

FIREBOT SOFTWARE

CS 350 SOFTWARE ENGINEERING II

SPRING 2018 PROJECT

ALI KALKANDELEN

SAMUEL OWUSU-BINEY

"Firebot" Fire Extinguishing Robot

There are more than 20,000 house fires and more than a dozen wildfires in the United States each year. Most of the time these fires are not immediately controlled or rescued due to lack of resources and personnel.

The firebot is a fire-extinguishing robot designed to assist firefighters in times of wildly spreading, uncontrollable fires, where seconds and minutes will make the difference between life and death. Firebot will be a self-sufficient, portable, powerful robot able to fight even the highest of blazes.

The robot will essentially be a water or foam tank attached to a rover along with a hose to spread the liquid evenly among the fire. The water or foam will depend on the situation and will easily be able to get filled through the firetruck or hydrant. The hose will be special stainless steel pipe with a custom tip that can change the spread of the solvent as appropriate for the situation.

The robot will detect the fire using changes in heat signatures on its high field of view camera. The camera will also detect high concentrations of carbon dioxide, which is a common molecule found in almost all fires. Once detected the robot will engage the fire using its reserve and hose as necessary and calculate the best angle and procedure for the extinguishing method.

Stake Holders

- **Government**
- **Fire Departments**
- **Municipalities**
- **Rescue Teams**
- **Non-Profit Organizations**

6 - Interfaces

<u>Movement</u>		
<u>Field Name</u>	<u>Data Type</u>	<u>Data Shape</u>
Yaw	Int	xxx.xxx
Pitch	Int	xxx.xxx
Roll	Int	xxx.xxx
Speed	Int	xxx.xxx
Wheel_RPM	Int	xxx.xxx
Hose_Speed	int	xxx.xxx
Robot_Turn	int	xxx.xxx
<u>DB</u>		
dataID	int	xxx
dataName	varchar	xxxx xxxx
heatSigVal	int	xxx xxx xxx
userID	varchar	xxxxxx
userName	varchar	xxxxxx
userPhone	varchar	XXX-XXx-XXX
userRegion	varchar	xxxxxx
connectionID	int	xxxxxx
<u>UI</u>		
menuID	int	xxx
menuName	varchar	XXXX
batteryLvl	int	xxxx
hoseAngle	int	xx.xxxx
statusID	int	xx.xxxx
statusName	varchar	xxx
emergencyStopTrig	int	x
<u>Process</u>		
settingID	Int	xxx
settingName	varchar	xxx
radiusCalc	int	xxx
chemToHeatFactor	int	xxx

7 – Functional Requirements

- **The Database:** Store Data such as:
 - User information such as name, address, position
 - Search Histories based on demographic data.
 - Data about current and past campaigns
 - Logs
- **Front-End Framework:**
 - Display the webpages, buttons, forms, etc..
 - Display the Graphics and Reports it receives from back-end.
 - Create functionality for all the intractable UI.
- **Back-End Framework:**
 - Crawl through external webpages, polls, news articles, google searches, databases, etc. to pull out data relevant to the area searched.
 - Make judgements based on all the data received. Use mathematical algorithms to determine hot topics and flashes. Categorize data effectively.
 - Connect to the Database to store and collect data.
- **External Login System:**
 - Handle Username/Password data.
 - Authenticate data entered and show error messages
 - Authenticate Person Position, Authority.
 - Handle changes to members.

8 – Non-Functional Requirements

- **Performance:** The robot will run on a 4000maH lithium Iode battery that will last about half an hour. The battery will be easily replaced. The directional speed of the robot will max out at 15 mph and a rotational speed of 15rpm. The instruction(system) to action speed is about 1/60 s.
- **Reliability:** Since the system is directly connected to the robot a reliability of 99% can be achieved. Maintenance on the caching system plugins will need to be performed once every 3 months.
- **Capacity:** The robot will be equipped with a 20 liter tank for either water or fire fighting foam. The tank will be easily refillable through a firetruck or a hydrant. The software will need to be installed individually to each robot.
- **Security:** Since the robot is used under very specific circumstances, with a supervision of at least 1 officer, security systems will not be put into place for the software. But the user will need to be authenticated and verified before the software is initiated and ultimately robot is used.
- **Portability:** A copy of the software will need to be in each robot individually and not be tampered with or installed without the consent of a professional technician. The robot will be easily portable due to its ability to move and the petite size to fit into trucks and beds.

9- Select a Process Model

Pick: Agile-EP

About Agile:

Agile is mostly described as an iterative waterfall model. This is because the software is delivered in iterations as opposed to delivering it as a whole. Software is developed in Sprints that last from 1 week to 4 weeks, and then presented and evaluated with the client before proceeding to the next sprint. This way, the changes or errors in development can be done much earlier, saving a lot of money from post-delivery maintenance.



Why choose Agile for Project:

- Made mostly for small groups working in a tight environment on a single project.
- Extreme Programming allows building a software without knowing what the client wants.
- Less prone to errors and bugs after delivery important when time and effort after delivery is not possible.

10 – SAFETY CONCERNS

Safety Concern #1: Software Compromised

Implications:

- Hacker can access database from an external network
- Hacker can add and delete data from database including entire table
- Hacker can steal user sensitive information such as full name and address.

