Smart Pot

An intelligent way to manage gardens

Mansi KARODE Rahul KUMAR

Marcelo ARDILES Blessie DURAIRAJ Naveen RAJAPPA

Introduction

“Know more, grow more”

Who are we: greentech

- Founded in 2013
- Located in Paris with a team of 35 employees.
- Area of expertise:Technologies involving sensors and embedded systems with application development
- Some achievements:

  In 2015 Go green-innovative award
  Member of Green Code Technology
  Member of GE Global research and innovation technology
- We believe in:

Innovation = Curiosity + Idea + Team + Nature driven + Technology → *Satisfaction*

Introduction

Our Team:

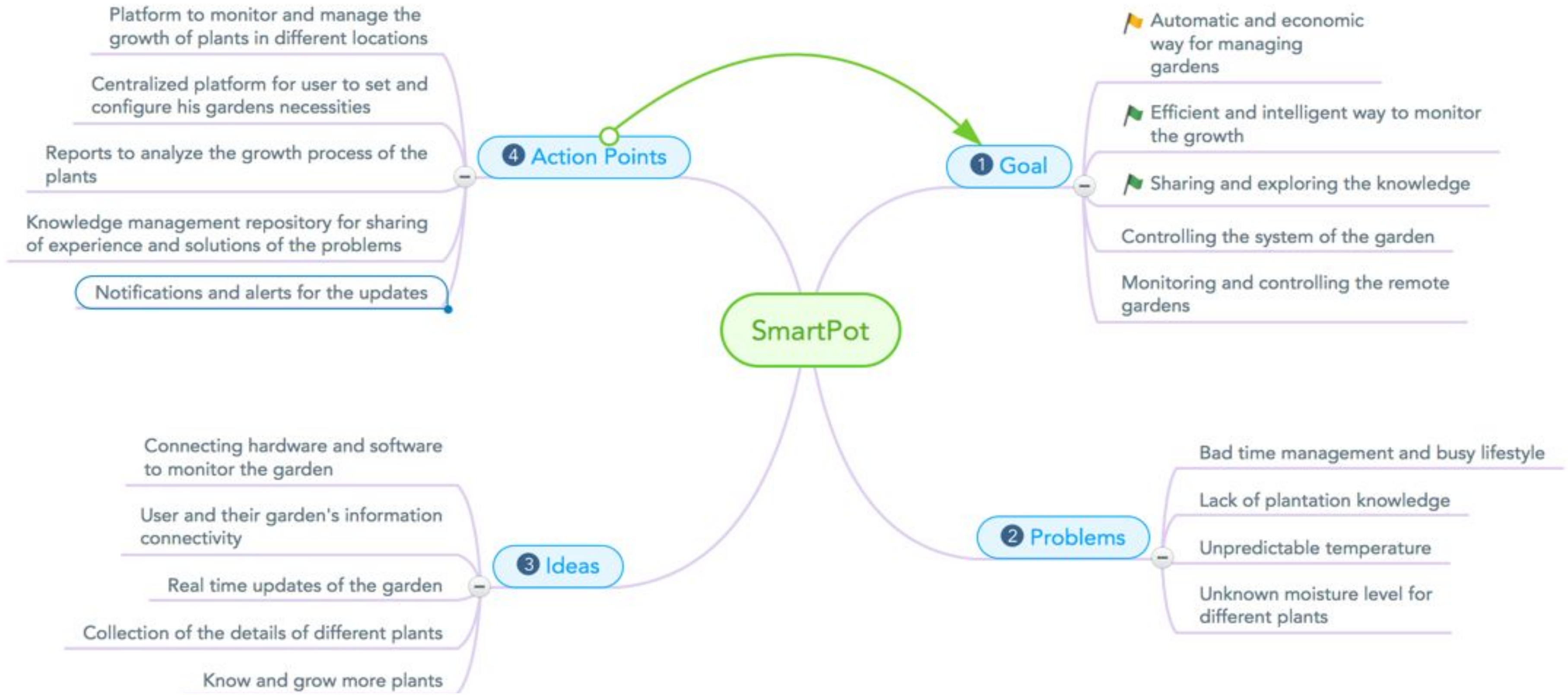
- Olivier Berthet : Mentor
- JF Bonnet : Sponsor
- Mansi Karode : Project Manager (ISM @ Epita)
- Marcelo Ardiles : Embedded system and security expert (Computer security @ Epita)
- Rahul Kumar Thai Valappil : Software development expert / UX UI (SE @Epita)
- Naveen Rajappa : Software developer (SE @Epita)
- Blessie Durairaj : Software developer (SE @Epita)

- Xavier Dupont : Botanist
- Brent Wagner : Sales and marketing expert

Jean Francais Bonnet
(Ceo and Sponsor)

Project Overview

Goal:An innovative solution with an automated and economical way to manage gardens.



SmartPot

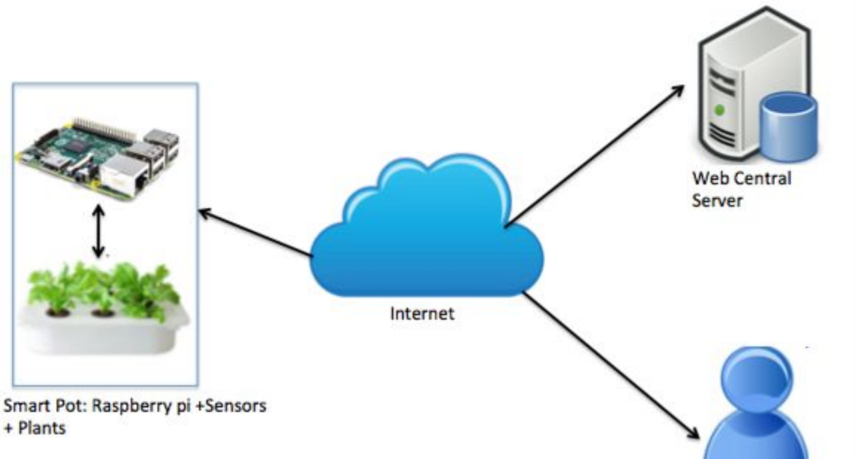
1 Goal
- Automatic and economic way for managing gardens
- Efficient and intelligent way to monitor the growth
- Sharing and exploring the knowledge
- Controlling the system of the garden
- Monitoring and controlling the remote gardens

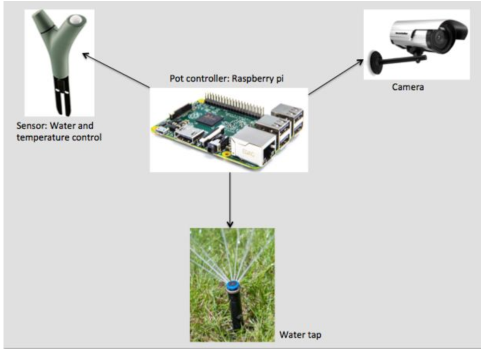
2 Problems
- Bad time management and busy lifestyle
- Lack of plantation knowledge
- Unpredictable temperature
- Unknown moisture level for different plants

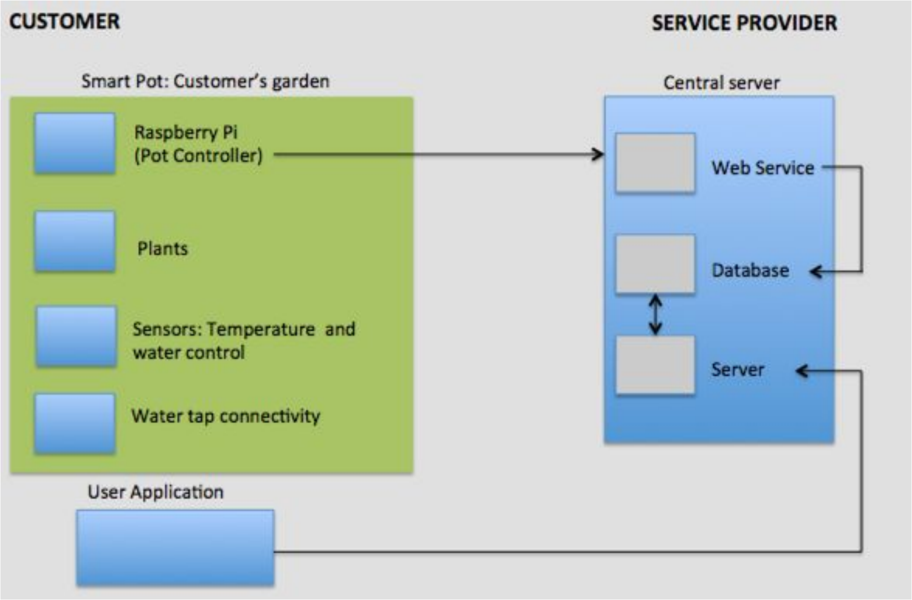
3 Ideas
- Connecting hardware and software to monitor the garden
- User and their garden's information connectivity
- Real time updates of the garden
- Collection of the details of different plants
- Know and grow more plants

