CEN 4010 Principles of Software Engineering

Milestone 3: More Detailed Requirements, Architecture and a Vertical Software Prototype

Florida Atlantic University

Dr. Shihong Huang

shihong@fau.edu

# Instruction:

Milestone 3 consists of two parts:

1. Milestone 3 document – an expanded version of Milestone 1
2. A vertical software porotype

**Part 1: Milestone 3 document**:

Milestone 3 has to be reasonably consistent with Milestone 1 and instructors’ feedback but it can also differ from Milestone 1 based on what you discover and develop in your design process in spirit of iterative software engineering process and based on the feedback you get.

The difference between M1 and M3 DO NOT need to be edited in Milestone 1 document which remains frozen. You should start with Milestone 3 only after you have incorporated instructors’ feedback on Milestone 1. Milestone 3 document is a separate document from Milestone 1.

**Part2: Vertical software prototype**

In addition to the Milestone 3 document, the team will create a “vertical software prototype” to test the infrastructure and chosen frameworks and to jumpstart the coding effort. The vertical prototype is the code that exercises full deployment stack from browser, via middleware, to DB and back-end, including your chosen framework. It has to be deployed from team account, in the same way that the final product will be deployed. For example, it shall allow one to enter a search term in the browser, then get a response form the DB and render it back on the browser. GUI for this can be simple one field entry and DB can have only a few items. The items in your DB shall be encoded with full schema as it is defined by now. The purpose of vertical prototype is to early and quickly test basic software components and deployment infrastructure and frameworks as well as the key architecture patterns and thus to serve as a basic “scaffolding” for final product. It also serves as “teaching and training” tool to bring the rest of the team up to speed on development, frameworks etc. We recommend that back-end team be assigned the task of constructing this vertical prototype.

Milestone 2 Document must be in the following order:

# Title page

Software 6

# Executive Summary

Access Control Device

* This project is designed simply to grant access, monitor and keep track of users that want to use a workstation or any lab equipment (devices) located in some of the lab rooms in the Engineering East and West buildings. The app will grant two level of access Student or Admin. The Admin user will be able to add delete or block a user from accessing the equipment. The Student user will only be able to access the site for lab equipment or bench use. This will be implemented through a friendly user mobile app, that will require the user’s Z-number for authentication to log into the site. The site will be accessed via an internet browser on a mobile device (smartphones, tablets, etc.). This will keep track of all workstations and lab equipment that are being used or available, by having an identifiable number assigned to it. The app will support different types of workstations (soldering station, drill press station, microscope station, etc.). The app will randomly assign an available workstation to the student and will show a sample picture of an operable clean workstation. The site will provide the user a selection of two choices to choose from, one if bench is clean and operable and the other if is not. If the bench is not in good shape the user will be asked to take a picture of the workstation which will then be send to EE management team, while the student will be assigned a new available workstation. Once the condition of the workstation is verified, the user will be allowed to use the workstation for the allotted time. Afterwards the user will be required to take a picture of the bench to verify the condition of it. The site will periodically remind the user of their remaining time and will be warned when the time is approaching 0 min. The site will also keep track of over 200 devices (tools, soldering workstations, etc.) that are located in the Engineering East and Engineering West buildings. The site will periodically ping these devices to keep status of their state so in case of an outage the service will remember the state of this devices before the outage. Only students that take a university course at FAU will be granted access to this workstations and lab equipment.

# Competitive analysis

Modify based on Milestone 1. Add or change as you see necessary.

|  |  |
| --- | --- |
| Our Features | Competitors Features |
| Mobile Browser Support | Requires App Download |
| Photo Upload Support | Text-only responses |
| Admin and Student Accounts | Only admin accounts stored in database |
| Supports different types of station | Only support 1 or 2 types of stations |
| Authentication via Z-number | Authentication via Username/Password |

While there are several competitors already on the market, they are not specifically tailored for the education use of several different types of physical devices. Most competitor’s software focuses either factories, which consist of mostly one type of station (for example there could be a soldering factory, or a drill press factory), or they focus on education station reservation, which is usually specific to desktop computers.

