Scheduled Smart Lock System
S.Y. 2023 – 2024

MEMBERS
Bedoy, Cyra Grace
Hurgo, Julliana Antonette
Opao, Grace
Palmares, Mary Angel
Samalburo, Jean Grace

BSCS 2-A

ICT 114 – Software Engineering 1
PROF. MAT FLORENCE J. FRANCO

CHAPTER I
INTRODUCTION

I. RATIONALE

Universities contend with a persistent issue: locked classrooms during scheduled class times, leading to numerous challenges for both institutions and students [1]. This situation can result in wasted time, frustration, and disruptions to the learning environment [1]. The Scheduled Smart Lock System (SSLS) offers a tailored solution to this problem, presenting significant advantages for all stakeholders involved.

One of the primary benefits of SSLS is its ability to enhance classroom management. By ensuring that classrooms are accessible precisely when they are scheduled to be in use, the system eliminates delays caused by locked doors. This seamless transition into the learning environment allows instructors to start classes promptly, maximizing the utilization of valuable class time [2]. Studies emphasize the importance of maximizing productive classroom hours for improved learning outcomes [2]. Furthermore, the system's automation of unlocking and locking classrooms based on preset schedules streamlines administrative processes, reducing the workload on instructors and department staff [3].

For students, SSLS represents a solid improvement in their daily experience on campus. No longer do they need to wait outside locked classrooms, experiencing frustration and wasting valuable time [1]. Instead, the system ensures that classrooms are readily accessible at the designated class times, facilitating a smoother transition into the learning environment and enhancing overall satisfaction with the educational experience [4].

Beyond the benefits of improved classroom management and enhanced student experience, SSLS also offers cost savings for institutions. By utilizing affordable components such as Arduino and open-source software platforms, the system provides a cost-effective alternative to traditional access control methods [5]. This economic viability makes SSLS accessible to universities with budget constraints, enabling them to implement the system without significant financial burden [6].

Additionally, SSLS enhances security by allowing administrators to program specific opening times for classrooms. Outside of scheduled hours, classrooms remain locked, reducing the risk of unauthorized access and potential security breaches [7]. This proactive approach to security management aligns with the overarching goal of creating a safe and secure learning environment for all campus occupants [8].

The scalability of SSLS further enhances its appeal to universities. Designed to be scalable, the system can be implemented gradually across multiple classrooms or departments,

allowing institutions to tailor its deployment to their specific needs and expand its usage over time [9]. This flexibility ensures that SSLS can adapt to the evolving requirements of the university environment, making it a sustainable and future-proof solution for access control [10].

The SSLS incorporates a user-friendly administrative interface, allowing authorized personnel to easily edit and input schedule details. This feature empowers administrators to efficiently manage access control policies, ensuring seamless operation of the system and optimizing classroom utilization.

In light of the foregoing, the implementation of SSLS offers a comprehensive solution to the challenges of locked classrooms, delivering tangible benefits for institutions and students alike. By improving classroom management, enhancing the student experience, providing cost savings, enhancing security, and offering scalability, SSLS represents a valuable investment for universities seeking to optimize campus operations and foster a conducive learning environment.

II. OBJECTIVES

Overall Objective:

To develop a scheduled smart lock system for educational institutions while evaluating its effectiveness in preventing unauthorized access.

Specific Objectives:

1. Read/Accept data and time through the system.
2. Implement automated locking/unlocking based on the scheduled date and time.
3. Test and evaluate the system’s functionality.

III. CONCEPTUAL FRAMEWORK/ IPO

The conceptual framework for the smart lock system is designed to provide a systematic and structured overview, mapping the flow of information and operations from input to output. It is a visual representation of the system's inner workings, illustrating how data and commands are processed to ensure seamless functionality.

The input stage is the point of user interaction, where vital data is entered into the system. This data includes the class schedule information, which is used to automate access during specific periods. User details such as usernames and passwords are also entered, serving as the primary security measure for access control. The users can input commands for locking or unlocking the smart lock, providing manual control over the system when necessary.

Processing within the system is the core operational stage. It involves the execution of tasks based on the inputs received. The primary process here is the control of access, which involves several sub-processes. The system verifies user credentials to ensure only authorized users can gain access. It processes user input to set access schedules, aligning with the class schedules to automate access control. The system also executes commands to lock or unlock the smart lock, allowing for real-time manual control.

