AMERICAN SOCIETY FOR ENGINEERING EDUCATION

Paper ID #35043

Time-Based Door Access Control System Capstone Senior Design Project

Drew Battison, Ohio Northern University

I am a Computer Science senior at Ohio Northern University, creating a prototype time-based door access control system as a Capstone Senior Design Project. During my team's Capstone Project, I have been the active leader. I also am in charge of the database portion of the project, along with integrating it into the created web pages and hardware code.

Adam Cook, Ohio Northern University

Senior computer science major from Ohio Northern

Mr. Mason Williams, Ohio Northern University

I am a senior at Ohio Northern University majoring in Computer Science.

Creating a Time-Based Door Access System:
A Capstone Senior Design Project

Abstract

The goal of this project is to create a competitive alternative to the current door access system utilized by the Ohio Northern Engineering building. Our team is creating a time-based door access system and reservation software to enable authorized users to reserve rooms. This system will allow for more privacy within classrooms and conference rooms because authorized users can no longer enter occupied rooms that they should not access(at a specific time). Key competitors were identified and researched in order to improve upon their designs and introduce time-based granular permissions. We also identified the project's stakeholders: Ohio Northern University and their public safety, information technology, and engineering buildings, allowing for alternative viable solutions to be drafted using constraints provided by our client.

Our first viable solution consists of integrating a reservation system with the Pro-Watch system installed at Ohio Northern University, abbreviated as ONU, the second consists of creating our own software and using similar hardware to what is installed, and the third consists of creating our own software and purchasing cheaper alternatives to create a prototype using Raspberry Pis. After applying evaluation metrics, we decided to continue our project using viable solution 3. To create this viable solution, hardware components were purchased and integrated with a server and database. The hardware includes Raspberry Pis, identification cards, RFID readers, door locks, and power supplies. The software is being completed by our team using a LAMP Stack(Linux, Apache, MySql, PHP), including the server, database, and webpages. The hardware and software will then be linked over a local network, creating a prototype time-based door access system for Ohio Northern University's Engineering Building.

1. Introduction

As a capstone senior design project, our team was tasked with improving Ohio Northern University’s door access system by adding time-based functionality. The access control system is intended to support the same functionality as the currently installed access system; additionally, the access control system will be limited by the time of day and allow for reservations. A reservation acts as a higher-level authorization for rooms, meaning it has a higher priority than “all or nothing” permissions, though reservations can only be made when authorized permissions exist. When a room is not reserved, any authorized persons that normally have access can still enter a room. The potential customers for our time-based door access system are organizations and businesses that would like a door access system that allows reservations.

2. Background

To create our own time-based door access system, our team chose to research competitive products to get an idea of what is expected and determine improvements that could be made. The competitors for our project were companies that sell their own door access control systems. The competitors that we researched were: Kisi [1], Openpath [2], Dicsan Technology [3], and Honeywell [4]. Our main competitor was Honeywell because their Pro-Watch integrated management software and hardware are what is currently installed at ONU.

We also researched applicable codes, standards, and regulations for implementing a door access system. Our project must comply with the codes, standards, and regulations for doors with access control [5] and door locking requirements [6]. These codes, standards, and regulations are necessary requirements for the installation and building safety of hardware, though our project consists of making a prototype rather than a complete installation. The applicable codes are the International Building Code (IBC), National Fire Protection Association (NFPA), Life Safety Code (LSC), and the Underwriter Laboratories Certification for access control standard requirements. Otherwise, all ACM, IEEE, and applicable ethics codes are followed.

Lastly, we researched the major components that make up typical control access systems, leading our team to the creation of viable design solutions. A door access control system consists of software including at minimum: a database and management software, a controller for communication between hardware components, RFID wireless scanners used to read the information supplied by identification RFID credential cards, electronic locks, and the power supplies necessary for the components. With this information, our team was able to consider different alternatives to design our project.

3. Possible Viable Solutions

The last step before viable solutions can be created is understanding the constraints, evaluation metrics, and ABET-Specified considerations that should be applied for our project. The constraints determine the aspects that must be satisfied for a solution to be considered "viable". The constraints that were considered for the design of this project are safety, the allocated budget of $400, the reliability such as functionality and availability, and the scalability of the system. Safety constraints can be satisfied by following the codes, standards, and regulations. Cost constraints are satisfied by staying under and minimizing the budget. The reliability constraints cause the design to emphasize performance and availability. Availability constraints impact the design's ability to be applied to larger situations such as adding additional hardware components.