4 Action Points
- Platform to monitor and manage the growth of plants in different locations
- Centralized platform for user to set and configure his gardens necessities
- Reports to analyze the growth process of the plants
- Knowledge management repository for sharing of experience and solutions of the problems
- Notifications and alerts for the updates

How to do this : Architecture


Smart Pot: Raspberry pi +Sensors + Plants
Internet
Web Central Server
End user-Web or Mobile Application


Sensor: Water and temperature control
Pot controller: Raspberry pi
Camera
Water tap



Tools and Technology

Project Management	Embedded system	Application Development
● ProjectLibre ● Google Slides ● Google Documents ● Daily Meeting	● Raspberry Pie (Nano Pi M1) ● Sensors (NHT11) : Temperature and humidity ● Debian Operating System ● Server:Ubuntu ● Port knocking ● IP Tables ● Firewalls ● HTPC Server ● Certificates for authentication	● IDE : Eclipse Mars ● Framework : Maven, Hibernate ● Frontend : JavaScripts, CSS, Bootstrap ● Database : Mysql ● Server : Apache Tomcat 7 ● DAO Methods ● Java programming language ● Testing :JUnits

Project Scope

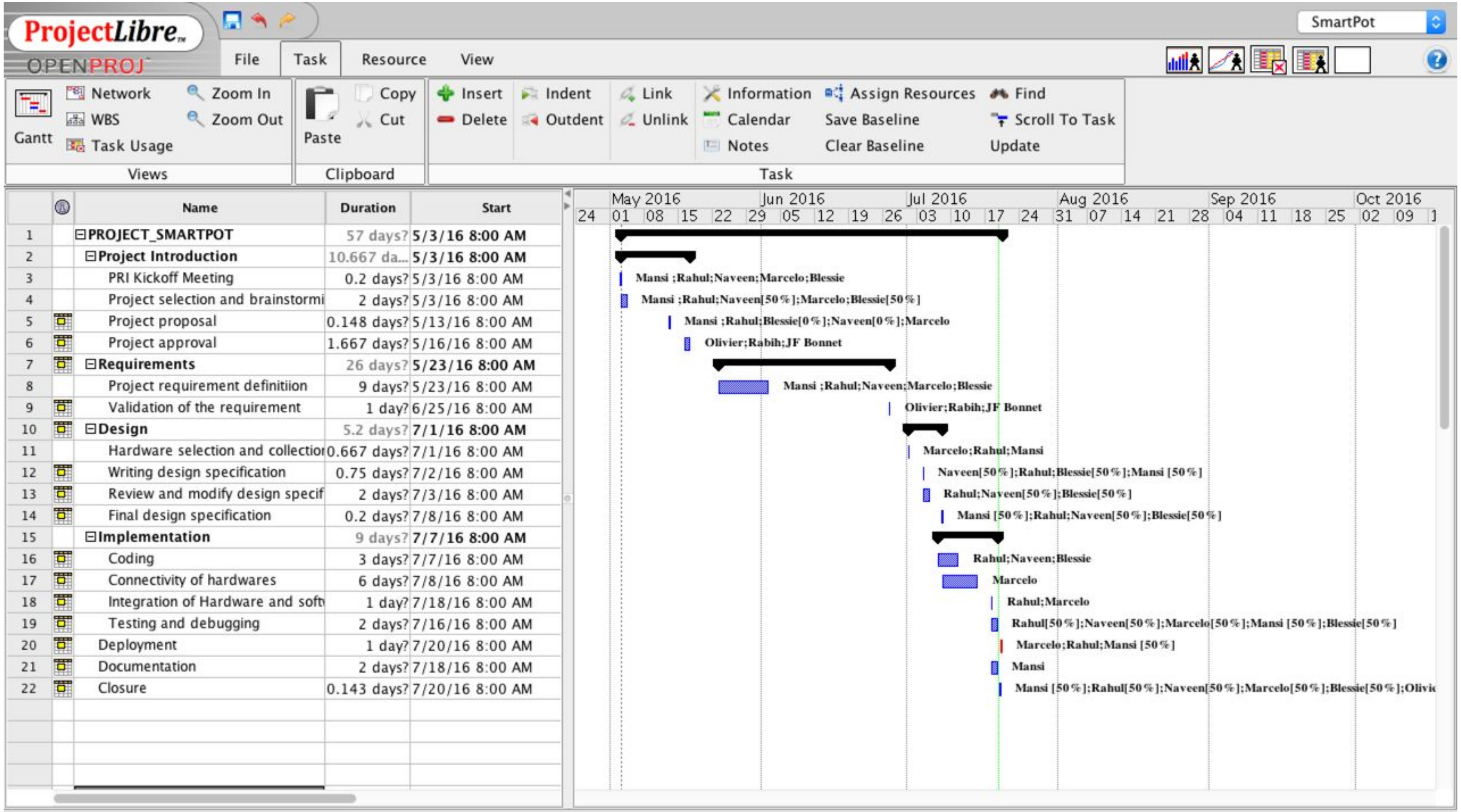
- Registration and login module
- User profile module
- User Edit profile
- Pot details module
- Temperature/Humidity set module
- Logout Module
- Gallery Module
- Forum Module
- Report and dashboard Module
- Automate tap control

Project Management

- Software : Project Libre
- Clear idea of the concept
- Planning and scheduling the process
- Daily meetings
- Open discussion of project
- Knowledge sharing
- Work distribution as per the competencies
- Time management

Project Libre™

Project Management:Project Libre



		Name	Duration	Start
1		⊟PROJECT_SMARTPOT	57 days?	5/3/16 8:00 AM
2		⊟Project Introduction	10.667 da...	5/3/16 8:00 AM
3		PRI Kickoff Meeting	0.2 days?	5/3/16 8:00 AM
4		Project selection and brainstormi	2 days?	5/3/16 8:00 AM
5	▦	Project proposal	0.148 days?	5/13/16 8:00 AM
6	▦	Project approval	1.667 days?	5/16/16 8:00 AM
7	▦	⊟Requirements	26 days?	5/23/16 8:00 AM
8		Project requirement definitiion	9 days?	5/23/16 8:00 AM
9	▦	Validation of the requirement	1 day?	6/25/16 8:00 AM
10	▦	⊟Design	5.2 days?	7/1/16 8:00 AM
11		Hardware selection and collectio	0.667 days?	7/1/16 8:00 AM
12	▦	Writing design specification	0.75 days?	7/2/16 8:00 AM
13	▦	Review and modify design specif	2 days?	7/3/16 8:00 AM
14	▦	Final design specification	0.2 days?	7/8/16 8:00 AM
15		⊟Implementation	9 days?	7/7/16 8:00 AM
16	▦	Coding	3 days?	7/7/16 8:00 AM
17	▦	Connectivity of hardwares	6 days?	7/8/16 8:00 AM
18	▦	Integration of Hardware and softv	1 day?	7/18/16 8:00 AM
19	▦	Testing and debugging	2 days?	7/16/16 8:00 AM
20	▦	Deployment	1 day?	7/20/16 8:00 AM
21	▦	Documentation	2 days?	7/18/16 8:00 AM
22	▦	Closure	0.143 days?	7/20/16 8:00 AM

Mansi ;Rahul;Naveen;Marcelo;Blessie

Mansi ;Rahul;Naveen[50 %];Marcelo;Blessie[50 %]

Mansi ;Rahul;Blessie[0 %];Naveen[0 %];Marcelo

Olivier;Rabih;JF Bonnet

Mansi ;Rahul;Naveen;Marcelo;Blessie

Olivier;Rabih;JF Bonnet

Marcelo;Rahul;Mansi

Naveen[50 %];Rahul;Blessie[50 %];Mansi [50 %]

Rahul;Naveen[50 %];Blessie[50 %]

Mansi [50 %];Rahul;Naveen[50 %];Blessie[50 %]