Defense:

- Symmetric Database Encryption
- Hashing sensitive data
- Transparent Data Encryption

Safety Concern #2: Radio Signal Compromised

Implications:

- Wrong information is provided to the user
- Hacker can disrupt the communication between robot and operator

Defense:

- Create proper Firewall protocols.
- Encrypt Signals before being sent out
- Monitor signal disruptions

Safety Concern #3: Robot gets too close to fire

Implications:

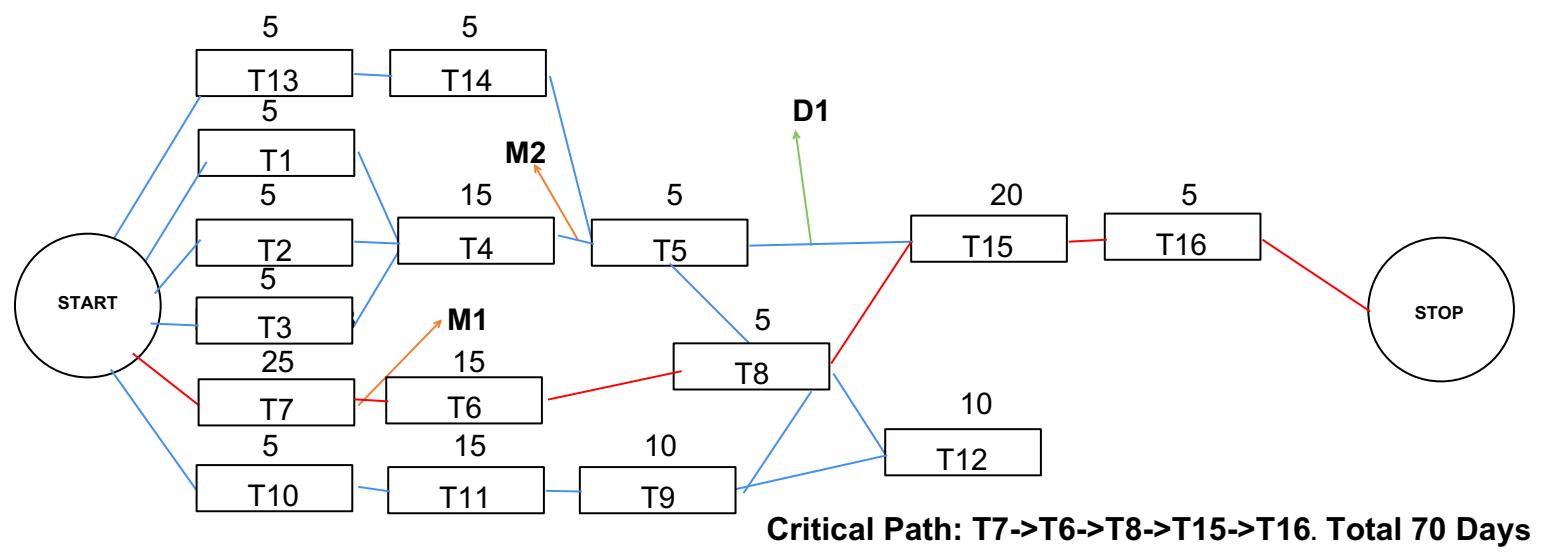
- The metal can melt releasing lead into the air
- The Pressurized tank inside of robot can burst
- The battery can explode

Defense:

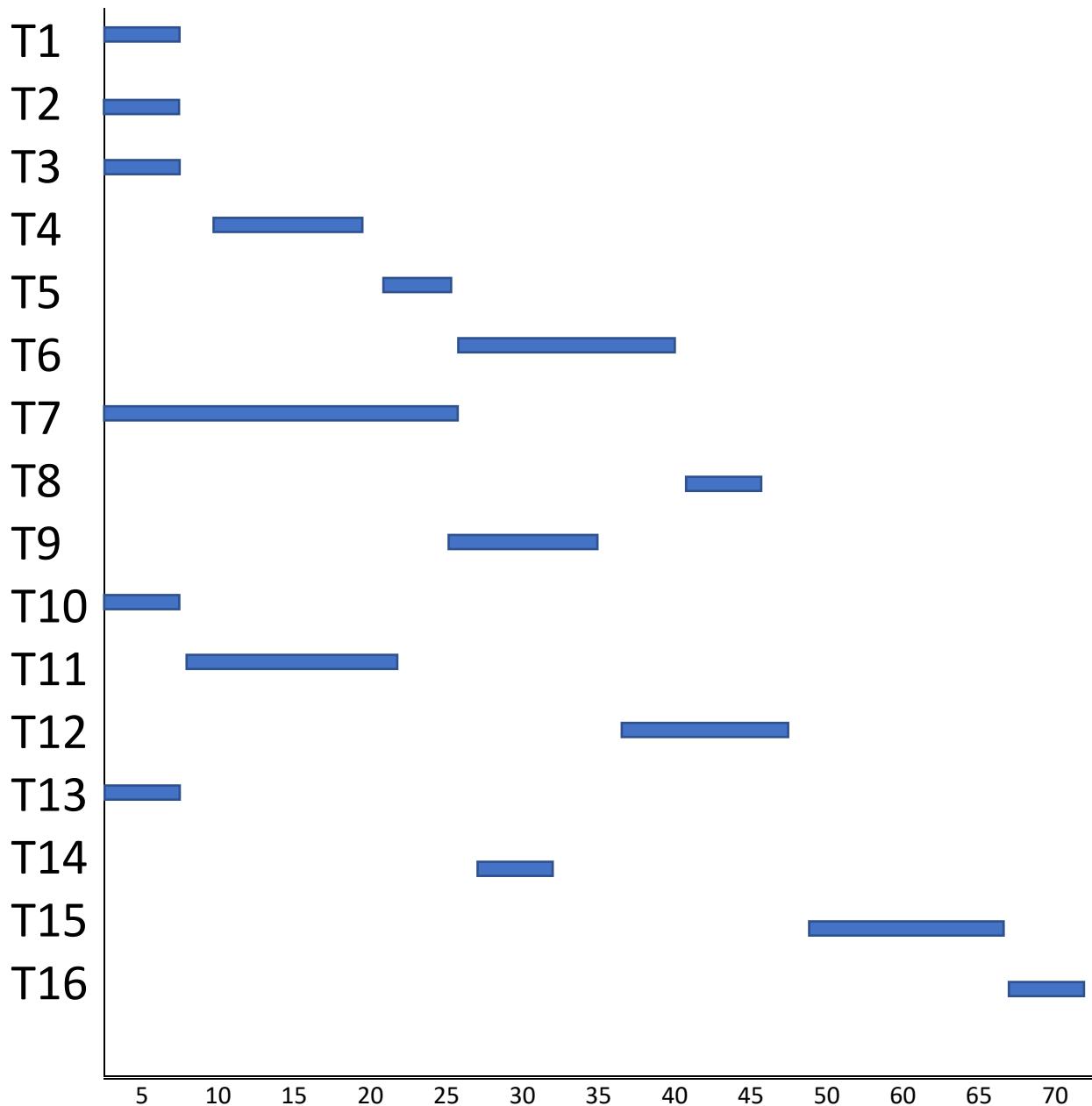
- Explosion proof casing of the battery and water tank.
- Calculating distance to heat signature and alerting if too low.

