|  |
| --- |
| Jan Fontanosa, Vyacheslav Perepelytsya, and Maasha Maheson |
| HVAC |
| Internet of Things Capstone Project |

|  |
| --- |
| Humber College  4-23-2018 |

# Declaration of Joint Authorship

We, Jan Fontanosa, Vyacheslav Perepelytsya, and Maasha Maheson, confirm that this work submitted for assessment is our own and is expressed in our own words. Any uses made within it of the works of any other author, in any form (ideas, equations, figures, texts, tables, programs), are properly acknowledged at the point of use. A list of the references used is included.

# Proposal

2018-02-05

***Proposal for the development of HVAC***

Prepared by Jan Fontanosa, Vyacheslav Perepelytsya, and Maasha Maheson  
*Computer Engineering Technology Students*https://github.com/fntj0052/HVAC

**Executive Summary**

As a student in the Computer Engineering Technology program, I will be integrating the knowledge and skills I have learned from our program into this Internet of Things themed capstone project. This proposal requests the approval to build the hardware portion that will connect to a database as well as to a mobile device application. The internet connected hardware will include a custom PCB with the following sensors and actuators Touch sensor, moisture sensor, LCD touchscreen, and sound sensor. The database will store Operational status/condition and maintenance reminder. The mobile device functionality will include Ability to toggle operational mode and set maintenance schedule and will be further detailed in the mobile application proposal. I will be collaborating with the following company/department School of Applied Technology. In the winter semester I plan to form a group with the following students, who are also building similar hardware this term and working on the mobile application with me Jan Fontanosa, Vyacheslav Perepelytsya, and Maasha Maheson. The hardware will be completed in CENG 317 Hardware Production Techniques independently and the application will be completed in CENG 319 Software Project. These will be integrated together in the subsequent term in CENG 355 Computer Systems Project as a member of a 2 or 3 student group.

**Background**

The problem solved by this project is creating and improving upon the monitoring of Heating, Ventilation and Air Conditioning (HVAC) systems with a user-friendly interface, with the ability to remotely control the system activities using a mobile device, and to fetch stored information on the system's condition from a cloud database.. A bit of background about this topic is HVAC systems are useful in all kinds of building applications: a smarter system can provide significant energy and financial savings while scheduling usage and allowing more granular control for systems used in specific applications (a HVAC system used to monitor an industrial refrigerator room will require different settings in comparison to one used in a residential building). By providing the ability for remote control using an Internet of Things(IoT)-based HVAC system, administrators of the system can ensure that the system is working as intended and can administrate changes to the system in a secure manner..

Existing products on the market include [1]. I have searched for prior art via Humber’s IEEE subscription selecting “My Subscribed Content”[2] and have found and read [3] which provides insight into similar efforts.

In the Computer Engineering Technology program we have learned about the following topics from the respective relevant courses:

* Java Docs from CENG 212 Programming Techniques In Java,
* Construction of circuits from CENG 215 Digital And Interfacing Systems,
* Rapid application development and Gantt charts from CENG 216 Intro to Software Engineering,
* Micro computing from CENG 252 Embedded Systems,
* SQL from CENG 254 Database With Java,
* Web access of databases from CENG 256 Internet Scripting; and,
* Wireless protocols such as 802.11 from TECH152 Telecom Networks.

This knowledge and skill set will enable me to build the subsystems and integrate them together as my capstone project.

**Methodology**

This proposal is assigned in the first week of class and is due at the beginning of class in the second week of the fall semester. My coursework will focus on the first two of the 3 phases of this project:  
 Phase 1 Hardware build.  
 Phase 2 System integration.  
 Phase 3 Demonstration to future employers.

*Phase 1 Hardware build*

The hardware build will be completed in the fall term. It will fit within the CENG Project maximum dimensions of 12 13/16" x 6" x 2 7/8" (32.5cm x 15.25cm x 7.25cm) which represents the space below the tray in the parts kit. The highest AC voltage that will be used is 16Vrms from a wall adaptor from which +/- 15V or as high as 45 VDC can be obtained. Maximum power consumption will be 20 Watts.

