CEN 4010 Principles of Software Engineering, Spring 2018

Milestone 2 Project Proposal and High-level description

Flight Control: Internet of things

Group 12

Members:

Ernesto D. Rivero

(riveroe2015@fau.edu)

Michael Anderson

Shawn Ponciano

Kevin Vallejo

Date: 2/10/2018

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| History Table | |  | | | |
|  | Revision # | | Revision Date | Revision Description | Revision Notes |
| 1. | V.1 | | 2/10/2018 | Completed M1 requirements | N/A |
| **2** | V.2 | | 3/24/2018 | Data definitions sub data items added | M3 requirment |
| **3** | V.2 | | 3/24/2018 | Expanded high level functional requirements | M3 requirement |
| **4** | V.2 | | 3/24/2018 | Modified high level system architecture and included DB organization | M3 requirement |
| **5** | V.2 | | 3/24/2018 | included UML diagram | M3 requirement |
| **6** | V.2 | | 3/24/2018 | identified actual risks | M3 requirement |

# 2. Executive Summary

We are developing an application to network individual devices together and wrap their interfaces into one concentrated control center. Under the working name, “Flight Control”, the application will give access to various nodes and stations to the user despite them not being physically close to them. A user will have a control center streamlined that will expand into each device's sub-menu and user controls. For example, a user might want to authenticate themselves at a workstation or clock their time spent in the lab. The application would provide the tools to record clock in times matched with their authentication time stamp to give a verifiable record and trusted documentation. Additionally, a user may wish to modify settings on a particular device such as the light intensity of a controlled bulb or the volume to a speaker. The app should allow similar integration of functionality in its evolution cycle.

# 3. Competitive Analysis

|  |  |  |
| --- | --- | --- |
| Key Features | Us | Competitors |
| In the case of power failure, automatically reset to their last known state | No safety measures in place in the case of a power outage |
| Two-factor authentication | Vulnerable security measures |
| Optimized for mobile | Un-optimized for mobile |
| Automatically assign available work station | Requires user interaction |

The advantages of our planned features, as opposed to those of competitors, is the ability to provide consumers with a secure, optimized, and user friendly product. We aim to have the users efficiently, and effortlessly use our product as opposed to the already available applications with obscure user interface and inaccessibility. Also, as a means to cut down the need for interference by an administrator, it is our goal to offer the user an easy automated system.

# 4. Data Definition

|  |  |
| --- | --- |
| Data Definitions |  |
| ESP8266 | A WI-FI enabled microcontroller |
| E-WeLink (App) | A smart home remote control software that can remotely control hardware products regardless of its type and brand. |
| API | (Aka application programming interface) is how a remote server communicates with a client. When an API receives a request from a client, it transfers that data to the remote server, the server will then send a response back to the client based on the clients’ initial request through the API. |
| SDK | (AKA software development kit) are tools used by developers in order to help them program applications for a specific operating system. |
| Frameworks | A basic structure (or template) developers use in order to add a greater functionality to that system. |
| Internet of Things | The interconnection of everyday objects with computers via the internet. |

|  |  |
| --- | --- |
| Database | A server used to store and organize information so that it can be easily accessed and managed |
| Server | Computer hardware or software that manages access to a centralized resource or service in a network. |
| Table | A collection of related data consisting of rows and columns used by the database. |
| myPHPAdmin | A tool written in PHP to handle the administration of MySQL over the web. |

# 5. Overview, scenarios, and use cases

Students would utilize their mobile device to access a website that would be used to gain access to work benches and machines in the Engineering buildings. They would be assigned an available bench. To gain access to work benches and machines, the student would either have to be in a university course that permits the use of the facilities or they would have to be in a special “learn to use” class offered by some faculty members. In any other case the student can show an admin that they can operate the machinery at a competent level, and the admin can allow access. Student use their Z number to log-in. If a assigned bench is not to the student’s liking then they can ask for a reassignment. The student would take a picture of the bench to send, so that an admin can confirm that the bench is not as it should usually be.

Administrators would oversee the benches and allow benches to be used by students who do not meet the formal requirements to use a work bench. They can also allow more time for students whose bench time is close to done.

# 6. ~~Initial list of~~ high-level functional requirements

The system must be able to respond to specific situations and inputs:

1. If connections have been interrupted
2. If connections have become unsecured
3. An internal inventory used by admin:
   1. Admin will be able to view a specific lab equipment or workstation reservation history
   2. A list of current students and the option to view their profile.
   3. Flag students for misconduct in order to temporarily suspend them from lab equipment.
   4. Add a new device / equipment.
   5. Remove a device  / equipment
   6. Access history of using devices
   7. View a list of all devices
   8. Favorite a set of devices
4. Control Center for workbenches will allow student users to:
   1. Check in/out of a workstation
   2. report any issues regarding the workstation
   3. Check in for class use when requested by professor.
5. Reservations on lab equipment / workbenches:
   1. Students will be allowed to reserve specific equipment or workstations based on the courses they’ve taken in the past.
   2. Admin will be able to request to reserve lab rooms if they we require it for a lecture or a workshop purposes.
6. Login for both students and admin
   1. Student and admin profiles will be uniquely different based on their roles and/or their prior history.

