**SOFTWARE REQUIREMENTS SPECIFICATION**

****

**Members:**

**Mohamed Almaazmi (20170850)**

**Nabila Sindi Wulandari (20180744)**

**Chrysan Angela Piarso (20180949)**

**KAIST School of Computing CS457**

**Spring 2020**

**Table of Contents**

[**1. Introduction**](#_ksq1jqn2tuz9) **4**

[1.1. Purpose](#_j30i0hdv5v9) 4

[1.2. Scope](#_qni7fpr5q390) 4

[1.3. References](#_1ouhnm4h3ukb) 5

[1.4. Definitions, Acronyms, and Abbreviations](#_m5wpxge2zh1r) 5

[1.5. Constraints](#_dtnh7ilfhlbw) 5

[1.6. Assumptions and Dependencies](#_1j98qkhl50w4) 6

[**2. Use-Case Model Survey**](#_tjar2f479xrd) **6**

[2.1. Domain Model](#_miz3ke61dlpf) 6

[2.2. Use Case Model](#_yekdyrg0d7as) 7

[2.3. Use Case Descriptions](#_wktvyddfj71t) 7

[**3. Actor Survey**](#_ajp0f6h9doof) **11**

[3.1. Home owners](#_8y2fm46jtinb) 11

[3.2. Supermarket owners](#_fkx0yyjzd12z) 11

[3.3. IoT developers](#_j8vec7z1441a) 11

[3.4. Smart devices controlled system](#_n4dzqcc44j76) 11

[3.5. Sensor-based data collection system](#_d095itxq89nw) 12

[3.6. Inventory management system](#_y7n8kpotolmj) 12

[3.7. Authentication](#_5bmqtrtolp69) 12

[**4. Contextual Design**](#_29y6yv3gy3u7) **12**

[4.1. Flow Model](#_8l6hx3of5p0i) 12

[4.2. Sequence Model](#_kw55ugpww0px) 13

[4.3. Artifact Model](#_7gx0tc9gegvr) 13

[4.4. Cultural Model](#_vgw8tqsnbev7) 14

[4.5. Physical Model](#_u6fuutj6reep) 15

[4.6. Consolidated Flow Model](#_lbaxfiejk0l4) 16

[4.7. Consolidated Sequence Model](#_f3bd3n2fqckp) 17

[4.8. Consolidated Artifact Model](#_1shk0adkqlkp) 17

[4.9. Consolidated Cultural Model](#_wvejl9do0dh2) 18

[4.10. Consolidated Physical Model](#_m2lf1dhvtahj) 19

[4.11. Storyboards](#_mvj2j4sh1pn6) 19

[4.11.1. Storyboard & Scenario 1](#_p46qm036pl2h) 19

[4.11.2. Storyboard & Scenario 2](#_elya3sbppiuy) 20

[4.11.3. Storyboard & Scenario 3](#_nqp7zw2x66pn) 21

[4.12. Vision Diagram](#_l3safxi0natw) 22

[4.13. User Environment Design](#_pzm5ce37wlzx) 23

[**5. Requirements**](#_kjgporafnhk8) **23**

[5.1. Functional Requirements](#_xv4ntmg39669) 23

[5.2. Nonfunctional Requirements](#_3s2dq7ta71b) 23

[5.2.1. Usability](#_819jk043qq51) 24

[5.2.2. Internationalization](#_2a4op62s8feg) 25

[5.2.3. Privacy](#_aouxq32rmab) 26

[5.2.4. Reliability](#_i4qomedyhtk0) 27

[5.2.5. Performance](#_en3wd44x0i0r) 27

[5.2.6. Safety](#_b1hw0quoma51) 28

[5.2.7. Maintainability](#_fx8f3di5eozl) 28

[5.2.8. Portability](#_jyqp4ihgtmip) 28

[5.2.9. Scalability](#_rnp600myxirw) 29

[5.2.10. Security](#_z5c523j5fw6) 29

[5.2.11. Supportability](#_u2xhxq1fo87f) 29

**6**[**. Interfaces**](#_zgoulyuuz9pj) **30**

6[.1. User Interface](#_6bna2ljdc8rv) 30

6[.2. Software Interfaces](#_e79bswkgdaxl) 31

6[.3. Communications Interfaces](#_ny1v87xw5wn2) 31

# 

# 1. Introduction

This software requirements document specification provides complete information about a system called TTYK (Talk to Your Kitchen) which will be developed by our project team, Team Perry. The system is visioned as a smart device and sensor integrated kitchen system. The purpose of this document is to present a detailed description of the TTYK system. It will explain the purpose and features of the system, the interfaces the system will employ, what the system will do and the constraints under which it must operate. This document is intended for users and developers of our system.

## 1.1. Purpose

According to the NFPA (National Fire Protection Association), 173,300 home structure fires are started per year by cooking activities in 2013 - 2017, leading to an average of 550 deaths, 5,020 injuries, and 1.2$ billion in property damage per year. Unattended cooking was the leading cause of the fire. Another problem worth mentioning is food waste. Food waste happens when edible food is discarded after consumers fail to consume them before it goes past its expiry date. Food waste also happens due to oversupply in markets. They end up in landfills and produce a tremendous amount of methane, a greenhouse gas more powerful than CO2, fastening the rate of global warming and climate change.

Food waste also takes a toll on the environment due to its agricultural preparation, which accounts for 70% of water used throughout the world. That would mean throwing 1 kg of beef is equal to wasting 50,000 of water used to produce the meat. Another comparison would be 1000 liters of water wasted when one glass of milk is poured down the drain. The 1.4 billion hectares of land used to grow the food is also wasted, creating negative impacts in biodiversity and wasting millions of gallons of wasted oil. Our team will attempt to tackle these problems.

The purpose of this SRS document is to provide the needed guidelines in order to design and implement the software that fulfills all the given requirements. In an event that the current team cannot finish the software, a future team can use this document to build upon the already created system or to create another version of the software that follows the requirements given. This document represents the requirements for the software that can be used by clients and stakeholders in an event of a contract.

## 1.2. Scope

TTYK is a smart & holistic kitchen system which helps kitchen users more efficiently make use of their kitchen and reduce the waste that comes out of it. The system will employ a number of sensors to record in real-time the kitchen conditions such as what devices are on, what items are currently in the kitchen, the temperature in the kitchen, smoke level, etc. This information will then be used to control the kitchen conditions by allowing the user to control the devices based on the real-time data measured. The application will have both a mobile and a desktop platform allowing users to interact with the system through either one. In addition, nearby supermarket owners can provide their stock information using a supermarket owner portal. This information will be available to users who can use it to more efficiently manage their kitchen stock. IoT developers will also be able to access user data, from a special IoT developer dashboard, on users’ smart devices to further improve their products. Furthermore, the system will need an internet connection at all times to function properly. All system information is maintained in a database, which is located on a web-server. The system will be designed so that users can easily add/remove/modify the devices and items in their kitchen after system set-up. The application also has the capability of representing both summary and historical information about the user’s kitchen allowing for even better efficiency.

## 1.3. Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| **Term** | **Definition** |
| User | A person or organization who uses our application |
| System | A group of hardware units and software programs working together to achieve an objective |
| Environment | The overall setting in which the system and user interact with one another |
| Device | Gadgets used for a certain purpose |
| Requirement | A condition or capability that must be met in order to solve a problem or fulfill a certain objective |
| Constraint | A limitation or restriction that prevents the system from having a certain property |

## 1.4. Constraints

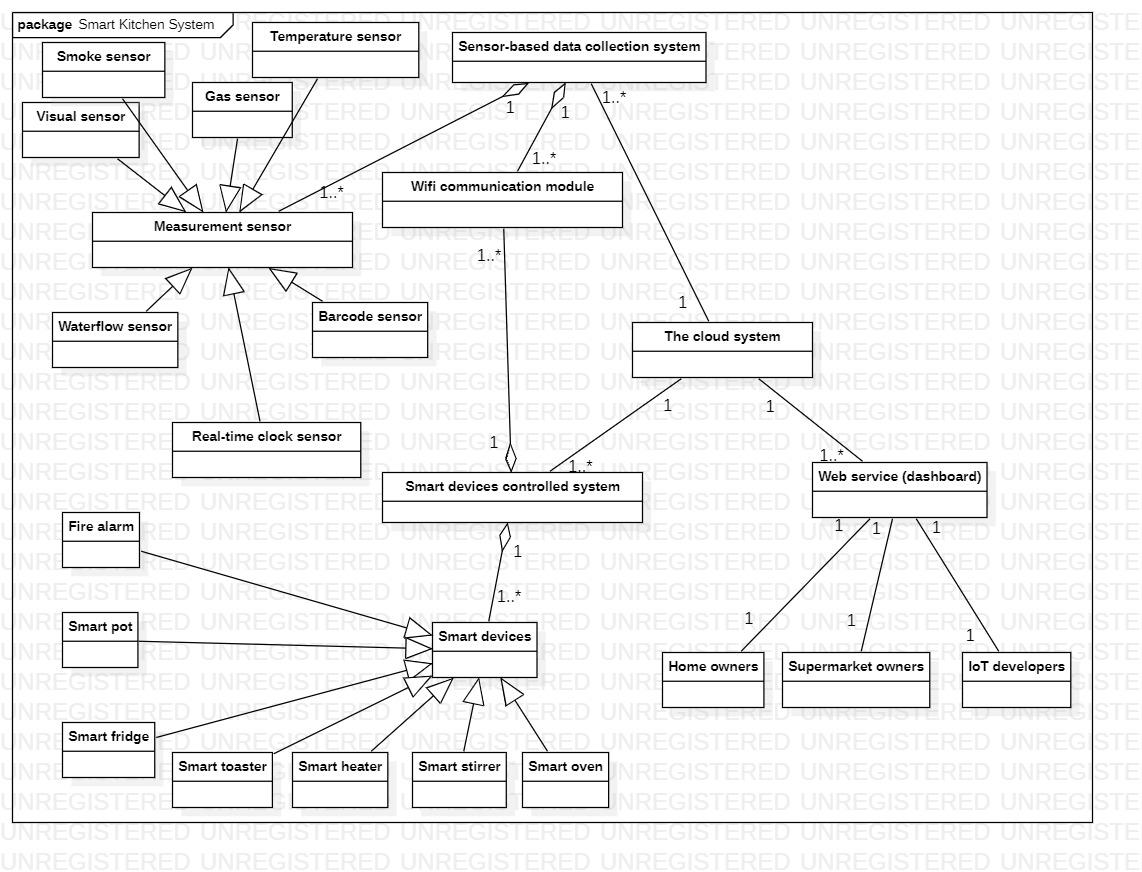
* Internet connection sets a constraint for the application. Since the application fetches data from the database over the Internet, it is crucial that there is an Internet connection for the application to function.
* Both the web portal and the application will be constrained by the database used. Since the database is external, the functioning of the database is wholly dependent on a thrid party vendor. In addition, it may be limited by capacity; it may be forced to queue incoming requests and therefore increase the time it takes to fetch data.
* The smart devices are a constraint because their malfunctioning or their inefficiency will either completely disrupt or slow down the system.
* Sensors are another constraint because all the information for the system will come from them, and their malfunctioning will mess up the data-centric nature of the system.
* The electrical connections are also a constraint since the system may be disrupted by blackouts and other electricity losses.
* Lack of technological infrastructure in certain places may prevent important features, such as supermarket owners displaying their stock, from working.
* Regulation in certain countries may also make it impossible to send user data to IoT developers for feedback and product improvements.