*Phase 2 System integration*

The system integration will be completed in the fall term.

*Phase 3 Demonstration to future employers*

This project will showcase the knowledge and skills that I have learned to potential employers.

The brief description below provides rough effort and non-labour estimates respectively for each phase. A Gantt chart will be added by week 3 to provide more project schedule details and a more complete budget will be added by week 4. It is important to start tasks as soon as possible to be able to meet deadlines.

No additional planned purchases

**Concluding remarks**

This proposal presents a plan for providing an IoT solution for for a user-friendly, Internet of Things-based HVAC system. This is an opportunity to integrate the knowledge and skills developed in our program to create a collaborative IoT capstone project demonstrating my ability to learn how to support projects such as the initiative described by [3]. I request approval of this project.

# Abstract

This technical report is about the build progress and other documentation about our Internet of Things (IoT)-based Heating, Ventilation, and Air Conditioning (HVAC) system. The HVAC system controls the temperature, moisture, and sound generated from the system depending on the user’s desire. The HVAC system is eco-friendly and optimizes energy and financial savings in a residential, commercial or industrial environment. These smart HVAC systems increases the efficiency of the working and production environment.

Table of Contents

[Declaration of Joint Authorship 1](#_Toc506207534)

[Proposal 2](#_Toc506207535)

[Abstract 5](#_Toc506207536)

[List of Figures 8](#_Toc506207537)

[Introduction 9](#_Toc506207538)

[Software Requirements Specification 10](#_Toc506207539)

[Introduction 10](#_Toc506207540)

[Purpose 10](#_Toc506207541)

[Document Conventions 10](#_Toc506207542)

[Intended Audience and Reading Suggestions 10](#_Toc506207543)

[Product Scope 10](#_Toc506207544)

[References 10](#_Toc506207545)

[Overall Description 11](#_Toc506207546)

[Product Perspective 11](#_Toc506207547)

[Product Functions 11](#_Toc506207548)

[User Classes and Characteristics 11](#_Toc506207549)

[Operating Environment 11](#_Toc506207550)

[Design and Implementation Constraints 11](#_Toc506207551)

[User Documentation 11](#_Toc506207552)

[Assumptions and Dependencies 11](#_Toc506207553)

[External Interface Requirements 12](#_Toc506207554)

[User Interfaces 12](#_Toc506207555)

[Hardware Interfaces 14](#_Toc506207556)

[Software Interfaces 14](#_Toc506207557)

[Communications Interfaces 14](#_Toc506207558)

[System Features 14](#_Toc506207559)

[Temperature Reading 14](#_Toc506207560)

[Sound Reading 14](#_Toc506207561)

[Moisture Reading 15](#_Toc506207562)

[Other Nonfunctional Requirements 15](#_Toc506207563)

[Performance Requirements 15](#_Toc506207564)

[Safety Requirements 15](#_Toc506207565)

[Security Requirements 16](#_Toc506207566)

[Software Quality Attributes 16](#_Toc506207567)

[Business Rules 16](#_Toc506207568)

[Conclusion 17](#_Toc506207569)

[Recommendation 18](#_Toc506207570)

[Bibliography 19](#_Toc506207571)

[Appendix A: Glossary 20](#_Toc506207572)

# List of Figures

[Figure 1 Android application concept GUI pictures. Consists of the initial design for the Login, Power Management, Register, Sound, and Status activity pages. 14](#_Toc506207647)

# Introduction

The HVAC system solves the technical problem of a more efficient heating and cooling of a building. Smart HVAC systems do monitor temperature and moisture levels in the room but the noise generated from the HVAC systems is not monitored. The noise can make it difficult for room occupants to hear, which can be a safety hazard as the listening to prolonged noise can damaged the occupants hearing. The noise can also make it difficult for occupants to hear noise generated from other monitoring systems (for example, an alarm system).

