SE 410 – Senior Design Project 1

System Design Document

Uğur Cem Öztürk	*12070006006*
Pelin Fidangül	*12070001011*
Ahmet Uğur İlter	*13070006026*

Advisors:

Yrd. Doç. Dr. Korhan Karabulut

Yrd. Doç. Dr. İbrahim Zincir

January, 2017

İZMİR

Table of Contents

Salem

A Multi-Featured Smart Pet Feeder

1. Introduction

Most of the pet owners are facing the difficulty of feeding their pets when they are not at home or they forget to feed if they are too busy. Therefore, they usually choose to leave more than enough food for their pets. This can cost more than necessary and yet pets can remain without food for the coming days if they eat the food in one go. One of the options is to ask someone to take care of pets and it is not an available option all the time. Moreover, another issue is the pets try to deal with many problems such as obesity, diabetes, and stomach problems as humans. As a solution to some of these problems, automated feeding machines have been presented due to care of such pets and helping the owners.

1.1 Purpose and Scope

Salem is basically a pet feeder for feeding pets at predetermined times during an owner's absence. The feeders that have been made so far generally dispense only food, not water. Salem will extend this by providing water dispensing capabilities. Also, since they are only based on a timer that triggers the system to fill the bowl with food, they usually lead to food excess if pet does not eat any. In our project, we will design a pet feeder capable of feeding without any food excess and also it will give the owners opportunity to see and check their pets through live web stream.

1.2 Goals and Objectives

Salem's main objectives are being able to determine the amount of food/water inside a pet food bowl in real-time and to reliably provide food or water at the time the owner wishes and to prevent the unwelcome insects by closing the bowl cover in the non-eating time. The bowl cover will be able to open when the pet wants to eat by detecting if it is close to Salem. Meanwhile, it will let the owner watch his/her pet with a mounted camera. This will happen through the Salem's web and mobile application. The system can also keep track of the vaccination of pets and notify the user when vaccination or vet time is getting closer. We seek to build it both functional as well as good looking at the same time.

1.3 References

All the applicable and referenced documents being used in this project are listed below.

- "The Perfect Petfeeder" Internet: http://www.perfectpetfeeder.com/, 2007 [Sep. 16, 2007].
- Own, Chung-Ming, Haw-Yun Shin, and Chen-Ya Teng. "The Study and Application of the Iot in Pet Systems". Advances in Internet of Things 03.01 (2013): 1-8.
- Own, Chung-Ming. "For The Pet Care Appliance of Location Aware Infrastructure on Cyber Physical System". International Journal of Distributed Sensor Networks 2012 (2012): 1-8.
- Kim, Seungcheon. "Smart Pet Care System Using Internet of Things". International Journal of Smart Home 10.3 (2016): 211-218.
- Heil, Rachel et al. The Smart Pet Feeder. Boston: Wentworth Institute of Technology, 2008. Print. The Design and Building of an Automated Pet Feeder Capable of Preventing One Pet from Eating another Pet's Food.

1.4 Overview

The Software Design Document is divided into 7 sections with various subsections. The sections of the Software Design Document are:

1 Introduction: Project description, scope and purpose of the project.

2 System Overview: Briefly explanation of how the system works.

3 System Architecture: List of system components and explanation of how these components interact with each other and subsystem mechanisms.

4 Diagrams: Demonstration of class diagram and activity diagrams.

5 Human Interface Design: Demonstration of mobile and web user interface designs.

6 System Development Environments: Components of both remote server and Raspberry PI environments.

7 Testing: Testing techniques that will be used in Web, server and mobile sides of the project.

2. System Overview

Salem will be based on a Raspberry PI computer as the brain of our system. This type of system comprises a base, a feeding bowl with pie-shaped divisions, a timer module, a bowl cover, and a locking mechanism to hold the entire unit in place. The timer provides programmed feeding schedules that determine the time the bowl cover closes or opens. Food dispenser mechanism will work as a system that works automatically according to determined schedule by the owner. There will be also a camera for live stream, two storage bins for food and water, RFID tag reader onto pets' collar and a LCD screen which displays food and water levels in the storage.

We will use RFID tag to close or open the bowl cover. This will be embedded onto pets' collars to determine if pet is eating or not. When pet is not around the feeder, the bowl cover will close automatically. This keeps dust, flies, and bugs from reaching the food and keeps the food fresh. The bowl cover is actuated by a battery-operated electric motor.

Users will be able to set their own schedule for feeding using a web interface and a mobile application. The interface also will be designed to include user login page, pet profiles (consist of age of the pet, last vaccination date and type, sicknesses and their dates), live stream and to display levels of food and water in the storage. In addition, user will be able to upload any kind of sound file. For example, greeting voice of an owner to alert the eating time can be uploaded to the system which will play it automatically as scheduled via a speaker.

If owner has more than one pet, each pet will have a unique profile. All necessary data will be kept on a database; some of these data will be entered by the pet owner manually.

There will be stepper motors controlling some mechanic parts through an Arduino to dispense food and water to pet bowls.

All of the data will be processed on Linux operating system on top of our Raspberry PI. Live video will be streamed to both our web interface and mobile application using a web server.

3. System Architecture

3.1 Connection

First and foremost, the Raspberry PI inside SALEM has a CRON job running on the background that records or updates its "Global IP Address" and "MAC ID" to a DB running on the remote server. This remote server will work as a bridge between the user and their product and will solve the problem caused by Dynamic IP Address services given by the ISPs.

As soon as the user connects to the Web Application (i.e. http://www.salem.com.tr/login) or Mobile Application, they will be welcomed by a login screen. With valid credentials provided, either application will be replied with IP addresses related with the user.

Once the information including the IP address is gathered, user will be redirected to the home screen of related application. All relevant information on this and further screens will be provided through the web service running on the Raspberry PI using the afore mentioned IP address. Please refer to Figure 1 for the system connection.