## 1.5. Assumptions and Dependencies

The system will be compatible with Windows, MacOS, and Linux. The web components of the system will be compatible with every web browser. The smart devices and sensors are regularly checked and kept up to date. They are also always plugged in. All components of the system are connected to the Internet at all times. The external stock management systems supermarket owners use are compatible with the TTYK system.

# 2. Use-Case Model Survey

## 2.1. Domain Model



## **2.2. Use Case Model**

## 2.3. Use Case Descriptions

|  |  |
| --- | --- |
| **Use case** | **Description** |
| Control kitchen conditions | User can control the conditions in the kitchen by setting thresholds for selected metrics, and by sending instructions to smart devices connected to the system |
| View supermarket data | User can view the offers and stock of nearby supermarkets |
| Plan grocery list | User can plan their grocery list using various information available from their dashboard such as available items in the kitchen and nearby offers & stock of supermarkets |
| View kitchen data | User can view real time data about their kitchens such as available ingredients, consumption rates, status of devices, and kitchen conditions. |
| Post stock data | Supermarket owners post information about their stock on the system for home owners to view. |
| View home owners’ items statistics | Supermarket owners can view data about items home owners have in their kitchens as well as their history of item purchases |
| Send warning to home owners | Sensor-based data collection system checks collected data against set thresholds and sends a warning if thresholds are exceeded |

**2.3.1. View kitchen data**

|  |  |
| --- | --- |
| Name | View the kitchen data |
| Summary | Home owners can view real time data about their kitchens such as available ingredients, consumption rates, status of devices, and kitchen conditions. |
| Actor | Home Owners |
| Preconditions | * User is validated * User is connected to Wi-Fi * Sensor-based data collection system is functional |
| Description | 1. Users open their dashboard and choose the view real time data menu. 2. Users can choose out of the several viewable data, such as status of device, ingredients, consumpution rates, or kitchen conditions. |
| Alternative Scenarios | Wi-fi connection lost: At any time, system requests User to check their connection to the Wi-fi and restart the system. |
| Postconditions | Users are able to view the desired data. |

**2.3.2. Send instructions to devices**

|  |  |
| --- | --- |
| Name | Send instructions to devices |
| Summary | Home owners can view data regarding their kitchen devices, such as their energy consumption or status of the device, and send some instruction to the device, such as turning it on/off, reducing its energy usage, increasing the time (such as cooking time) etc. |
| Actor | * Home Owners * Smart devices controlled system |
| Preconditions | * User is validated * User is connected to Wi-Fi * Smart devices controlled system is functional |
| Description | 1. User opens their dashboard and chooses the control kitchen conditions menu. 2. Users get a view of their kitchen devices and their status. Users can choose an device to view their data. 3. Users get a list of the available instructions for the selected 4. Instruction is sent to the device which then executes it |
| Alternative Scenarios | Wi-fi connection lost: At any time, system requests user to check their connection to the Wi-fi and restart the system. |
| Postconditions | The user’s selected device executes the desired instruction the user chose |

**2.3.3. Plan grocery list**

|  |  |
| --- | --- |
| Name | Plan grocery list |
| Summary | Home owners can plan their grocery list using various information available from their dashboard |
| Actor | * Home Owners |
| Preconditions | * User is validated * User is connected to Wi-Fi * Sensor-based data collection system is functional |
| Description | 1. User uses his dashboard to view list of available items in the kitchen 2. User searches dashboard for items he needs and their prices based on posted supermarket offers 3. User uses this data to plan his grocery list accordingly |
| Alternative Scenarios | Real time list of supermarket offers is not available: System sends notification to supermarket owners informing them. |
| Postconditions | User’s creates a grocery list to use for when they go grocery shopping |

**2.3.4. Send warning to home owners**

|  |  |
| --- | --- |
| Name | Send warning if some metric exceeded the preset threshold |
| Summary | Sensor-based data collection system checks collected data against set thresholds and sends a warning if thresholds are exceeded |
| Actor | * Sensor-based data collection system * Home owners |
| Preconditions | * Sensor-based data collection system is connected to Wi-Fi * Sensor-based data collection system is functional |
| Description | 1. Sensor-based data collection collects real-time data on a number of metrics 2. System checks this data against existing thresholds 3. If threshold is exceeded, a warning is sent to the appropriate parties |
| Alternative Scenarios | Data exceeds thresholds and reaches dangerous levels: System automatically notifies authorities about this emergency |
| Postconditions | Metrics are kept within appropriate levels, and users have a quick response system available in case metrics exceed thresholds. |

**2.3.5. Post stock data**

|  |  |
| --- | --- |
| Name | Post data about stock |
| Summary | Supermarket owners post information about their stock on the system for home owners to view. |
| Actor | * Supermarket Owners * Inventory management system |
| Preconditions | * Users are validated * Users are connected to Wi-Fi * Inventory management system is functional |
| Description | 1. Users open their dashboard and choose to post information on new stock 2. Users connect their inventory management systems to the smart kitchen system |
| Alternative Scenarios | Inventory Management system could not be connected: System sends an error message to the supermarket owners informing them of the error (if known) and requesting they try reconnecting and restarting the system. |
| Postconditions | Stock data for supermarket owners is posted on the smart kitchen system and is available for home owners to view |

**2.3.6. View home owners’ items statistics**

|  |  |
| --- | --- |
| Name | View statistics about items homeowners buy |
| Summary | Supermarket owners can view data about items home owners have in their kitchens |
| Actor | * Supermarket Owners * Inventory management system |
| Preconditions | * User is validated * User is connected to Wi-Fi |
| Description | 1. User opens their dashboard and chooses to view homeowner statistics 2. Users can filter the representation to see the entries they are interested in |
| Alternative Scenarios | Wi-fi connection lost: At any time, system requests User to check their connection to the Wi-fi and restart the system.  User statistics not available: An error message is sent to the supermarket owners, and a notification is sent to the technician of the system informing them of a possible malfunction. |
| Postconditions | A visual representation of home owner statistics with filters applied is shown to the user |

# 3. Actor Survey

## 3.1. Home owners

Home owners are the main actors of the system. They install the TTYK system in their kitchens. They use the system to increase the efficiency and reduce the waste of their kitchens. They do this by using the system to better manage their kitchen stock of items and their devices by setting automated instruction routines and operations. In addition, they can use the system to plan and optimize their grocery shopping by having a holistic of the items in their kitchens and of the offers from nearby supermarkets. They interact with the system through both mobile and desktop platforms based on the web. They interact with the system through a web dashboard for home owners.

## 3.2. Supermarket owners

Supermarket owners interact with the system by posting a snapshot of their inventory which is update in real-time for home owners nearby to view. In addition, they can view the purchases and current items in the kitchens of nearby home owners and can use it to better manage their stock. They interact with the system through a web dashboard for supermarket owners.

## 3.3. IoT developers

IoT developers are people who develop the smart devices used in the system. They can use the system to view the statistics of the smart devices home owners use. They can then use this information to further improve the devices. They interact with the system through a web dashboard for IoT developers.

## 3.4. Smart devices controlled system

The subsystem which handles the smart devices of the system. It interacts with the main system by sending real-time information on the status of the smart devices connected. In addition, it receives instructions from the main system and sends them to the devices to be executed.

## 3.5. Sensor-based data collection system

The subsystem which handles the data collection of the main system. It consists of a number of different sensors which measure in real-time the current conditions and metrics in the kitchen such as humidity, temperature, etc. The system then sends this information to the main system for further processing and use in decision making.

## 3.6. Inventory management system

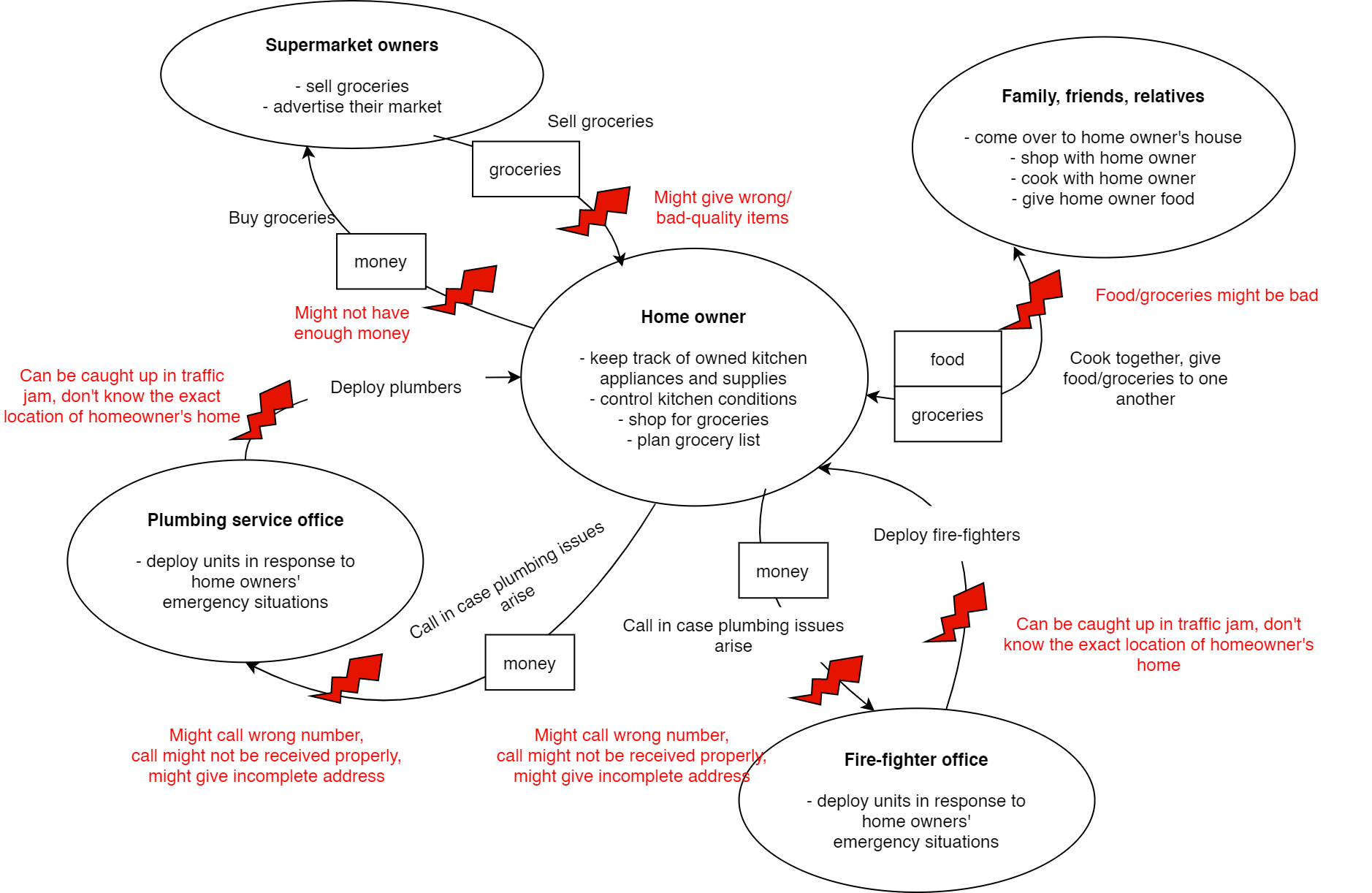
The external inventory management system supermarket owners use to keep track of and manage their stock. This system connects to the TTYK to give real-time data on the stock of the supermarket. This information can then be used by home owners in their decision making processes.