This IoT system will monitor and control the temperature, moisture and sound levels in the room using a Broadcom-based development platform (also known as the Raspberry Pi 3 [RPI]), temperature, moisture, and sound sensors, and will transmit the readings from the sensors to a cloud-computing-based service (Amazon Web Services [AWS]) and NoSQL-based database (DynamoDB). The database will interface with an Android application and a website for easy remote access for the user.

The report will detail the process for the concept, building, and feasibility of the project for other users to develop and improve upon the project. Instructions on where to find and how to build the HVAC system are included along with information on issues that we have encountered while working on the project.

# Software Requirements Specification

# Introduction

## Purpose

The purpose of this document is to detail the requirements for the HVAC Android application, and the website component, and the Amazon AWS/DynamoDB database component. The Android application’s purpose is to generate a user-friendly graphical interface for our HVAC hardware and to provide a status report containing the temperature readings, sound-based detection functionality of the system with the maintenance report and scheduler. The database’s purpose is to collect readings from the HVAC system for the Android application and website to fetch from.

## Document Conventions

This document is written in English following software requirements specifications standards and occasionally uses bullet formatting as well as some concept images of the software product. Software development requirements featured in this document are listed with their own priority and weight.

## Intended Audience and Reading Suggestions

The intended audience for this document is the project supervisors and application users. This document assumes that the audience has basic technical knowledge with Internet of Things (IoT) projects, Android mobile programming, WordPress/web programming, and AWS/DynamoDB.

## Product Scope

The Android application scope is to help HVAC users to connect to, control and monitor the system remotely, and receive notifications about the system condition on the screen.

The database is there to store and retrieve information of the status of the hardware for every user-specified time interval.

# Overall Description

## Product Perspective

The product is the latest in the family of smart HVAC products, which monitor temperature and moisture. While maintaining traditional functionality it implements new technologies and provides new features and a new price.

## Product Functions

* Read Temperature level
* Read Moisture level
* Determine System Functionality via Sound Sensor
* Store Status Reports in the database for the user to look back on

## User Classes and Characteristics

This product will be used in residential, commercial and industrial buildings. The product will most frequently be used by homeowners and industrial workers.

## Operating Environment

The operating environment for the project will be the Raspberry Pi 3 with Raspbian Linux, the Android mobile operating system, AWS & DynamoDB for the database environment, and finally, WordPress/HTML for the website.

## Design and Implementation Constraints

Hardware limitations in terms of the sound sensor sensitivity (cannot be used in an extremely noisy environment), needs embedded C, Internet access for database & website, minimum of Android API 5.0 (Lollipop), app encryption/security, hardware susceptible to extreme weather.

## User Documentation

The technical report will be provided and the build instructions will be available online: <https://github.com/fntj0052/HVAC>

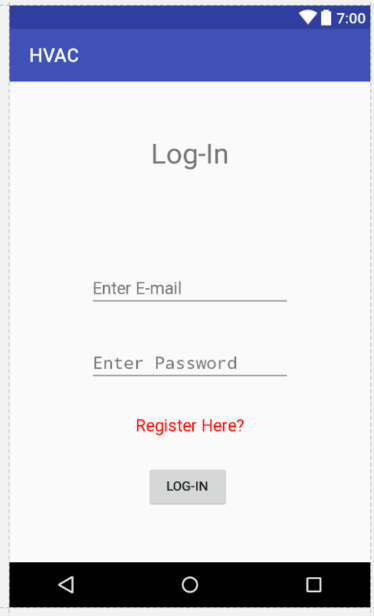
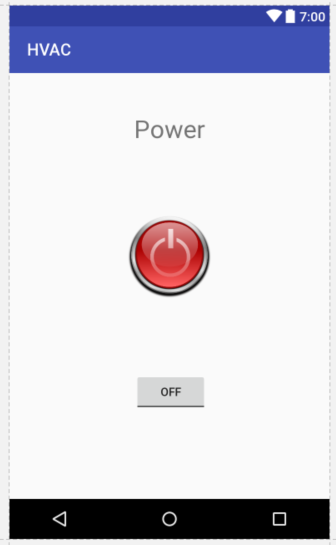
## Assumptions and Dependencies

Updates and changes to AWS/DynamoDB software and policies, Github, and Android may affect the functionality of the product.