Rahul;Naveen;Blessie

Marcelo

Rahul;Marcelo

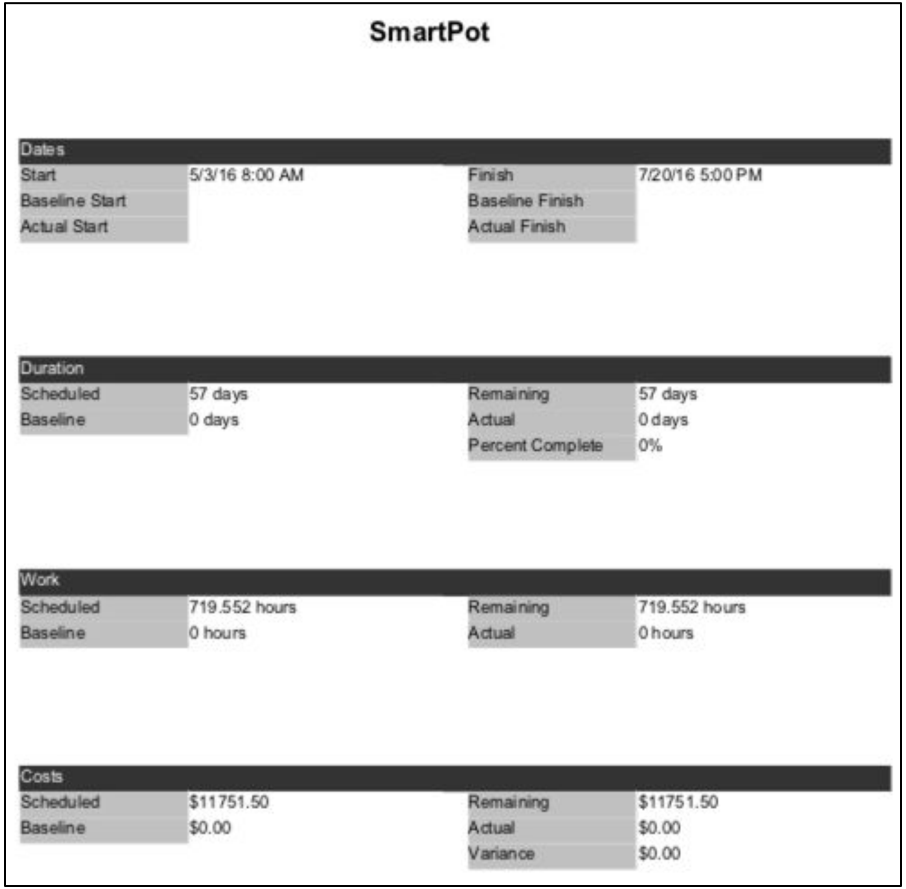
Rahul[50 %];Naveen[50 %];Marcelo[50 %];Mansi [50 %];Blessie[50 %]

Marcelo;Rahul;Mansi [50 %]

Mansi

Mansi [50 %];Rahul[50 %];Naveen[50 %];Marcelo[50 %];Blessie[50 %];Olivie

Project Management:Project Libre



SmartPot

Dates			
Start	5/3/16 8:00 AM	Finish	7/20/16 5:00 PM
Baseline Start		Baseline Finish	
Actual Start		Actual Finish	

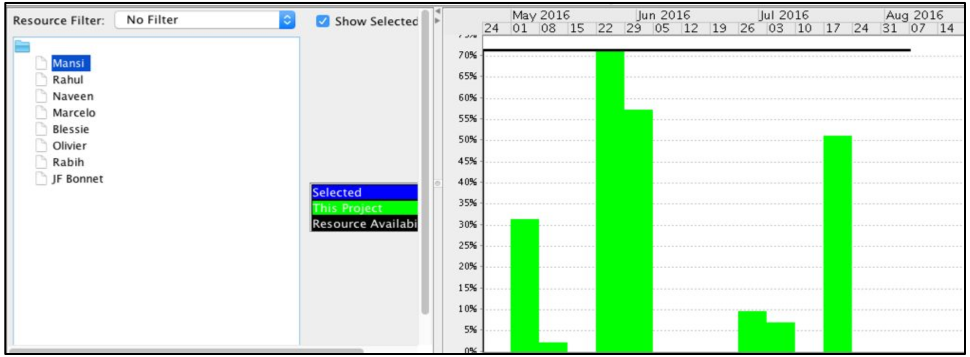
Duration			
Scheduled	57 days	Remaining	57 days
Baseline	0 days	Actual	0 days
		Percent Complete	0%

Work			
Scheduled	719.552 hours	Remaining	719.552 hours
Baseline	0 hours	Actual	0 hours

Costs			
Scheduled	$11751.50	Remaining	$11751.50
Baseline	$0.00	Actual	$0.00
		Variance	$0.00

Advantages of ProjectLibre

- Open source
- Agile Project Management
- Reporting
- Issue Tracking
- Task and resource management
- Cost management

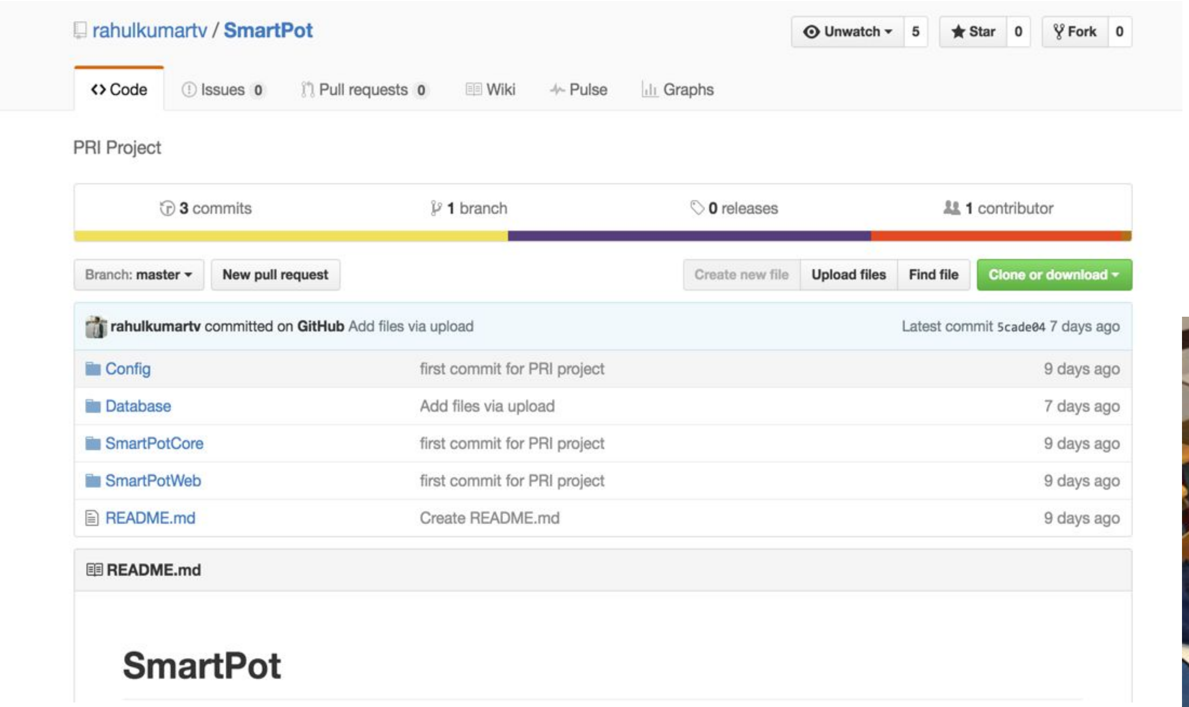


Resource Filter: No Filter

☑ Show Selected

- Mansi
- Rahul
- Naveen
- Marcelo
- Blessie
- Olivier
- Rabih
- JF Bonnet

Project Management:Code exchange



rahulkumartv / SmartPot

PRI Project

3 commits · 1 branch · 0 releases · 1 contributor

rahulkumartv committed on GitHub Add files via upload — Latest commit 5cade04 7 days ago

Config	first commit for PRI project	9 days ago
Database	Add files via upload	7 days ago
SmartPotCore	first commit for PRI project	9 days ago
SmartPotWeb	first commit for PRI project	9 days ago
README.md	Create README.md	9 days ago

README.md

SmartPot

Code exchange:

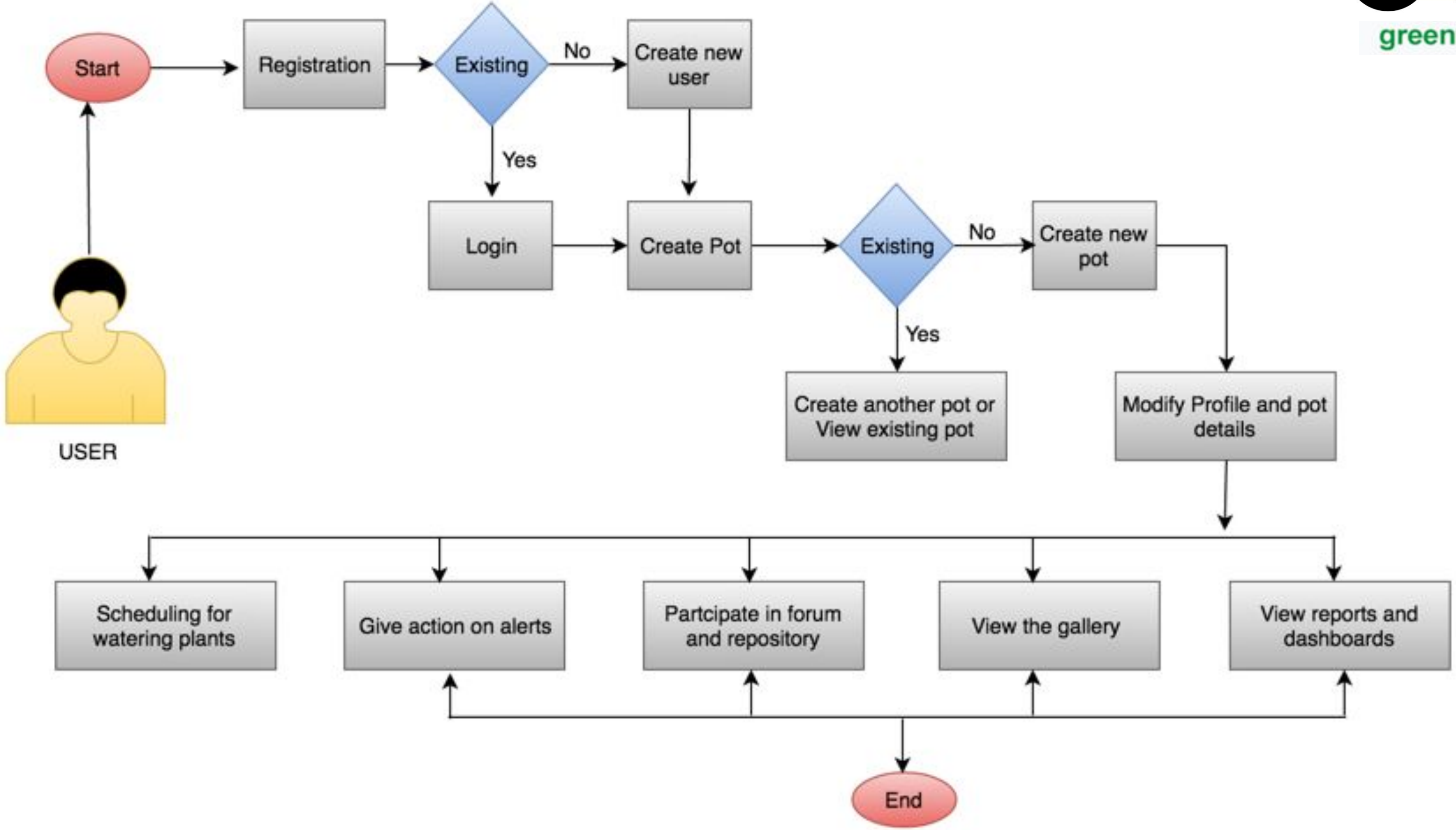
- Git Repository
- On campus group work

Participants and functionality

SMART POT	CENTRAL SERVER	USER
● Initial setup assistance. ● Receive configuration from central server. ● Send statistics to central server. ● Schedule activities in the pot. ● Send alerts to central server. ● Receive news from central server. ● Controlling devices like camera,tap and sensors.	● Register users. ● Receive and Store statistics. ● Send alerts to users. ● Exchange of data between user and smartpot. ● Continuous communication with smart pot. ● Maintain user forum.	● Register account. ● Activate garden. ● Create pot,select plant or vegetable to grow. ● Monitor the single and multiple gardens through camera ● Generate dashboards and reports ● Share the experience and solutions in forum.

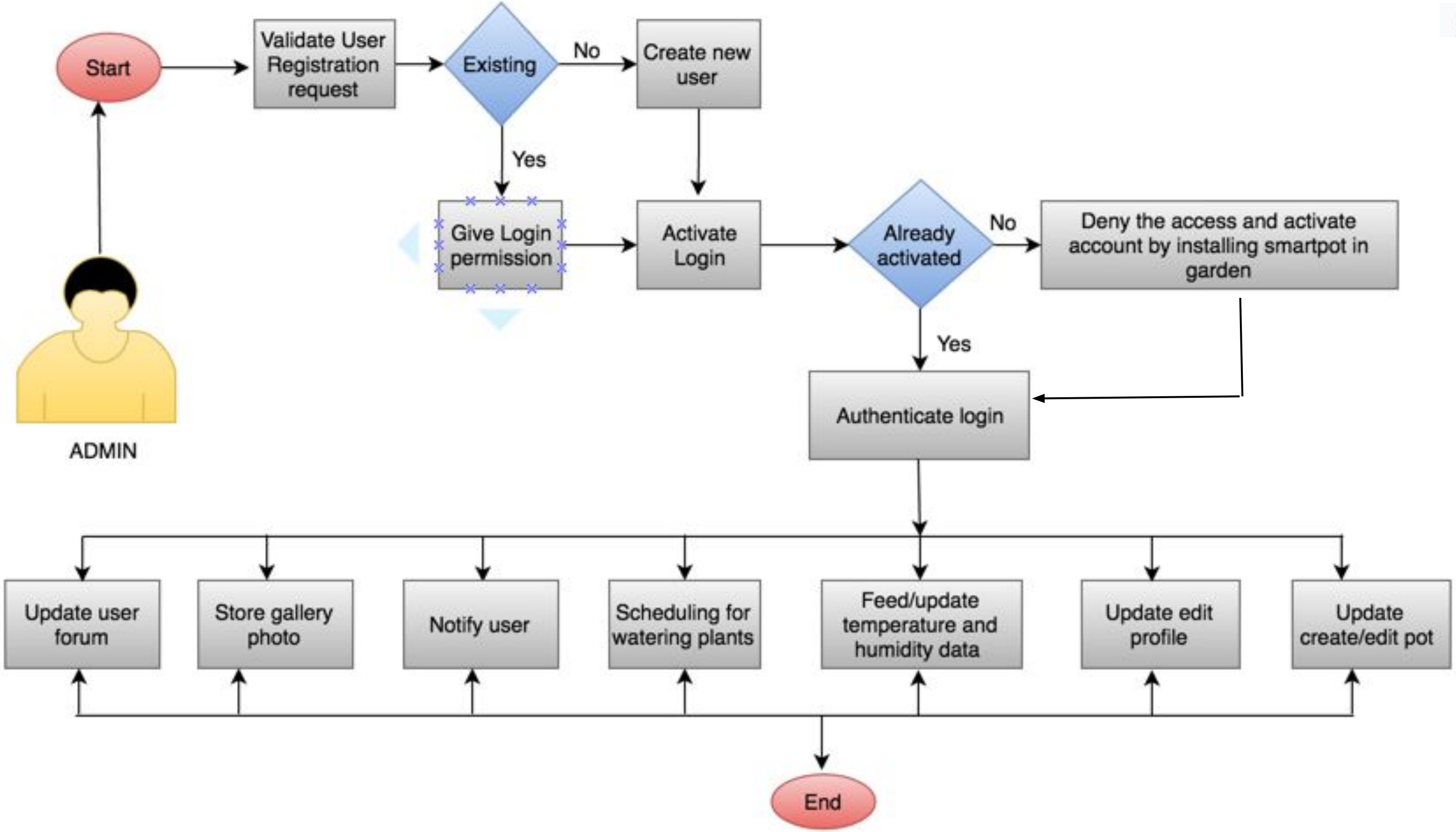
Data Flow Diagram:User side

Smart Pot greentech

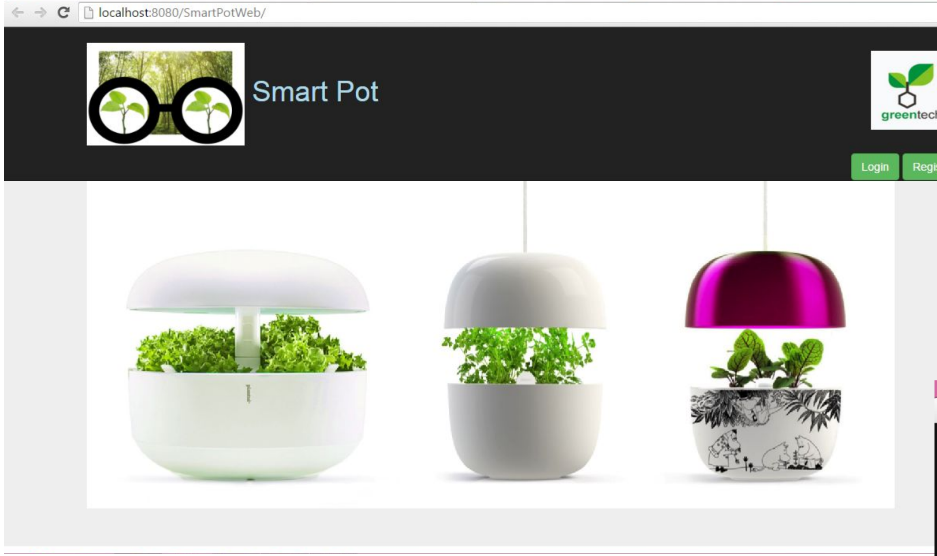


Start
Registration
Existing
No
Create new user
Yes
Login
Create Pot
Existing
No
Create new pot
Yes
USER
Create another pot or View existing pot
Modify Profile and pot details
Scheduling for watering plants
Give action on alerts
Partcipate in forum and repository
View the gallery
View reports and dashboards
End