## 3.7. Authentication

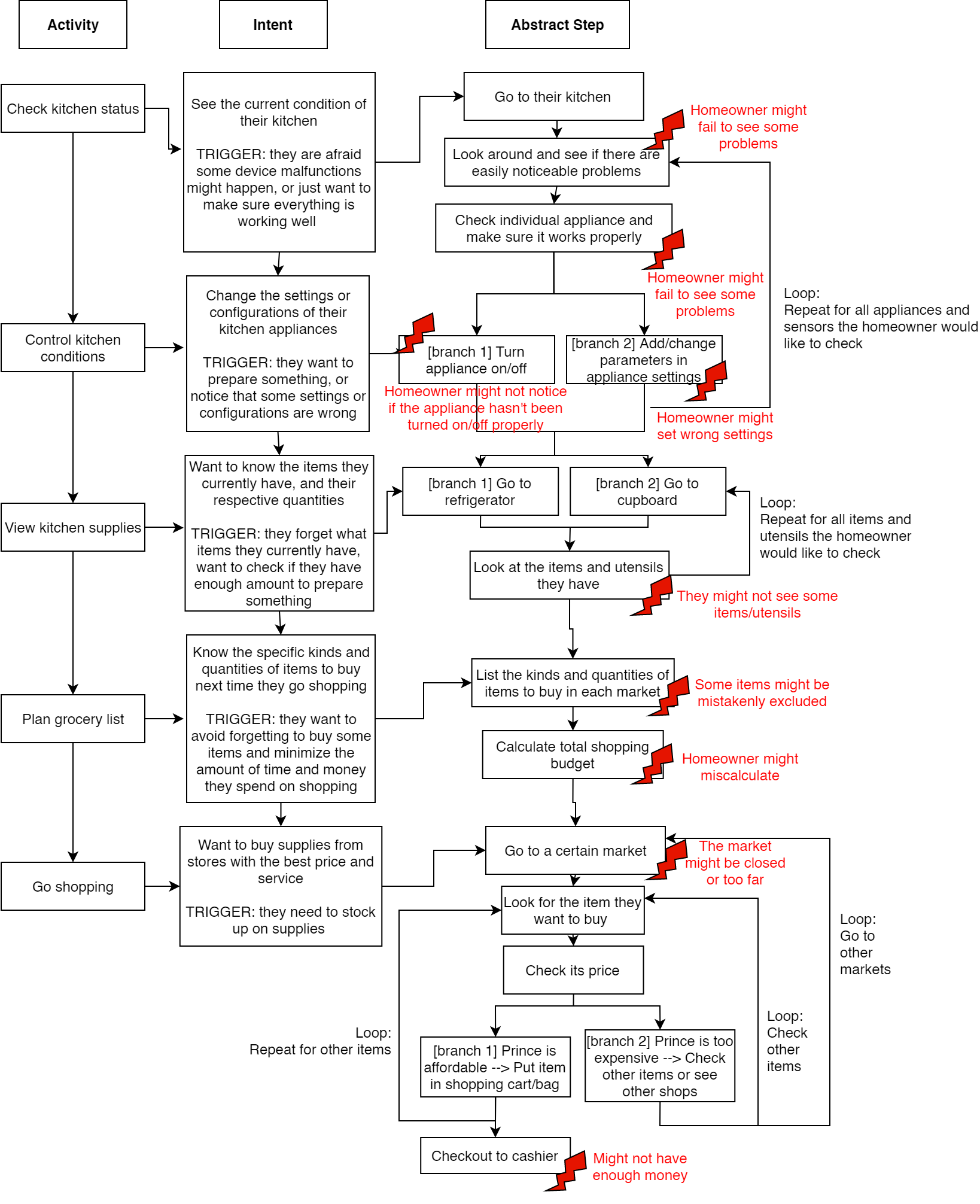
An authentication system used to validate the various stakeholders, home owners, supermarket owners, and IoT developers. It ensures that the right people access the right parts of the system and have access to the right set of functions.