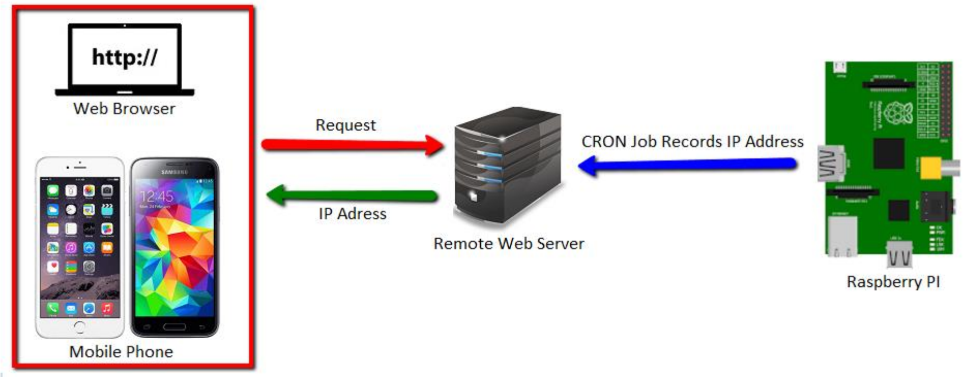


Figure 1.Redirection of the user to the feeder

3.2 System Components

The base of our system is a credit card sized computer, namely Raspberry PI. Other components included are;

- 1x Arduino Nano (for reading sensor data),
- 1x Web Camera (for live video streaming),
- 4x Load cells (to measure food and water levels inside bowls and storages),
- 1x Servo Motor (for food dispensing mechanism),
- 1x Solenoid Valve (for water dispensing mechanism),
- 2x Stepper Motors (to open and close bowl covers),
- 1x LCD Display (to display food and water percentages left inside the storages),
- 1x Passive RFID tag and Reader (to toggle bowl covers per pet proximity)
- 1x Red and Green LED (to indicate storage levels inside the bins)
- 1x Raspberry PI

3.3 Subsystems

3.3.1 Webcam and Live Stream

Salem will use software called "Motion" for webcam live streaming. Motion is a free software application developed for Linux, mostly used for CCTV solutions. Once the system is up and running, live stream will start subsequently. At any time, user can watch the stream from the embedded player on the home page.

3.3.1 Cover Mechanism

The bowl cover will be actuated by the RFID reader. The reader detects the presence of the pet and then triggers the cover to open it, let the pet eat or drink. When the pet is out of RFID range, the bowl cover closes automatically.

RFID tags will be attached to the pet's collar and an RFID reader will be placed in front of each bowl. RFID reader reports the presence or absence of the pet to Raspberry PI and then it triggers the stepper motors to open/close the covers.

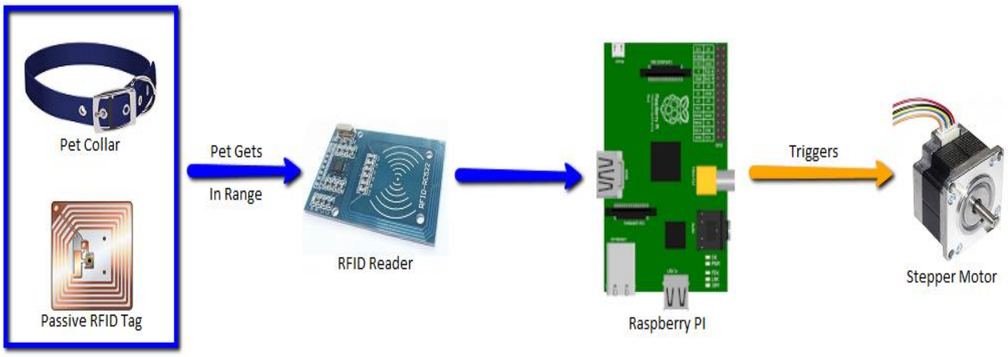


Figure 9.How RFID works to open/close the bowl covers

3.3.2 Food and Water Dispensing

The water and food dispensing subsystem will be actuated in daily scheduled times by Raspberry PI. The servo motor will take place in opening/closing the food storage bin’s lid when the eating time is indicated. (Figure 8a) For water dispensing, solenoid valve will be used. Please refer to Figure 8b for the dispensing mechanism.

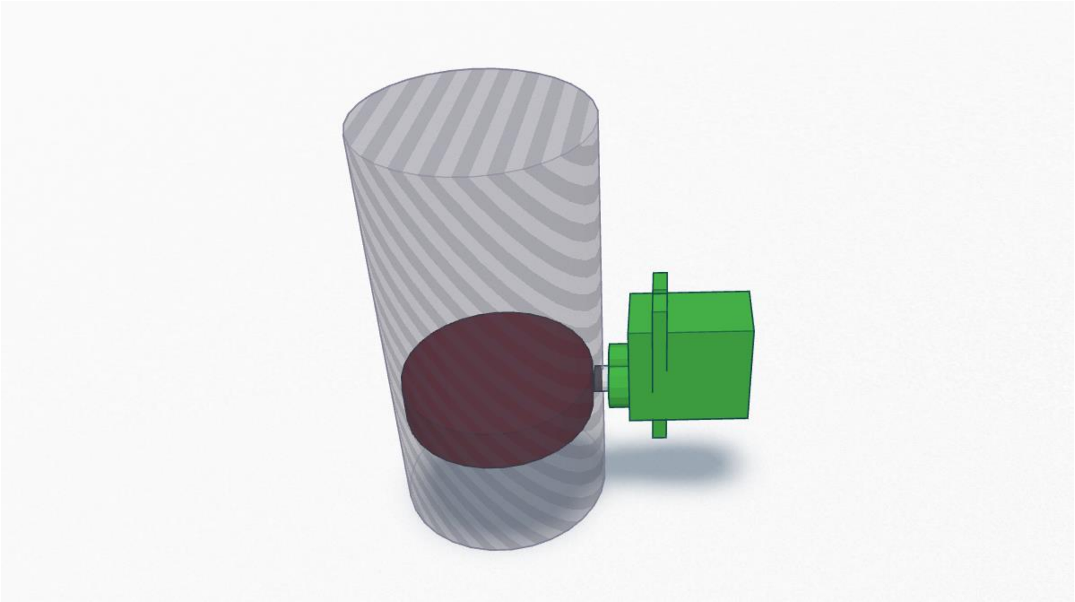


Figure 8a.Food storage bin’s lid with servo motor

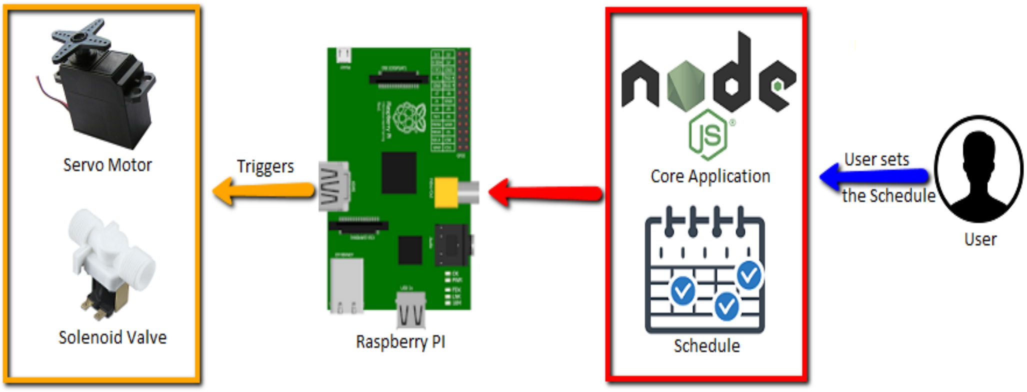


Figure 8b.How dispensing mechanism works

3.3.3 Storage Mechanism

Salem has two storage bins inside to store food and water which are shown in Figure 7. The levels of both food and water inside the bins will be tracked in real-time using load cells. If levels go below the specified threshold values, a notification will appear on LCD Display stating that related storage is below the critical level and a red LED will start blinking.