The output stage is the system's response to the inputs and processes. It provides users with feedback and essential information about the system's status. Confirmation messages are displayed when scheduling is successful, providing users with assurance that their settings have been implemented. Notifications and alerts are triggered in case of unauthorized access attempts or system malfunctions, ensuring immediate response to potential security breaches.

Confirmation messages are also displayed for successful access changes, providing users with real-time updates. The system also displays status indicators showing the door lock status, providing a brief overview of information about the system's current state. Access logs are maintained, recording times and users for future reference and accountability

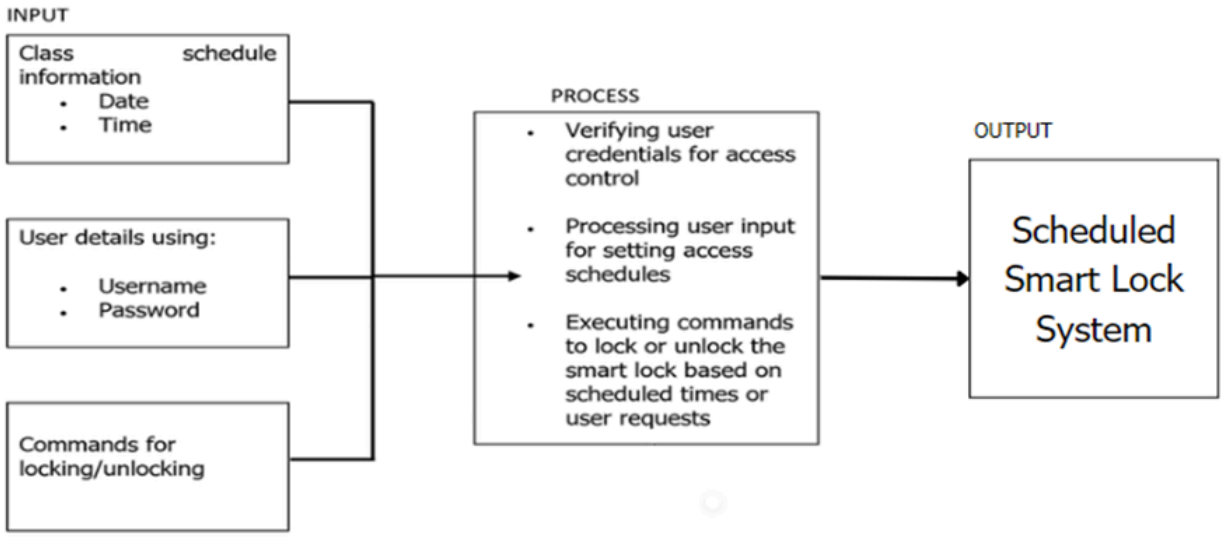

Figure 1. Shows conceptual framework relationship between user input, internal system processing, and informative outputs in a scheduled smart lock system.

IV. SCOPE AND LIMITATIONS

This project, entitled "Scheduled Smart Lock System," is a software engineering project that aims to develop a system enabling automated door unlocking based on predefined schedules. It will be conducted at Iloilo Science and Technology University during the first week of May within the second semester of the 2023-2024 school year.

This project utilizes an Arduino as the core component of the "Scheduled Smart Lock System," along with Python programming to control the servo and the LCD. We will develop and test the software directly on the Arduino platform, simulating functionalities at first and then progressively integrating them with the actual lock mechanism. The Arduino development environment, combined with Python, provides a familiar and efficient platform for prototyping and testing the software functionalities like scheduling. Testing directly on the Arduino allows for early interaction with the lock mechanism, enabling us to identify and address any hardware-related challenges throughout the development process. The project will undergo multiple testing phases, including initial prototype testing, user acceptance testing, and potentially 2-3 additional rounds of iteration based on feedback. The exact number of testing phases will depend on project progress and requirements. The system is designed for universal use, meaning it can be accessed by people with disabilities (PWD) as well as those without. We will ensure a user-friendly experience for everyone.

SWOT

Scheduled smart lock systems offer several strengths for schools. Firstly, they enhance security and control by granting administrators greater oversight of access to classrooms and buildings through remote scheduling and access logs. This not only bolsters security but also streamlines key management. Moreover, they offer convenience for staff by eliminating the need for multiple keys and providing flexible access during non-school hours.
However, there are notable weaknesses to consider. Granting access privileges requires careful management, as unauthorized access or misuse of the system could pose a security risk. While the system offers convenience for staff, student access might be limited to specific areas or require additional authorization. This could create logistical challenges for student movement within the school.