The evaluation metrics are intended to choose which viable solution should be chosen using comparison. The metrics chosen are the cost, scalability, implementation and testing, and system integration. Cost is measured by the initial and upkeep costs for each solution. Scalability is measured by the ease of adding additional components. Implementation and testing are impacted by the limitation of hardware and software, the ease of testing, and the ease of making changes at the client's discretion. System integration is measured by the compatibility between the hardware and software and can be quantified by the difficulties of creating a system framework. Lastly, weights are also assigned to each metric depending on our client's needs.

Our team proposed a total of three viable solutions based on constraints and ABET-considerations. Simple diagrams are provided for the viable solutions under the figures heading. The first viable solution consists of expanding upon Honeywell's Pro-Watch access control system currently installed at ONU. The same hardware, software, time-based functionality, reservation, and scheduling software would be used and integrated. This solution benefits from the availability of the existing hardware and software. However, the ability to alter the software is limited by the documentation and licensing making it difficult to add functionality. The sketch for Viable Solution 1 is shown in Figure 1.

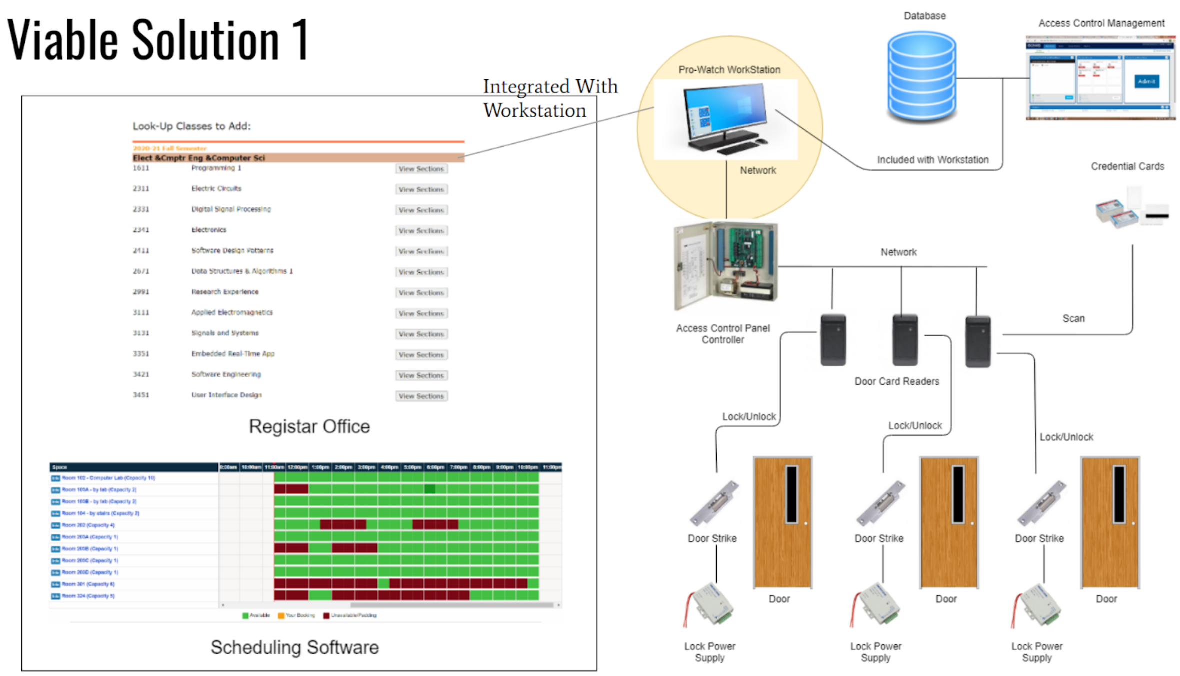

Figure 1: Viable Solution 1 Sketch

The second viable solution is to use the professional hardware installed at ONU and create all necessary software. This solution allows the software to be created specified by the client's needs and does not require the purchasing of hardware. The biggest limitation of this solution is the lack of spare hardware components for testing. Without using hardware already installed, testing cannot be performed and the professional components are expensive. Figure 2 shows viable solution 2 where the database and control management software are added to the project's workstation.

