens the fridge door and does what he needs to. During the time that the door is open, the app will swap the screen for a non-interactable screen which will warn the user that the fridge is open. After the door closes, then the screen will return to the normal main screen and the contents will be updated.

4.3.3 Changing user profiles

Whenever a person comes close to the fridge and the embedded fridge app notices this presence through either cameras or sensors, it can automatically login the person to their own profile. Besides this, there is also a way to change user profiles while the app is unlocked. We implemented this in order to offer more flexibility and control of the app. In particular, if John is currently logged in but Mary steps in side by side to John and the fridge app detects her presence, then, by simply selecting the current user bubble on the main screen, they are able to swap user profiles from John to Mary. Note that by changing user profiles then the recommended recipes as well as the preferences change accordingly.

4.3.4 Favoriting a recommended recipe

Since each user can have their own favorite recommended meals from the app, a user can simply click the favorite icon which is represented by a star or by using voice commands.

4.3.5 Selecting the preferences

Just like the favorite recipes, users can also select their own preferences by clicking the preferences bubble and choosing what they prefer. This can be done by touch or by using voice commands.

4.3.6 App notifying smartphone

Because interoperability is a great aspect of ambient intelligence, we decided to incorporate it into our smart fridge by allowing the user to connect any smartphone to their account. After such action, the user can select, in the preferences tab, if we wishes his smartphone to be notified in case of an ingredient running low. If the particular case when this preference is selected, then whenever someone in the house removes a particular ingredient, then the user with such preference will be notified in the following moments.

5 Evaluation

5.1.1 Evaluation methods

In order to properly evaluate and critique our project and determine whether or not our functionalities were well developed, flexible and allowed for exploration and easy execution from the user we came up with many possible test scenarios so that we could fully test the quality and flexibility of our functionalities and app as a whole. Some of these have been previously mentioned in the last section and others were tested to allow for exceptions, specific cases and fix any issues around these.

An interface-based project like this one should have been tested with potential users, but because of the pandemic situation we are currently living in, such approach was not possible, so we had to resort to methods and analysis by usability inspection. We decided to use a Heuristic Evaluation by Nielsen which are probably the most used usability heuristics for user interface design and evaluate the design of the application as a whole.

Also, we try to check the loading and rendering time of the application. According to some studies, 53% of the people quit trying to access a website if it takes more than 3 seconds (VI). Despite being executed in a localhost, we want the performance of our application to be good so it can help the users in theirs tasks otherwise it will only add frustration and make it tiresome.

5.1.2 Results analysis

After testing with our diverse test scenarios, we were able to come up with positive results, both in performance speed and robustness since the app allowed us to perform all tests without incoherence or bugs/issues attached. This means that the current developed functionalities are all working as intended and have room for improvement and new functionalities as well.

Since it was developed using the React library, it runs smooth and quickly without need of page reloading every time there is interaction with the user, allowing for hot module reloading.

Analyzing according Nielsen's Heuristics:

1. Visibility of system status:

   The system keeps the users informed about what is going on, by displaying the food quantity, the preferences chosen and the recipes selected; in the simulator there is a helper so one can insert/remove the exact amount wanted.

2. Match between system and the real world:

The system speaks the user's language: the icons represent concepts already known by the user, such as the foods and return icons, the simulator icons, the pagination o the fridge and the changing between users is similar to other widely known applications.

3. User control and freedom:

Users can decide functionality of the application anytime, such as preferences or recipes.

4. Consistency and standards:

The application follows consistency by using the same icons and components have the same behavior across screens.

5. Error prevention:

In this system, there are not conflicting behaviors of different users that can generate errors. Through our test case scenarios, we fixed bugs as were implementing the application.

6. Recognition rather than recall:

The icons and text provide better context for users to successfully complete their tasks. However, voice commands must be learned by the user.

7. Flexibility and efficiency of use:

In this application, due to its simplicity, we didn't find a need to different types of users do the same tasks in different ways.

8. Aesthetic and minimalist design:

The interface is simple and clean, and it was avoided to add items that would not have significance or help the users to accomplish their tasks.

9. Help users recognize, diagnose, and recover from errors:

The same argument as #5.

10. Help and documentation:

Due to its simplicity, there is no documentation (other than this report). However, if it would be expanded, it could be considered.

Despite all this, the results shown from the test scenarios did not cover all the intended functionalities and behavior the smart fridge could have in a real-world scenario. All the developments were mainly for the concern of the project and allowed to properly demonstrate its capability and potential.

6 Conclusion

6.1 Global balance

In this project regarding ambient intelligence, we went for an approach heavily related to our homes, specifically our fridge environment since it is something we have in high regard and use everyday for our personal needs.

Through the development of the fridge smart app we were able to properly demonstrate and cover the main aspects that should be taken into account when developing functionalities which allow for interoperability, self-learning ability, minimalistic design and interaction associated with it, whether it's through touch, voice, sensors or even cameras.

By developing this smart fridge, besides improving our knowledge and ability to further understand the concepts underneath ambient intelligence, also allowed us to improve our own skills in what comes to being able to problem-solve, work as a team and most importantly, discover new technologies and be able to explore and learn more about them.

6.2 What is missing

Our main goal while developing this project was to fully demonstrate the functionalities and capabilities present in an embedded smart fridge right in our homes and take full advantage of that situation to further improve it. However, due to time constraints, some details and functionalities weren't developed as we wanted from the beginning. For example, some aspects such as login verification with inputs, a fully developed back-end which handled https requests and attached to a database. Besides these more technical details, some functionalities such as allowing a user to adapt and change his goal in what concerns his diet in a personalized manner, having a history of already cooked meals as well as having detailed information about the ingredients' properties were also a part of our initial plan, but since they didn't have as major of an impact like the ones we implemented, their priority was on a lower level.

6.3 Final Thoughts

Overall, the project development went well and as expected even if some aspects weren't accomplished due to time constraints. Since we developed the app using technologies such as the React library, our performance was very successful and the interactions were all very smooth and easy to execute, while also focusing on important factors such as user experience and interaction.

Since nothing ever is its best self, there is always room for improvement. In our specific case, we would prioritize the development of the missing functionalities we couldn't achieve due to time constraints but also come up with an even more design friendly approach, since the main idea of the smart fridge is for it up be a part of the skeleton of the fridge, like an incorporated screen in the big fridge door.

7 References

I. W., Weber, J., Rabaey & E., Aarts. 2005. Ambient Intelligence.

II. José, Rouillard. 2012. The Pervasive Fridge A Smart Computer System Against Uneaten Food Loss. Available at: https://www.researchgate.net/publication/229071826_The_Pervasive_Fridge_A_Smart_Computer_System_Against_Uneaten_Food_Loss.

III. Murali, N. G. 2017, IoT Based Interactive Smart Refrigerator. Available at: https://www.researchgate.net/publication/322653199_IoT_Based_Interactive_Smart_Refrigerator.

IV. Michael, F., Olivier, D.C., Yves, P. & Petteri, A. 2005. Perspectives of ambient intelligence in home environment. Available at: https://www.researchgate.net/publication/222681195_Perspectives_of_ambient_intelligence_in_home_environment.

V. https://www.nngroup.com/articles/ten-usability-heuristics/, Nielsen Norman Group UX consulting firm webpage, accessed 14th May 2020.

VI. https://www.hobo-web.co.uk/your-website-design-should-load-in-4-seconds/, Hobo-Web SEO consulting firm webpage, accessed 14th May 2020.