12 & 13 - ACTIVITY CHART, MILESTONES, DELIVERABLES, CRITICAL PATH

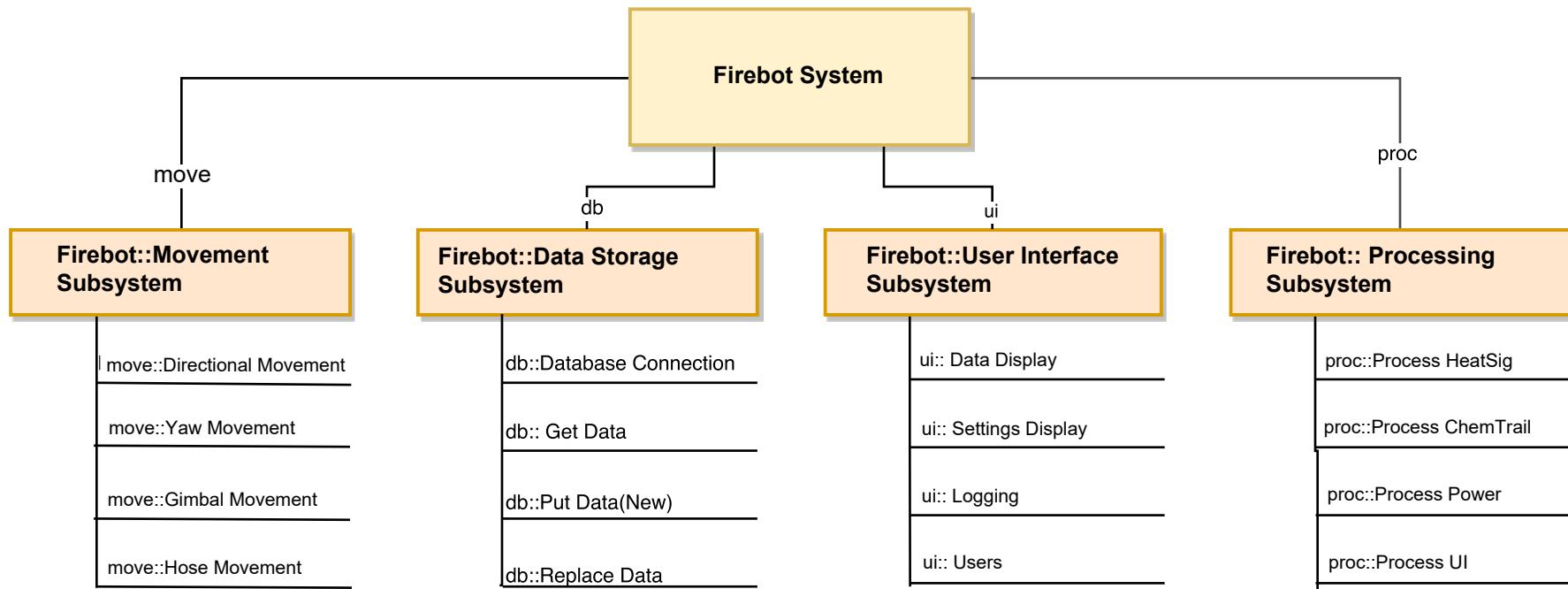
Task	Description	Duration (days)	Dependencies
T1	Directional Movement	5	
T2	Yaw Movement	5	T2
T3	Gimbal Movement	5	
T4	Setup Database Connection	15	T1,T2,T3
T5	Hose Movement	5	
T6	Set database systems and variables	15	T4
T7	Code out processing algorithms for the heat signatures and chemTrail signatures	20	
T8	Process algorithm on data and setup organization of data in db	5	T5,T6,T7
T9	Code script to process user input from controller	10	T11
T10	Design the user interface outlook	5	
T11	Code the menus according to design	15	T10
T12	Code out display of robot information such as battery or power	10	T8,T9
T13	Setup connection with 3rd Parties	5	
T14	Code authentication of user using 3rd party softwares	5	T13, T5
T15	Create Settings Framework(Back-End)	20	T5,T8,T9,T14
T16	Code Encryption, Analytics, and maintenance code	5	T5,T9