# 4. Contextual Design

## 4.1. Flow Model

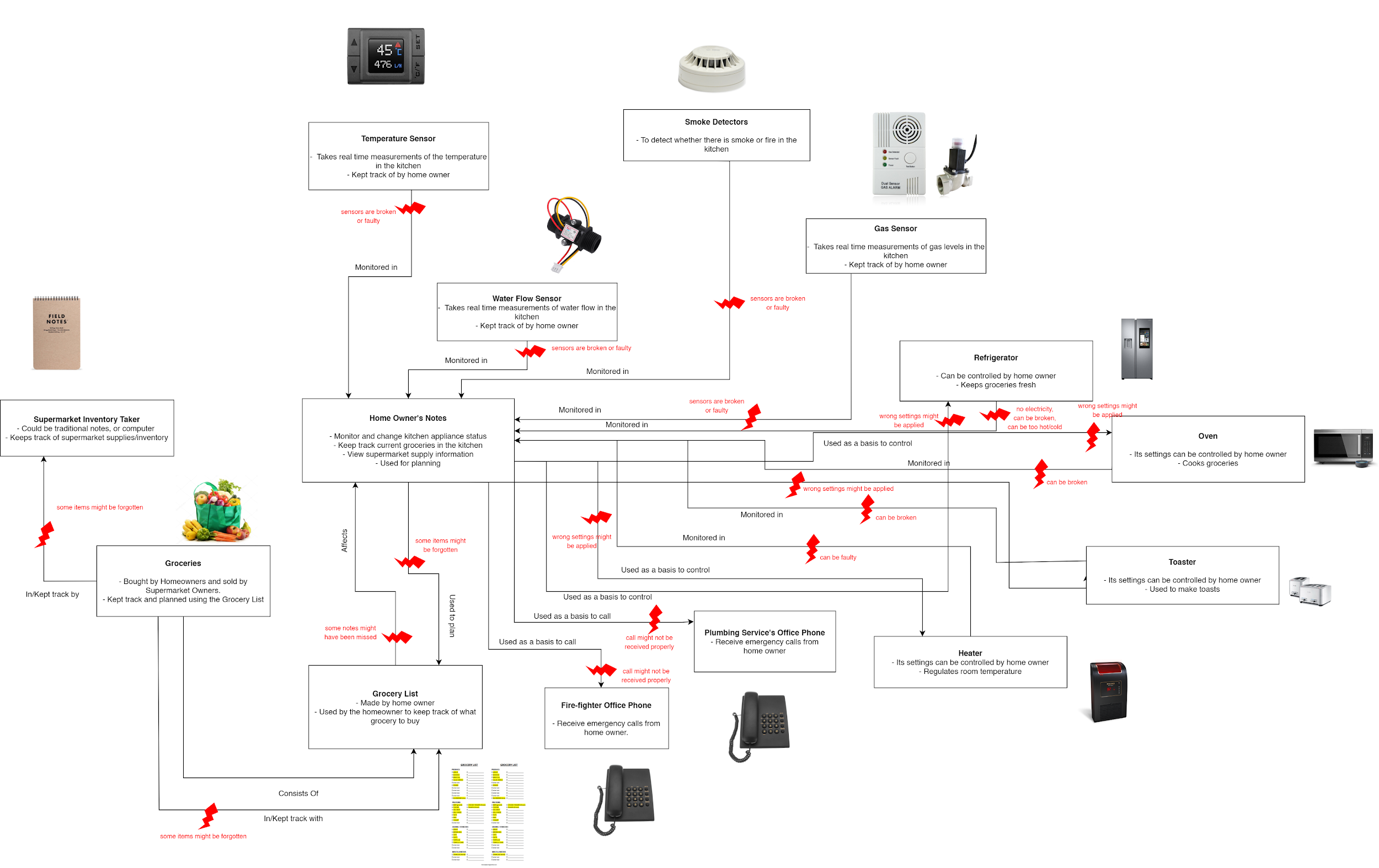
****

## 4.2. Sequence Model

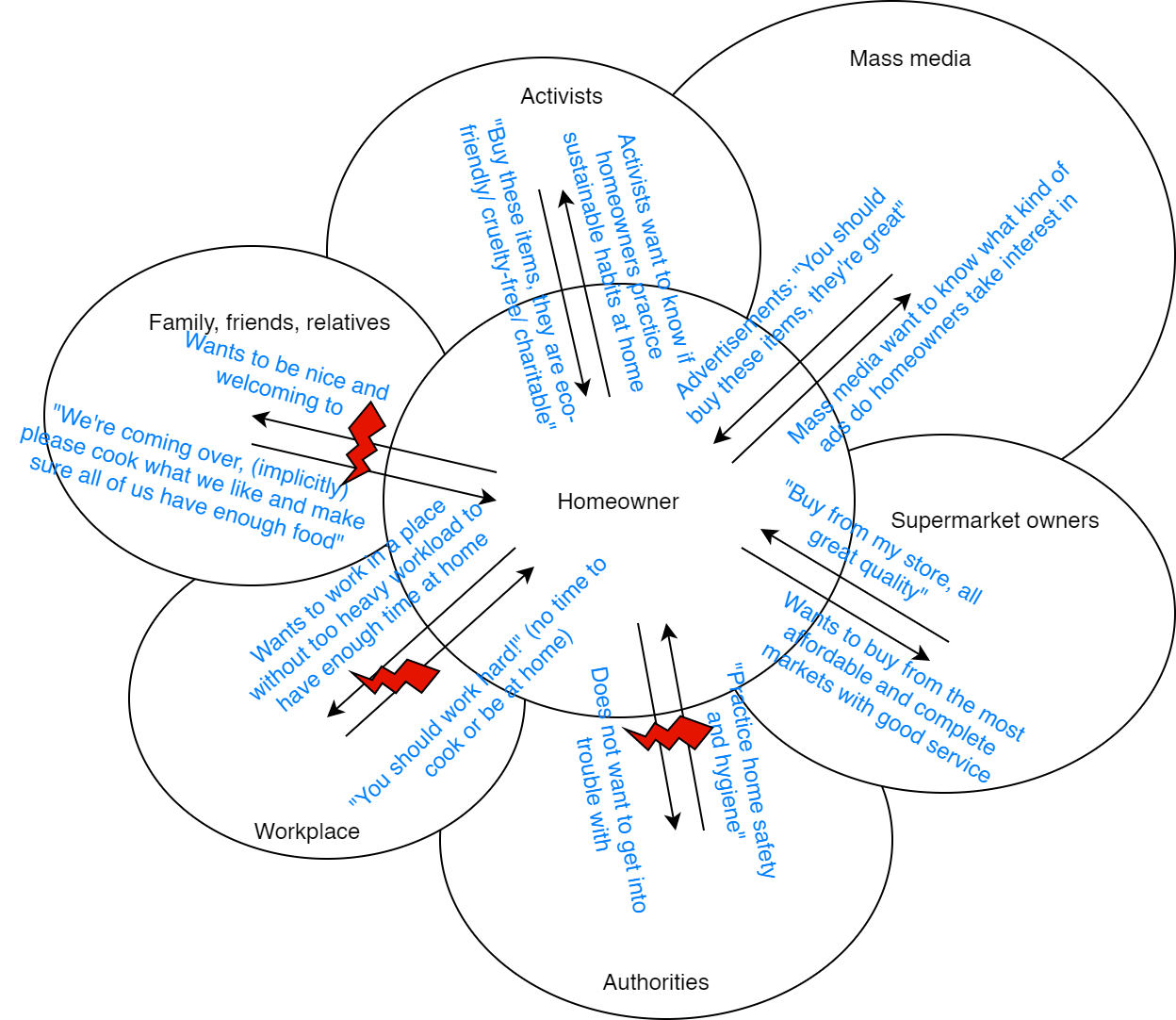
****

## 

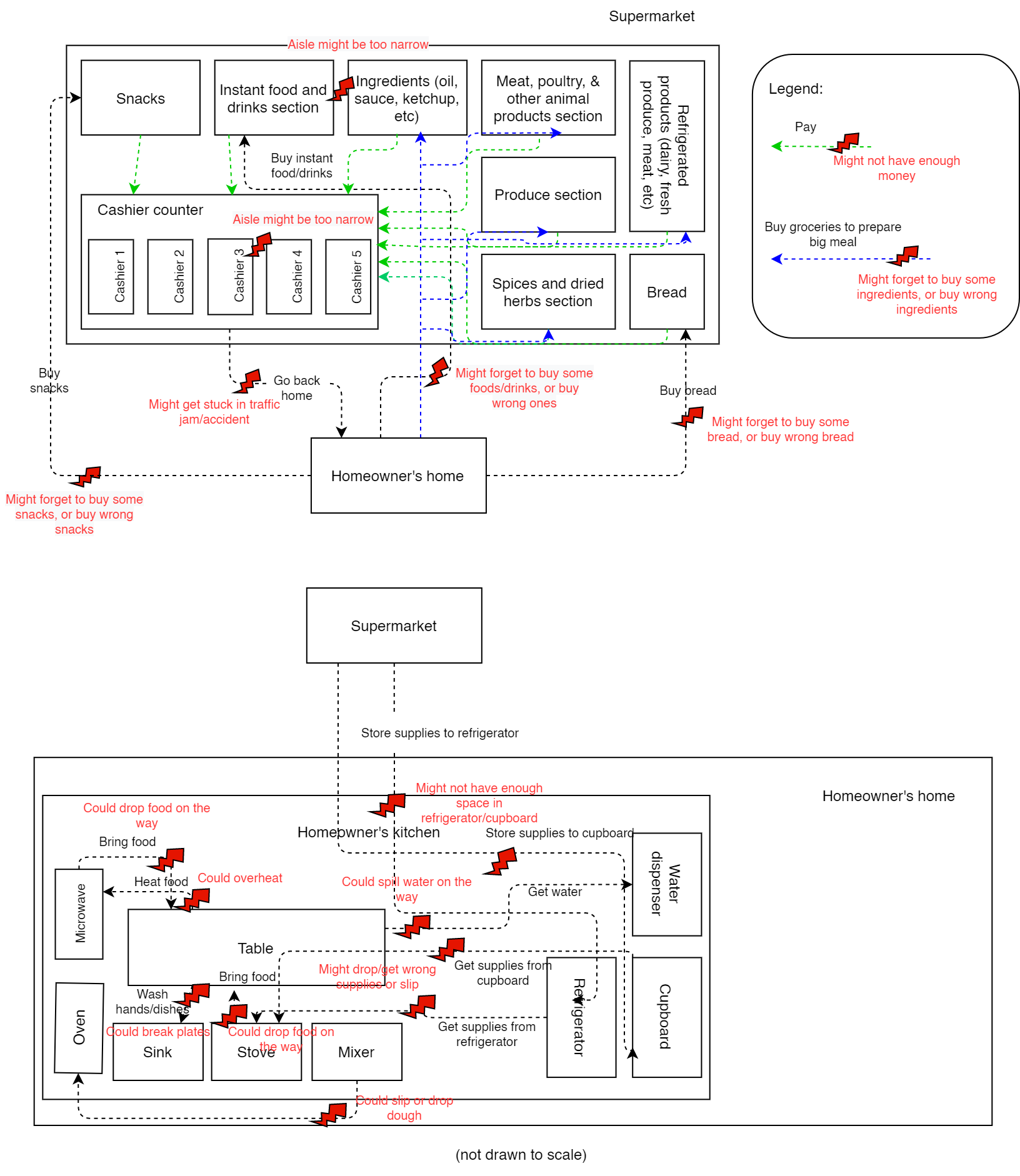
## 4.3. Artifact Model

****

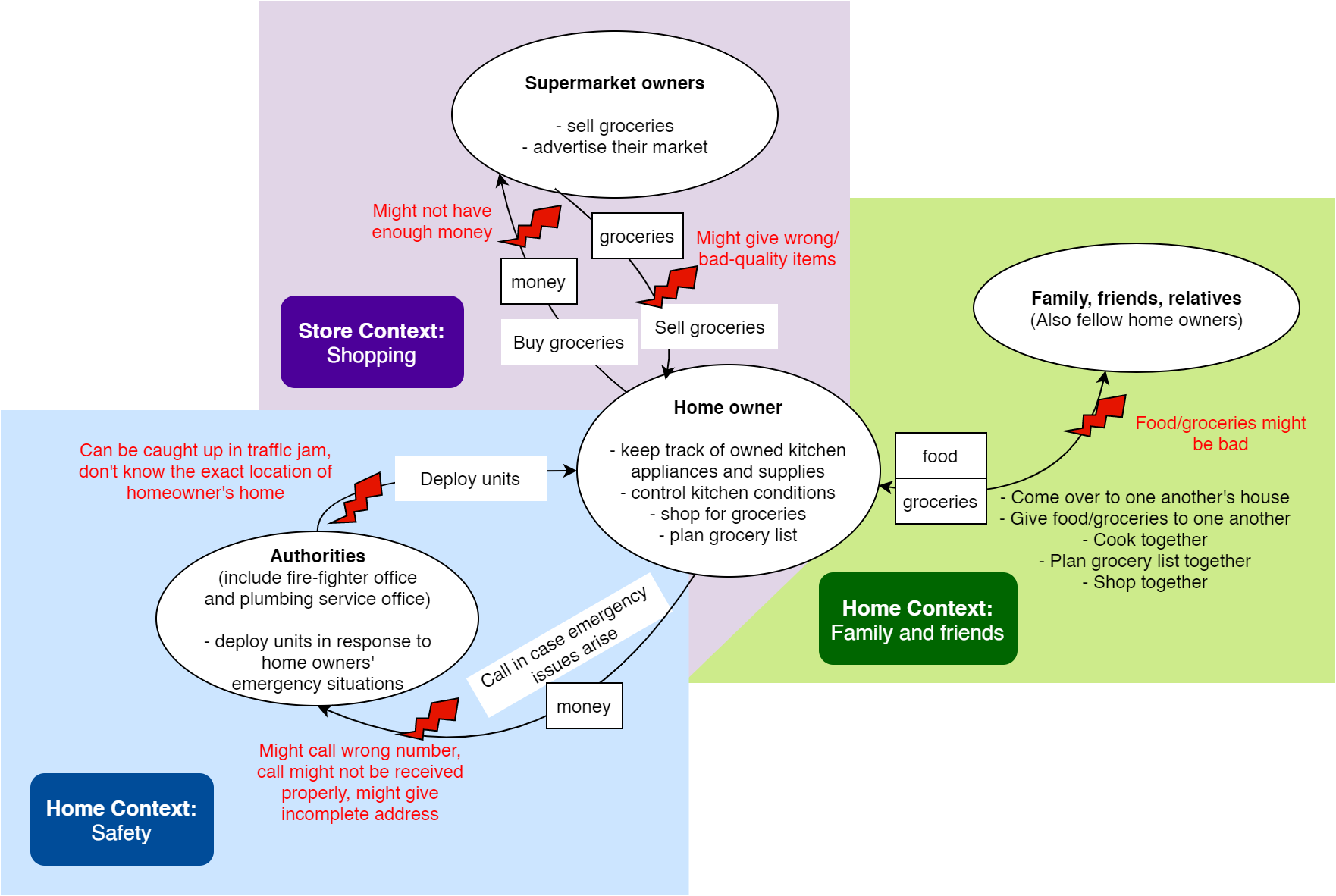
## 4.4. Cultural Model

****

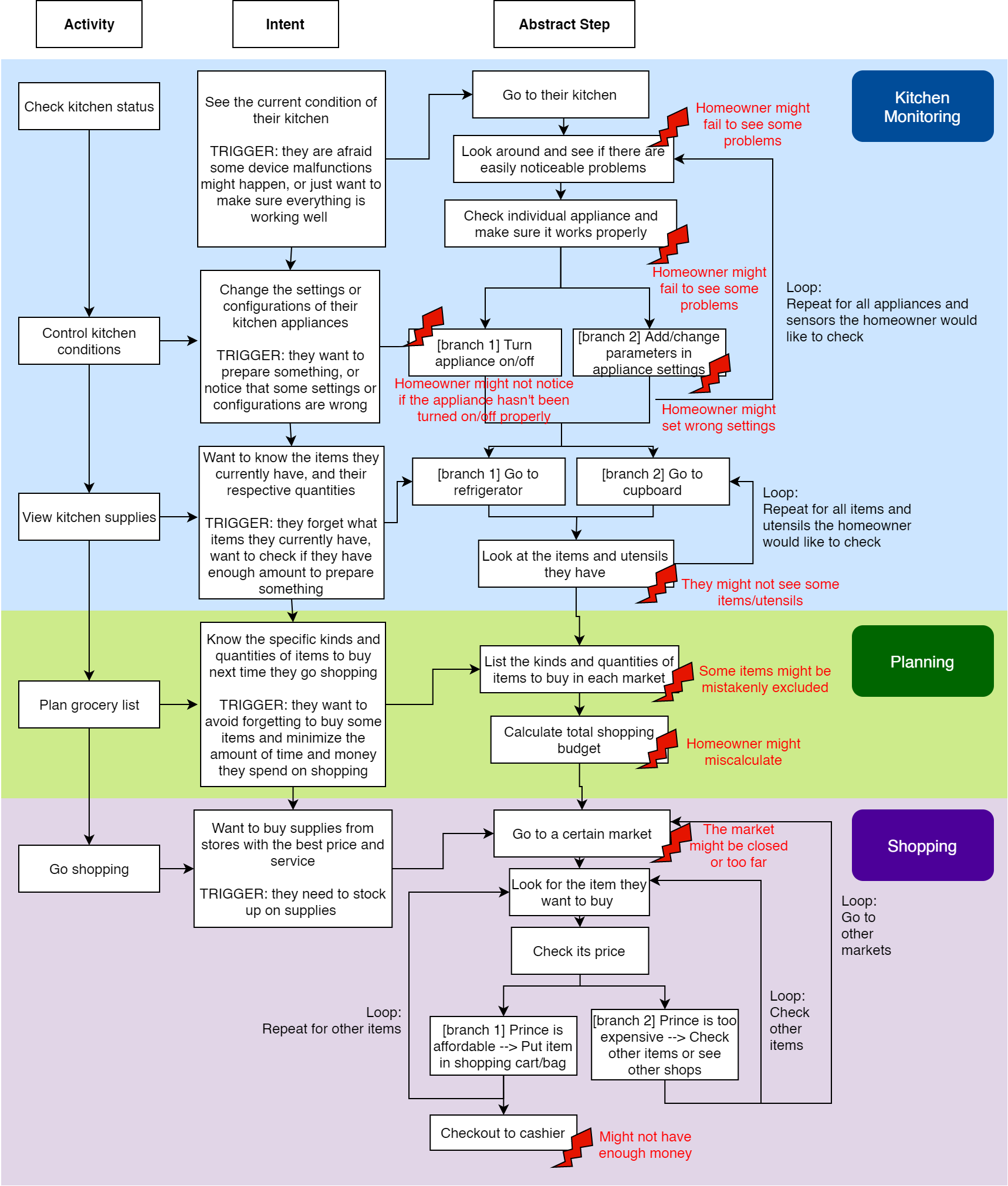
## 4.5. Physical Model

****

## 4.6. Consolidated Flow Model

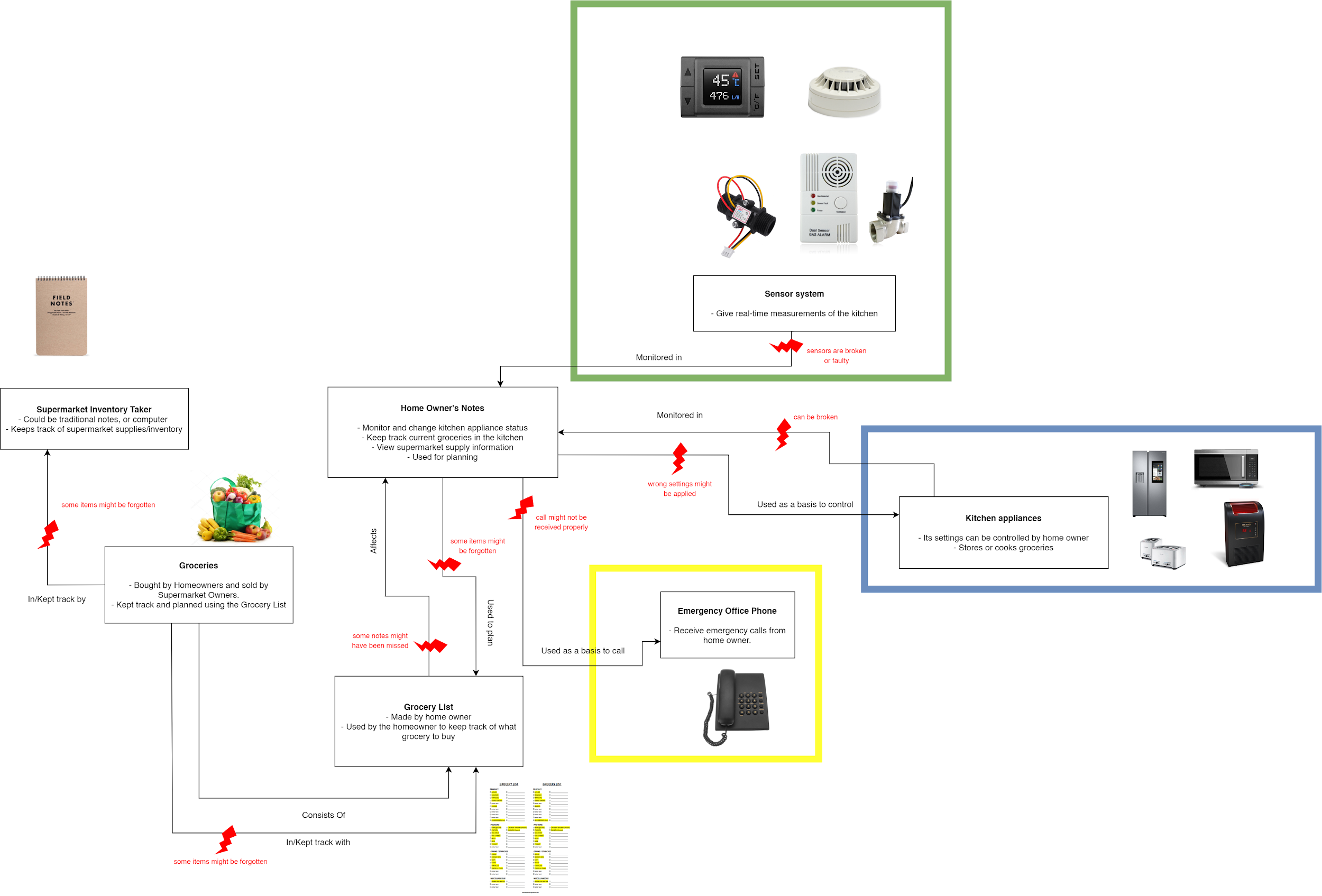
****

## 4.7. Consolidated Sequence Model

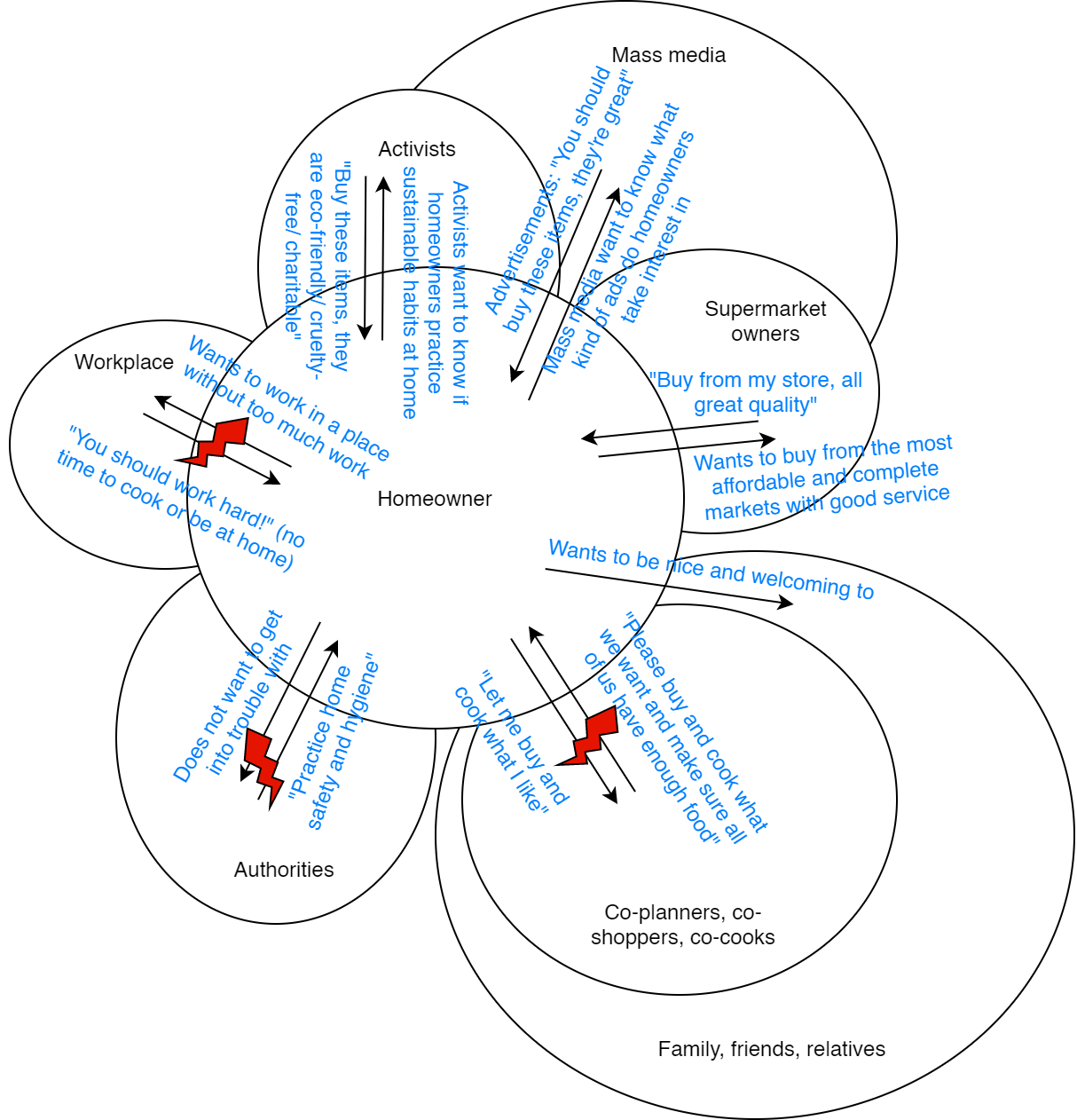
****

## 

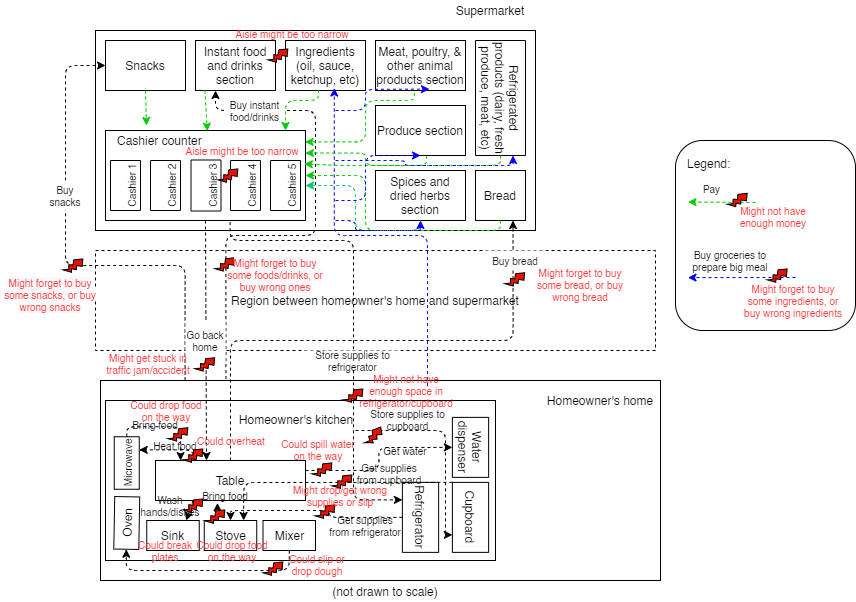
## 4.8. Consolidated Artifact Model

****

## 4.9. Consolidated Cultural Model

****

## 4.10. Consolidated Physical Model

****

## 4.11. Storyboards

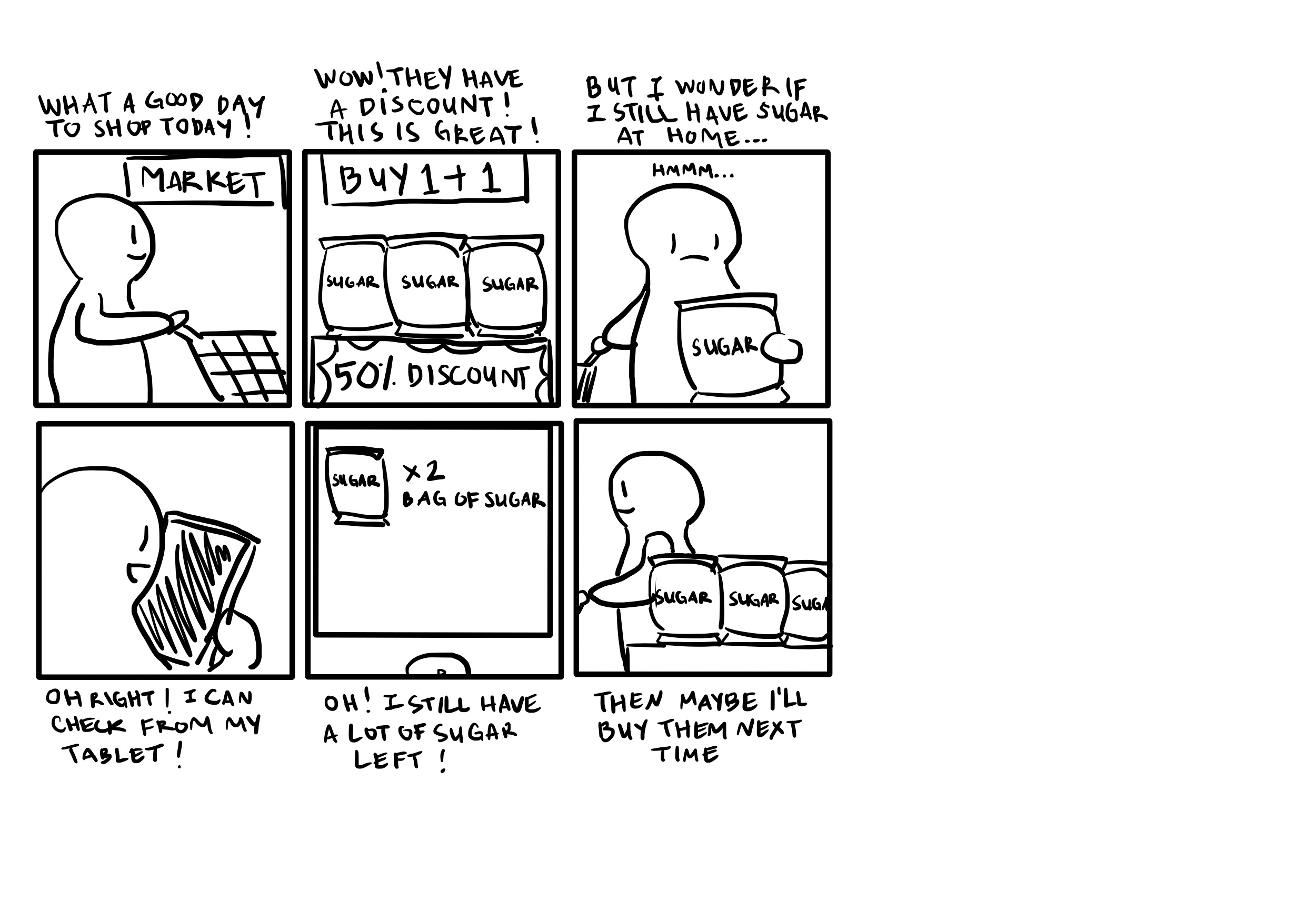
### 4.11.1. Storyboard & Scenario 1

|  |  |
| --- | --- |
| Initial Assumption | 1. Sensor Based for Physical Things is turned on and connected to Cloud Server. 2. User is connected to Wi-fi and is logged in to the System. |
| Normal | 1. Sensors take measurements and push data to Cloud server via Wi-fi. 2. User opens their dashboard and is given the kitchen appliances stats, such as energy consumption, its on/off state, and many more. 3. User can choose to interact with the kitchen appliance through the dashboard. |
| What Can Go Wrong | 1. Wi-fi might not be connected -> Prompt user to check W-fi connection. 2. User is not logged in -> Prompt user to log in. 3. Error in measurements. Ex: Measurements of two nearby sensors vary extremely. -> Prompt Owner to check for the malfunctioning of the sensor. |
| Other Activities | Sensors take measurements and constantly update the data on the dashboard. |
| System State on Completion | Collection of data from the sensors is pushed to Cloud and the User was able to see the data on the dashboard. |



### 4.11.2. Storyboard & Scenario 2

|  |  |
| --- | --- |
| Initial Assumption | 1. Sensor Based for Physical Things is turned on and connected to Cloud Server. 2. User is connected to Wi-fi and is logged in to the System. |
| Normal | 1. Sensors take measurements and push data to Cloud server via Wi-fi. 2. The User can open their dashboard and choose list of items menu. 3. The User can use the list to plan a grocery list. |
| What Can Go Wrong | 1. Wi-fi might not be connected -> Prompt user to check W-fi connection. 2. User is not logged in -> Prompt user to log in. 3. The System does not generate the list of items -> Prompt user to restart the system. |
| Other Activities | Sensors constantly push the changing data to the Cloud. |
| System State on Completion | User successfully used system to plan their grocery list. |



### 4.11.3. Storyboard & Scenario 3

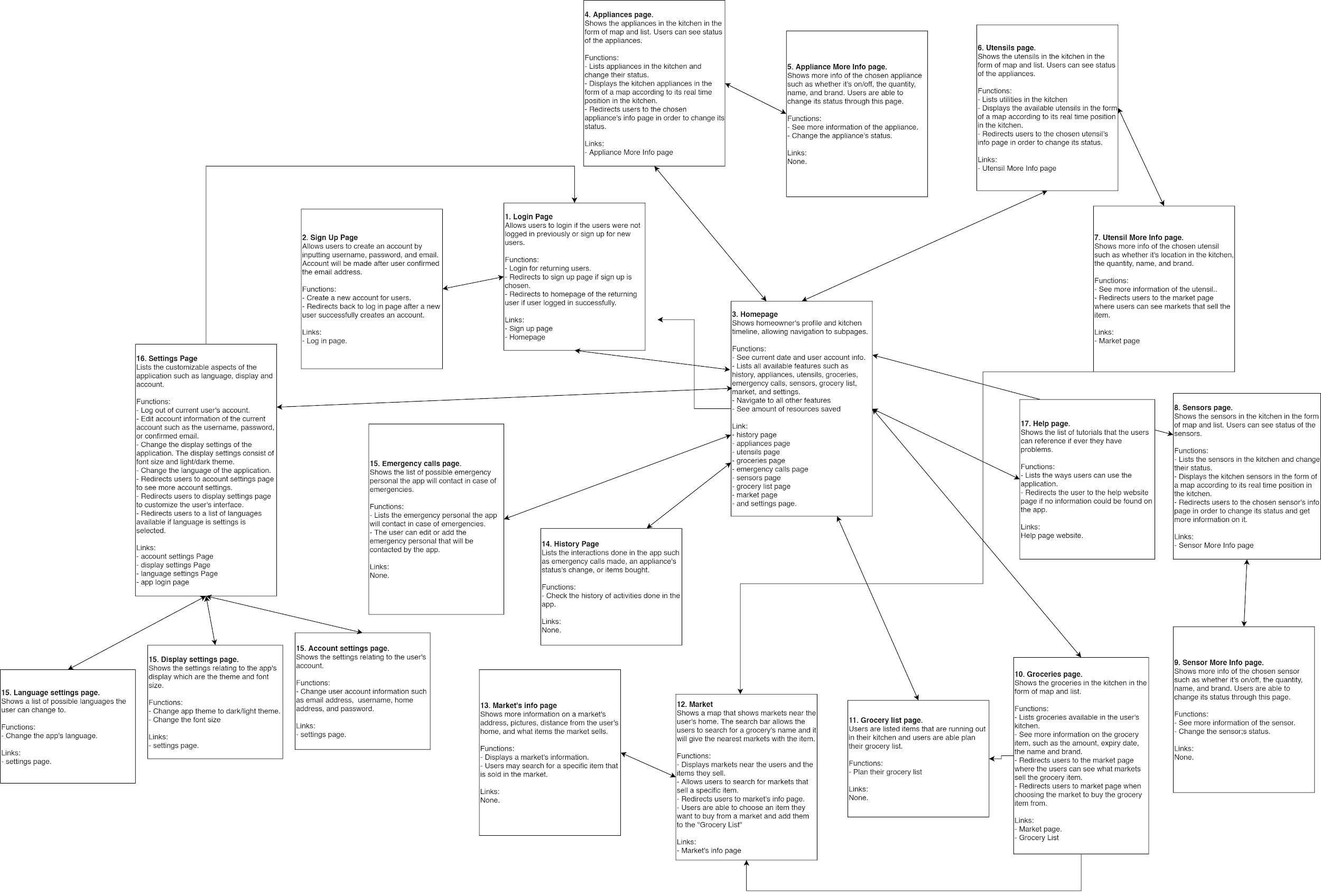
|  |  |
| --- | --- |
| Initial Assumption | The Supermarket owner is connected to Wi-fi and is logged in to the System. |
| Normal | 1. The Supermarket owner opens their dashboard. 2. The Supermarket owner chooses the menu to view statistics on what items homeowners buy. |