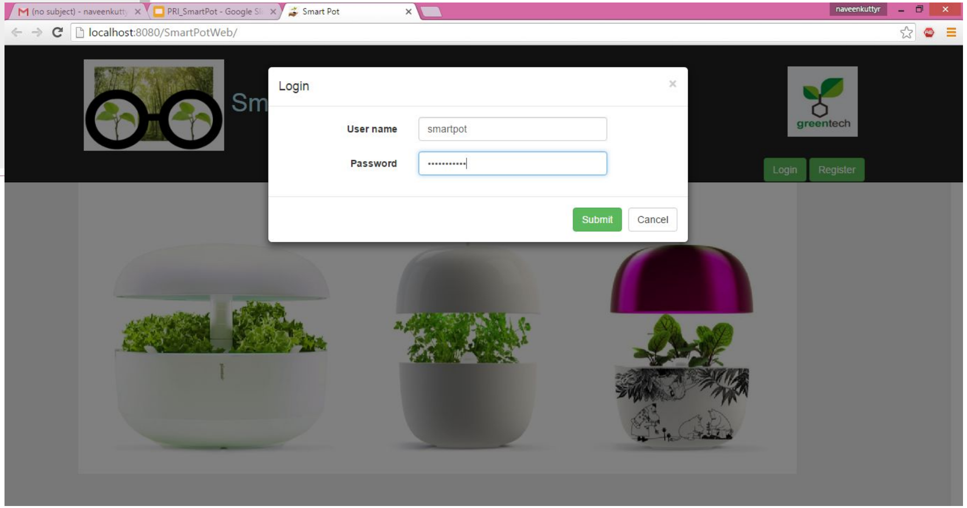
Data Flow Diagram:Admin side



Application Development: Interface



Home Page of Smart Pot



Login

User name smartpot

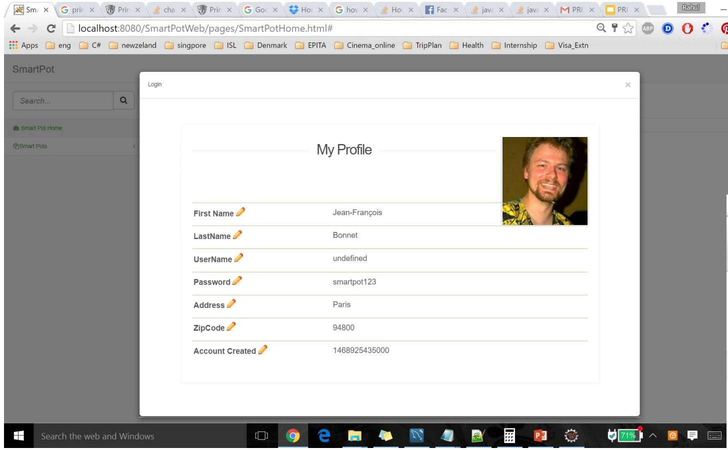
Password ••••••••••••

Login page after registration

Application Development Phase

Smart Pot
greentech

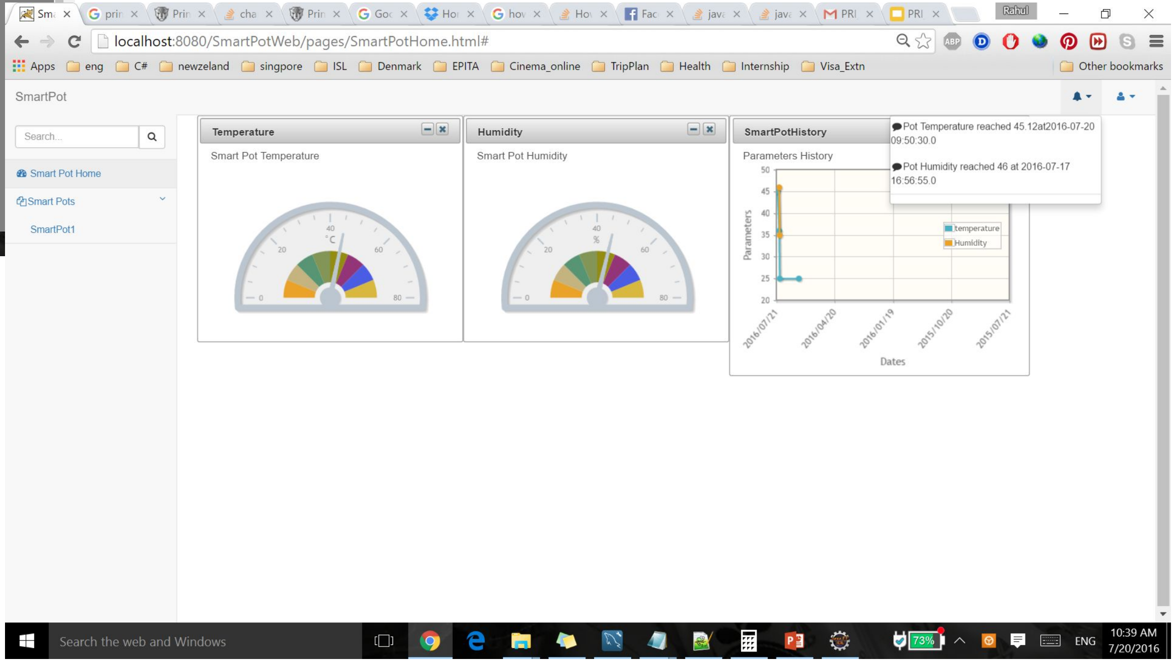
User Profile view and edit


localhost:8080/SmartPotWeb/pages/SmartPotHome.html#
SmartPot
Search...
Smart Pot Home
Smart Pots
Login
Search the web and Windows

My Profile

First Name	Jean-François
LastName	Bonnet
UserName	undefined
Password	smartpot123
Address	Paris
ZipCode	94800
Account Created	1468925435000

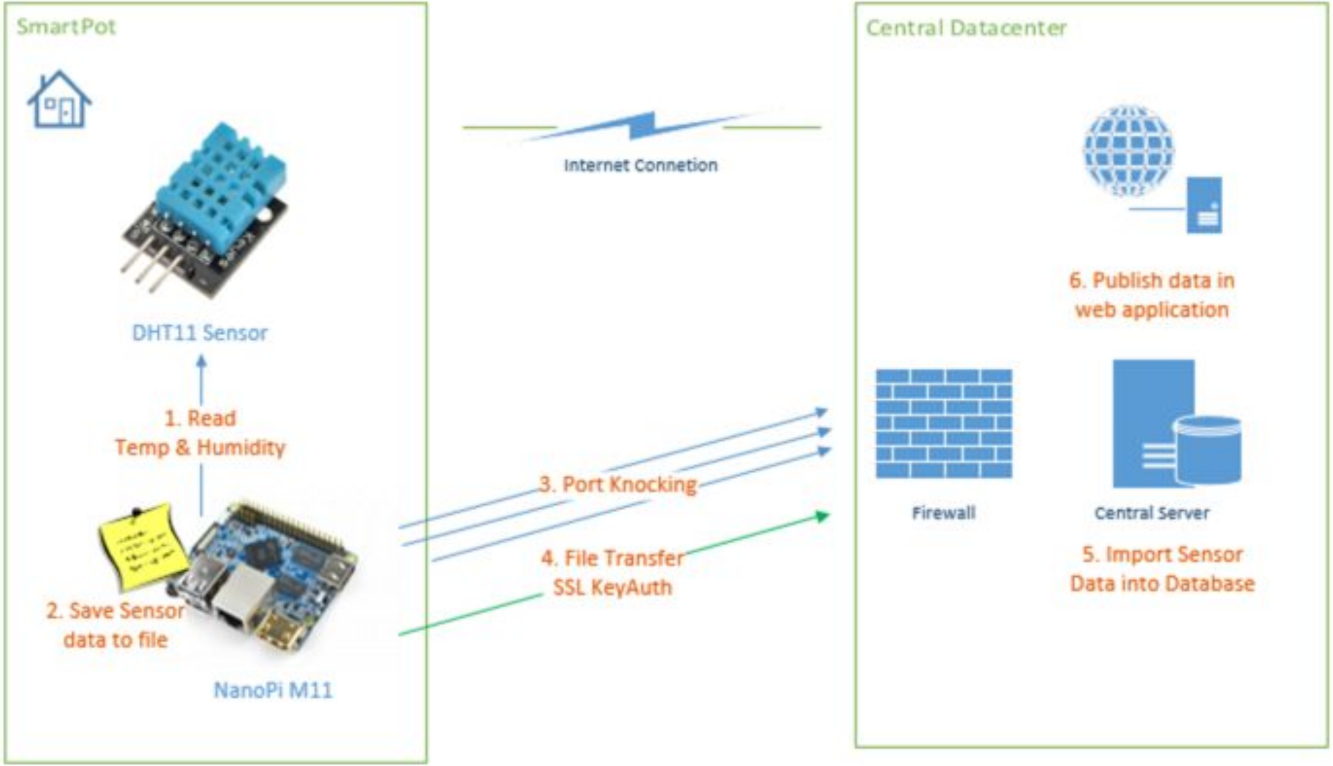
Dashboard and reports of the garden depending on temperature and humidity level