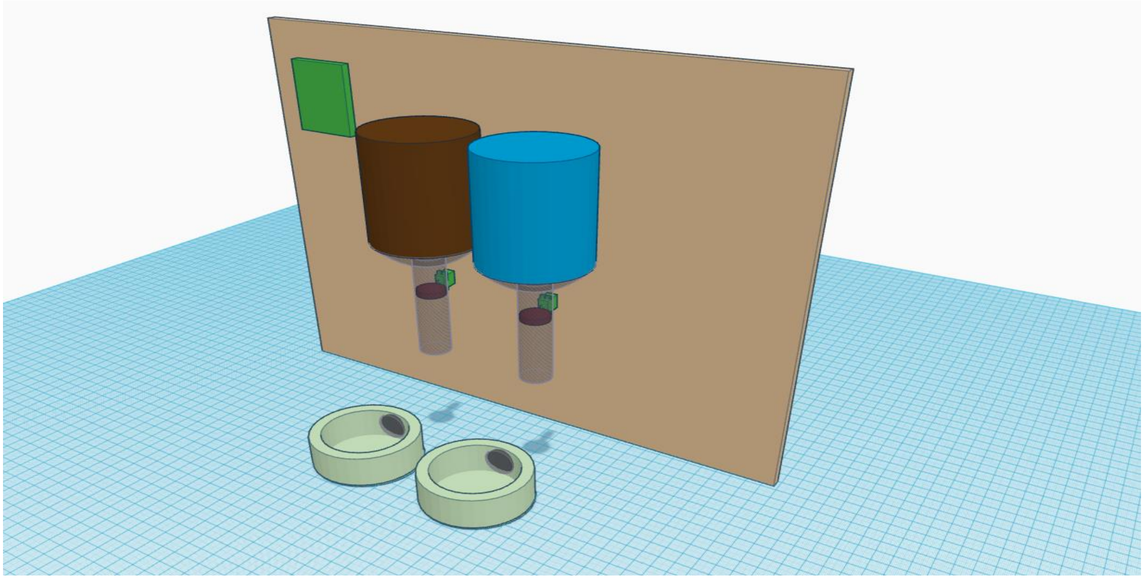
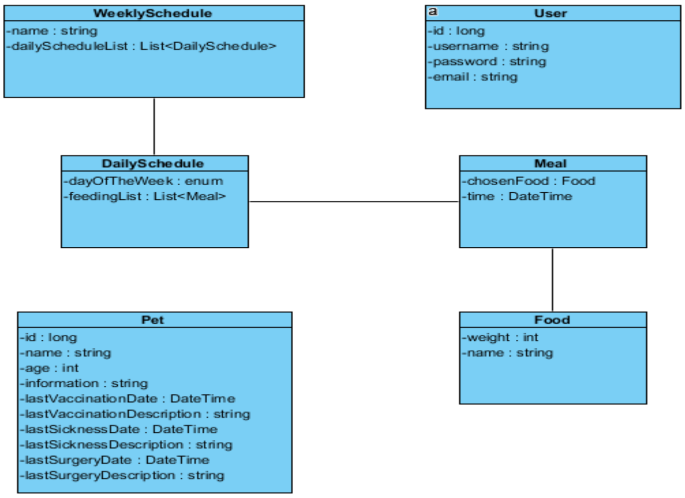


Figure 7.Demonstration of food and water bins inside the feeder

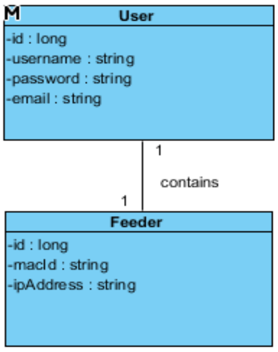
4. Diagrams

4.1 Class Diagram

- Class diagram of Salem web and mobile applications run at the Raspberry PI;