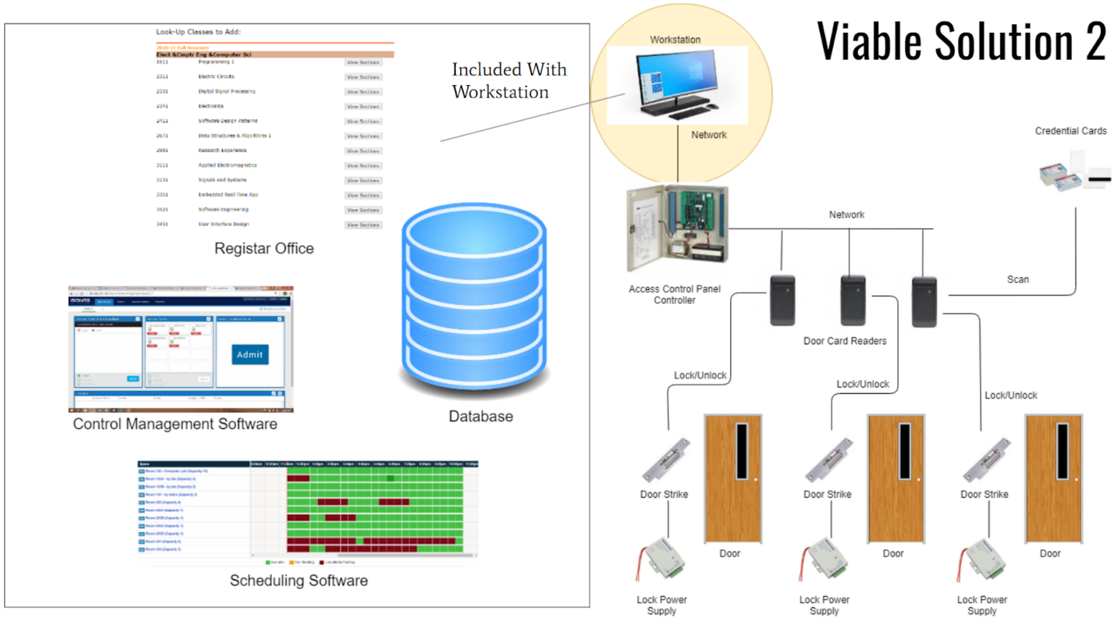

Figure 2: Viable Solution 2 Sketch

The third solution consists of creating the software and using cheap alternatives for hardware to make a prototype. Because the goal of the capstone project is to create a proof-of-concept, the hardware can be scaled-down and simple parts can be used. The benefits of this solution is the software can be specified to the client's needs and the hardware is cheap. The biggest limitation would be the transition from a prototype to a real installation, though our capstone project is not planned for installation. Figure 3 shows an altered version of the previous sketch by now choosing to focus on both the workstation and controller.