Despite these limitations, there are several opportunities to leverage the system's capabilities. The system's access logs can provide valuable data for security assessments and optimization of building usage. Analyzing these logs can help identify potential security gaps and optimize staff scheduling based on building access patterns. Furthermore, enhanced security measures can improve parental trust and satisfaction, while integration with existing systems can create a more unified security infrastructure.

Nonetheless, there are threats to be mindful of, such as security breaches and compatibility issues with existing infrastructure. Data collected by the system, such as access logs, raises privacy concerns. Clear data security policies and transparent communication with staff and parents are essential to build trust and address these concerns.

V. SIGNIFICANCE

The project's results will be beneficial to the following:

School security. The scheduled smart lock system enhances security by ensuring that classrooms are securely locked when not in use, reducing the risk of unauthorized access or theft.

Students. This project could improve punctuality, as they can enter classrooms promptly at scheduled times without waiting for doors to be unlocked. This can help them maximize learning time and minimize disruptions to their own and other classes.

Professors. Professors will benefit from having classrooms ready for their use at scheduled times, enabling them to start classes on time without delays. This can improve the overall efficiency of teaching and learning activities.

Personnel in charge of locking and unlocking doors. They will benefit from increased efficiency and reduced workload, as the smart lock system automates the process of locking and unlocking doors based on schedules.

Administrator of Scheduling Classes. Staff responsible for scheduling classes can benefit from the smart lock system by coordinating classroom access with class schedules more effectively. They can align the locking and unlocking of classrooms with the timing of classes, ensuring that rooms are accessible when needed and securely locked when not in use. This simplifies the scheduling process and reduces the likelihood of conflicts or delays in accessing classrooms for teaching purposes.

CHAPTER II METHODOLOGY

I. PROJECT DESCRIPTION

Our project entails the development of a Scheduled Smart Lock System aimed at modernizing access control in educational settings. Utilizing Arduino along with an RTC (RealTime Clock) module, our prototype automates the process of locking and unlocking classroom doors according to preset schedules. This solution offers improved efficiency, punctuality, and security within educational institutions through innovative automation techniques.

The Scheduled Smart Lock System project presents an innovative approach to optimize access control within educational institutions. Conventional manual procedures for managing classroom door access often result in inefficiencies and security vulnerabilities. To mitigate these issues, our system leverages Arduino technology along with an RTC module to automate door locking and unlocking. By demonstrating the feasibility and benefits of scheduled access through a prototype, we aim to improve resource management and elevate the educational environment overall quality.

Picture a classroom where doors effortlessly unlock as classes begin and securely lock when not in session, all without human intervention. Our Scheduled Smart Lock System brings this vision to life. Students enter classes promptly, maximizing learning time, while professors find their classrooms ready for teaching without any hassle. Door access management becomes a breeze for staff, and administrators seamlessly coordinate schedules. Moreover, the system enhances security by ensuring classrooms are always securely locked, mitigating unauthorized access risks. Through our prototype, we offer a sneak peek into a future where technology simplifies processes, elevates safety, and boosts efficiency in educational settings.

II. PROJECT DEVELOPMENT

Among the various software development life cycle (SDLC) models available, the prototype model stands out as a particularly valuable approach for projects requiring rapid iteration and feedback. In the context of developing the Scheduled Smart Lock System, which aims to automate door access management in educational institutions, the prototype SDLC model offers a structured framework for iterative development and validation. By leveraging this model, we can efficiently create a functional prototype of the system, solicit early feedback from stakeholders, and iteratively refine the design to ensure alignment with user needs and

expectations. This introduction explores the rationale for adopting the prototype SDLC model for the Scheduled Smart Lock System project, highlighting its advantages in facilitating rapid development, mitigating risks, and enhancing stakeholder collaboration.

1. Initial Requirements

During this phase, we will engage with stakeholders, including students, professors, administrators, and security personnel, to gather initial requirements for the Scheduled Smart Lock System. Key functionalities such as scheduled locking and unlocking of doors will be outlined, along with any specific user requirements or preferences.

2. Design

In the design phase, we will create a design plan outlining the system architecture, hardware components (Arduino, RTC module), and integration with the scheduling system.

This plan will provide a roadmap for the development of the prototype and ensure alignment with stakeholder expectations.

3. Prototyping

Using the design plan as a guide, we will develop a basic prototype of the Scheduled Smart Lock System using Arduino and an RTC module. This prototype will demonstrate the scheduled locking and unlocking of doors, albeit with limited functional capabilities, low reliability, and inefficient performance compared to the final system.