| What Can Go Wrong | 1. Wi-fi might not be connected -> Prompt user to check W-fi connection. 2. User is not logged in -> Prompt user to log in. |
| Other Activities | Sensors in home owners’s houses push data to the Cloud, updating the database. |
| System State on Completion | The Supermarket owner successfully viewed statistics on what items homeowners buy and may choose to create decisions based on that. |



## 4.12. Vision Diagram

****

## 4.13. User Environment Design

****

# 5. Requirements

## 5.1. Functional Requirements

* The system should be able to perform validation and authentication to prevent misconduct.
* In order to do (1), the system should be connected and has access to the relevant identity databases.
* The system should be connected to Wi-Fi and cloud database.
* The system should be able to process data from sensors.
* The system should be able to perform measurements.
* The system should be able to compare measurement results with certain thresholds.
* The system should be able to record, display, and process the stakeholder’s data. This includes statistical analysis and visualization.
* The system should be able to generate notifications and warnings and give them to users.
* The system should be able to receive input from IoT developers and send them as notifications to users.
* The system should be connected to relevant authorities, such as policemen, firemen, and healthcare facilities, in order to respond to emergencies.

## 5.2. Nonfunctional Requirements

### 5.2.1. Usability

The system must be easy to learn for all users irrespective or background. Within a week, at least 80% of all users should be able to use the system to fulfill their desired needs. The web interface elements (e.g. login, video display page) will be simple & easy to understand conveying only the necessary content, following standards and conventions for displaying content, and using terminology that users will follow and understand. All content will be easily reachable. There will be a help page and complete user documentation which will explain how to achieve common tasks. Errors will not exceed 1 bug per 1000 lines of code. Error messages must give the user specific instructions for recovery. For expert users, special commands and accelerators will be available to increase their efficiency. We expect that experts should exhibit at least 50% more efficiency (measured by speed of completing tasks) than novice users. Users can flexibly traverse the system and perform actions in whatever order they require. System will, at all times, clearly make visible it’s current status.

|  |  |
| --- | --- |
| Requirement | Priority |
| Within a week, at least 80% of all users should be able to use the system to fulfill their desired needs | High |
| The web interface elements (e.g. login, video display page) will be simple & easy to understand conveying only the necessary content | High |
| All content will be easily reachable via appropriate links. Content should be accessed from anywhere | High |