# External Interface Requirements

## User Interfaces

Android GUI standards will be implemented for the mobile application.



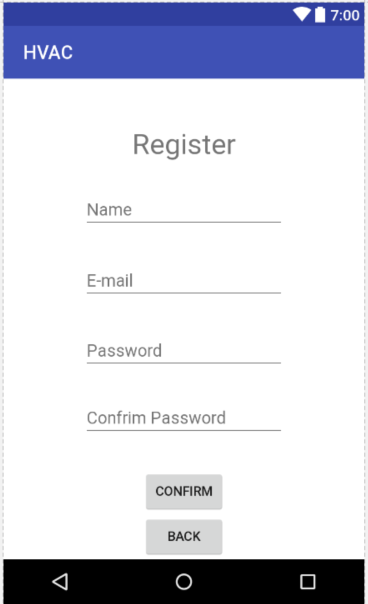
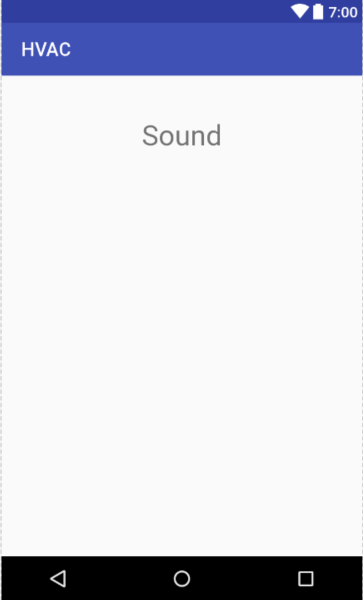
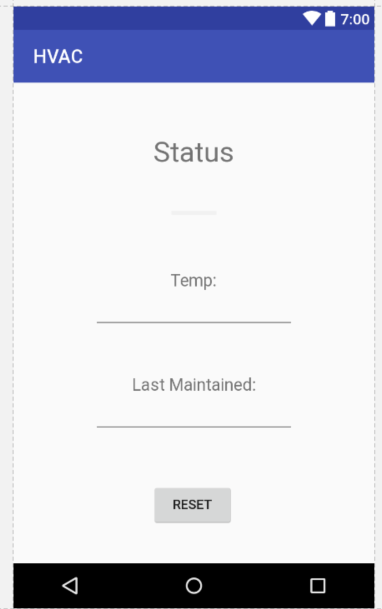


Figure 1 Android application concept GUI pictures. Consists of the initial design for the Login, Power Management, Register, Sound, and Status activity pages.

## Hardware Interfaces

For the mobile application, any Android device running Android 5.0 or up will be able to use the application. The software on the Raspberry Pi 3 can be used through the LCD and physical buttons on the keypad.

## Software Interfaces

For the web interface, a connection to the website through a browser will be necessary. The database can be viewed through the AWS website using an AWS account. The operating system used will be a Raspbian Linux. The product will rely on analogRead to get the analog data and will be processed on C to display to the user through text/images.

## Communications Interfaces

The product website can be accessed through any modern browser that uses the standard HTTP protocol. The website will be protected through free WordPress protection system but may be vulnerable to security issues.

# System Features

## Temperature Reading

4.1.1 Description and Priority

The temperature reading is an essential component of the HVAC which will provide a reading of the room temperature to the user and determine whether the HVAC system needs to turn on or off. It’s of HIGH priority.

4.1.2 Stimulus/Response Sequences

The user is entitled to read the temperature as well as stop and start or restart the system.

