Cat feeder

Shimon Mimoun
Naor Eliav
Nahama Weill

Instructor : Yossi Zaguri

Software Requirements Specification

Document

Version: 1.0 **Date: 21/01/2020**

Table of Contents

1. Introduction

1.1 Purpose

The SRS document is intended to explain to the customer about the product and its purpose.
The document is intended for the target audience of the product and this is animal shops, cat institutions and especially for people who have cats in their home.

1.2 Scope

(1) The system is divided into 2 parts: hardware (sensors, weight, etc.) and software (data base, application, etc.)
(2) The system will track the cat's behavior and update the cat owner.
(3) The system will absorb the cat's arrival, decide whether to allow it to eat, and weigh the cat. The system studies the cat's behavior and sees if anything is abnormal in its behavior.

1.3 Definitions, Acronyms, and Abbreviations.

There are several possible events for the product.

- The cat eating event: As the cat approaches the bowl, the bowl sensor detects the collar sensor and checks if its ID is appropriate and can be approved to eat.
  (1) If the cat is allowed to eat: The bowl opening will open and the cat will be able to eat the food from bowl.
  (2) If the cat is not allowed to eat: The bowl opening will not open and a red light will light up.

- The cat weighing event: As the cat approaches the bowl, the bowl sensor detects the collar sensor and checks if its ID is appropriate and can be approved for weighing.
  (1) If the cat has received permission: The system will weight the cat and send the data to the database.
  (2) If the cat has not received permission: The system will not weigh the cat and a red light would light.

- The food weighing event: When the cat gets permission and the bowl opening is opened, we will weight the food in the bowl. As the cat leaves the area and the opening closed, we will weigh the food in the bowl, and so we can know the amount of food the cat ate.

1.4 Overview

The SRS document contains detailed information about the product and the system, both for the client and the programmer.

SECTION 2 is for the customer. It lists all the requirements the customer needs for the product, as well as the best use of the product.

SECTION 3 is for programmers. It lists all the functional and non-functional requirements of the product, and explains the system setup.

2. The Overall Description

2.1 Product Perspective

The cat is equipped with a unique collar that allows him to access a special food bowl for him, and as he approaches the bowl, the possibility of eating out of the bowl opens to him. In addition, the cat is weighed.
The cat owner can see the cat activity in the application and track whether something is unusual in its behavior and weight. It can also set cat eating hours and other cat eating permissions from a particular bowl.
This product is different from other products on the market because, in addition to allowing a cat to eat from its bowl, there is deep learning about its behavior. There is also a possible to track the cat with GPS.

2.1.1 System Interfaces

User login will be done with data from Firebase, in addition we will use the API of the hash key to encode the user interface.

2.1.2 Interfaces

The system has a user interface (GUI), there are no specific requirements to use this interface.
The system supports all types of devices and computers, but only those with an Internet connection.
The system optimization will be using Google Chrome.

2.1.3 Hardware Interfaces

(1) ESP 32
(2) RFID
(3) WAVGAT 5V 1A
(4) Diode LED
(5) HX711 AD (weight sensor)
(6) CD-ROM Component

2.1.4 Software Interfaces

We will use the Cloud, where we will save the information we want to display on the WEB, run the algorithm and perform the deep learning.
In the Cloud we will save the data of the Database and with this we will make the connection between the hardware and the software.

2.1.5 Communications Interfaces

We do not use protocols directly.

2.1.6 Memory Constraints

There are no memory constraints.

2.1.7 Operations

There is one user for the system, it's the cat/s owner:

(1) The user can give permissions to cats for certain bowls, and allow access to bowls at specific times.
(2) The system works 24/7.

2.1.8 Site Adaptation Requirements

The user must reboot the sensor and register to the site to use the system.

2.2 Product Functions

We will install a sensor on the cat collar, when the cat approaches the bowl, the bowl sensor will detect him and if the sensors are matched the system will open the bowl opening.
While the cat is eating it will stand on the weight and it will weigh him.
When the bowl open, the system will send the data to the database and update the application.