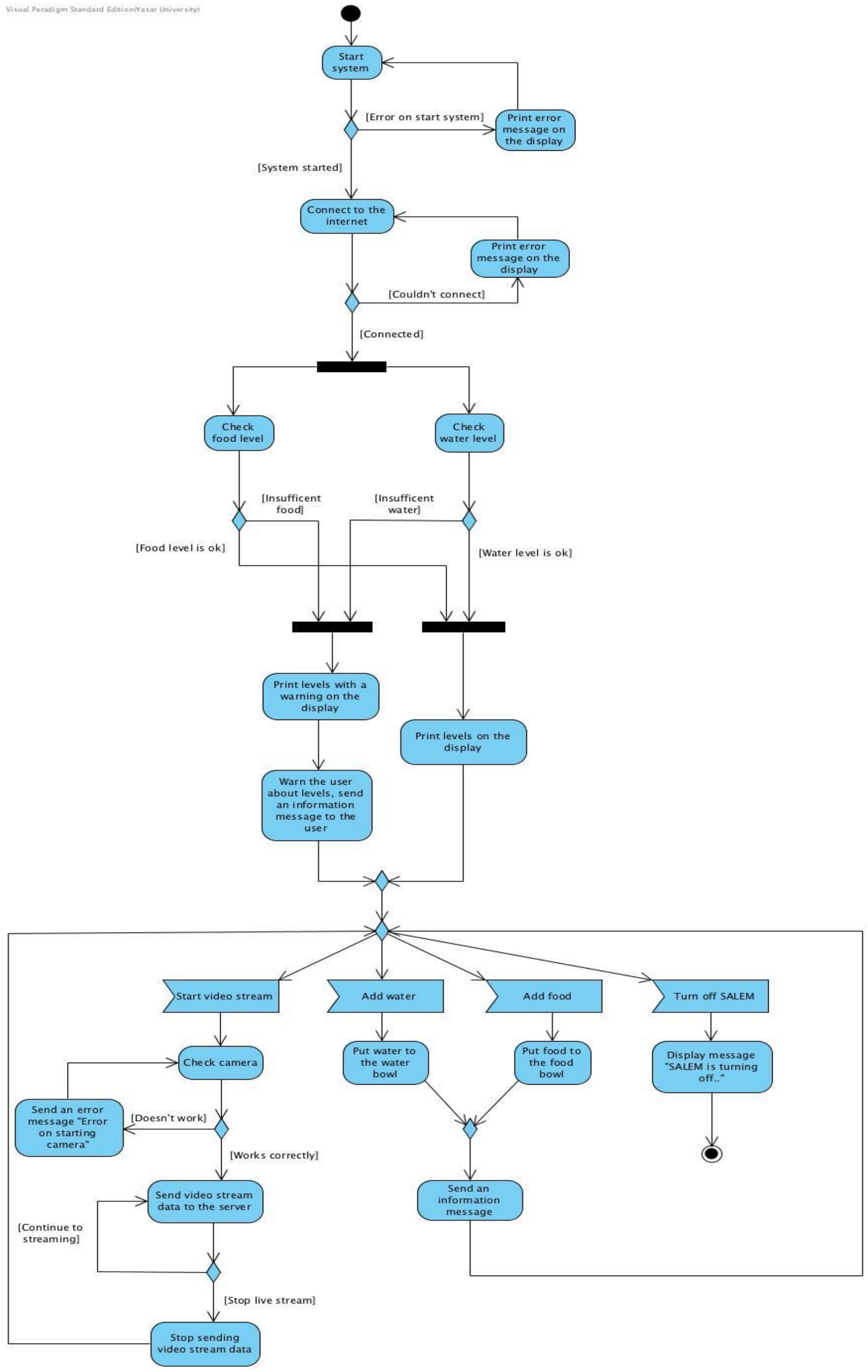


- Class diagram of the code on the remote server side;

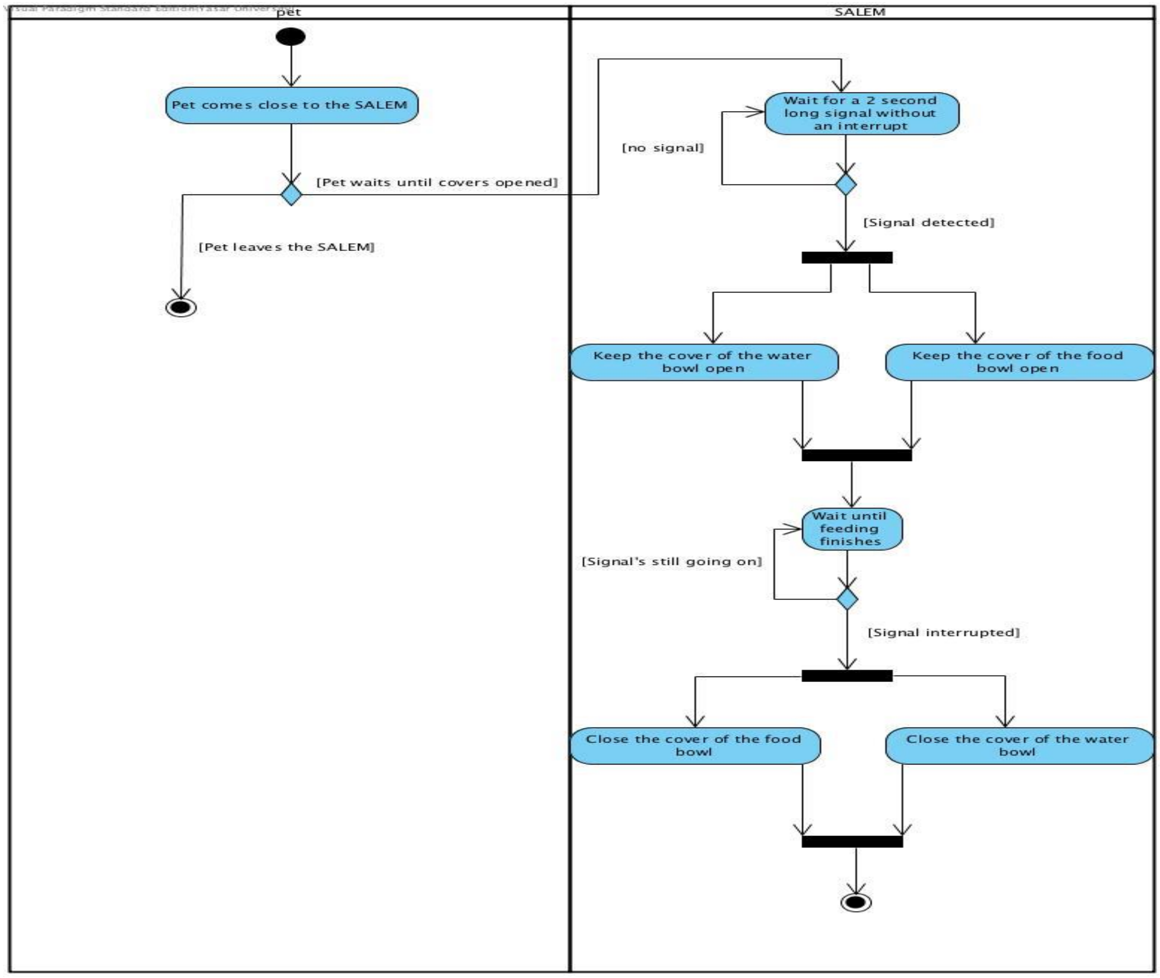


4.2 Activity Diagram

- Salem main activity;


Visual Paradigm Standard Edition(Yasar University)
Start system
[Error on start system]
Print error message on the display
[System started]
Connect to the internet
Print error message on the display
[Couldn't connect]
[Connected]
Check food level
Check water level
[Insufficent food]
[Insufficent water]
[Food level is ok]
[Water level is ok]
Print levels with a warning on the display
Print levels on the display
Warn the user about levels, send an information message to the user
Start video stream
Add water
Add food
Turn off SALEM
Check camera
Put water to the water bowl
Put food to the food bowl
Display message "SALEM is turning off.."
Send an error message "Error on starting camera"
[Doesn't work]
[Works correctly]
Send video stream data to the server
Send an information message
[Continue to streaming]
[Stop live stream]
Stop sending video stream data

- RFID signal reading;


pet
SALEM
Pet comes close to the SALEM
[Pet waits until covers opened]
[Pet leaves the SALEM]
Wait for a 2 second long signal without an interrupt
[no signal]
[Signal detected]
Keep the cover of the water bowl open
Keep the cover of the food bowl open
Wait until feeding finishes
[Signal's still going on]
[Signal interrupted]
Close the cover of the food bowl
Close the cover of the water bowl

5. Human Interface Design

The user interface overview of the project along with explanations is demonstrated below.

- Login Page

Users can log in with username and password, which is illustrated in Figure 2. Only the administrator can change the IP setting of feeder.