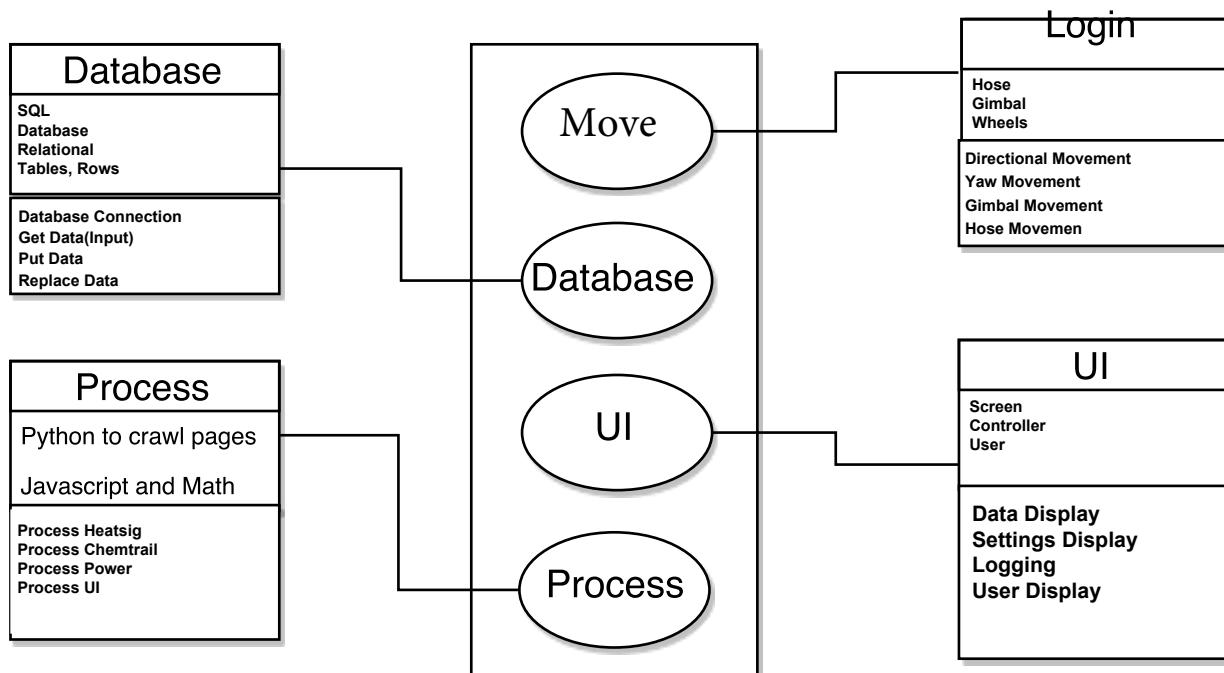
12 – BAR CHART



4. IDENTIFY SUBSYSTEMS



#14 - UML



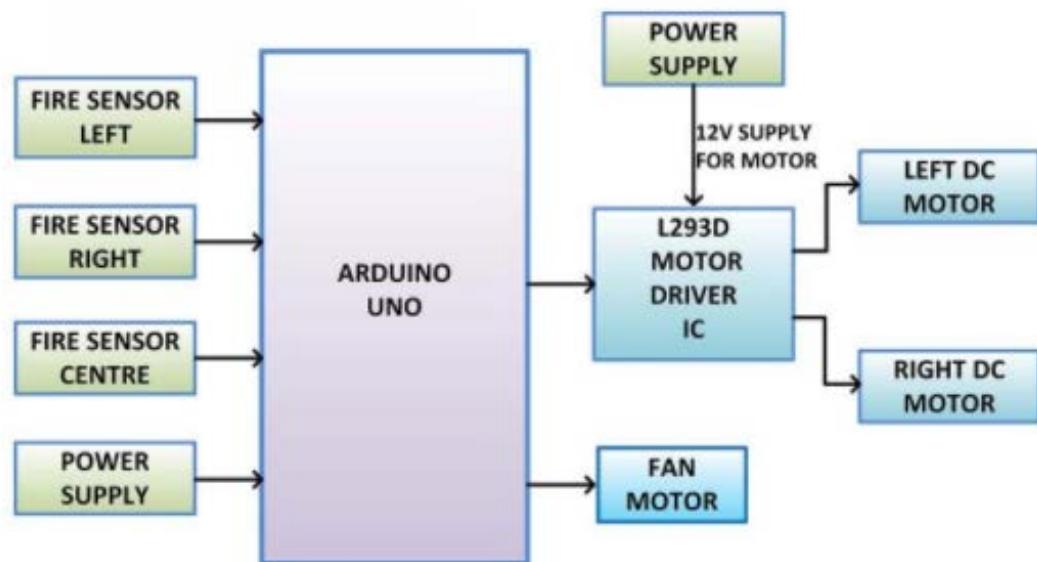
#21 - Hardware

Power Supply - 5V regulated DC for Circuit. 12V DC for motor. A 12V NIMH battery is used as the primary source of power.