# Data definition

This should be reasonably consistent with Milestone 1 but should be expanded as needed and refined as per feedback. Major data items that comprise of sub-data items have to be defined in full (list all its sub-data items, and for images/video list formats, max size etc.). You must use all the data definitions and names consistently in all documents, including GUI text. Focus on data items unique and important to your application and avoid explaining obvious things like Internet,, Browser, Cloud, etc. Be sure to cover ALL items critical to your project and especially those providing a competitive advantage. At this stage data describing user privileges, registration info and main info (raw data, metadata, supporting data) have to be fully defined (as much as it is possible at this stage)

Station – Synonymous with “Work Bench” - One setup of equipment designed for the use of one person. For example, a soldering station, drill press station, or microscope station.

Equipment – Any individual device, usually combined with other devices to make a station.

WiFi Module – A device that connects to a wifi network and accepts commands via an API to turn a 110V AC relay on or off.

Soldering Station – Consists of a soldering iron, roll of tin, bottle of flux, loop, and wire holder.

Z-Number – A unique numerical identifier for each student.

Smartphone – Cellular phone with the capabilities to display a web page.

Website – Publicly accessible HTML page.

Tablet – Handheld device capable of displaying a web page.

Engineering East/West – A building location on FAU campus.

Picture – A photograph taken by a camera or phone.

# Overview, scenarios and use cases

Modify based on Milestone 1. Add or change as you see necessary.

Product overview and its usage:

* Each workbench is assigned a number and when a user needs a bench the app will automatically search for ones that is available and assign it. The user will be shown a picture of a clean bench and will require to select if the bench is clean or not. If the bench is not clean then the user will be asked to take a pic of it and would also have the option of being reassigned.
* The site should also be mobile friendly since users will be accessing the site via an internet browser on their mobile device (smartphones, tablet, etc.) The user will be granted access to the bench or lab equipment only if they are enrolled in a University course. Users will be using their student Z-number to log into the site. Users will also be allowed access for a period of time only for a course session scheduled.
* All devices will have to be updated every few minutes in case of a power failures or other issues, so the device will resume its last state.
* This site will keep track of 200+ devices in Eng. West and East.

# ~~Initial list of~~ High-level functional requirements

Expand functional requirements from Milestone 1 into Milestone 3, with more details as necessary. Keep the same reference numbers with respect to Milestone 1 (i.e. if high level requirement was number 3 in Milestone 1, then in Milestone 3 more detailed requirements are 3.1, 3.2 etc.). Be sure to cover ALL and especially unique features of your product. OK to add new or delete previous functional requirements from Milestone 1, if you can justify it.

Prioritize each requirement/spec with 1, 2, 3. (1-must have; 2 – desired; 3 – opportunistic as defined in the class). To develop these priorities think of the user, use cases, and making your application complete from usability, marketing and business aspects. Base this also on your skills, resources and schedules. Instructors will check final priorities. The priorities you set later in Milestone 4 will constitute your commitment (especially priorities of 1), so be very careful.

1. Users will select their user type - 1
   1. Two total users
      1. Students: have limited access
      2. Admin: have access to more sensitive information
2. Users should be able to login to system with their FAU credentials -1
   1. Users will enter in Z-number and password into the site or scan their barcode with the barcode reader
3. Can scan the barcode on their owl card instead of entering Z-number and password - 3
4. The system will validate their credentials. -1
   1. The system will check the users FAU credentials to make sure they are a valid user.
   2. Valid users include Students, TA, and admin.
   3. This process will also pull any relevant additional information on the user such as the courses they are taking and the equipment that they can use.
5. Admin users can then select from various options -2
   1. Add User: Admins can add users who can use the equipment in the room
      1. Admin will fill out the user to be added information such as, their z-number and they type of bench they have access to.
   2. Remove User:
      1. Admin can remove user and deny them access to the lab. They must provide a reason why they are removing them. This will be reviewed by outside staff
   3. Block Access:
      1. Admin can choose from active users and control the types of benches that they can use
   4. Lab Settings:
      1. Admin can control the equipment allowed at certain benches and how long students can use the bench
6. Students will select the room number of the lab they are using - 1
7. Students will then be given a bench number -1
   1. The user will be asked if the workbench they have been assigned is clean. If it is, then they will be assigned that workbench. If not, then they will be asked to take a picture and be assigned a new workbench.
8. Once a valid work bench is assigned, users will have a set time limit for how long they can work for and limited access to the tools that they can use. - 1
   1. A timer will start once the user is assigned a valid workbench.
   2. They will only have access to equipment that they are qualified to use.
9. Once user is done with the bench, they are expected to clean up. -2
   1. Users are expected to clean the work area when they are done. This will be validated later when another user uses the same work area.
10. The system will then turn off any lights or devices after a certain amount of inactivity. - 2
    1. After the inactivity period, the system will turn of any active devices or equipment.