| Users can flexibly traverse the system and perform actions in whatever order they require | High |
| System will, at all times, clearly make visible it’s current status | High |
| Errors will not exceed 1 bug per 1000 lines of code | High |
| Error messages must give the user specific instructions for recovery and must speak the user’s language | High |
| Follow standards and conventions when designing the interface | Medium |
| Use terminology that users understand | Medium |
| Add a help page and complete user documentation which will explain how to achieve common tasks | Medium |
| Add special commands and accelerators to increase the efficiency of experienced users. Efficiency increase should be roughly 50% compared to the average | Medium |

### 5.2.2. Internationalization

The system will be used by varied users of different backgrounds and expertise. We must adapt the system to deal with these issues related to internationalization. All interface text and messages will be translated to adapt to the user’s language. All documents related to the system will also be translated. All documents must allow for both page formats, A4 and 8.5x11. Icons, maps, and colors must be adapted to fit the user’s culture as well making sure to avoid any signs that may offend or misrepresent the culture. User’s currency, date and time format, measurement units, and address formats will also be used and implemented in the code. From the technical side, users will be able to display, input, store, print, and save file names using characters from their native languages. The system will classify and sort characters as well use typesetting rules appropriate for the users’ native languages. Use globalization management systems to help manage all the different global versions

|  |  |
| --- | --- |
| Requirement | Priority |
| Adapt interface text and messages to the user’s language | High |
| Adapt system related documents to the user’s language | High |
| Adapt icons, maps, and colors to fit the user’s culture avoiding offending signs. | High |
| Allow users to display, input, store, and print characters from their native languages | High |
| Classify and sort characters according to users’ native languages rules | High |
| Use typesetting rules appropriate for the users’ native languages | High |
| Use Unicode (UTF-8) to encode all scripts and symbols in the system | High |
| Use globalization management systems to help manage all the different global versions | High |
| Use page format that works with both A4 and 8.5x11 | Medium |
| Use user’s currency, date and time format, measurement units, and address formats in all content information (documents and interface) | Medium |
| Implement user’s currency, date and time format, measurement units, and address formats in the code | Medium |
| Adapt cooking devices and utensils functions to the user’s culture (for example Asian cultures usually use chopsticks as utensils whereas non-Asian cultures don’t as much) | Medium |
| Allow users to save file names using characters from their native languages | Low |