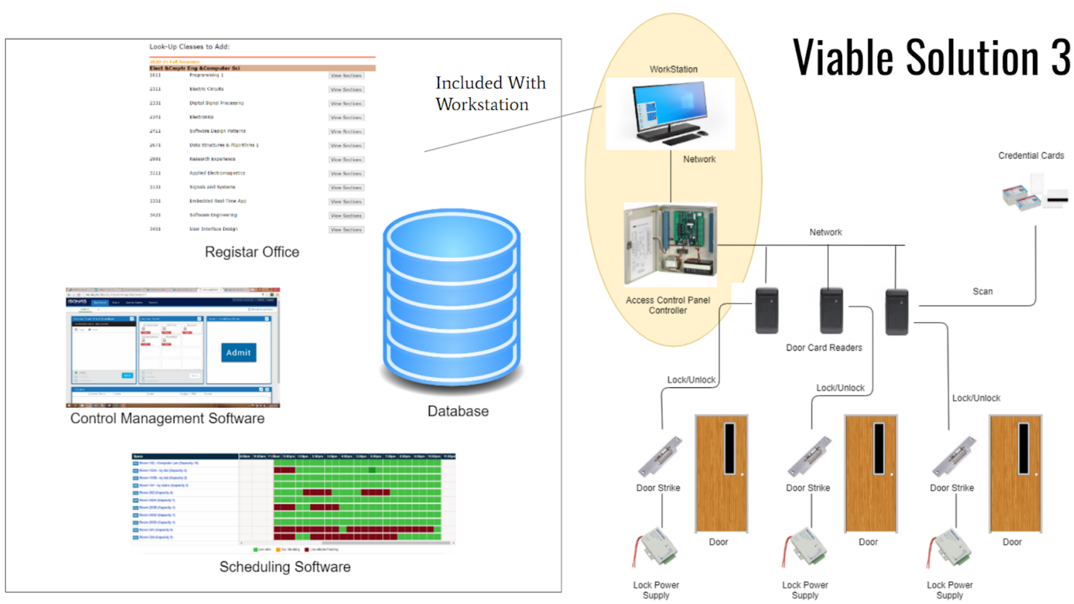

Figure 3: Viable Solution 3 Sketch (Chosen Solution)

Now that alternatives have been drafted, the evaluation metrics were applied to determine which solution should be built. The applied Evaluation Metrics can be seen in Figure 4.

Evaluation Metrics:	Weight	Viable Solution 1	Viable Solution 2	Viable Solution 3
Implementation and Testing	9	4	8	8
System Integration	8	3	10	7
Cost	7	2	5	10
Scalability	4	9	6	3
	Total:	110	211	210

Figure 4: Applying Evaluation Metrics

The weights for the evaluation metrics were determined by the priority of the client's needs and it is assumed a higher weight is the best option. Implementation and Testing are deemed the most important metric because the ability to design our project to our client's needs and create feasible prototypes for testing is the goal of the capstone project. System Integration is the next important metric because it places limitations on how we can design the framework and influences the software and tools available. Cost is also important because a prototype should not strain the budget and prototypes do not need to be fully implemented into a working environment and therefore should be less expensive. Scalability is deemed less important because we are focusing on a prototype rather than a full implementation. A proof of concept can always be expanded on in the future to meet the needs of a new client. Along with the weights, the assignments of metrics for each solution were justified.

Each solution is evaluated and justified based on the solution plan and the description of the metric. Brief justifications are provided to explain the result without going into too much detail. For implementation and testing, solutions 2 and 3 allow for the complete control of software whereas solution 1 is limited by documentation. For system integration, solution 1 already has the software and the existing system cannot be easily changed. Solution 2 used existing software and hardware and therefore has no compatibility issues. Solution 3 is limited by the compatibility of cheap hardware components. For the cost, the price is considered to utilize each alternative. Solution 1 is extremely expensive because a full system and software management system would need to be bought. Solution 2 requires professional level hardware while solution 3 can use chapter alternatives. Lastly, scalability is impacted by the software and hardware. Solution 1 has large scalability as it is the scale of the university. Solution 2 may be impacted by the software created and. Solution 3 is impacted by both the software and the cheap and limited hardware. After performing the evaluation metrics, we were left to consider solutions 2 and 3 as the results were close. Our team then discussed with our advisor and public safety, our client, and decided to continue with solution 3 because it was the cheaper alternative for prototyping.

4. Designing the Prototype

The first note before discussing the implementation of our project is that the capstone project is a work in progress. Our team is currently in the beginning stages of implementation and therefore have finalized plans but are still in the process of creating the database, web pages, and setting up hardware. With that in mind, the design and plan will be discussed for implementing our capstone project.

Our team decided that viable solution 3 was the best solution for a capstone project because of the low budget and small prototyping possible. For this implementation, we will be using Raspberry Pis for the main controller and microcontrollers. Our project will be built off Raspberry Pis using RFID readers, compatible identification tags, door locks, power supplies for those door locks, and breadboards with cables to connect our components together. The plan for combining the hardware is to have RFID readers, door locks, and power supplies connected to Raspberry Pis directly. These hardware components will be connected to each Raspberry Pi and each Pi will communicate with the others and the main controller over a network. The software will be integrated into the Pis where necessary and will be hosted on the main system controller using an Apache server. The software will be implemented using the LAMP Stack Framework [7] which uses Linux as the operating system, Apache to host the server, a database using MySQL, and the web pages running off PHP, HTML, and CSS. The software consists of a database and two web pages, one on the user side for reservations of rooms and the other for admins to access the control management software.

