Doggie Doorman

Commented [CW1]: Only dog? Cat?

California University of Pennsylvania

CSC 490: Senior Project I

Specification Document

Dr. Chen

11/12/2020

Group Members

XXX

XXX

XXX

XXX

Instructor Comments/Evaluation

Outline is well followed.

Writing is smooth.

Reference & Citation is good.

Statechart is well done.

Dataflow diagram needs improvement.

Table of Contents

Commented [CW2]: ToC can be improved with intended titles based on their levels.

Commented [CW3]: ToC normally only consists of section/subsection titles, no Figures.

Abstract

Our project, Doggie Doorman, is an automatic dog door that can be installed in a home or kennel. It is a pet door that will automatically unlock when the user's pet decides that it wants to enter or exit the house. The project will use an electrified magnetic lock, RFID reader, and RFID chip. A Raspberry Pi microcontroller will be used to drive the hardware and act as the brains of the project. The Pi device will have the ability to connect to the owners WIFI network, giving the user the ability to control the door remotely. The user will be able to control the door based on different variables such as day, time, and weather. This document will cover all necessary details for the end user as well as the development team regarding the functionality of the final product.

Description of the Document

Purpose and Use:

The purpose of this document is to define the scope and specific requirements for the project. These requirements will include, but are not limited to, the product's functionality and system requirements. The developers and client will use this document to state the terms of their arrangement. The client has a right to dispute the contents of this document, so the product properly reflects their desired requirements. After acceptance of the terms stated in this document, it shall be known that this document becomes a binding contract.

Intended Audience:

This document's intended audience include the client and the product developers. The product developers have created this document to make an outline of their finished product. By doing so, it helps the client to know exactly what product, functionality, and the cost that will be associated with this project. The contents in this document create a set of guidelines for the developers to follow for the duration of the project. It also assists the development team to know which direction they are headed in and states their priorities for the project. Any uncertainty in this document needs to be addressed as this serves as a binding contract.

System Description

Overview:

The Doggie Doorman is a combination of physical hardware integrated with a software application, accessed via a web app, to control the hardware. The main part of the Doggie Doorman is the application that allows the user to interact with the door. A user will be able to connect to the door to control its ability to be locked and unlocked based on different situations that may arise. The hardware portion of the Doggie Doorman is how the RFID chip will work with the door. Only pets with the correct RFID chip will be permitted to activate the door given that the user has not overridden the lock feature for the entire door. The use of a unique RFID signal from each of the chips will allow the user to control which pet will have access to the door.

Environment and Constraints:

End User Profile

The most common end users of the Doggie Doorman are pet owners, animal shelters, and kennels. Users will need to be capable of using mobile applications to operate and interact with the door. With a basic understanding of how to navigate a simple mobile application, anyone will be able to operate the door. It is expected that the Doggie Doorman will be used primarily in a residential setting, however it can be scaled up to accommodate a commercial setting.

User Interaction

The user interaction will be through the software application that controls the door. After downloading the application, the user will have the option to enter RFID chip tags into the database, allowing their pets to have access to the door. Once the user enables the specific RFID chip signal, they will then be able to control access to the door based on a particular RFID sequences or a complete door lock override. The user will also be able to set times and dates for when they would like the door to be locked on a reoccurring basis.

Hardware Constraints

The hardware constraints will require that the user has a mobile device that is able to download the mobile app that controls the door. Without a mobile device, the user will

not be able to connect to the door and use all its intended features. It is also required that the user has enough RFID chips for all their pets that they wish to grant access to the door. The door will only open if the correct RFID sequence is detected by the door. Therefore, if a pet does not have an RFID chip on their collar, they will have no way to activate the door on their own.

Commented [CW4]: Pets activate the door?

Software Constraints

The software constraints are limited to the operating system and the programming languages that allow us to have control over our hardware components. Raspberry Pi OS, also known as Raspbian and Linux Ubuntu are two of the leading choices for the operating system software. Both are open source and have active online communities for troubleshooting. We need to have a software language that can relay RFID signals and allow the maglock to release, therefore enabling the door to open. Python is among the leading languages due to its simple syntax. Another software constraint is the mobile application that the user will use to interact with the device. The device that the user has must be compatible with the software application being created. We will focus our efforts on developing the mobile app for one platform, such as ~~android~~Android, and one browser such as Google Chrome. To implement the mobile app, we will utilize an open-source web app framework such as React or Angular ("Raspberry PI RFID" 2020).