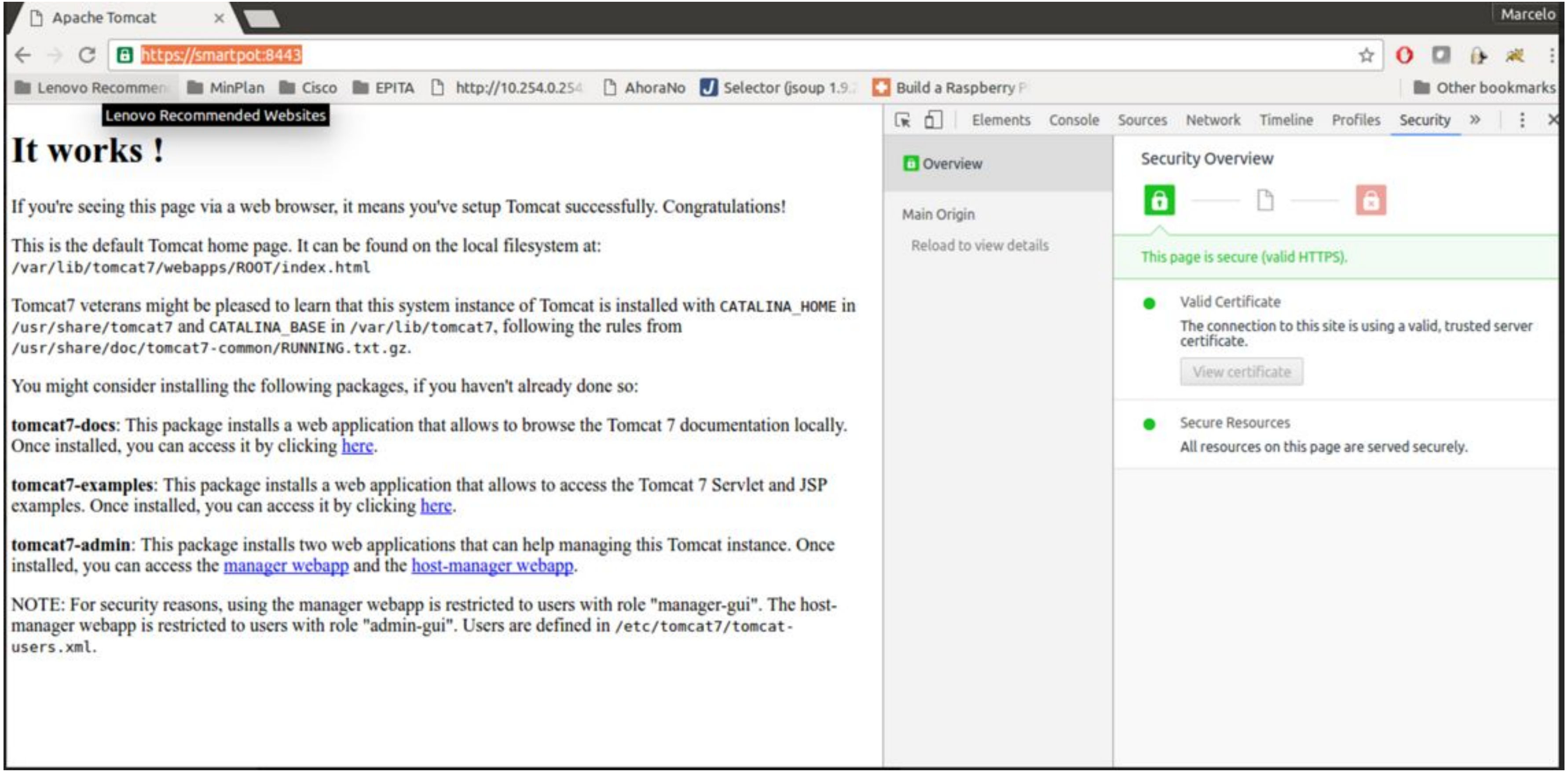
Pot Temperature reached 45.12at2016-07-20 09:50:30.0

Pot Humidity reached 46 at 2016-07-17 16:56:55.0

Smart Pot Communication flow



Smart Pot:HTTPs Connection and certificates



It works !

If you're seeing this page via a web browser, it means you've setup Tomcat successfully. Congratulations!

This is the default Tomcat home page. It can be found on the local filesystem at: `/var/lib/tomcat7/webapps/ROOT/index.html`

Tomcat7 veterans might be pleased to learn that this system instance of Tomcat is installed with `CATALINA_HOME` in `/usr/share/tomcat7` and `CATALINA_BASE` in `/var/lib/tomcat7`, following the rules from `/usr/share/doc/tomcat7-common/RUNNING.txt.gz`.

You might consider installing the following packages, if you haven't already done so:

tomcat7-docs: This package installs a web application that allows to browse the Tomcat 7 documentation locally. Once installed, you can access it by clicking here.

tomcat7-examples: This package installs a web application that allows to access the Tomcat 7 Servlet and JSP examples. Once installed, you can access it by clicking here.

tomcat7-admin: This package installs two web applications that can help managing this Tomcat instance. Once installed, you can access the manager webapp and the host-manager webapp.

NOTE: For security reasons, using the manager webapp is restricted to users with role "manager-gui". The host-manager webapp is restricted to users with role "admin-gui". Users are defined in `/etc/tomcat7/tomcat-users.xml`.

Security Overview

This page is secure (valid HTTPS).