Behind the scenes we will run an algorithm that will learn about the cat's behavior (amount of eating, eating times and weight) and alert the user when something unusual happens.

2.3 User Characteristics

No character is required to use the system, it is accessible to anyone with an Internet connection at home.

2.4 Constraints

- Put a sensor that is compatible with the cat and that it will not disturb him.
- create a user interface (web and mobile) where we could display the information on the chat and be able to interact with the interface.
- Learn about cat's behavior

(1) For the purpose of system integrity, put the collar with the sensor on the cat. The system should be in an indoor and remote location from water sources. The system should be connected to the Internet.
(2) To activate the system, it is necessary to reboot the sensor and register for the site to be used as the user interface. For the bowl sensor to be able to detect the cat sensor it is necessary that they be less than a meter away. The weight will work with cats weighing up to 10 kilo.
(3) Browser and Internet connection are required.
(4) The deep learning function will monitor the behavior and control the data.
(5) The cat sensor works according to a unique number tailored to a specific user and the weight works reliably and has a 98% accuracy.
(6) The sensors, the weight and the behavior of the cat will be checked by the programmers.
(7) The login will be done using a username and password. The system will not cause any physical or mental damage to the cat.

2.5 Assumptions and Dependencies

In case there is no internet connection, it's not possible to access the database and check if the cat has permissions to the bowl. Therefore, a light will light up and it will be possible to open the bowl manually.

2.6 Apportioning of Requirements.

It was decided to prioritize the cat GPS option at a low level.
The system will study the amount of the usual food that the cat eats and accordingly will pour an accurate amount of food.
These options will be possible in version 2 of the product.

3. Specific Requirements

The system has 2 main parts that eventually they communicate.

hardware:

- The system will use a smart component (for example in our case ESP 32) through which we perform all the functionality. On this component we will connect the components that will help us to collect information about the cat and by which we will present the data to the user.
- The system will include RFID sensors for identifying the cat and fitting it to its bowl.
- The system will include weight for weighing the cat and learning about its health.
- We will use the push and pull component (in our case CD ROM) to make the food accessible to the cat.
- The cat will be equipped with a GPS sensor that will track its location.
- The programming part of the components will be via C ++ or Python.

Software:

- The system will be built on client-server:
  - Client side: will include PHP, JS, CSS, HTML.
  - Server side: will include node.js, react.js.
- The system will use the SQL database to store the cat data.
- The system will include a visual interface for the user (website, app, etc.) where the user will be able to track the cat's behavior and health.
- The system will study the cat's behavior and condition and present it with graphical tools.

3.1 External Interfaces

[SKIP THIS PART]