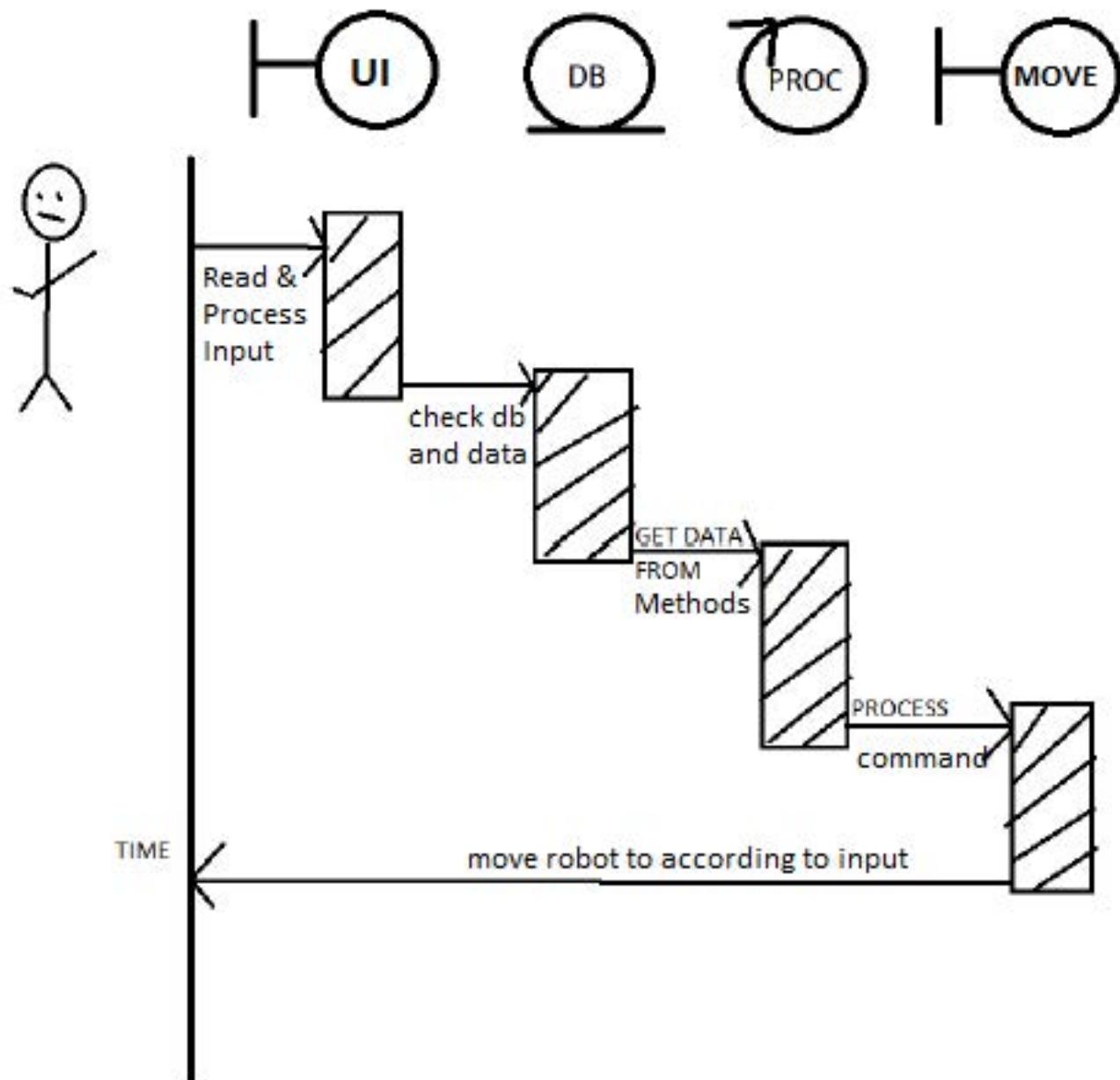
Arduino UNO - Computer Board. Built-in arduino boot loader. Atmega 328 based controller board w/ 14 GPIO pins, 6 PWM pins, 6 Analog inputs and on board UART, SPI and TWI interfaces.

L293D DC Motor Driver IC - Dual H-bridge motor driver integrated circuit (IC). 16 pins.

Fire Sensors - A Fire Detection Sensor. wavelenghts 760 nm to 1100 nm. 60 degrees detection point, distance about 1 M to 2 M. IR receivers are used as fire detection sensors in the circuit.



18 - SEQUENCE DIAGRAM



17 – TEST PLAN

- **System:**
 - Click Login System
 - Click Movement system
 - Click User Interface Systems
 - Click Processing/Calculations.
- **Subsystem:**
 - **Login:**
 - Click User Authentication
 - Click password encryption
 - Click password recovery system
 - Click security and authentication systems
 - **Data Storage:**
 - Click Proper DB Connection
 - Click Data Retrieval
 - Click Data replacement
 - Click Insert of new data and relationships
 - **User Interface:**
 - Click Charts and reports and figures
 - Click all buttons
 - Click all hyperlinks
 - Click forms and all text inputs
 - **Processing:**
 - Click proper calculations
 - Click for proper calculation input
 - Click Functions on buttons and links
 - Click Algorithms used for calculation
- **Unit (Login System):**
 - U: Enter “John Smith”
 - S: Error “john smith” is not recognized
 - U: Enter “JohnSmith”
 - S: Ok. Enter Password
 - U: “*****”
 - S: Error “Must be longer than 8 digits”
 - U: “*****”
 - S: OK.