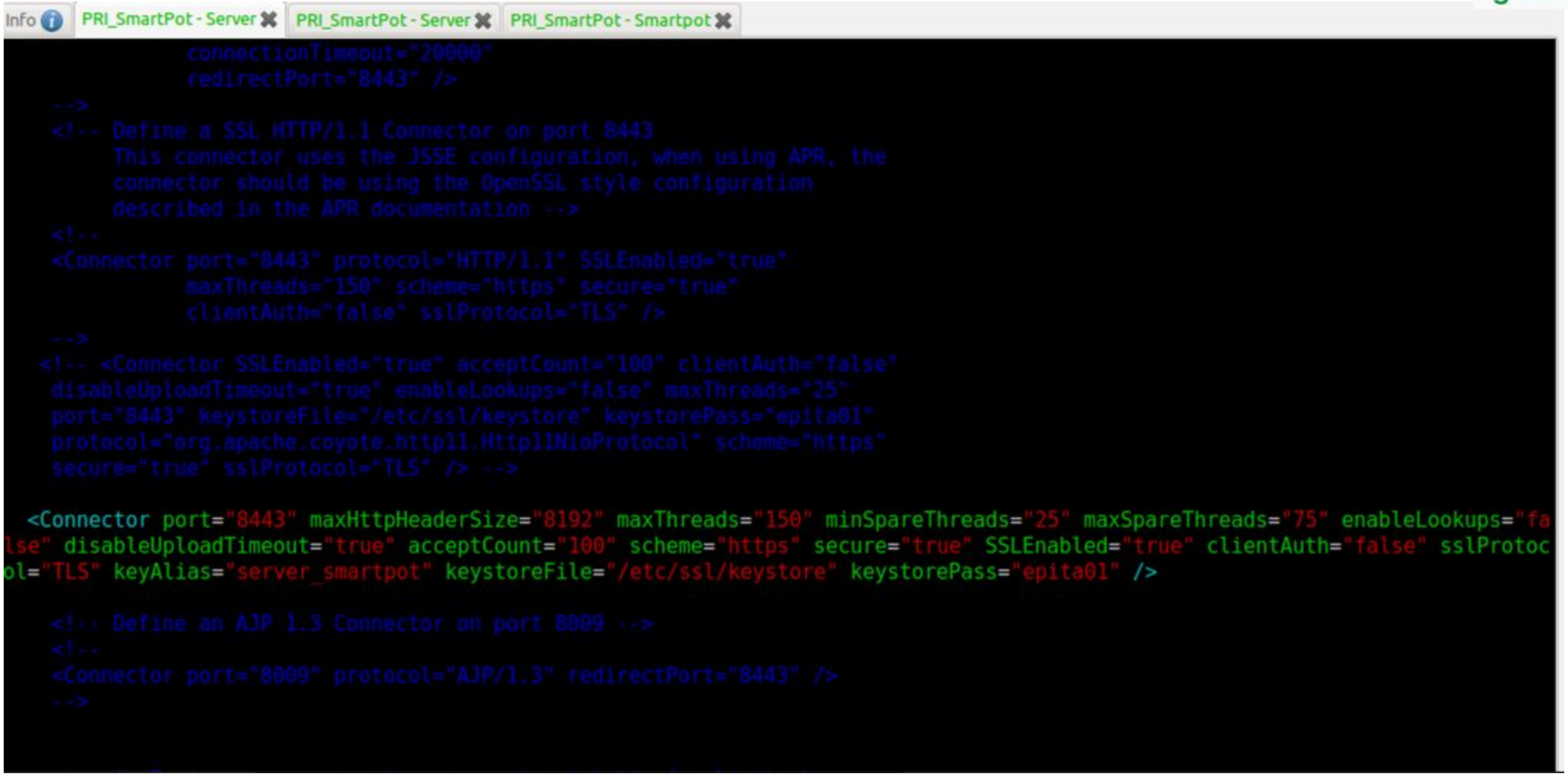
Valid Certificate

The connection to this site is using a valid, trusted server certificate.

Secure Resources

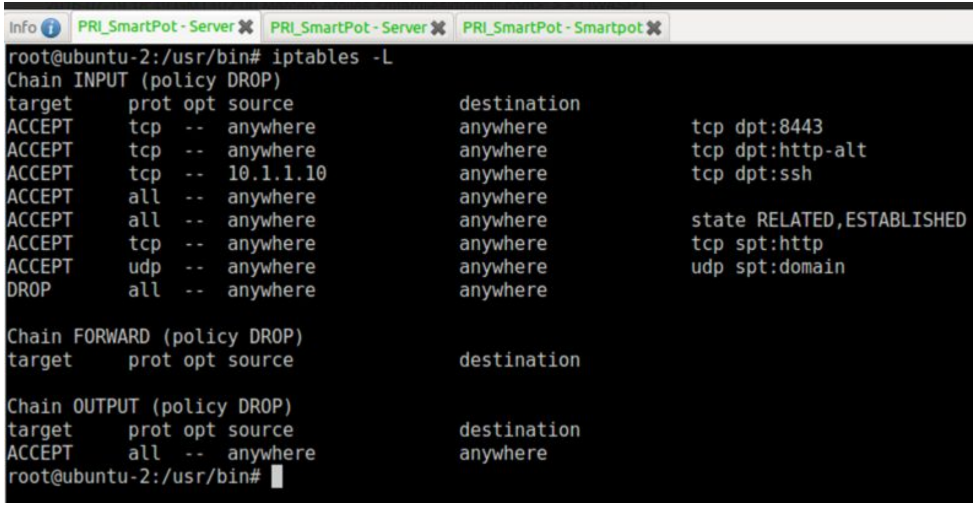
All resources on this page are served securely.

Smart Pot:Configuration of Tomcat Server

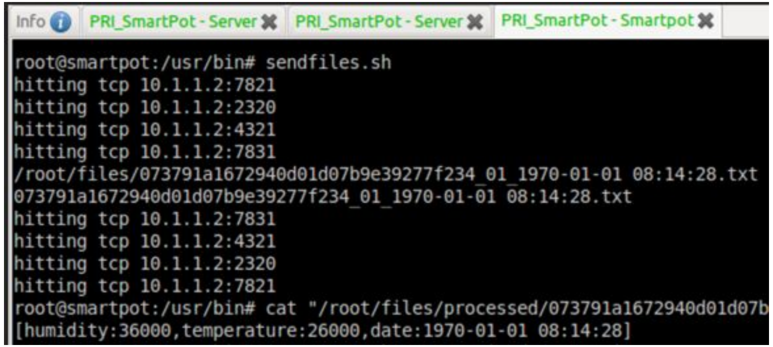


```
               connectionTimeout="20000"
               redirectPort="8443" />
    -->
    <!-- Define a SSL HTTP/1.1 Connector on port 8443
         This connector uses the JSSE configuration, when using APR, the
         connector should be using the OpenSSL style configuration
         described in the APR documentation -->
    <!--
    <Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"
               maxThreads="150" scheme="https" secure="true"
               clientAuth="false" sslProtocol="TLS" />
    -->
   <!-- <Connector SSLEnabled="true" acceptCount="100" clientAuth="false"
    disableUploadTimeout="true" enableLookups="false" maxThreads="25"
    port="8443" keystoreFile="/etc/ssl/keystore" keystorePass="epita01"
    protocol="org.apache.coyote.http11.Http11NioProtocol" scheme="https"
    secure="true" sslProtocol="TLS" /> -->

  <Connector port="8443" maxHttpHeaderSize="8192" maxThreads="150" minSpareThreads="25" maxSpareThreads="75" enableLookups="fa
lse" disableUploadTimeout="true" acceptCount="100" scheme="https" secure="true" SSLEnabled="true" clientAuth="false" sslProtoc
ol="TLS" keyAlias="server_smartpot" keystoreFile="/etc/ssl/keystore" keystorePass="epita01" />

    <!-- Define an AJP 1.3 Connector on port 8009 -->
    <!--
    <Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />
    -->
```

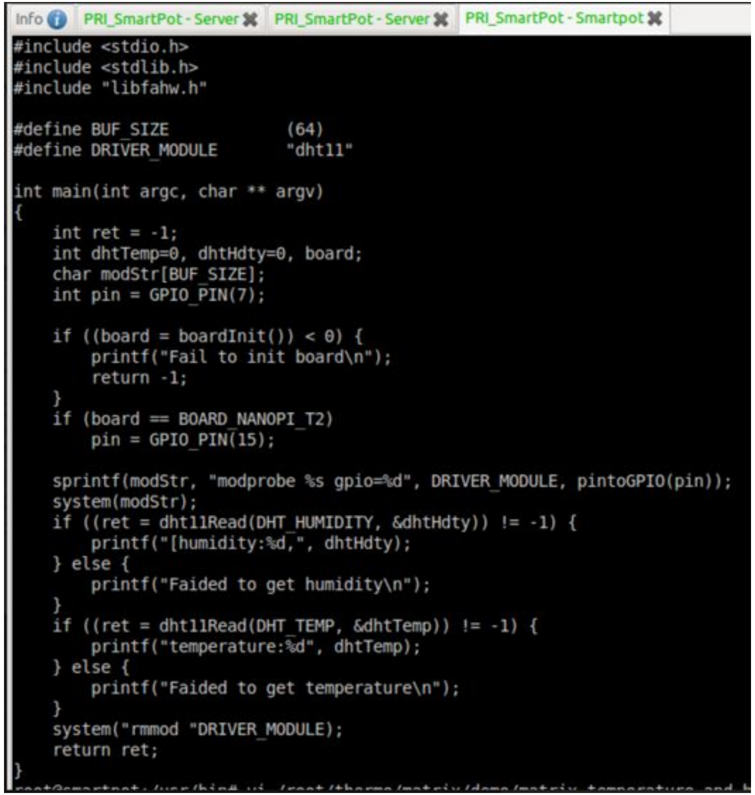
Smart Pot:Low level Programming



```
root@ubuntu-2:/usr/bin# iptables -L
Chain INPUT (policy DROP)
target     prot opt source               destination
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:8443
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:http-alt
ACCEPT     tcp  --  10.1.1.10            anywhere             tcp dpt:ssh
ACCEPT     all  --  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere             state RELATED,ESTABLISHED
ACCEPT     tcp  --  anywhere             anywhere             tcp spt:http
ACCEPT     udp  --  anywhere             anywhere             udp spt:domain
DROP       all  --  anywhere             anywhere

Chain FORWARD (policy DROP)
target     prot opt source               destination

Chain OUTPUT (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere
root@ubuntu-2:/usr/bin# █
```