# List of non-functional requirements

Reference to your final high-level functional requirements, modify based on Milestone 1. Add or change as you see necessary.

Product requirements:

1.) The access control device needs to be simple to use and most users should be able to intuitively figure out how to operate it.

2.) The access control device should activate and deactivate the assigned equipment within 15 seconds of being requested.

3.) The access control device should resume its previous state when recovering from a power failure.

Organizational requirements:

4.) Users will be required to use their z number to access the system.

External requirements:

5.) The access control device should be able to operate across multiple browsers.

6.) Users should not have access to other users information except for staff for privacy concerns.

# High-level system architecture and database organization

Modify M1 accordingly, and add the following:

1. High level Architecture of the code must be consistent with UML class diagram (see below).
2. DB organization: Describe the main database schema/organization (high level), e.g. list main DB tables and items in each DB table
3. Media storage: Decide if images and video/audio will be kept in file systems or in DB. Describe any other special data format requirements like for video/audio/GPS etc.
4. Search/filter architecture and implementation: what will be the algorithm for search; what DB terms will be searched, how it will be coded and organized in the DB. Similarly, say what DB items will be filtered/sorted
5. Your own APIs: Describe and define at high level any major APIs that you will create
6. Describe any significant non-trivial algorithm or process (like rating, ranking, automatic prioritizing of items etc.)

For this project we will be using HTML5 and JavaScript for the front end of the website. The backend will be a web socket server and a SQL database for the users who are allowed access to the system. The tools we will be using are node.js which has a MIT license and uWebsockets which has a zlib license. The sonoff switches that will be used in the labs have an API to control them which is given in the following link: <https://blog.ipsumdomus.com/sonoff-switch-complete-hack-without-firmware-upgrade-1b2d6632c01>. We will have 2 databases, one is for authorized users and the other is for the devices being controlled. In the user database we will have the users name, Z#, class crn, email, flag for unclean workbench, flags for what bench they are authorized to use, flag for admin, and password for admin. This DB will be sorted and searched by Z#. The device database will have station type, station number, flag for clean, status, time activated, room number, and picture of last state. This DB is going to be sort by room number, then station type for simplicity.

# High-Level UML diagrams

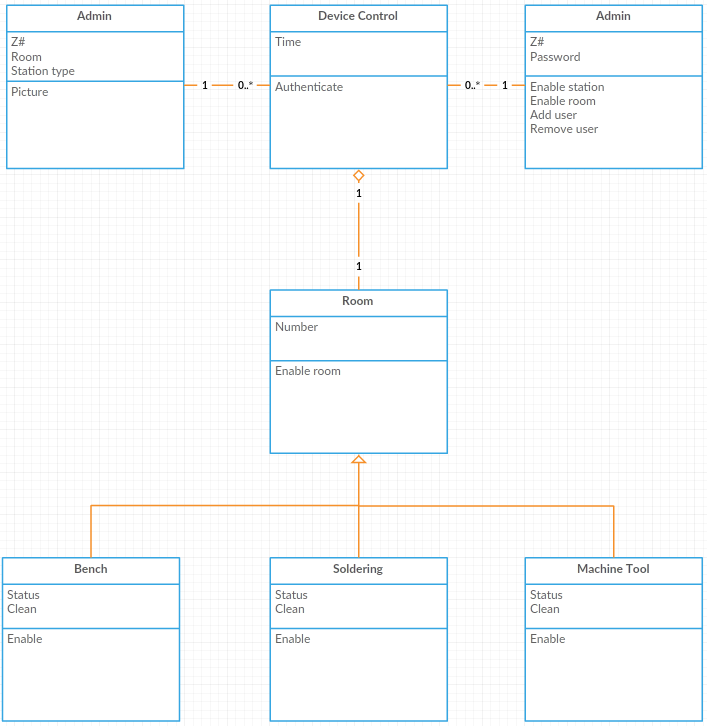
Familiarize yourself with Unified Modeling Language (UML). Find your favorite UML tutorials from the Internet. One good one is <http://edn.embarcadero.com/article/31863>