Time Constraints

The time constraints for this project will be in an academic setting meaning that we will have approximately fifteen weeks to complete this project. There will be several hours per week dedicated to the development of the project amongst all members of the group. Given that this is in an academic setting, all group members are not able to dedicate their entire workload to the project as there are other classes that require attention. It is expected that the team will have a set of markers to indicate how far along they are in the development of the project. As a team we anticipate that we have the appropriate amount of time to execute all the features expressed in this document in our given time frame.

Cost Constraints

The cost constraints will be exclusively on the hardware portion of the project. The software will be open source, so no cost will be involved. The hardware will include the Raspberry Pi board and all necessary components to include but not limited to, an enclosure and power supply. A mag lock, RFID reader, and RFID chip will factor in the remaining cost. The total anticipated cost of the project is roughly $160 dollars to produce, to include sourcing a pre-manufactured pet door (“RFID” 2019).

Other Concerns

The end-user will need to have WIFI access to best utilize all the functions of their Doggie Doorman. If they are unable to connect to the software via the mobile app, the user will be unable to customize the product to their specific needs. The user would

also not be able to add/remove RFID tags without directly connecting to the Raspberry Pi device with a hardwired connection. The user would then need to understand how to navigate a coded environment to manually add/remove or further customize their device.

Acceptance Test Criteria:

Testers

The project team members for the Doggie Doorman project will also be the main testing and quality assurance team for the project. All team members will perform individual testing as well as group testing on all portions of the project to include the hardware, software, and mobile app interface. Testing will be done consistently to ensure that each newly completed portion of the project integrates successfully into the larger project. Due to the nature of this project, the team will be following a linear project management method known as the waterfall methodology.

Criteria for User Acceptance

Successful user acceptance for the Doggie Doorman is when the customer can use the mobile application to access the door and set permission with an easy-to-use mobile application interface. The following criteria will make the Doggie Doorman a success per the project team's internal requirements.

- The physical door and hardware should be easily installed.

Commented [CW5]: How about the door open and close correctly? i.e., only valid RFID tag can open the door and the door will close after certain amount of time?
Also these criteria do not include the weather info, which is included in Figure 1

- The end-user can log into the mobile app and create an account and sign into their device.
- The end-user can create and remove RFID tags for pets as needed.
- Individual door permissions can be set for each tag.
- An override feature for the door should be easily accessible.
- Door statistics should be made available to the end-user.

Integration of Separate Parts and Installation:

The software will be pre-loaded on the Raspberry Pi device which will serve as the main communication channel with all the attached hardware devices. The Raspberry Pi will communicate through WIFI with the front-end mobile app. The mobile app will be accessed through a free to download web app. All permissions and door statistics will be stored locally on the Raspberry Pi device.

System Modeling

Functional: Use Cases and Scenarios

Upon opening the Doggie Doorman app, the user can choose between four options. These options are shown in the diagram below.