The web pages are split into two sections, the control management side intended for administrator usage and the reservation scheduling software intended for use by students and others who want to reserve meeting or conference rooms. Figure 5 shows a rough GUI of the administrator's control management software. This GUI is meant to allow the administrators to view all relevant information about any given user or room. The idea is that a user could be looked up to see the student's information and any rooms they have reserved. A room could also be viewed to see the reservations and who has a reservation for given times. Figure 6 shows the rough GUI for the reservation scheduling software. This GUI is intended to allow for users to reserve a room by simply matching the calendar to the room they wish to reserve. The search can also be narrowed down by searching for specific times, room types, or a building.

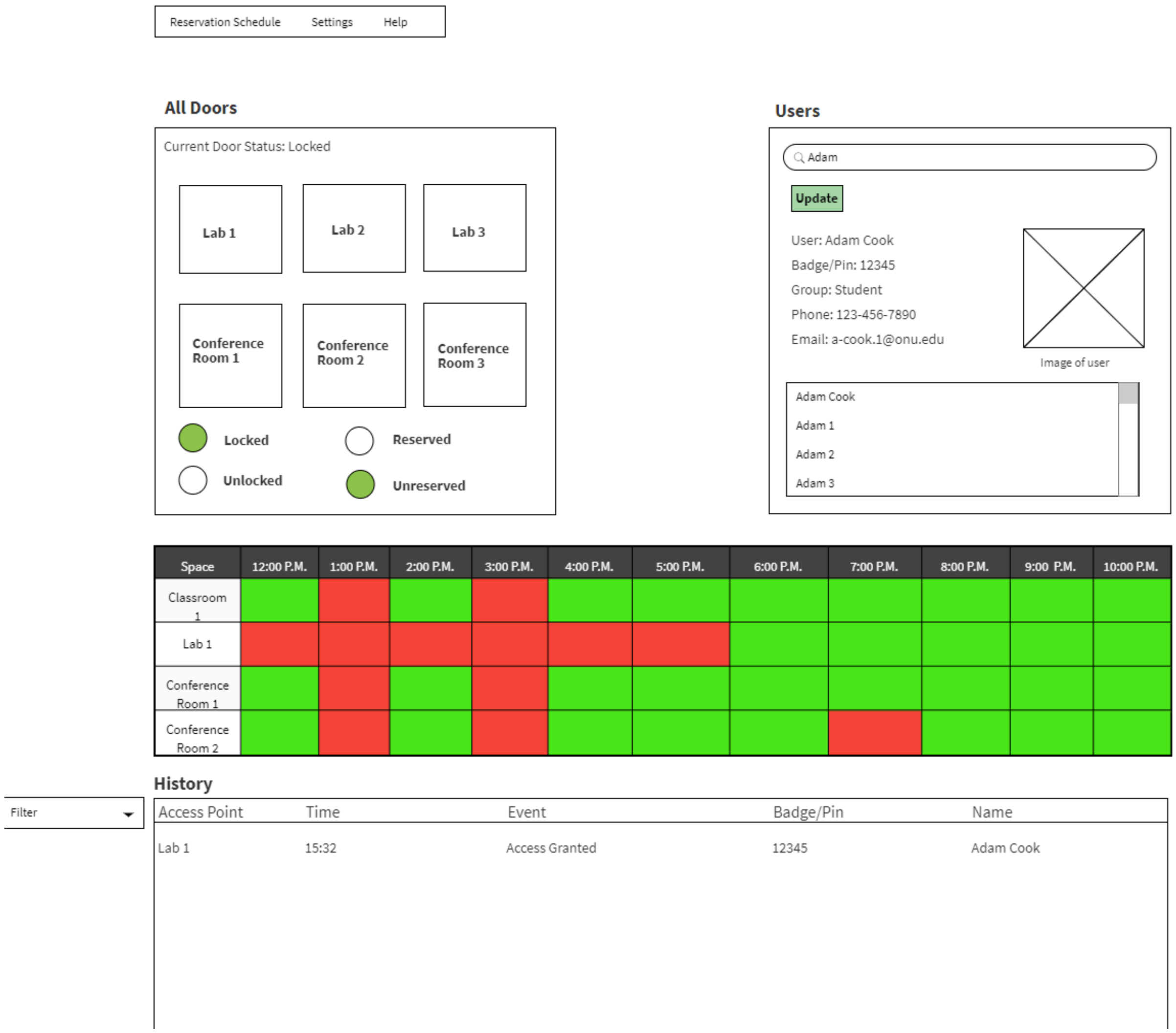

Figure 5 : Chosen Solution - GUI For Control Management Software

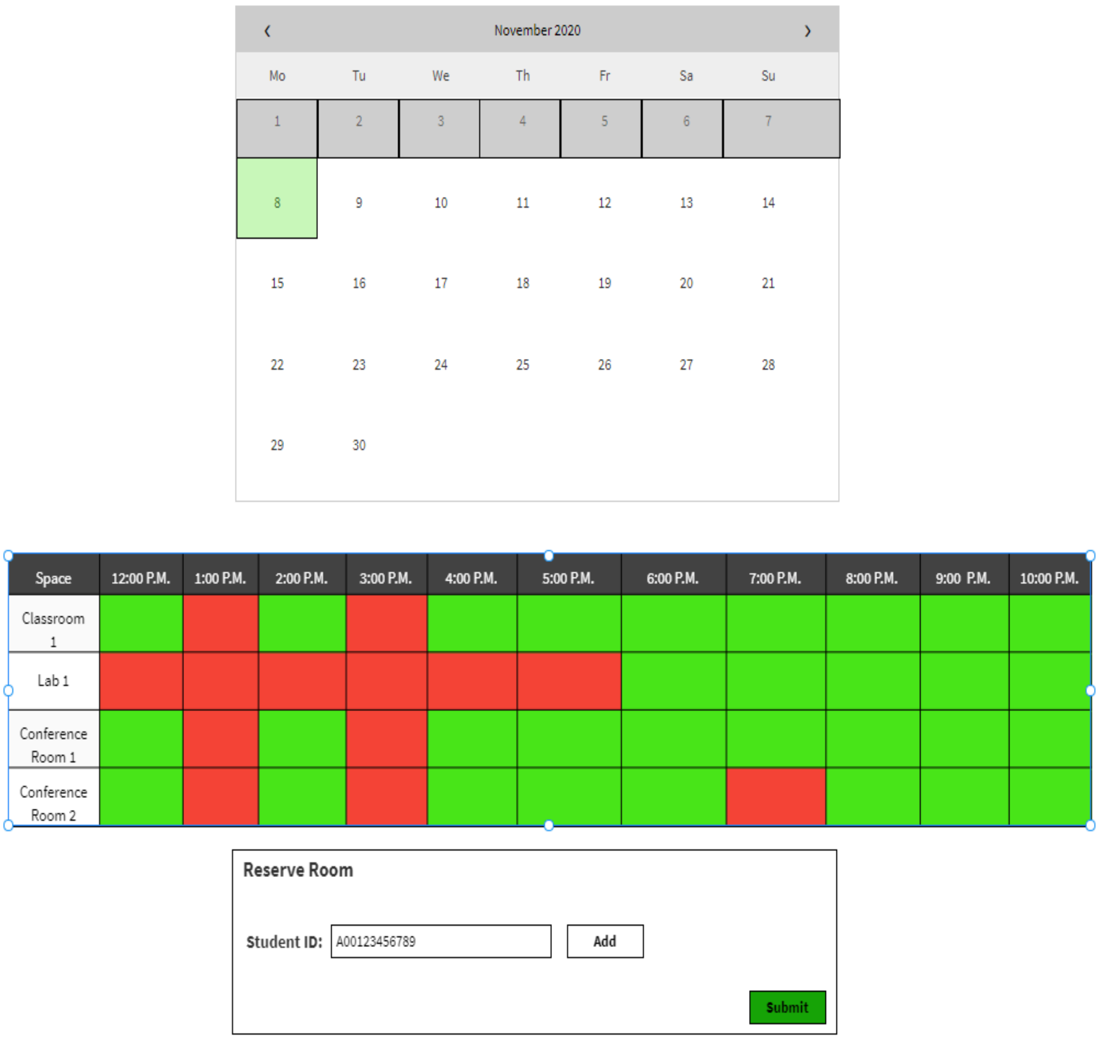

Figure 6 : Chosen Solution - GUI For Reservation Web Pages

The web pages will pull and push information to and from a database. The database stores all relevant information about students, rooms, reservations, permissions, and more. The database is shown in Figure 7 as an entity relationship diagram. The figure is meant to show the layout of how the database is organized, including the attributes and connections to be expected within the database.