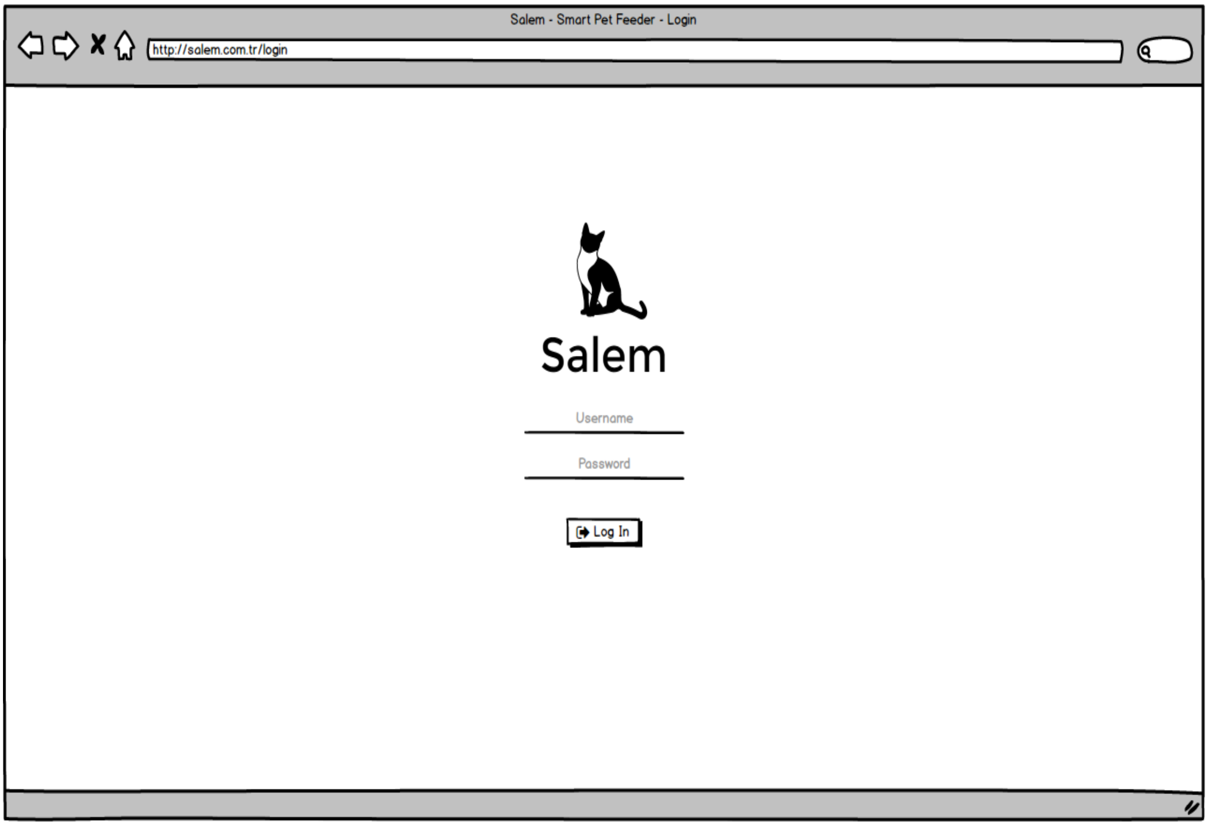


Figure 2.Web Login Page

- Home Page

  Users will be able to see their pets through a web cam mounted on the system and also to see the level of food and water inside the storage section, after they logged in. Please refer to Figure 3 for web application and to Figure 4 for mobile application home page interface design.