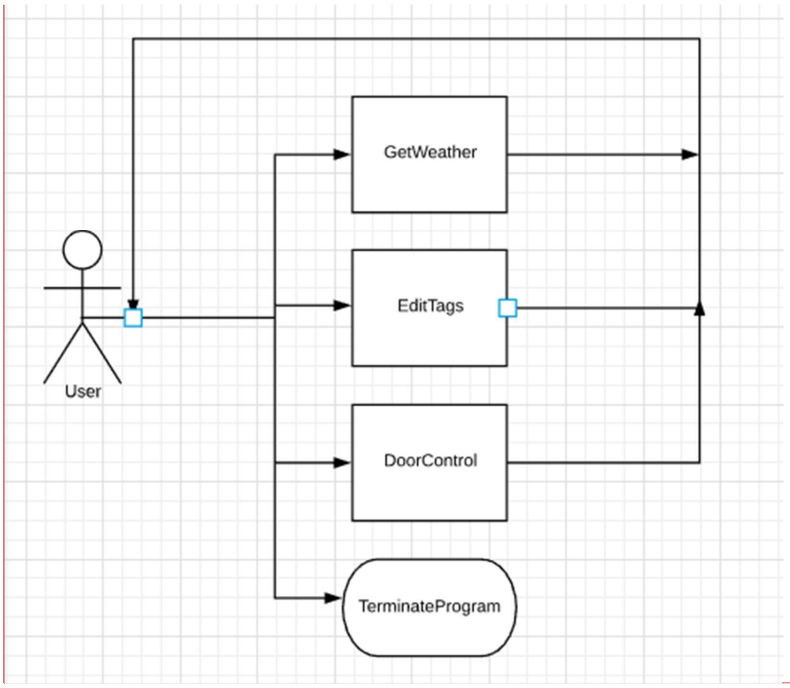

Figure 1: General Flow of DoggieDoorman App

Figure 1 above displays a simplified Use Case Diagram as the User enters the app. GetWeather will display for the user details about the current weather. EditTags allows the User to add or remove tags (or pets) from the database. The DoorControl option allows the user to set a lockdown on the DoggieDoorman or remove such a lockdown. After finishing these cases, the user is returned to the starting menu. TerminateProgram allows the user to close out the app.

Run Scenario

Commented [CW6]: Where is the user case for the Pets?

- User enters app
- User chooses one of the following options from the main menu:
    a. EditTags
    b. DoorControl
    c. GetWeather
    d. Terminate (End Program)

a. If EditTags is chosen, user then chooses between:

    i.) Add Tag

    ii.) Remove Tag

    iii.) View Tags

b. If DoorControl is chosen, user must choose between:

    i. Unlock door

    ii. Lock Down door

c. If GetWeather is chosen, the weather class is returned to the user in a viewable format.

d. If Terminate is chooses, the application will end and be closed.

Entity: Class Diagrams:

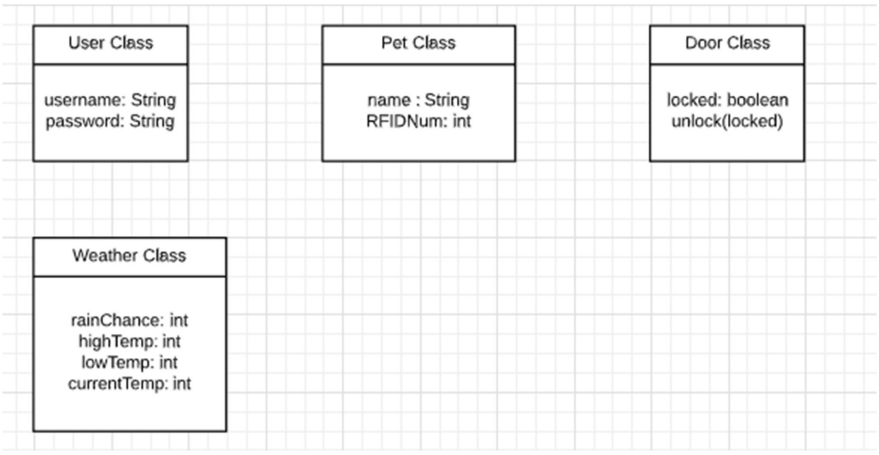

Figure 2: Class Diagrams

Class Descriptions

User Class

The User class holds the user's password and username data, which is used during the initial connection process.

Pet Class

The Pet class contains two variables. The first is a string housing the name of the pet, and the second is an integer number that matches to the RFID chip on the pet's collar.

Door Class

The Door class holds a Boolean value *locked*, when this variable is true the door is in lock down mode. The Door class also has the function *unlock(locked).* This function will unlock the door for a set amount of time if the *locked* variable is false.

Weather Class

The Weather class houses four variables. The first in an integer value representing the percent chance of rain for the day. The second is an integer value denoting the broadcasted high temperature of the day, and the third represents the low temperature of the day. The fourth variable is an integer representing the current temperature. When called upon, this data is displayed in a readable format (Karlin 2019).