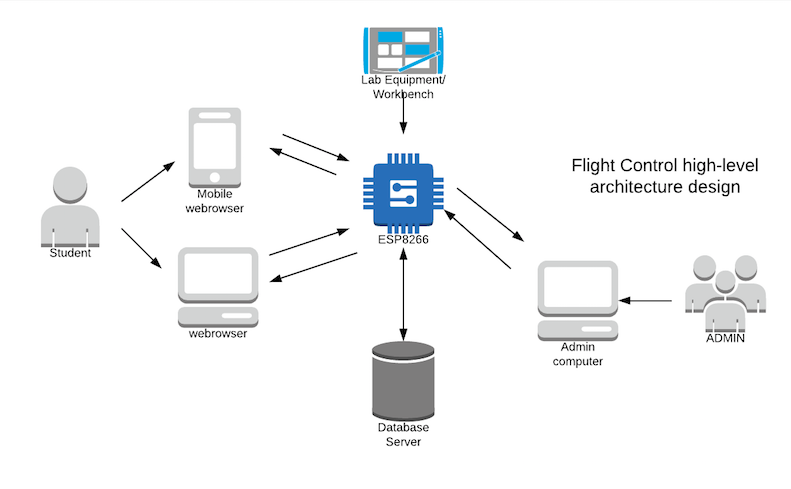
# 7. List of non-functional requirements Initial list of high-level functional requirements

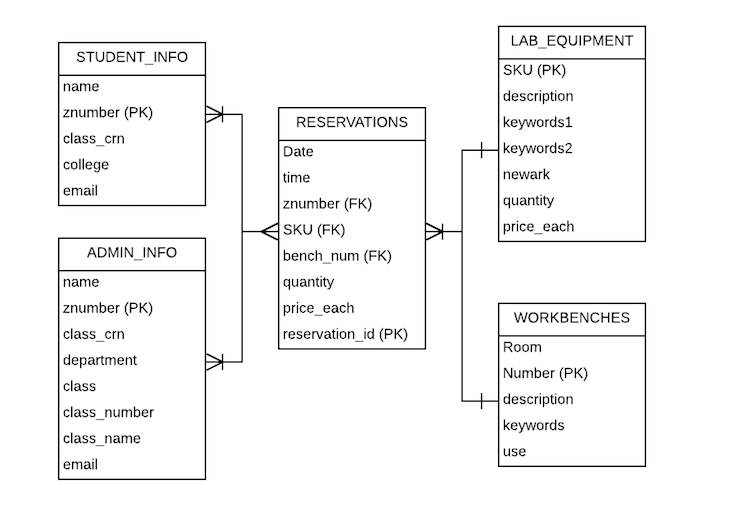
The application will have required parameters to be met:

* Connection with authentication server must be secured and private
* Connection with devices must be maintained
* A well-organized user interface must be interact-able intuitively
* The system must be flexible for added functionality
* System must have database interactions capable

# 8. High-level system architecture and database organization

1.) **High Level architecture: (next page)**

****2.) **DB organization:**



# 3.) Media storage: Video/audio will be kept in a file system, no format requirements as of yet.

4.) **Search / filter architecture and implementation:** The data stored in the databased will be coded and organized as B+ trees, thus B+ searching algorithms will be used in order to find / filter/traverse the tree.

5.) **Our own API’s:** A log in page for all users will be developed and communicate with the servers using mySQL and PHP, if the criteria matches with whats stored inside the myPHPAdmin then users will be allowed access to all of the programs internal functions.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | | |
| Software Products/Tools | **Languages** | **APIs** | Supported Browers | **Frameworks** | |
| E-WeLink App | Arduino Language | E-Welink API | Chrome |  | |
| ESP8266 | HTML |  | Safari |  | |
| SONOFF Modules | CSS |  | Firefox |  | |
|  | PHP |  | Mobil Web browsers (ex: Samsung Internet) |  | |

6.) **Significant Non-trivial algorithm or process:** N/A as of yet

# 9.) High-Level UML diagram: (View next page for UML)

# 

# 10.) Identify actual key risks for your project at this time

1. Skill risks:

Within our group we have some members with the necessary experience and understanding to build specific components during the development and testing stages. But at the same time, we have otherwise highly competent members who are coincidentally lacking in a specific set of skills. We found that one member did not have the prior experience with web design in PHP, making it foolish to assign that member to work on the practical prototype for querying a live database. To remedy this simple issue, a common sense approach of delegating this member to focus on documentation while the rest of the group's resources were spent to quickly code the working prototype. We then reconvened as a group in a voice chat to facilitate a proper debrief and bring all members up to speed and on the same page.

1. schedule risks

As far as schedule risks, every group is vulnerable to scheduling conflictions. Healthy communication with proper facilities to support it resolves this issue cleanly. If each member knows what work they have and the timetable is broken down and posted in a pinned message, there is little room for confusion

3.) Technical Risks

In a prior assignment's development cycle, we found that our group was vulnerable to miscommunications as a result of the chosen medium for our group chat. SMS proved to drop messages at key points and resulted in a week of setbacks. Without a centralized hub of communication with multiple channels, information could be lost due to dropped packets or lost within the very sea of text with constant updates and unrelated posts. Additionally, work done on multiple machines in potentially different programs with unique configurations can lead to formatting issues of the final output. To remedy this issue, we established a centralized hub of communication in a shared discord server. There are specific channels for each vector of development specific to each assignment. Each member can see who is available and easily get their attention. Messages are consolidated on a server and thus will refresh until displayed on each client if anything was ever skipped or missed due to internet connectivity issues. Lastly, important posts can be pinned and work formatted in a unified way. This centralized formatting and communication allows us to keep the formatting and setting consistent as well by having one dedicated person to creating the final output with input and suggestions after posting pushes to the proper development channel.