### 5.2.3. Privacy

Since the system deals with real-time and potentially sensitive data such as the user's kitchen layout and details, purchasing history home address, phone number etc., privacy is a major concern for the system. System should make good use of this information by providing a customized and efficient smart kitchen for the user and passing useful data to IoT developers and supermarket owners to help them improve and better manage their work. The system should not use this information to spam, profile, or disclose personal data to 3rd parties. To ensure privacy of the user, the system should ensure data quality, be transparent, correctly treat sensitive data, and enforce privacy guidelines. System needs to ensure proper accountability is maintained for all personal info collected from the user. System should limit information collection to just the necessary information. For each information collected from the user, make sure to identify the purpose of the information and get consent to collect the information. Ensure that the info is not reused or disclosed anywhere else outside the system and retain the information only as long as necessary. Ensure that all the collected information is accurate, complete, and up to date and protect it with appropriate security safeguards. System should allow users access to their information at will. Above all, be open about privacy and data collection policies and practices to the users. To do this, system will standardize the data protection technology. Data protection options will be added to the Internet architecture and protocols will be developed to automatically negotiate privacy requirements. A digital rights management system will be used for managing personal information, and W3C’s Platform for Privacy Preferences Project will be used to ensure and assure users that their privacy standards are being met.

|  |  |
| --- | --- |
| Requirement | Priority |
| Ensure proper accountability is maintained for all personal info collected from the user | High |
| System should limit information collection to just the necessary information | High |
| For each information collected, identify the purpose of the information and get consent to collect the information | High |
| Ensure that the info is not reused or disclosed anywhere else outside the system | High |
| Retain the information for only as long as necessary | High |
| Ensure that all the collected information is accurate, complete, and up to date | High |
| Protect information with appropriate security safeguards | High |
| Make privacy and data collection policies and practices public and easily available to the user | High |
| Allow users access to their information at will | Medium |
| Data protection options will be added to the Internet architecture | Medium |
| Protocols will be developed to automatically negotiate privacy requirements | Medium |
| Use a digital rights management system for managing personal information | Medium |
| Use W3C’s Platform for Privacy Preferences Project to ensure and assure users that their privacy standards are being met. | Medium |

### 5.2.4. Reliability

The reliability of the wearable device essentially depends on the software tools, hardware tools, and services (database, hosting, etc) used for the system development. It is expected that, camera, sensor and other tools works perfectly until their life time expires. The reliability of the web interface has also a crucial importance. In case the user needs help, web interface should be 100% percent available at least 165 hours per week. If the web interface crashes, it must be repaired in at most 30 minutes. We expect that external services will also be reliable if a premium 3rd party is used.

|  |  |
| --- | --- |
| Requirement | Priority |
| Devices should be plugged in at all times | High |
| Web interface should be 100% percent available at least 165 hours per week | High |
| If the web interface crashes, it must be repaired in at most 30 minutes | High |
| Devices work perfectly until their life time expires | Medium |
| External services are at least 95% reliable every week | Medium |

### 5.2.5. Performance

Real-time interaction with devices is crucial for the proper functioning of the system, so performance is crucial. Transfered data size, speed of connection, response time, processing speed must all be considered. In addition, devices should have wifi adapter fast enough to transfer data almost real-time

|  |  |
| --- | --- |
| Requirement | Priority |
| System should work real-time which means there should be an acceptable time delay such as max 4-5 seconds between request and response. | High |
| Wi-Fi adapter needs to be fast enough to transfer data almost real-time | High |
| Web server should be able to handle multiple device and user connection simultaneously. Web server may initially handle roughly 100-1000 users which could be scaled if the need arises | High |

### 5.2.6. Safety

System handles important safety conditions for the kitchen on behalf of the user. As such, it should be working properly & safely at all times

|  |  |
| --- | --- |
| Requirement | Priority |
| In case of malfunction, system should shutdown itself and reboot in order to prevent unpredicted results. | High |
| System should not let important functions such as smoke detectors be turned off by users | High |
| If device is malfunctioning, shutdown immediately and reboot in order to prevent disastrous results | High |

### 5.2.7. Maintainability

Users have little technical knowledge and need to have a system that is easy to maintain and requires minimal updates outside content.

|  |  |
| --- | --- |
| Requirement | Priority |
| System must be easy to maintain and update after installation | High |
| Provide full documentation that users can use if internal maintenance or updating is required; create a quick reference guide to troubleshoot common issues | High |
| Make it easy to add/remove/update devices without prior training or external help | High |

### 5.2.8. Portability

The web platforms and dashboards for the system will be viewed from several different platforms and browsers, so compatibility is crucial. In addition, both mobile and desktop platforms should be supported.

|  |  |
| --- | --- |
| Requirement | Priority |
| Check cross-browser compatibility (IE 7, IE 8, Firefox, Google Chrome, Safari) | High |
| Downloadable content must be compatible with different platforms (Windows, Mac, Linux) | High |
| Write software in a platform independent programming language | Medium |

### 5.2.9. Scalability

The back-end is expected to be able to accommodate an increasing number of users and traffic. The system itself should also allow users to add items and devices to their kitchens and handle this increased load.

|  |  |
| --- | --- |
| Requirement | Priority |
| Ensure that the database used has the capacity for the potential growth | High |
| Ensure that the system can handle potential growth in user’s kitchen items and devices | High |
| Check capacity requirements with host to ensure server scalability | Medium |

### 5.2.10. Security

A multi-level access system needs to be in place so only admins have full back-end access, and only users can access their respective kitchen systems. Also, the system includes a number of dashboards for the different users, so password logins are necessary.

|  |  |
| --- | --- |
| Requirement | Priority |
| Multi-level access for users (admins, users, supermarket owners, IoT developers) | High |
| Password login for each dashboard (users, supermarket owners, IoT developers) | High |
| Secure and private access for each user to to their kitchen systems | High |
| System should store user data on database securely and set access permissions to the these datas carefully | High |

### 5.2.11. Supportability

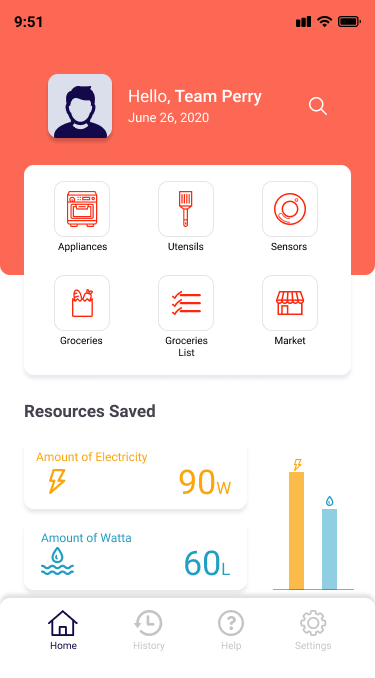
The system shall allow the system administrator to add additional features. The system needs to be cost-effective to maintain. There should be documents in requirements specification, design and implementation and validation steps.

|  |  |
| --- | --- |
| Requirement | Priority |
| Allow additional features to be added | High |
| Provide diagrams in the documents in order to improve stakeholder and developer understanding | Medium |
| Create and keep documents for all requirements steps; specification, design, implementation and validation. | Medium |

# 6. Interfaces

## 6.1. User Interface

The following are screenshots of the lo-fi prototype. The interface below was designed for an iPhone 8 and above. Here are the representative screens.

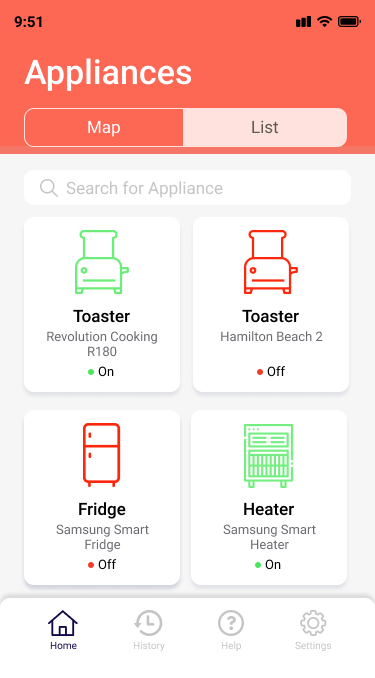
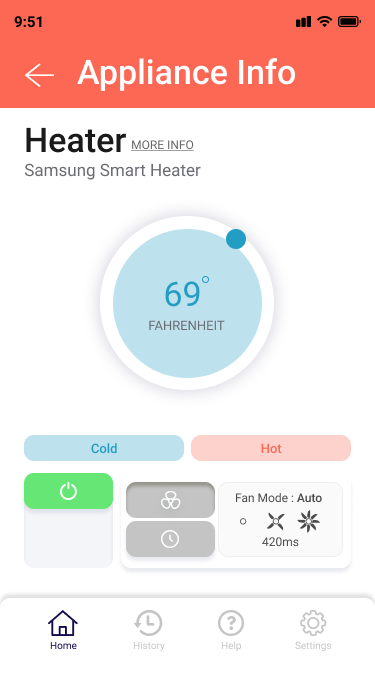


Homepage

Functions

1. See current date and user account info.
2. Lists all available features such as history, appliances, utensils, groceries, emergency calls, sensors, grocery list, market, and settings.
3. Navigate to all other features.
4. See amount of resources saved.