Dynamic State Chart

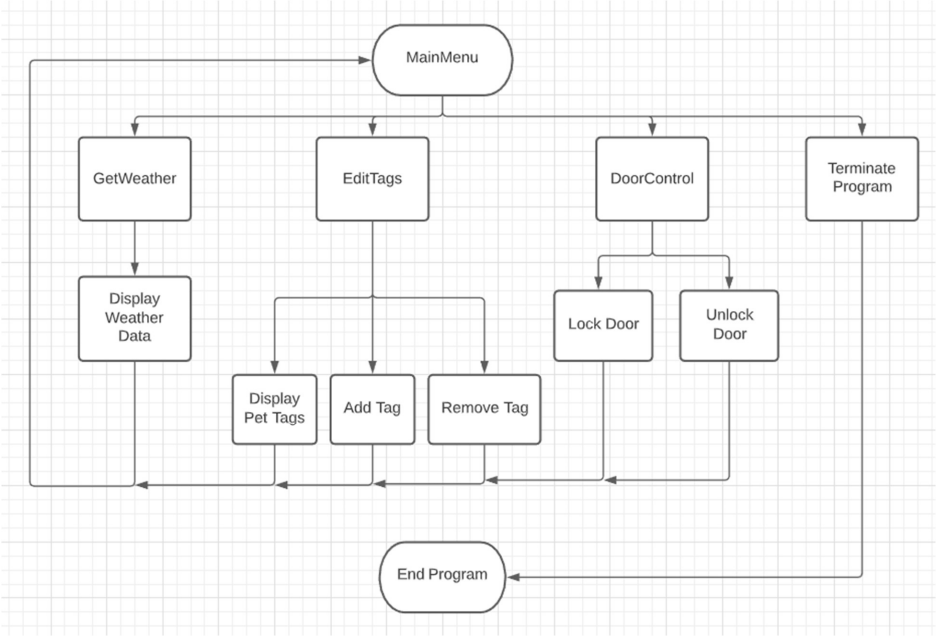

Figure 3: Dynamic State Chart

States

- The user selects the application from their mobile device's main menu.
- Once the application is loaded, the user has four options within the application
    - State 1: Weather interface is loaded for the user
    - State 2: Pet tag control screen is loaded and allows the user to make changes to what RFID chips are registered to the door.

- State 3: Door control screen is loaded for the user so that they can make adjustments to when the door is locked versus unlocked.
- State 4: This is the final state that will terminate the program and allow the user to exit the application.

Transitions

- Once the application is loaded onto the user's mobile device, there will be a series of options for them to choose. The initial state will always be the home screen which can lead to three other screens before termination. It is possible to loop as many times as the user would like between the GetWeather, EditTags, DoorControl, and Main Menu screen. Only if the user chooses the TerminateProgram option will they exit the program.

Dataflow Diagrams

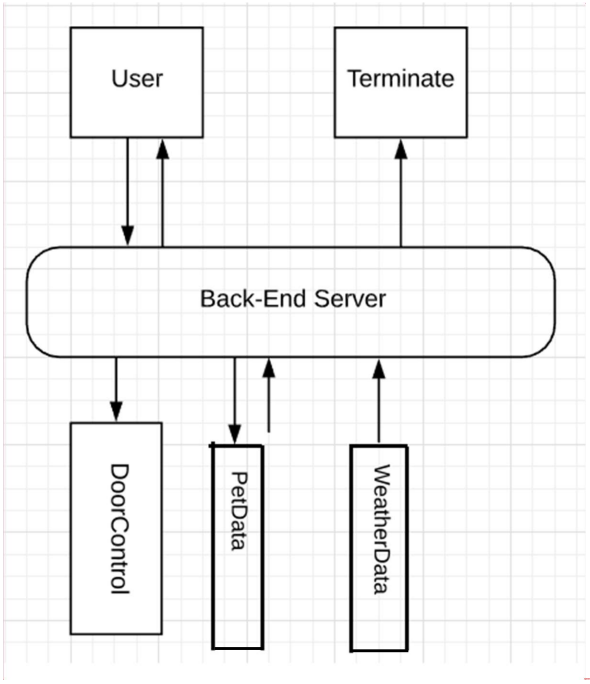

Figure 4: Data-Flow Diagram

Commented [CW7]: This is not collaboration nor sequence diagram

Components / Tools Needed

A Raspberry Pi microcontroller will be used to drive the hardware components of the Doggie Doorman ("Raspberry PI RFID" 2019). Connected to the microcontroller will be an RFID chip reader. This reader will be used to detect nearby RFID chips and communicate to the microcontroller when a pet is within the field of view of the device. The Doggie Doorman uses

an electrified magnetic lock is used to keep the pet door secured. An IDE will be used to create a GUI web app that runs on the Android operating system.