3.2 Functions

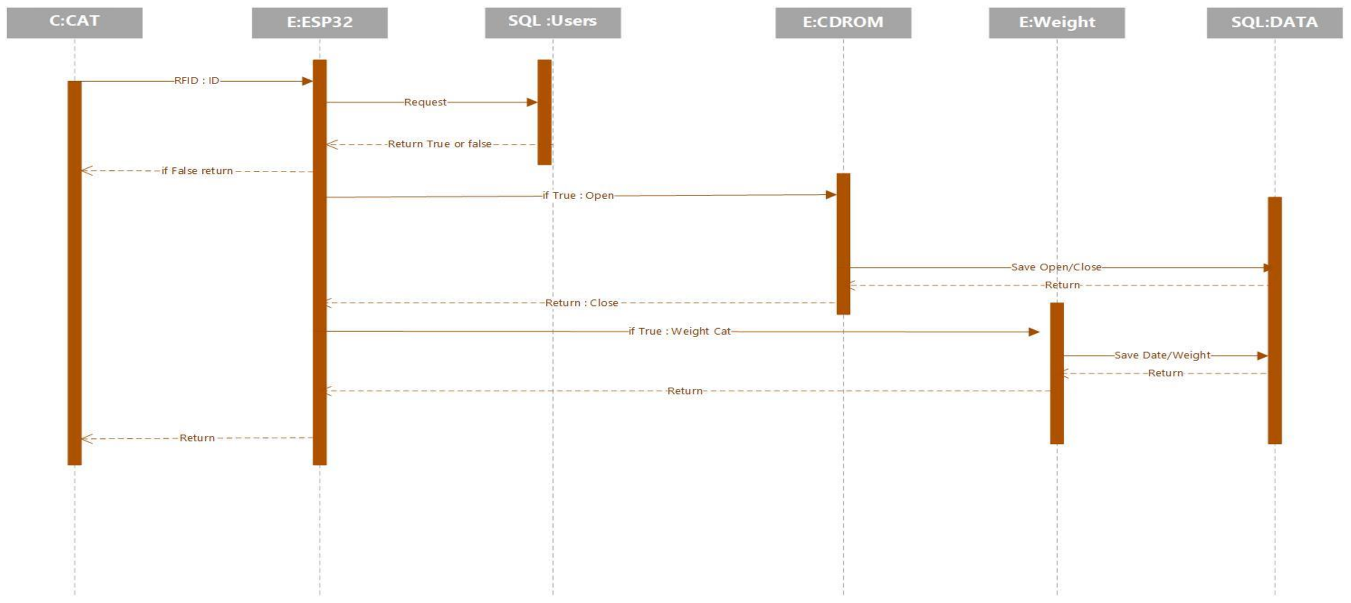
- The system will identify the cat based on its sensor and ID, and check if it has permission for the unique food bowl.
- The system will open the food bowl to the identified cat.

- The system will weight the cat while eating and send the data to a database.
- The system will display the data learned in a friendly UI.

3.3 Performance Requirements

[SKIP THIS PART]

3.4 Logical Database Requirements



3.5 Design Constraints

- One of the major limitations is the mismatch between the hardware components and the mismatch between the programming code and the hardware.
- Another limitation is that the hardware components will not fit our expectations as programmers (for example, the RFID component works at a smaller distance than we expected).

3.5.1 Standards Compliance

[SKIP THIS PART]

3.6 Software System Attributes

[SKIP THIS PART – FILL ONLY SECTION 3.6.3 ON Security]

<u>3.6.1 Reliability</u>

Specify the factors required to establish the required reliability of the software system at time of delivery. If you have MTBF requirements, express them here. This doesn't refer to just having a program that does not crash. This has a specific engineering meaning.

<u>3.6.2 Availability</u>

Specify the factors required to guarantee a defined availability level for the entire system such as checkpoint, recovery, and restart. This is somewhat related to reliability. Some systems run only infrequently on-demand (like MS Word). Some systems have to run 24/7 (like an e-commerce web site). The required availability will greatly impact the design. What are the requirements for system recovery from a failure? "The system shall allow users to restart the application after failure with the loss of at most 12 characters of input".

<u>3.6.3 Security</u>

We want every user in the system to have the ability to change only their data and not the data of other users in the system, this is done by username and password, which is encoded with the help of a hash key.
We will keep the user's last login to the site so that we can identify whether it is a login of the original user or a malicious connection.

<u>3.6.4 Maintainability</u>

Specify attributes of software that relate to the ease of maintenance of the software itself. There may be some requirement for certain modularity, interfaces, complexity, etc. Requirements should not be placed here just because they are thought to be good design practices. If someone else will maintain the system

<u>3.6.5 Portability</u>

Specify attributes of software that relate to the ease of porting the software to other host machines and/or operating systems. This may include:

- *Percentage of components with host-dependent code*
- *Percentage of code that is host dependent*
- *Use of a proven portable language*
- *Use of a particular compiler or language subset*

- *Use of a particular operating system*

Once the relevant characteristics are selected, a subsection should be written for each, explaining the rationale for including this characteristic and how it will be tested and measured. A chart like this might be used to identify the key characteristics (rating them High or Medium), then identifying which are preferred when trading off design or implementation decisions (with the ID of the preferred one indicated in the chart to the right). The chart below is optional (it can be confusing) and is for demonstrating tradeoff analysis between different non-functional requirements. H/M/L is the relative priority of that non-functional requirement.