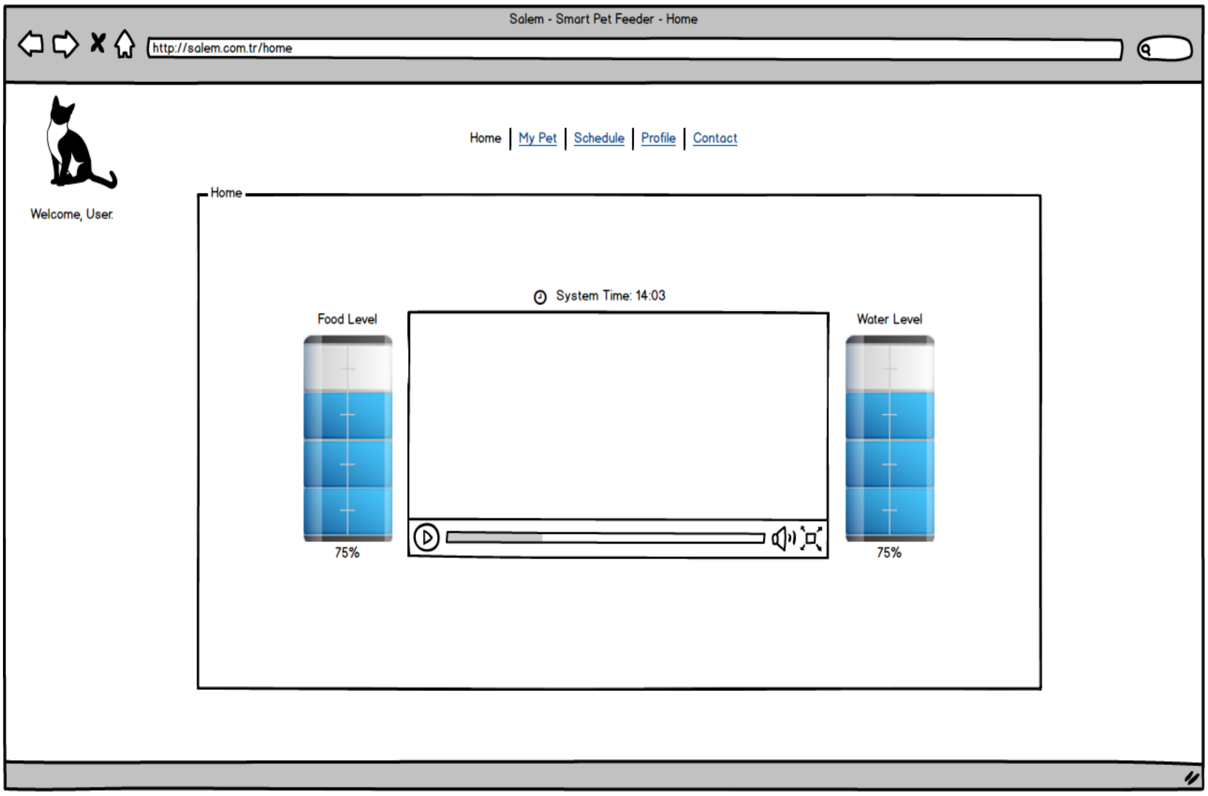


Figure 3.Salem web application home page

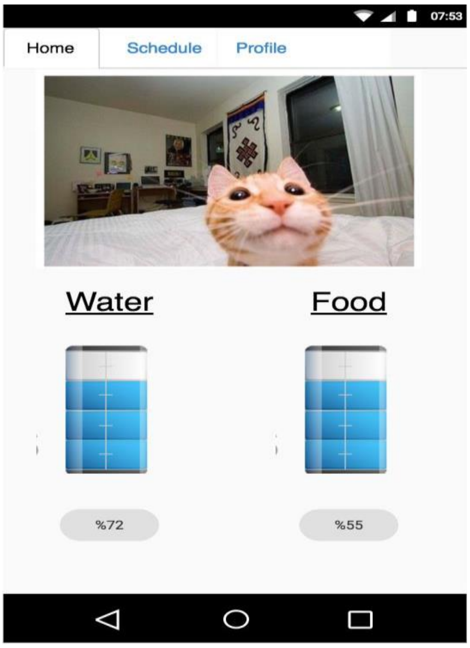
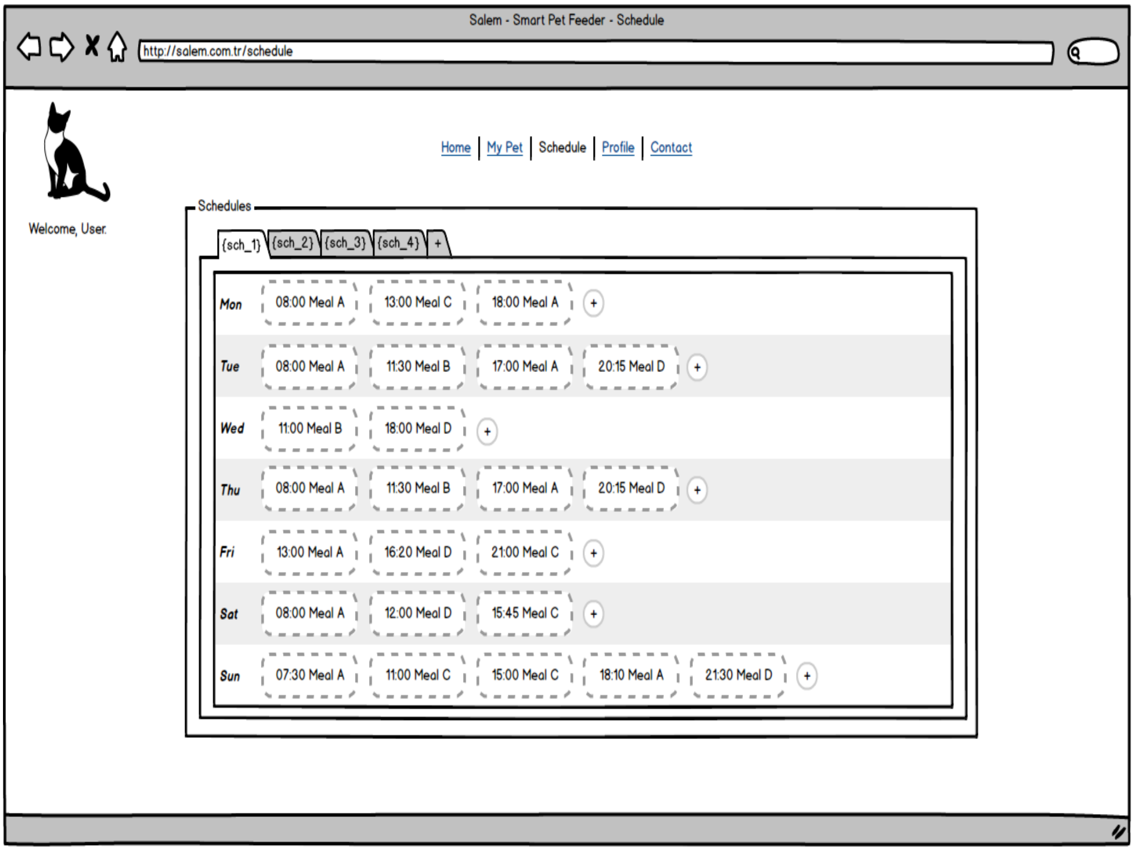