References

DevicePlus Editorial. (2019, November 22). Raspberry PI RFID Reader- How To Integrate RFID With Raspberry Pi. Retrieved October 19, 2020, from https://www.deviceplus.com/raspberry-pi/integrate-rfid-module-raspberry-pi/

Karlins, D., & Sahlin, D. (n.d.). How to Embed the Weather Forecast in Your Website. Retrieved October 19, 2020, from https://www.dummies.com/web-design-development/site-development/how-to-embed-the-weather-forecast-in-your-website/

RFID Systems. (2019, March). Retrieved October 19, 2020, from http://www.ia.omron.com/support/guide/47/introduction.html

Appendix I: Technical Glossary

API – Application programming interface

Discord – A communication application for messages and code files

JavaScript – A programming language that allows for development of mobile applications

Linux – An open-source operating system

Mobile Application – A portable application that can be run on a portable device

OneDrive – Microsoft's cloud-based service

Raspberry Pi – A single board, pocket sized computer running Linux

RFID – Radio frequency identification, used for access control

WIFI – Wireless networking technology. Allows devices to communicate over wireless systems

Maglock – Low voltage powered lock that interfaces with an RFID reader

React – A web framework to make web applications

Angular – A web framework to make web applications

Raspbian – An operating system exclusively used on Raspberry Pi devices

Linux Ubuntu – An open source operation system

Android – A mobile operating system directly competing with Apple

Waterfall Methodology – A typical development method in the software development industry

Front End – Typically denotes the part of the software which the user can interact with

Back End – Typically denotes where information and permissions are held, not accessible to end-users

Field of View – The area which the device is able to recognize

Open Source – Free to use, open source code, not requiring a license to modify

Appendix II: Team Details

The leader of this documentation was xxxx. All members took part in proofreading, formatting, and grammar evaluations. Each member of the team completed their section as well as helped to add to other members sections. The requirement document was created by all the following individual efforts:

X was responsible for creating the system model and data flow sections for the document.

X was responsible for the description of document system description part of the document.

X was responsible for the abstract, run scenarios, and class functions of the document.

X was responsible for taking the paper to the writing center and the components section of the document.

Appendix III: Workflow Authentication

I, , hereby attest that I have performed the work as documented herein.

Signature:

Date: 11/12/2020

I, hereby attest that I have performed the work as documented herein.

Signature:

Date: 11/12/2020

I, , hereby attest that I have performed the work as documented herein.

Signature:

Date: 11/12/2020

I, , hereby attest that I have performed the work as documented herein.

Signature:

Date: 11/12/2020

Appendix IV: Report from Writing Center

Cal U Writing Center Report

Client: xxxx

Staff or Resource: Sierra M.

Date: November 12, 2020, 5:00pm - 6:00pm

Did the student request that the instructor receive a visit report?: Yes

What course was serviced by this visit?: CSC490 - Sen. Proj. I: Software Engineering

What goals were established for this tutoring session?: Reviewing - rereading to see if the project is organized and developed

How did the process of this consulting session address the established goals?:

The client, xxxx, scheduled an appointment to review the project of Doggie Doorman. Andrew was the one that scheduled the editing appointment. His team members are xxxxx who are also able to read my suggestions. Doggie Doorman appears to be efficient to pet owners and pet places. There are a few grammar, APA formatting, and edit suggestions to apply to the project. The project seems to cover all the needed criteria. Overall, is a well organized and detailed idea and essay.

Please provide any additional comments relevant to this session

The project idea is very detailed with nice demonstrations through images!!!