4.1.3 Functional Requirements

The temperature sensor needs to be functioning as well as the Raspberry PI and the database. In case of a specific error there will be a specific error message to the user, in case of a power outage error or a system shutdown the user will experience a blank screen.

REQ-1: TEMP

## Sound Reading

4.2.1 Description and Priority

The sound reading is MEDIUM priority component of the HVAC which will provide a reading of the noise level of the system environment to the user and determine if the HVAC system is operating at a safe sound level.

4.2.2 Stimulus/Response Sequences

The user is entitled to read the sound level status as well as stop and start or restart the system.

4.2.3 Functional Requirements

The sound sensor needs to be functioning as well as the Raspberry PI and the database. In case of a specific error there will be a specific error message to the user, in case of a power outage error or a system shutdown the user will experience a blank screen.

REQ-1: SOUND

## Moisture Reading

4.3.1 Description and Priority

The moisture reading is MEDIUM priority component of the HVAC which will provide a reading of the room moisture levels to the user and determine whether the HVAC system needs to turn on or off.

4.3.2 Stimulus/Response Sequences

The user is entitled to read the moisture sensor status as well as stop and start or restart the system.

4.3.3 Functional Requirements

The moisture sensor needs to be functioning as well as the Raspberry PI and the database. In case of a specific error there will be a specific error message to the user, in case of a power outage error or a system shutdown the user will experience a blank screen.

REQ-1: MOIST

# Other Nonfunctional Requirements

## Performance Requirements

The Android device should be capable of running Android 5.0 (Lolipop) and have a good internet connection for the AWS/Dynamo DB database to transfer information properly.

## Safety Requirements

If the device is installed in an extreme weather environment it could damage or harm equipment and people. For example in an earthquake situation the device may affect or damage nearby equipment or people if it is installed improperly or unsafely. Floods and fires may also have unpredictable consequences for the system and nearby devices/people.

## Security Requirements

The user does not need to provide any personal information aside from the email and password for the Android application.

## Software Quality Attributes

Sensor accuracy, reliability and a user-friendly GUI are the main additional quality characteristics.

## Business Rules

The owner of the product and the people they share the device access with will be the only users of the product and as such perform all the possible user roles. In cases of malfunction an additional person may be involved for the repairs.

# Conclusion

# Recommendation

# Bibliography

Al-Ali, A. R., Alikarar, M., Gupta, R., Rashid, M., Zualkernan, I.A. (2017). A smart home energy management system using IoT and big data analytics approach. IEEE Transactions on Consumer Electronics, 63(4), 426-434. Retrieved from <http://ieeexplore.ieee.org/document/8246800>

Amazon Web Services, Inc. (2012, August 10). Amazon DynamoDB Developer Guide. Retrieved from <http://docs.aws.amazon.com/amazondynamodb/latest/developerguide/>

Amazon Web Services, Inc. (n.d). AWS IoT Developer Guide. Retrieved from <http://docs.aws.amazon.com/iot/latest/developerguide/>

Google LLC & Open Handset Alliance. (n.d). Android API Guide. Retrieved from <https://developer.android.com/guide/index.html>

Ecovent Systems Inc. (n.d.). Ecovent. Retrieved from <https://www.ecoventsystems.com/smart>

Institute of Electrical and Electronics Engineers. (2015, August 28). IEEE Xplore Digital Library [Online]. Retrieved from <https://ieeexplore.ieee.org/search/advsearch.jsp>

Wordpress.org. (n.d). WordPress APIs << WordPress Codex. Retrieved from <https://codex.wordpress.org/WordPress_APIs>

# Appendix A: Glossary

**AWS** – Amazon Web Services. Cloud computing-based software.

**DynamoDB** – A NoSQL database service that can be used with AWS

**Github** – A website to share and develop code/projects using version control

**IoT** – Internet of Things. Products/systems that connect to and communicate using the Internet

**Raspberry PI 3** – a recent version of a portable computer/Broadcom development platform used for practical projects and programming education.