ID	**Characteristic**	**H/M/L**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1	Correctness													
2	Efficiency													
3	Flexibility													
4	Integrity/Security													
5	Interoperability													
6	Maintainability													
7	Portability													
8	Reliability													
9	Reusability													
10	Testability													
11	Usability													
12	Availability													

Definitions of the quality characteristics not defined in the paragraphs above follow.

- *Correctness - extent to which program satisfies specifications, fulfills user's mission objectives*
- *Efficiency - amount of computing resources and code required to perform function*
- *Flexibility - effort needed to modify operational program*
- *Interoperability - effort needed to couple one system with another*
- *Reliability - extent to which program performs with required precision*
- *Reusability - extent to which it can be reused in another application*
- *Testability - effort needed to test to ensure performs as intended*
- *Usability - effort required to learn, operate, prepare input, and interpret output*

3.7 Organizing the Specific Requirements

3.7.1 System Mode

The system has 2 types of mod:

- Automatic mode - When the system is connected to the Internet.
- Manual Mode - When the system is disconnected from the Internet.

3.7.2 User Class

There are 3 different types of users:
Cat - It contains name and ID.
Cat's Owner - Contains a username, password and list of cats and bowls.
Vet - It contains a list of cat's owners.

3.7.3 Objects

There are 3 types of objects to the system:
Cat - This is the main object of the system, each cat contains its own data, with which we can track its behavior.
Cat's owner - He has the option to track the cat behavior and check his health. In addition, he can set the cat eating hours.
Vet - He can check when something unusual happens in the cat's behavior, check on his health and update the cat's owner if necessary.

3.7.4 Feature

In order for the system to work we will need an externally desired service that is the cat. The whole system is based on a cat that turns on the hardware and thus sending information to the database. If the cat does not access the hardware, the system will not operate.

3.7.5 Stimulus

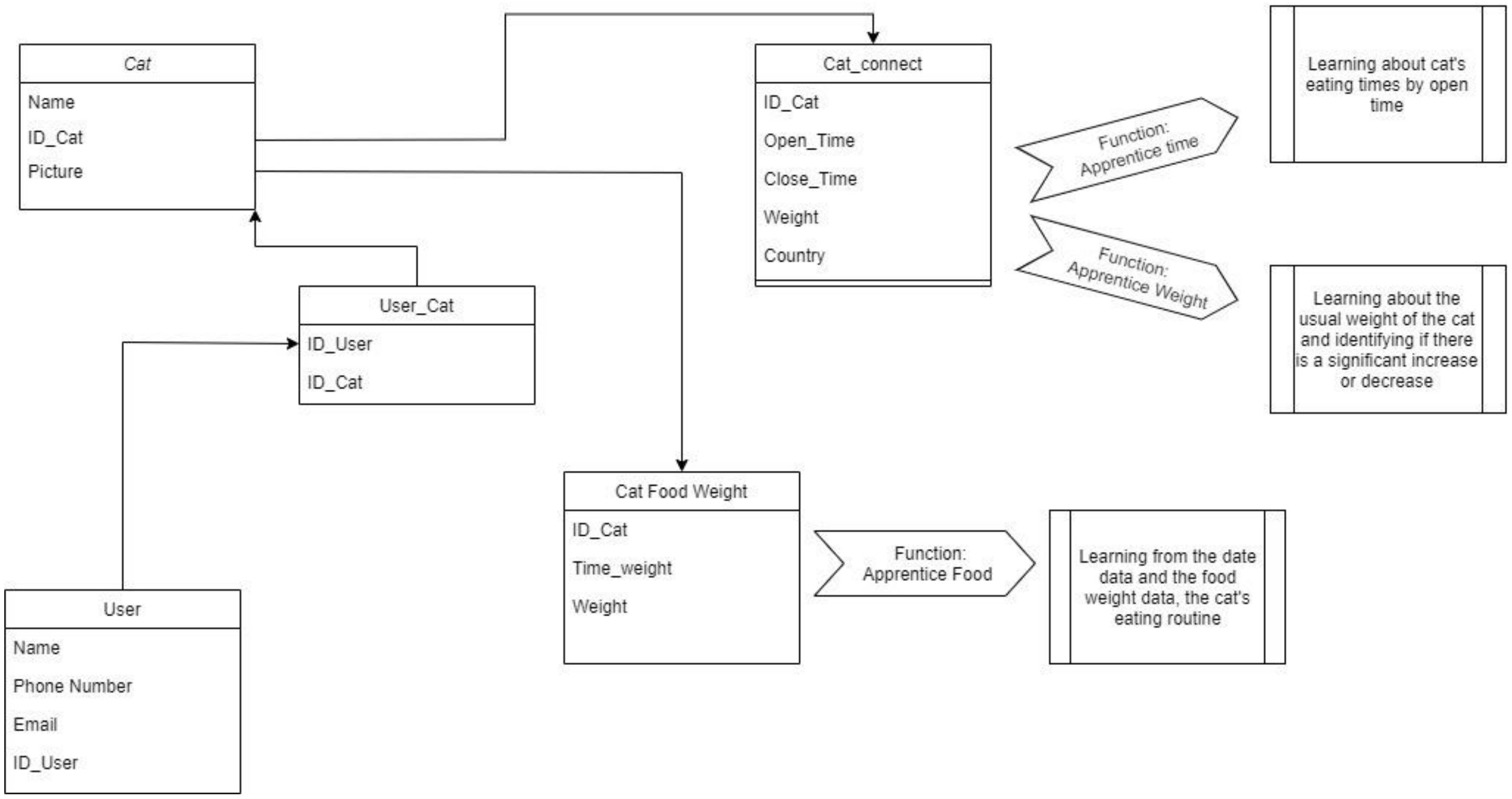
The deep learning function is a system that we can describe his functions in terms of stimuli, because when it receives information from the database it immediately starts to act.