18 – MAINTENANCE PLAN

Daily Maintenance

- Perform Backup(Automatic) of database and crucial sub-systems
- Update Plugins(Automatic)
- Check for spikes in Network
- Review any issues sent from Analytics, Security, or Email Servers

Monthly Maintenance

- Check if all pages load properly.
- Check for any broken links
- Check for 404 Errors.
- Check for any type of Console errors.

Monthly Maintenance

- Check website loading speed.
- Review security scans and resolve any issues
- Review website statistics and evaluate conversion rates.
- Review local visibility and search engine optimization.

Quarterly Maintenance

- Review website – what could be improved?
- Check for Outdated 3rd party libraries or software.
- Review and tweak meta title and meta description tags
- Test and tweak website to improve conversion rates.
- Test Website for browser compatibility.
- Check the uptime logs

Yearly Maintenance

- Update the copyright date in footer
- Review each page of the site for content accuracy
- Renew SSL certification if needed.
- Check backup health by restoring the most recent backup to a separate web server
- Validate your site. (CSS/HTML, Accessibility, Mobile-friendly)
- Review your website strategy to align with your business goals.

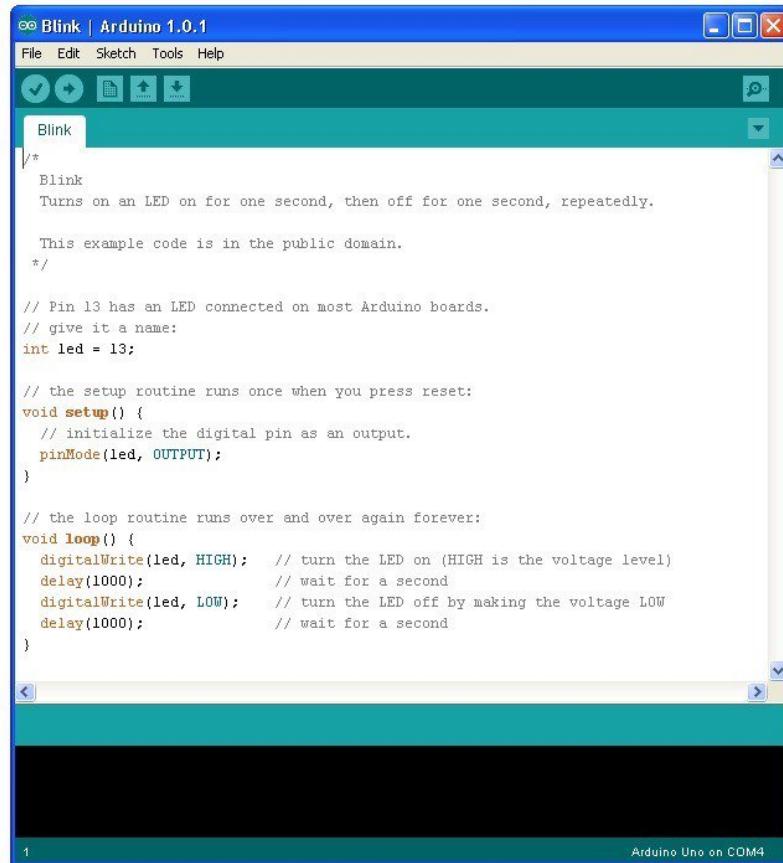
19 – PROGRAM DESCRIPTION LANGUAGE

Method: Hose Movement

```
void perform_hose_movement (int pitch)
{
    if (get_hose_pitch() is equal to pitch )
        print “error 1: already in position”;
    else if(pitch less than 30 or more than 120 )
    {
        print “error 2: pitch degree impossible to achieve”;
    else
    {
        set hose_pitch_deg equal to pitch;
        Move Hose To hose_pitch_deg.
    }
}
```

23 – Software and State Diagram

Arduino Sketch Code written and then compiled on Arduino IDE on Windows 10.



The screenshot shows the Arduino IDE interface with the title bar "Blink | Arduino 1.0.1". The menu bar includes File, Edit, Sketch, Tools, and Help. Below the menu is a toolbar with icons for checkmark, play, save, upload, and search. The main code editor window contains the "Blink" sketch. The code is as follows:

```
/*
Blink
Turns on an LED on for one second, then off for one second, repeatedly.

This example code is in the public domain.
*/

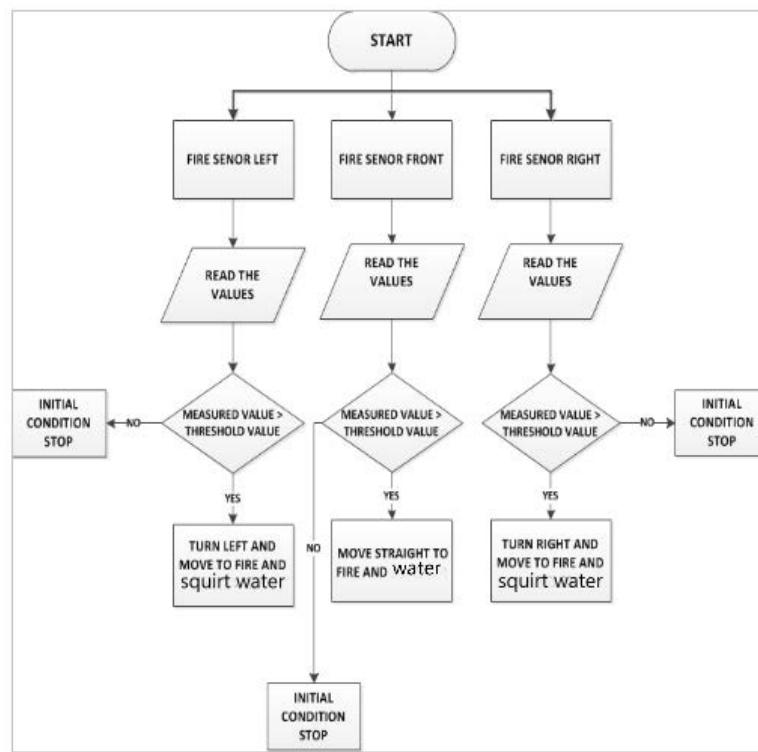
// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
    // initialize the digital pin as an output.
    pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
    digitalWrite(led, HIGH);      // turn the LED on (HIGH is the voltage level)
    delay(1000);                // wait for a second
    digitalWrite(led, LOW);       // turn the LED off by making the voltage LOW
    delay(1000);                // wait for a second
}
```