Figure 4.Salem mobile application home page

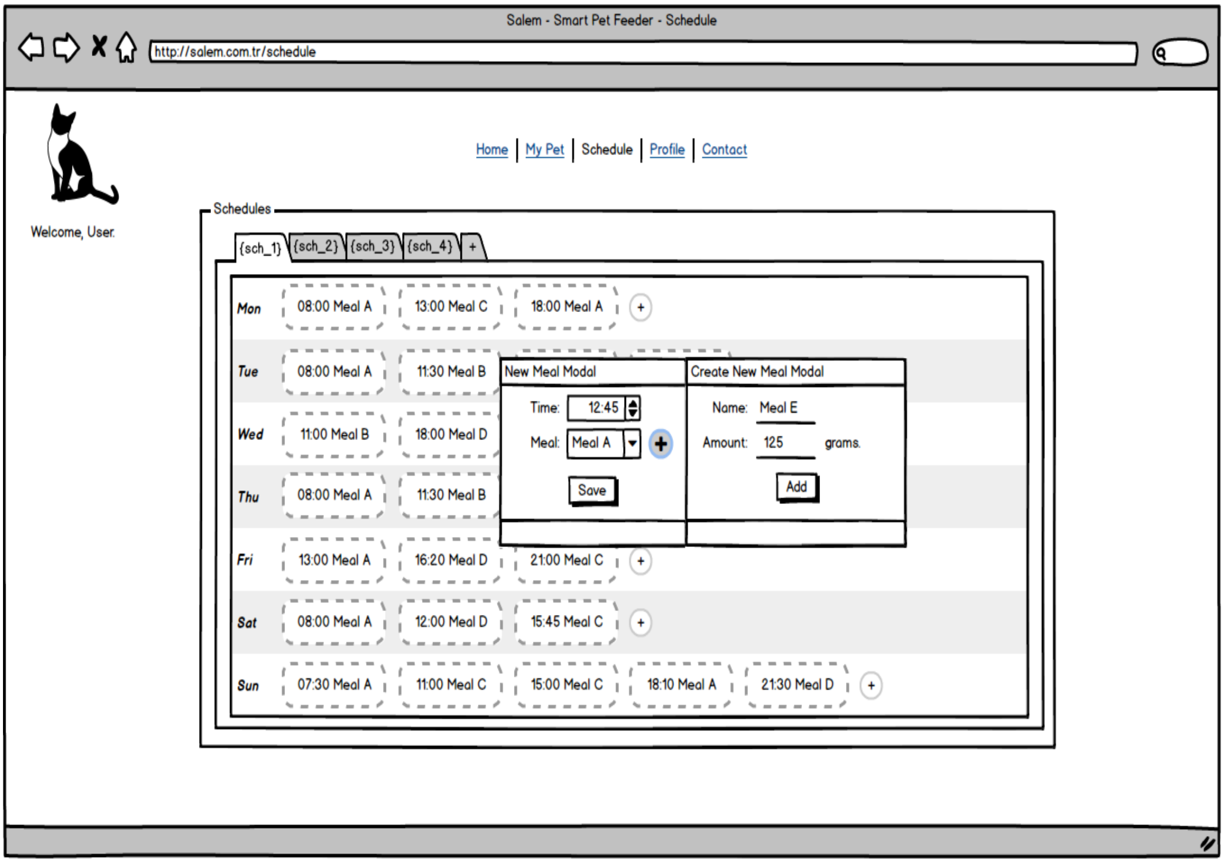
- Schedule Section

Users can set the feeding time and amount of the food. They will have a time table which is shown in Figure 5a, Figure 5b and Figure 5c. This interface will let them do these arrangements even they are away from the machine.


Salem - Smart Pet Feeder - Schedule
http://salem.com.tr/schedule
Welcome, User.
Home | My Pet | Schedule | Profile | Contact
Schedules
{sch_1} {sch_2} {sch_3} {sch_4} +

Mon	08:00 Meal A	13:00 Meal C	18:00 Meal A	+		
Tue	08:00 Meal A	11:30 Meal B	17:00 Meal A	20:15 Meal D	+	
Wed	11:00 Meal B	18:00 Meal D	+			
Thu	08:00 Meal A	11:30 Meal B	17:00 Meal A	20:15 Meal D	+	
Fri	13:00 Meal A	16:20 Meal D	21:00 Meal C	+		
Sat	08:00 Meal A	12:00 Meal D	15:45 Meal C	+		
Sun	07:30 Meal A	11:00 Meal C	15:00 Meal C	18:10 Meal A	21:30 Meal D	+

Figure 5a.The schedule table of feeding set by the user

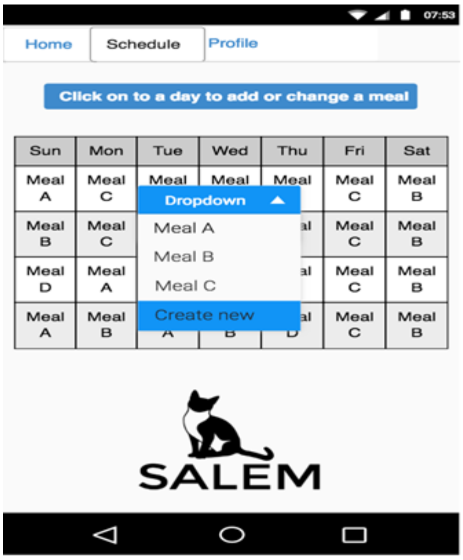

Salem - Smart Pet Feeder - Schedule
http://salem.com.tr/schedule
Welcome, User.
Home | My Pet | Schedule | Profile | Contact
Schedules
{sch_1} {sch_2} {sch_3} {sch_4} +
New Meal Modal
Create New Meal Modal
Save
Add

Mon	08:00 Meal A	13:00 Meal C	18:00 Meal A	+		
Tue	08:00 Meal A	11:30 Meal B				
Wed	11:00 Meal B	18:00 Meal D				
Thu	08:00 Meal A	11:30 Meal B				
Fri	13:00 Meal A	16:20 Meal D	21:00 Meal C	+		
Sat	08:00 Meal A	12:00 Meal D	15:45 Meal C	+		
Sun	07:30 Meal A	11:00 Meal C	15:00 Meal C	18:10 Meal A	21:30 Meal D	+

Time: 12:45
Meal: Meal A

Name: Meal E
Amount: 125 grams.

Figure 5b.Module to let user set time and amount of food

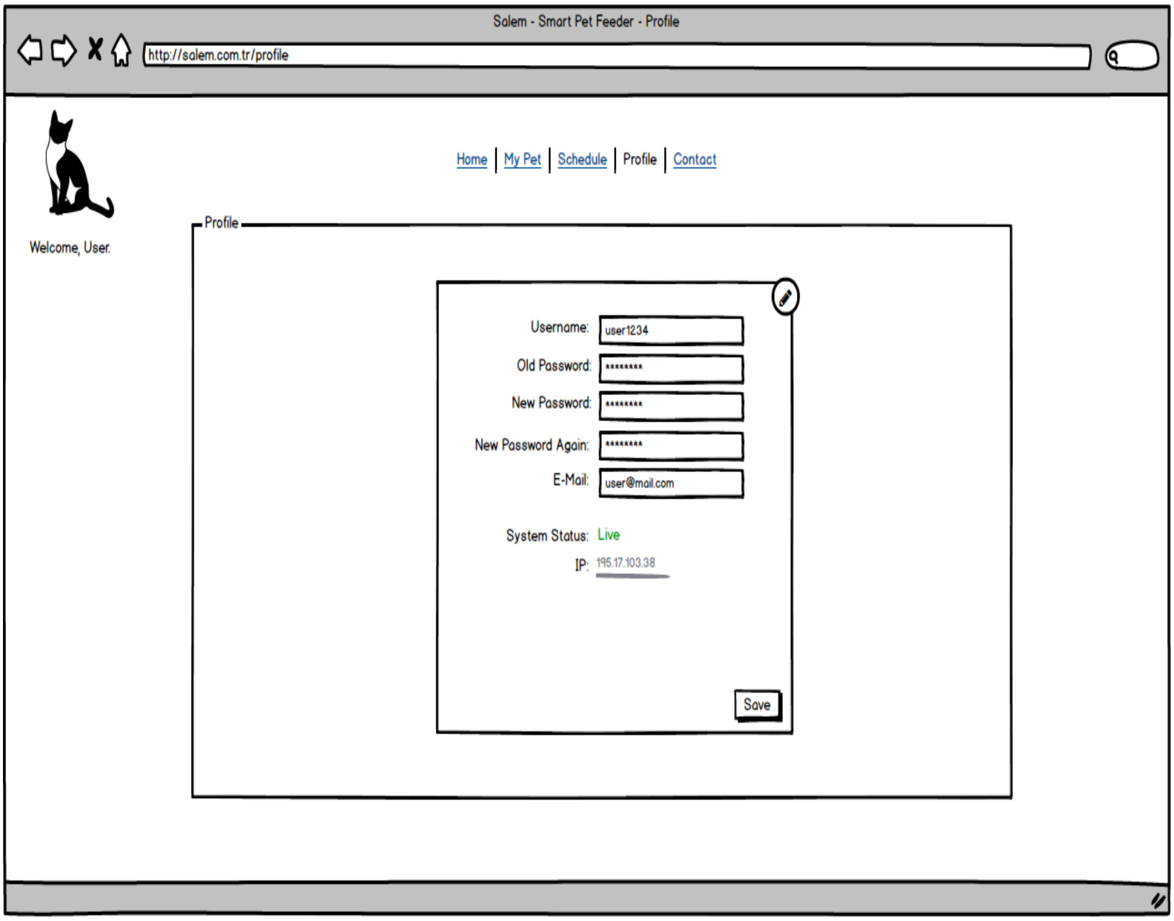


Sun	Mon	Tue	Wed	Thu	Fri	Sat
Meal A	Meal C	Meal	Meal	Meal	Meal C	Meal B
Meal B	Meal C			al	Meal C	Meal B
Meal D	Meal A			al	Meal C	Meal B
Meal A	Meal B	A	B	al D	Meal C	Meal B