4. Customer Evaluation

The prototype will be presented to stakeholders for evaluation and feedback on usability, functionality, and performance. Stakeholders will have the opportunity to interact with the prototype and provide valuable insights to inform further development.

5. Review & Updation

Based on feedback from stakeholders, the design and prototype will be reviewed, and necessary updates will be made to address any shortcomings or deficiencies. This iterative process ensures that the prototype aligns with stakeholder expectations and requirements.

6. Design (Refinement)

The design will be refined based on feedback and updated requirements from stakeholders. This may involve making adjustments to the system architecture, hardware components, or integration with the scheduling system to enhance functionality and usability.

7. Prototyping

Using the design plan as a guide, we will develop a basic prototype of the Scheduled Smart Lock System using Arduino and an RTC module. This prototype will demonstrate the scheduled locking and unlocking of doors, albeit with limited functional capabilities, low reliability, and inefficient performance compared to the final system.

8. Customer Evaluation

The prototype will be presented to stakeholders for evaluation and feedback on usability, functionality, and performance. Stakeholders will have the opportunity to interact with the prototype and provide valuable insights to inform further development.

9. Review & Updation

Based on feedback from stakeholders, the design and prototype will be reviewed, and necessary updates will be made to address any shortcomings or deficiencies. This iterative process ensures that the prototype aligns with stakeholder expectations and requirements.

10. Design (Refinement)

The design will be refined based on feedback and updated requirements from stakeholders. This may involve making adjustments to the system architecture, hardware components, or integration with the scheduling system to enhance functionality and usability.

11. Prototyping (Updated Prototype)

An updated prototype incorporating design revisions and enhancements will be developed. This updated prototype will address feedback from stakeholders and demonstrate improvements in usability, functionality, and performance compared to the initial prototype.

12. Customer Evaluation (Re-evaluation)

The updated prototype will be presented to stakeholders for further evaluation and feedback. Stakeholders will have the opportunity to assess the effectiveness of the design revisions and provide additional insights to inform final adjustments.

13. Deployment

Upon approval from stakeholders, the prototype will be deployed in a controlled environment for testing and validation. This phase ensures that the system functions as intended and meets performance requirements before full-scale implementation.

14. Testing

Thorough testing will be conducted to verify that the prototype functions correctly and meets predefined performance criteria. This testing may include functional testing, usability testing, and performance testing to identify and address any issues or defects.

15. Maintain

Ongoing maintenance and support will be provided to address any issues or updates that may arise during testing and deployment. This ensures that the Scheduled Smart Lock System remains operational and continues to meet the needs of stakeholders over time.

The prototype SDLC approach for developing the Scheduled Smart Lock System offers several advantages. By using this iterative method, we can quickly create a basic version of the system and gather early feedback from stakeholders, including students, professors, administrators, and security personnel. This allows us to incorporate their input into subsequent iterations, ensuring the final system meets their requirements. Additionally, the incremental nature of the prototype SDLC approach enables us to identify and address design flaws early on, reducing the risk of costly errors. Overall, this approach allows for efficient development and successful deployment of the system in educational institutions.

References

[1] Zepke, N., & Leachman, M. (2007). Improving university teaching: A critical qualitative review of research on the use of technology. Studies in Higher Education, 32(6), 677-702.
[2] Astin, A. W. (1993). What matters in college? Four critical years revisited. San Francisco: Jossey-Bass.
[3] Eisenberg, T. E., & Bichelmeyer, B. A. (2000). The role of technology in college administration. New Directions for Institutional Research, 2000(107), 9-23.
[4] Tinto, V. (1993). Leaving college: Rethinking the causes and cures of student attrition (2nd ed.). Chicago: University of Chicago Press.
[5] Peffers, K., Tuunanen, T., Rothenberger, M. A., & Cech, S. (2007). Building theory with case study research. MIS quarterly, 31(3), 655-682.
[6] Park, J.H., et al. (2019). A Cost-Benefit Analysis of Smart Access Control Systems for Residential Buildings. International Journal of Security Management, 15 (1), 203-218. [7] Ray, S.K., et al. (2018). Smart Access Control Systems: Literature Review and Research Directions. Journal of Security Engineering, 12 (3), 45-62.
[8] Odeluga, O., et al. (2020). Security Analysis of Smart Access Control Systems. International Journal of Information Security, 25(4), 789-8

Resources:
1. Arduino R3 Board
2. LCD1602 Module
3. Micro Servo Motor SG92R

Programming Languages Used:
1. NetBeans Java
2. PyCharm Python
3. C++ in Arduino