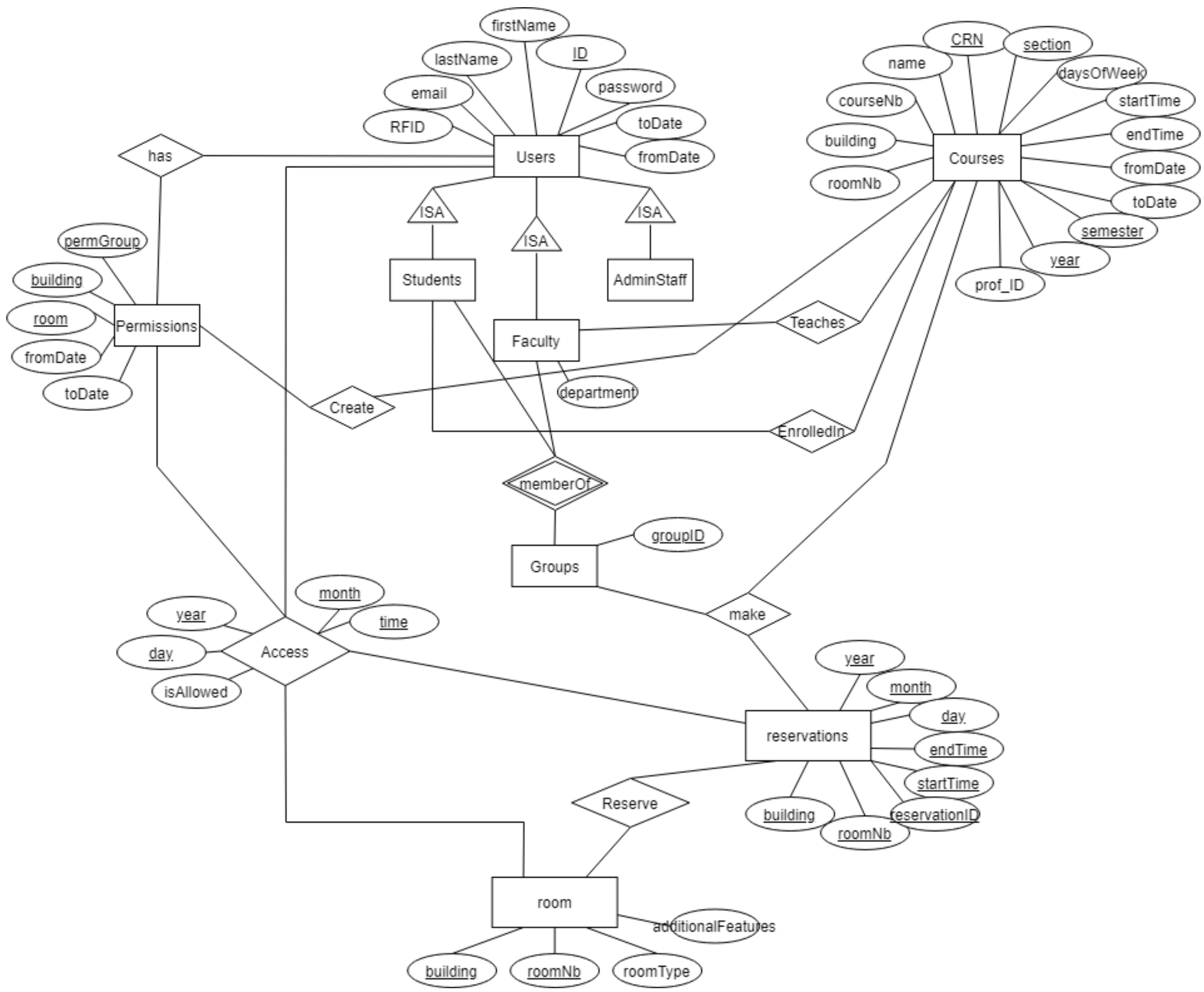

Figure 7: Chosen Solution - Database Entity Relationship Diagram

An example scenario to show how the project performs is as follows: A user will scan their RFID credential card on the wireless RFID reader hosted on a Raspberry Pi. The Raspberry Pi will then communicate with the main controller over the network in order to communicate with the server and access the database. The RFID information will be compared against the existing database and will check the authorization levels and reservations of the “room” trying to be entered. The main controller will then send the signal back to the Raspberry Pi which hosts the RFID reader being used and will signal for the door lock to be unlocked if authorized. Otherwise, a signal will be released to show permission was not authorized.

5. Lessons Learned

Throughout this project, our group has gained a deeper understanding of how control access systems work. We have learned more about what is actually happening when we swipe our student ids on the RFID readers attached to classrooms. We have learned how our university stores our information, including the authorization levels for granting or denying access when we swipe our student id. Throughout the process of this project, we have gained the experience of working with a client and designing a product to their needs. We have also gained experience in pitching our ideas and designs to a panel and proposing the best solution after weighing constraints and evaluation metrics. We have also researched the compatibility of hardware components and are eager to continue integrating software during our project's implementation.

6. Conclusion

In conclusion, our team is creating a time-based door access system to improve upon the Honeywell access control system installed at Ohio Northern University. We decided upon a project solution after considering evaluation metrics and constraints. Our project will allow our team to create the software and use cheap hardware to show proof-of-concept through a prototype. We will be running a LAMP stack on a main Raspberry Pi controller which consists of an apache server, a MariaDB database, and control management and reservation web pages made using HTML, CSS, and PHP. This controller will then locally connect to other Raspberry Pis which will each act as "rooms" in our prototype system. These Raspberry Pis will have a RFID reader and door locks so that users may swipe their proximity credentials card and unlock the door based on permissions and active reservations.