Figure 5c.Schedule table illustration for Salem mobile application

- User Profile Page

  Users will be able to change the username, password and email, if they wish. Please refer to Figure 6 for illustration.


Salem - Smart Pet Feeder - Profile
http://salem.com.tr/profile
Home | My Pet | Schedule | Profile | Contact
Welcome, User.
Profile
Save

Username: user1234
Old Password: ********
New Password: ********
New Password Again: ********
E-Mail: user@mail.com

System Status: Live
IP: 195.17.103.38

Figure 6.User profile page

6. System Development Environment

Raspberry PI:
- OS: Raspbian PIXEL (Lightweight Linux distro for Raspbery PI)

- Necessary Software:

- Motion (Live Streaming)
- PostgreSQL or MySQL (DBMS)
- Tomcat or Apache (Web Server)
- Python
- Node.js

Remote Web Server:

- OS: Ubuntu

- Necessary Software:

- Tomcat or Apache (Web Server)
- PostgreSQL or MySQL (DBMS)

7. Testing

On the testing process, we will use unit, GUI and cross platform browser testing tools for the web page. On the mobile applications side, we will use native application testing tools. Tests will be done automated and manual. Load, stress and security tests will be applied on the remote server. The testing tools will be as follows:

7.1 Web

- Selenium IDE -> GUI Testing
- BrowserStack -> Cross Platform Browser Testing

7.2 Mobile

7.2.1 Android

- Robotium -> Android UI Test

- Appium -> Test automation framework for IOS and Android

7.2.2 IOS

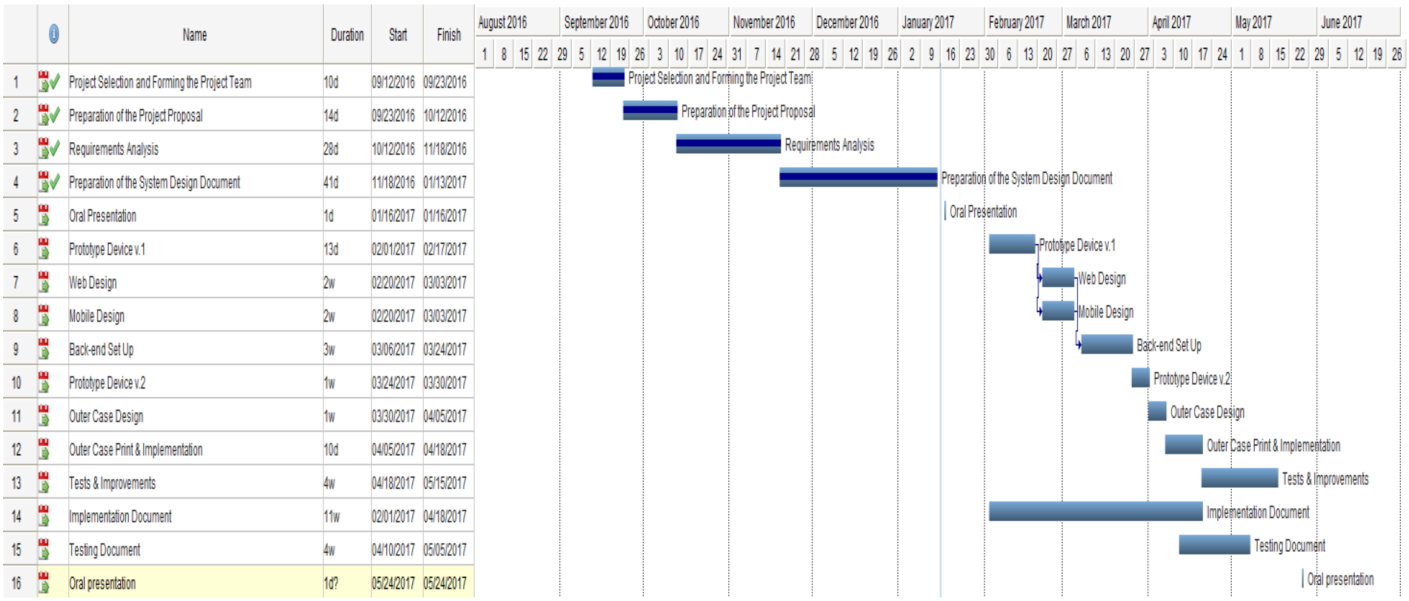
- XCTest -> Xcode testing framework
- Appium -> Test automation framework for IOS and Android

7.3 Remote Server

- JMeter -> Load & Stress Testing
- Detectify -> Automated web security test server

8. Project Schedule

The Gantt chart demonstrated below represents the timeline of completed and future tasks of the project.



		Name	Duration	Start	Finish
1	✓	Project Selection and Forming the Project Team	10d	09/12/2016	09/23/2016
2	✓	Preparation of the Project Proposal	14d	09/23/2016	10/12/2016
3	✓	Requirements Analysis	28d	10/12/2016	11/18/2016
4	✓	Preparation of the System Design Document	41d	11/18/2016	01/13/2017
5		Oral Presentation	1d	01/16/2017	01/16/2017
6		Prototype Device v.1	13d	02/01/2017	02/17/2017
7		Web Design	2w	02/20/2017	03/03/2017
8		Mobile Design	2w	02/20/2017	03/03/2017
9		Back-end Set Up	3w	03/06/2017	03/24/2017
10		Prototype Device v.2	1w	03/24/2017	03/30/2017
11		Outer Case Design	1w	03/30/2017	04/05/2017
12		Outer Case Print & Implementation	10d	04/05/2017	04/18/2017
13		Tests & Improvements	4w	04/18/2017	05/15/2017
14		Implementation Document	11w	02/01/2017	04/18/2017
15		Testing Document	4w	04/10/2017	05/05/2017
16		Oral presentation	1d?	05/24/2017	05/24/2017