The status bar at the bottom right shows "Arduino Uno on COM4".

STATE DIAGRAM



16 – WEB SITE CODE

```
<html lang="en-US"><head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="chrome=1">
<title>CS350PROJEKT</title>
<meta property="og:title" content="CS350">
<meta property="og:locale" content="en_US">
<meta name="description" content="CS 350 PROJECT FALL 2017">
<meta property="og:description" content="CS 350 PROJECT FALL 2017">
<meta property="og:url" content="https://alikalkandelen.github.io/CS350/">
<link href="https://fonts.googleapis.com/css?family=Arvo:400,700,400italic" rel="stylesheet" type="text/css">
<link rel="stylesheet" href="https://pages-themes.github.io/dinky/assets/css/style.css?v=46349262f43f70d5dc78cf83775850b0466aa573">
<script src="https://pages-themes.github.io/dinky/assets/js/scale.fix.js"></script>
<meta name="viewport" content="width=device-width, initial-scale=1, user-scalable=no">
<!--[if lt IE 9]>
<script src="//html5shiv.googlecode.com/svn/trunk/html5.js"></script>
<![endif]-->
<style>
header{
padding: 34px 80px 22px 50px;
}
</style>
</head>
<body>

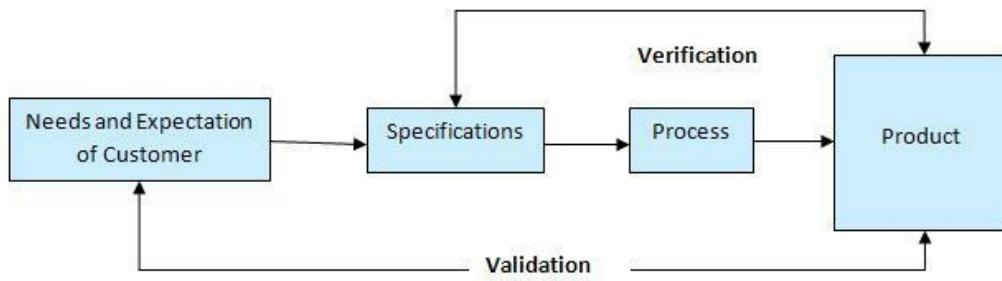
<div class="wrapper">
<header>
<h1 class="header">BRAINSTREAM</h1>
<p class="header">CS350 PROJECT FALL 2017</p>
<ul>
<li class="download"><a class="buttons" href="combinepdf.pdf">Download PDF</a></li>
<li class="download"><a class="buttons" href="projectfiles.zip">Download ZIP</a></li>
```

```
</ul>

<p>Ali Kalkandelen <br> Samuel Owusu-Biney <br> Stephen Kilnisan</p>

</header>
<div id="results" class="hidden"></div>
<section id="example1">
    HELLO WORLD
</section>
<script src="pdfobject.min.js"></script>
<script>
    var options = {
        pdfOpenParams: {
            pagemode: "thumbs",
            navpanes: 1,
            toolbar: 1,
            statusbar: 0,
            view: "FitV"
        }
    };
    var myPDF = PDFObject.embed("project.pdf", "#example1", options);
    var el = document.querySelector("#results");
    el.setAttribute("class", (myPDF) ? "success" : "fail");
    el.innerHTML = (myPDF) ? "" : "The PDF embed Didn't Work. Please try another browser or deleting your cache and trying again.";
</script>
</div>
<!--[if !IE]><script>fixScale(document);</script><![endif]-->
</body>
</html>
```

20 - VERIFICATION AND VALIDATION



	Validation	Verification
Questions:	<ul style="list-style-type: none">• Does the final product meet the business needs of the customer?• Does the product meet all the testing requirements?	<ul style="list-style-type: none">• Does the product according to the specifications?• Does the implementation meet the design?• Does the product follow the proper cycle?
To do:	<ul style="list-style-type: none">• Execute the software• Test executed software (alpha,beta,FAT)• Validate product with business	<ul style="list-style-type: none">• Review specifications and implementation, making sure nothing is missed.• Review the Requirements and match it with the product