At minimum provide:

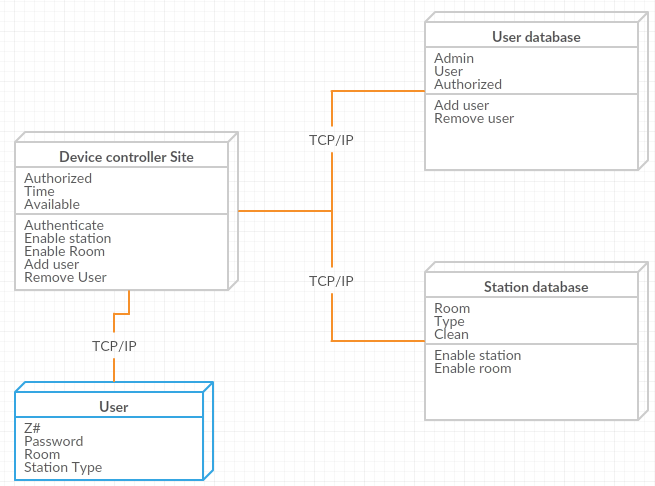
1. High-level UML class diagrams for implementation classes of core functionality, i.e. functionality with provided interfaces. Focus on main high-level classes only (one or at most two levels deep). This must reflect an OO approach to implementing your site.
2. UML Component and deployment diagrams

Use data terms and names consistently with Glossary/Data Dictionary.

UML class diagram:



UML component and deployment diagram



# Identify actual key risks for your project at this time

Identify only actual and specific risks in your current work such as (list those that apply:

1. Skills risks (do you have the right skills),

* Some of the risks involved are the use of HTML5, Java, node.js and SQL languages which will be required to implement this project. Some of us might not have enough knowledge and skills in using these languages.
* By communicating with team members and finding out who has more experience and who is more familiar with these types of languages and by assigning tasks to those individuals that are more proficient in certain languages listed above, will eliminate some if not all those risks listed.

1. Schedule risks (can you make it given what you committed and the resources),

* I believe the project can be implemented to some degree based on the information and resources we have (teacher, TA, teamwork).
* Since there is a short-limited time in completing this task, I believe this might contribute to not having it completely functional by the given deadline.
* Due to the time constraints, all team members will have to devote extra time and effort in completing this project.

1. Technical risks (any technical unknowns to solve),

* Other than the ones described above in section1, I cannot think of any technical unknowns to solve.

1. Teamwork risks (any issues related to teamwork);

* Lack of effort from certain team members can and will contribute to not completing the project successfully.
* This could be addressed by speaking directly to the individual involved or notifying the teacher that there is lack of effort from certain individual.

1. Legal/content risks (can you obtain content/SW you need legally with proper licensing, copyright).

* I would not think this would involve certain legal content, or proper licensing regarding the SW being used for our school project.

Tell us how do you plan to resolve risks? The key is to resolve risks as soon as possible. Categorizing risk as above helps a lot in managing them. Be brief: identify the risk and explain (2-3 lines), list how will you address these issues (2-3 lines)

1. **History Table**

3/26/18

* Added the ability for users to scan their owl card via a barcode application
* Added an admin user
  + Can add users
  + Can remove users
  + Can restrict access for users
  + Can change lab settings

# Submission

Store the modified Milestone 3 in your GitHub repo.

Each team submits one single word document with all the above required sections to Canvas by the due date. Must have a title page to your document.

# Grading criteria

Your document needs to be well-written, well-organized (formatted) and reads well. Grading is based on cohesiveness and completeness.

1. Executive Summary 10 points
2. Competitive analysis 10 points
3. Data definition 10 points
4. Overview, scenarios and use cases 10 points
5. High-level functional requirements 10 points
6. List of non-functional requirements 10 points
7. High-level system architecture (UML) 10 points
8. Identify risk and actions 10 points
9. Working with GitHub 10 points
10. Vertical demo 10 points