Our Project is currently in the implementation phase. The plans for completing the Capstone Project consists of finalizing the database, web pages, and hardware components and then integrating them all together. Once the integration is successful, the system should function as a prototype time-based door access system. Reservations will be created using a web page and the access management software will allow admins to alter room and user information. The system will then function by swiping a credential card against an RFID reader which will then connect to the database over the server. This database will then compare the RFID information against the tables in the database including the users, permissions, rooms, and reservations. The access attempt will then be logged and will determine access based on the comparison between the existing information in the database and the newly added RFID information. Finally, the door will be opened if access should be allowed.

7. References

[1] "Understanding Access Control Systems," Kisi. [Online]. Available: https://www.getkisi.com/access-control. [Accessed: 18-Sep-2020]

[2] "The Ultimate Guide to Access Control Systems," Openpath. [Online]. Available: https://www.openpath.com/the-ultimate-guide-to-access-control-systems. [Accessed: 18-Sep-2020]

[3] "Access Control Systems," Dicsan Technology. [Online]. Available: https://dicsan.com/access_control/access_control_controller/. [Accessed: 18-Sep-2020]

[4] "Pro-Watch Integrated Security Suite: Integrated Security," Honeywell Commercial Security. [Online]. Available: https://www.security.honeywell.com/All-Categories/integrated-security/pro-watch-integrated-security-suite. [Accessed: 18-Sep-2020]

[5] L. Greene, "Code requirements for doors with access control," Construction Specifier, 01-Mar-2013. [Online]. Available:

https://www.constructionspecifier.com/code-requirements-for-doors-with-access-control/. [Accessed: 18-Sep-2020]

[6] L. Peake, “Complying with door locking requirements,” Health Facilities Management, 04-Apr-2018. [Online]. Available: https://www.hfmmagazine.com/articles/3302-complying-with-door-locking-requirements. [Accessed: 18-Sep-2020]

[7] R. Sharma, “Difference Between XAMPP vs WAMP vs LAMP vs MAMP,” InkHive, 13-Aug-2018. [Online]. Available: https://inkhive.com/2018/08/13/difference-xampp-vs-wamp-vs-lamp-vs-mamp/. [Accessed: 30-Oct-2020].