3. 7.6 Response

The whole system operates according to a response it receives from the hardware or the database, and accordingly to this it runs different functions in the system.

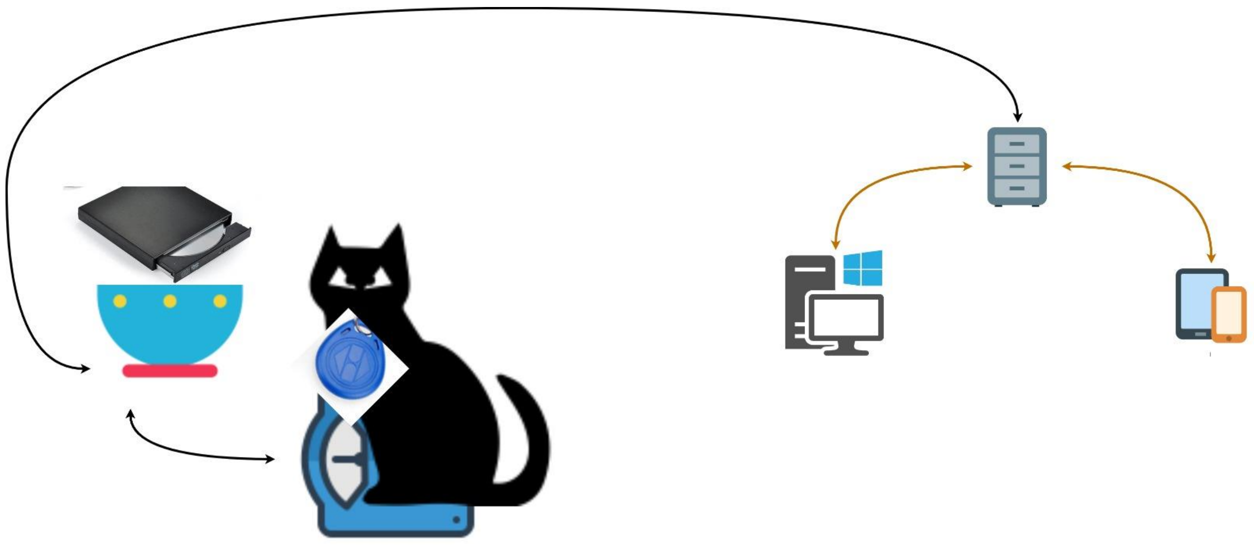
3.7.7 Functional Hierarchy

Data flow diagram to show the relationships between the functions and data:

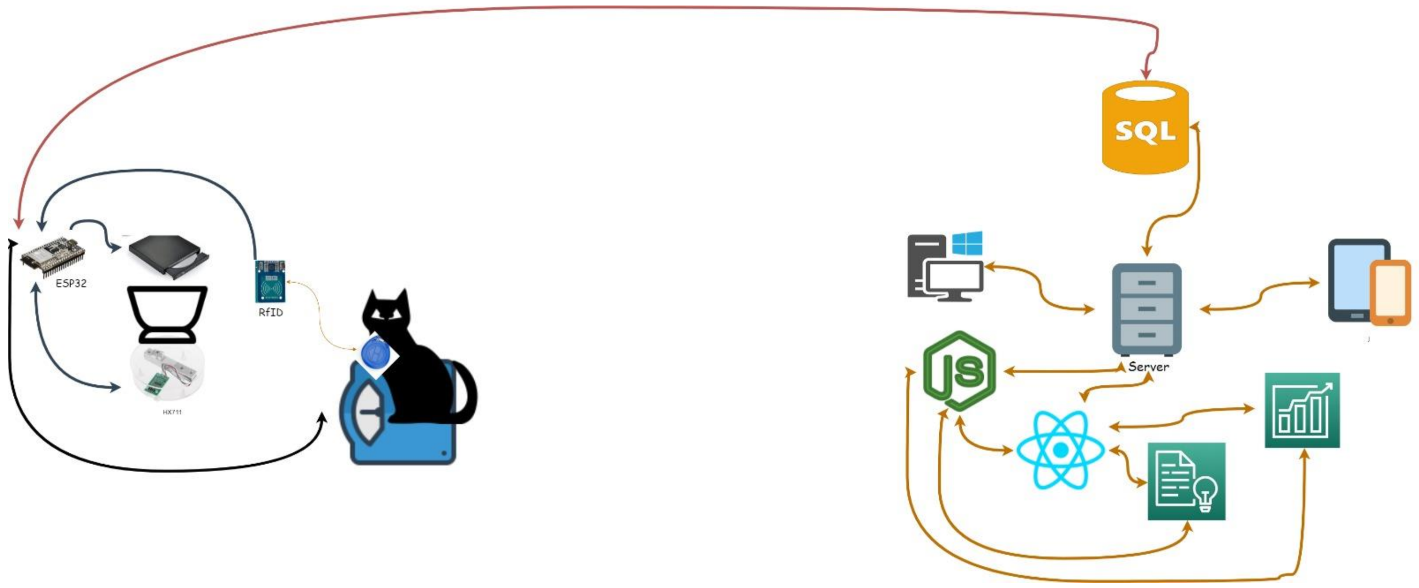

Cat
Name
ID_Cat
Picture
Cat_connect
ID_Cat
Open_Time
Close_Time
Weight
Country
Function: Apprentice time
Learning about cat's eating times by open time
Function: Apprentice Weight
Learning about the usual weight of the cat and identifying if there is a significant increase or decrease
User_Cat
ID_User
ID_Cat
Cat Food Weight
ID_Cat
Time_weight
Weight
Function: Apprentice Food
Learning from the date data and the food weight data, the cat's eating routine
User
Name
Phone Number
Email
ID_User

3.8 Additional Comments

User Explanation Diagram:



Developer Explanation Diagram:



4. Change Management Process
[SKIP THIS PART]

5. Document Approvals
[YOUR SUPERVISOR]

Identify the approvers of the SRS document. Approver name, signature, and date should be used

.

---------------------------------- [END OF RELEVANT PARTS]-------------------------

6. Supporting Information
[SKIP THIS PART]