21 - COCOMO

Nominal Effort: $3.2 \times (\text{KDSI})^{1.05}$

Estimated DSI = 4000

$$3.2 \times (4)^{1.05} = 13.72$$

$(1.15)(1.08)(.85)(1)(1)(1)(.87)(1.19)(.91)(.86)(1.1)(.95)(.82)(.91)(1.1)$

= .74

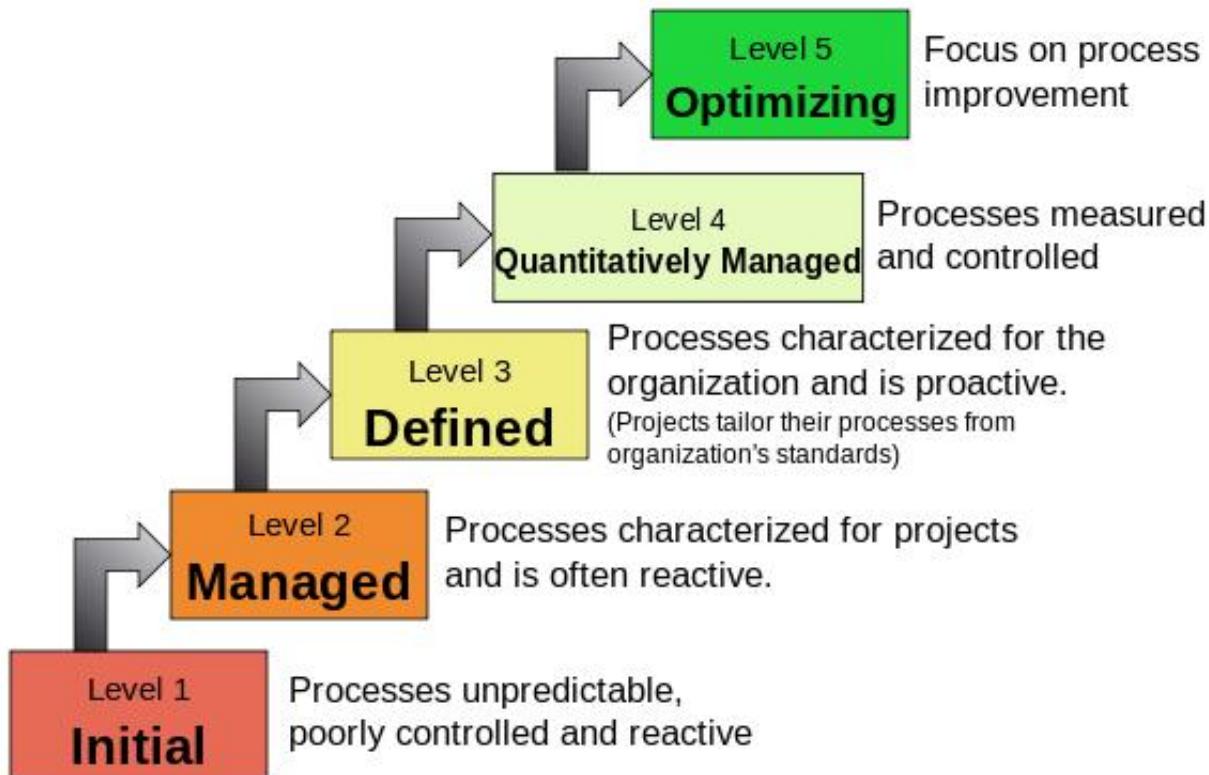
$14.72 \times .74 \approx 11 \text{ Person-Month}$

22 - Decommissioning Plan

- Determine Retirement Strategy
 - Is the program in full retirement or migrating to a newer system?
 - Will the retired program be backed up or purged forever?
 - How many people are involved in the current system?
 - Size of Database? Size of Code? Size of Traffic?
- Database:
 - Database Refactoring to newer software, or complete, safe deletion if full retirement.
 - Backups are deleted or migrated depending on the decision
- Documentation:
 - Docs are updated to include decommissioning portions and plans
- Users:
 - Users are informed of migration or removal.
 - Users are directed to a new system in a seamless manner
 - Users receive proper training into understanding new system
- Implementation:
 - Remove all code from server.
 - Delete proper files and folder structures.
 - Backup to another PC if needed
- Hardware:
 - Determine if or how the hardware will be used.
 - If reused replace all batteries.
 - Clean and replace motors if shown signs of metal fatigue or damage
 - If not reused dispose recycle or trash accordingly.

23 - CAPABILITY MATURITY MODEL

Characteristics of the Maturity levels



CMM LEVEL PICKED: **LEVEL 2 - MANAGED**

Reasons:

- The characterization of projects are well defined
- Processes for the project are clear and explanatory.
- Software is from a brand new group, so no organizational standards are set.
- Processes are defined but not well controlled.
- Processes have no definition for improvement or proactive for the future.

COST PER PERSON : \$20,000 Per Month Per Person. Based on 11

person month * 20K = **\$220,000 For Project.**

34 - DATED LOG

Person Name	Person Position	Description	Date
Alan Smith	Operations Manager	Set-Up Orgnization structure and personnel	09/01/2017
John Adams	Database Analyst	Made priliminary database structures	09/20/2017
John Doe	Programmer	Tech stack setup and analysis	09/22/2017
Jane Doe	Senior Analyst	Major change in life cycle method	10/10/2017
Sam Silverstein	Programmer	Changed tech stack to accomodate project	10/20/2017
Tom Turner	Project Manager	Added developers to project to speed up prod	10/31/2017
Adam almosa	Assistant VIP	provided additional funding to proj	11/01/2017
Laurie Comes	Secretary	Made updates to office space	11/02/2017