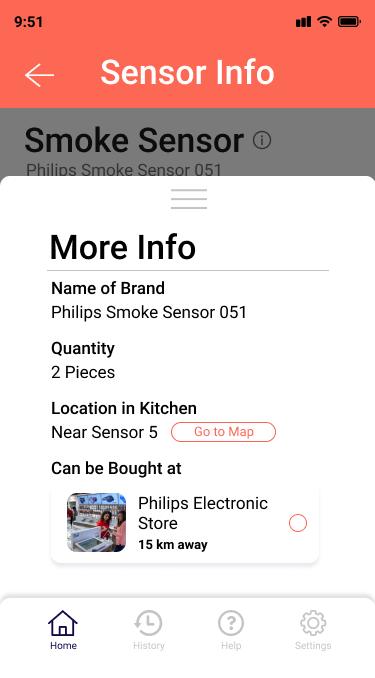
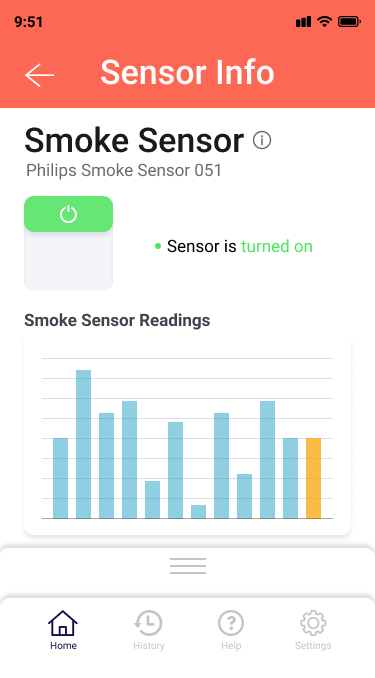
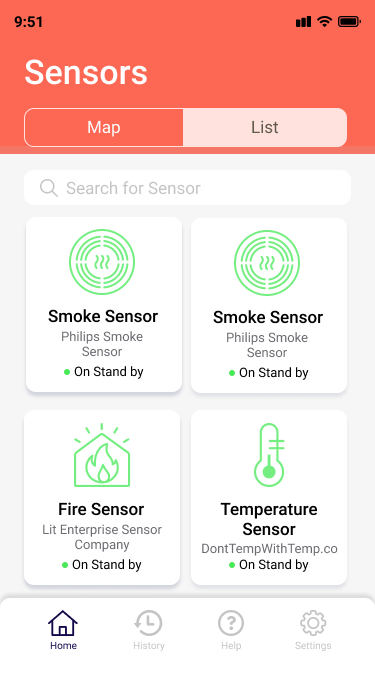
Appliances



Functions:

1. Lists appliances in the kitchen and change their status.
2. Displays the kitchen appliances in the form of a map according to its real time position in the kitchen.
3. Redirects users to the chosen appliance's info page in order to change its status.
4. See more information of the appliance.
5. Change the appliance's status

Sensors

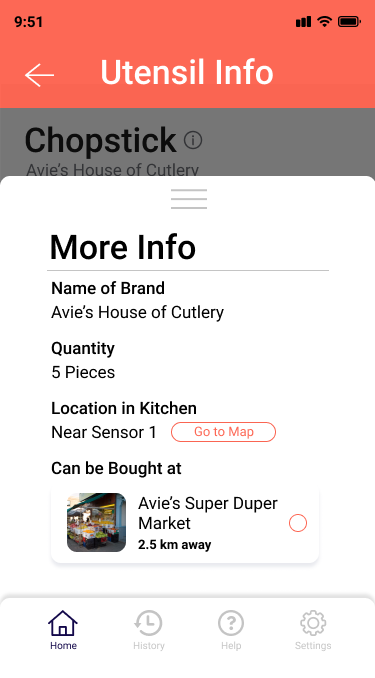
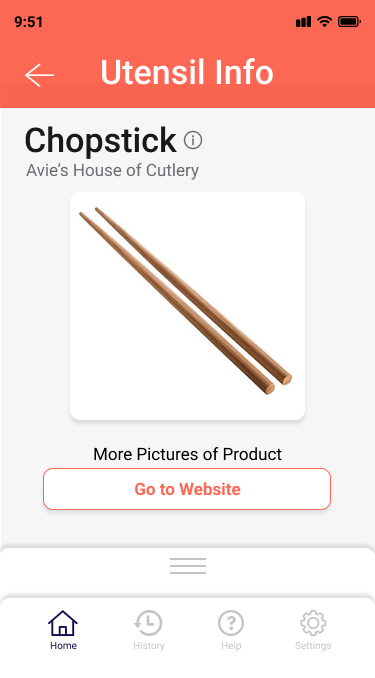
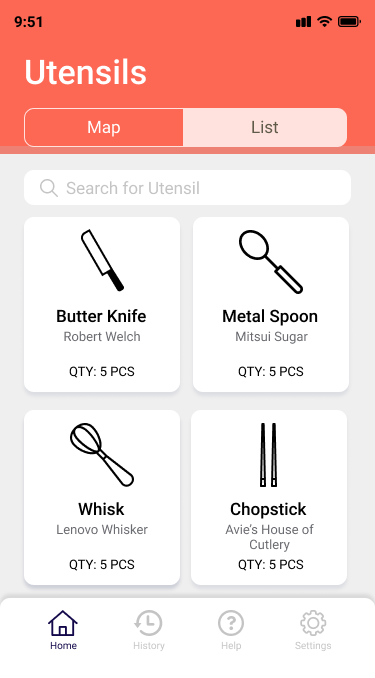


Functions

1. Lists all the sensors in the kitchen
2. Displays the available sensors in the form of a map according to its real time position in the kitchen.
3. Redirects users to the chosen sensor’s info page in order to change its status.
4. See more information of the sensor.
5. Redirects users to the market page where users can see markets that sell the item.

Utensils

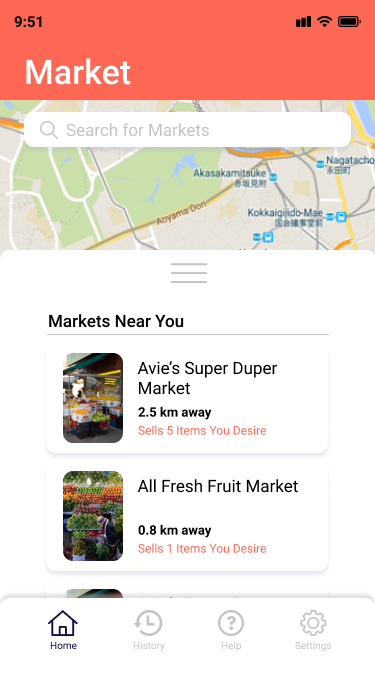
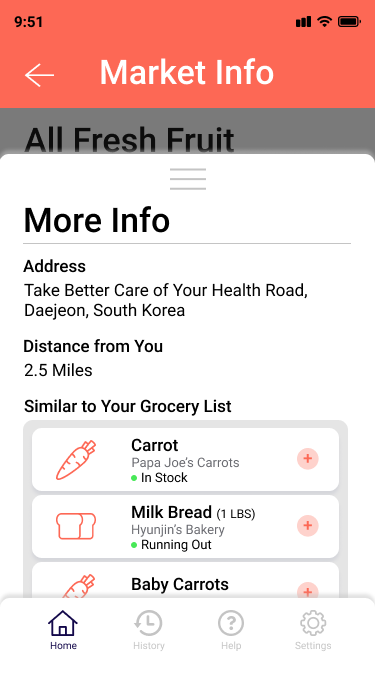
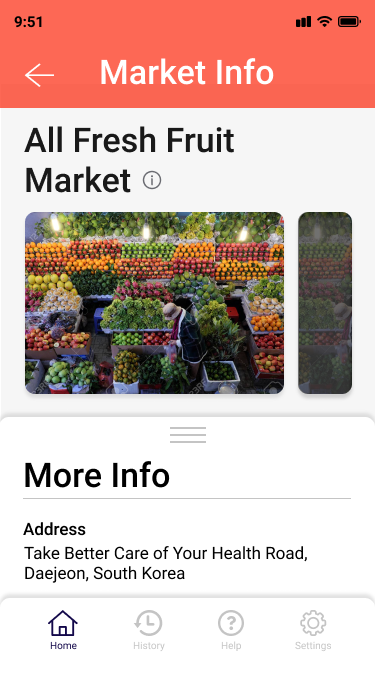
Function



Function

1. Lists utilities in the kitchen
2. Displays the available utensils in the form of a map according to its real time position in the kitchen.
3. Redirects users to the chosen utensil's info page in order to change its status.
4. See more information of the utensil.
5. Redirects users to the market page where users can see markets that sell the item.

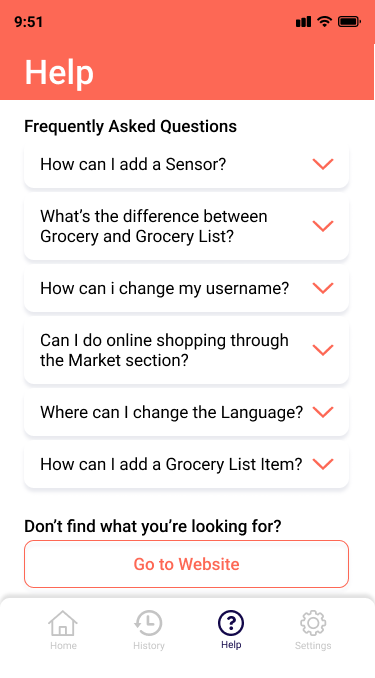
Market



Function

1. Displays markets near the users and the items they sell.
2. Allows users to search for markets that sell a specific item.
3. Redirects users to market's info page.
4. Users are able to choose an item they want to buy from a market and add them to the "Grocery List"

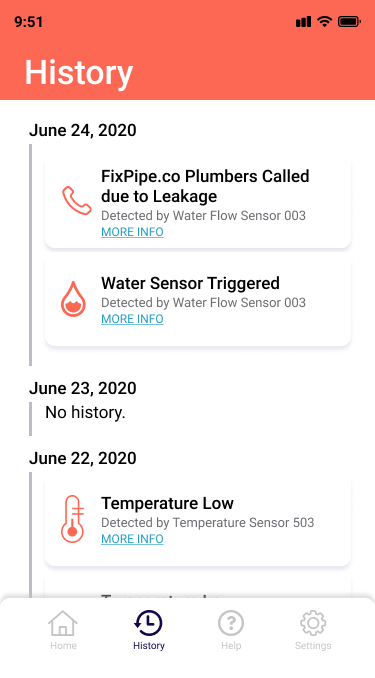
Help



Function

1. Lists the ways users can use the application.
2. Redirects the user to the help website page if no information could be found on the app.

History



Function

Check the history of activities done in the app.



Settings

Function

The settings page contains the basic settings of language, display, accounts, and log out. The language page enables users to change the language of the app according to their needs (the app, if location is not allowed), will automatically run in english at first run. The language setting will also change any features relating to internalization in the application such as the date and characters used. The display settings hold settings such as the font size or dark/light mode to accommodate the users preference. Accounts is to control the settings relating to the information relating to the user’s account, such as username, password, email, and location of address.

## 6.2. Software Interfaces

TTYK requires a cloud server to store all the data. Possible cloud servers could be Digital Ocean, Amazon AWS, or Alibaba’s cloud, depending on the country.

For Digital ocean, one must create a Droplets server and connect to it using the SSH. Users will not be required to install more software in order to connect to the created Droplet, since the application will help them do so.

For Amazon AWS, creating a EC2 Instance will suffice, using an already built-in security group by Amazon called tutorial-securitygroup. We will use short-lived SSH Keys to connect to the EC2 instance to ensure security and so will the new users. In an instance of a new user, a new IAM user will need to be set up for each individual user. Users will not be required to install more software in order to connect to the created Droplet, since the application will help them do so.

For Alibaba’s cloud, we a CentOS 7 Server is required to create the environment server. The root password will be: ‘TTYK’ + Randomized Characters of 6 digits. Users will not be required to install more software in order to connect to the created Droplet, since the application will help them do so. For initial testing stages, it would suffice to test it by connecting your laptop with a USB cable.

## 6.3. Communications Interfaces

TTYK requires an internet connection to install new plugins, update already installed ones and update some of its components (APIs, modules etc.). While the communication between the different parts of the system is important since they depend on each other, in what way the communication is achieved is not important for the system and is therefore handled by the underlying operating systems for both the mobile application and the web portal.

# 7. References

* NFPA's "Home Cooking Fires", Author Marty Ahrens, Issued November 2019.
* 2013 FAO (Food and Agriculture Organization of the United Nations) “Food wastage footprint, Impacts on natural resources” Summary Report.
* https://stackoverflow.com/questions/43759637/connecting-centos-7-to-android-phone
* https://www.alibabacloud.com/blog/how-to-set-up-your-first-centos-7-server-on-alibaba-cloud\_593743?spm=a3c0i.8276372.5363776690.4.56b26ed6zbzctr