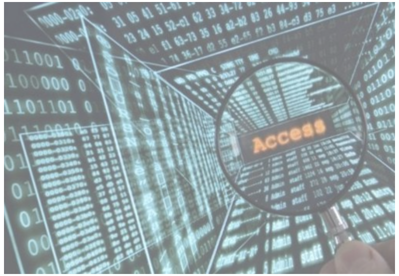
```
root@smartpot:/usr/bin# sendfiles.sh
hitting tcp 10.1.1.2:7821
hitting tcp 10.1.1.2:2320
hitting tcp 10.1.1.2:4321
hitting tcp 10.1.1.2:7831
/root/files/073791a1672940d01d07b9e39277f234_01_1970-01-01 08:14:28.txt
073791a1672940d01d07b9e39277f234_01_1970-01-01 08:14:28.txt
hitting tcp 10.1.1.2:7831
hitting tcp 10.1.1.2:4321
hitting tcp 10.1.1.2:2320
hitting tcp 10.1.1.2:7821
root@smartpot:/usr/bin# cat "/root/files/processed/073791a1672940d01d07b
[humidity:36000,temperature:26000,date:1970-01-01 08:14:28]
```



```c
#include <stdio.h>
#include <stdlib.h>
#include "libfahw.h"

#define BUF_SIZE            (64)
#define DRIVER_MODULE       "dht11"

int main(int argc, char ** argv)
{
    int ret = -1;
    int dhtTemp=0, dhtHdty=0, board;
    char modStr[BUF_SIZE];
    int pin = GPIO_PIN(7);

    if ((board = boardInit()) < 0) {
        printf("Fail to init board\n");
        return -1;
    }
    if (board == BOARD_NANOPI_T2)
        pin = GPIO_PIN(15);

    sprintf(modStr, "modprobe %s gpio=%d", DRIVER_MODULE, pintoGPIO(pin));
    system(modStr);
    if ((ret = dht11Read(DHT_HUMIDITY, &dhtHdty)) != -1) {
        printf("[humidity:%d,", dhtHdty);
    } else {
        printf("Faided to get humidity\n");
    }
    if ((ret = dht11Read(DHT_TEMP, &dhtTemp)) != -1) {
        printf("temperature:%d", dhtTemp);
    } else {
        printf("Faided to get temperature\n");
    }
    system("rmmod "DRIVER_MODULE);
    return ret;
}
root@smartpot:/usr/bin# vi /root/thermo/matrix/demo/matrix-temperature-and-h
```

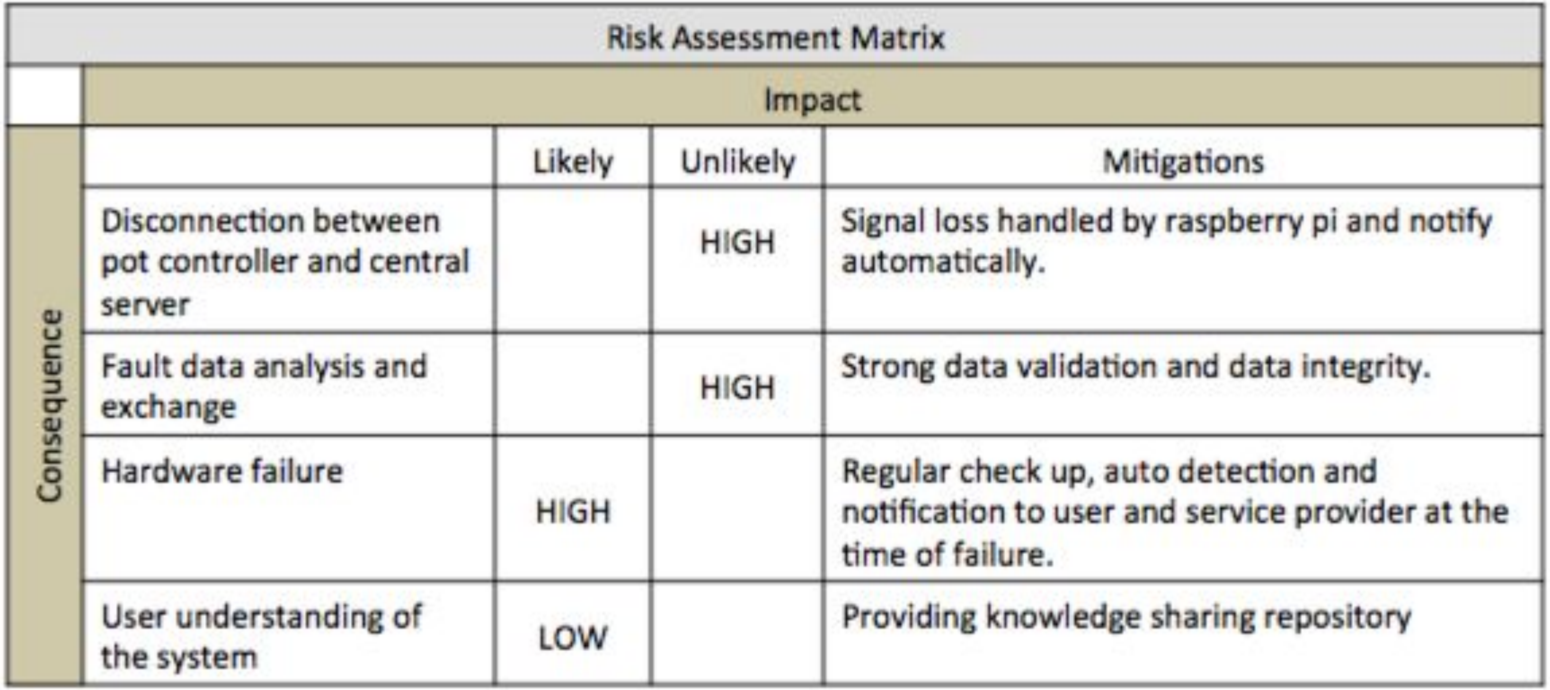
Project Security Issues

- **Secure Architecture**
- **Secure networking** using IP tables and firewall
- **Port knocking** for managing open connection
- Key authentication for **secure transfer** SSL
- HTTPS Tomcat server with **certificates**
- Operating system hardening
- **Security evaluation**
- Owasp Lapse Project for **secure scanning** of java **applications**



Risk and Threats

Smart Pot greentech



Risk Assessment Matrix

		Impact		
Consequence		Likely	Unlikely	Mitigations
	Disconnection between pot controller and central server		HIGH	Signal loss handled by raspberry pi and notify automatically.
	Fault data analysis and exchange		HIGH	Strong data validation and data integrity.
	Hardware failure	HIGH		Regular check up, auto detection and notification to user and service provider at the time of failure.
	User understanding of the system	LOW		Providing knowledge sharing repository

Extensions and future plans

- Automatic tap control
- Fertilizer control
- Camera and gallery for monitoring the growth of plants
- Notifications and alerts of the updates
- Knowledge sharing and solution repository
- Partnership with companies manufacturing protection shed for plants

Business Model

Smart Pot

greentech

Key partners:

- Suppliers and service provider
- Joint ventures
- Associations
- Experts

Key activities:

- Monitoring the garden activities
- Maintenance of the smartpot
- Analyzing the data
- Exchange of real time data

Key resource:

- Smartpot infrastructure
- IT infrastructure
- User information and account

Value proposition:

- Expert’s advise
- Maintenance of the system
- Installation at Home
- Technical help service
- Video tutorial for assistance
- Knowledge sharing platform
- Good customer service

Customer relationships:

- Customer retention
- Long term relations with customer and their gardens
- Customer feedback and solutions

Customer segmentation:

- Biological student
- Bio food lovers
- Indoor and outdoor gardens
- Busy lifestyle
- Middle or upper class
- Travel lovers

Cost:
Hardware cost, IT infrastructure,Maintenance,Staff and logistics

Revenue:
Hardware,Personalized technical help at remote location, Package,Personalized expert’s advice, garden material.

Challenges and Best Practice

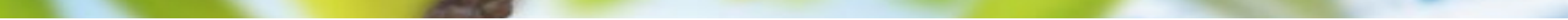
Challenges

- Compability of OS with sensors and hardware
- Low level programming i.e. related to electronics
- HTTPs configuration of Tomcat
- Fetching data from multi tables
- Using hibernate framework for joining multiple table
- Integration of prime phase servlet with spring servlet.

Best Practice

- Test driven development with framework like JUnits and Mockito
- Design patterns like singleton
- Data access object (DAO)
- Model View controller pattern
- 3 Tier architecture

Learning + Development



THANK YOU

Rabih Haddad   Olivier Berthet   JF Bonnet