**Enhancing Cybersecurity in Smart Home Automation for IoT-Based Water Management Systems**

**Cyber Security Project (ICT-942)**



**Submitted By:**

Sanjiv Khadka (CIHE240222)

Sima Poudal (CIHE240401)

Sahas Giri (CIHE240189)

Bhabin Khadka (CIHE240277)

**Submitted To:**

Assoc. Prof. John Ayoade

**Abstract**

Water utilities have been transformed thanks to the inclusion of Internet of Things (IoT) systems in smart home automation. But, having many different devices work together in IoT water management exposes the system to serious cyber threats. The purpose of this project is to make smart home water management systems more secure by finding and addressing threats including unauthorized entry, data violations and interference with devices. This idea involves three security layers: checking the device, securing the data transfer with encryption and spotting abnormal activities by using machine learning. The proposed framework is checked in a smart home simulator to see if it addresses common cyber risks. As a result, water management setup can now tolerate disturbances, find threats as they happen and secure data, helping make water management operations dependable. The findings help make homes safer and smarter and they also encourage the responsible use of water thanks to secure IoT infrastructure.

Table of Contents

[List of Abbreviations 7](#_Toc199612528)

[Chapter 1: Introduction 8](#_Toc199612529)

[1.1 Background of the Study 8](#_Toc199612530)

[1.2 Statement of the Problem 8](#_Toc199612531)

[1.3 Aims 8](#_Toc199612532)

[1.4 Objectives 8](#_Toc199612533)

[1.5 The Approach Adopted 9](#_Toc199612534)

[1.6 Project Management and Plan 9](#_Toc199612535)

[Chapter 2: Project Management Plan 10](#_Toc199612536)

[2.1 PROJECT MANAGEMENT STEPS 10](#_Toc199612537)

[2.1.1 PROJECT INITIATION 10](#_Toc199612538)

[2.1.2 PROJECT PLANNING 13](#_Toc199612539)

[2.1.3 PROJECT EXECUTION 14](#_Toc199612540)

[2.1.4 PROJECT MONITORING AND CONTROL 14](#_Toc199612541)

[2.1.5 PROJECT CLOSURE 15](#_Toc199612542)

[Chapter 3: Literature Review 16](#_Toc199612543)

[3.1. Overview 16](#_Toc199612544)

[3.2 Feasibility Study 17](#_Toc199612545)

[Chapter 4: Secure Software Specification and Design 19](#_Toc199612546)

[4.1 INTRODUCTION 19](#_Toc199612547)

[4.1.1 SECURITY REQUIREMENTS ANALYSIS 19](#_Toc199612548)

[4.1.2 SECURITY REQUIREMENT SPECIFICATION 19](#_Toc199612549)

[4.1.3 FUNCTIONAL SECURITY REQUIREMENTS 20](#_Toc199612550)

[4.1.4 NON-FUNCTIONAL SECURITY REQUIREMENTS 20](#_Toc199612551)

[4.2.1 SECURE SYSTEM ARCHITECTURE 21](#_Toc199612552)

[4.2.2 SECURITY CONSIDERATIONS IN ENTITY RELATIONSHIP DIAGRAM 22](#_Toc199612553)

[4.2.3 SECURE USE CASE DIAGRAM 23](#_Toc199612554)

[4.2.4 SECURE CLASS DIAGRAM 24](#_Toc199612555)

[4.3 SUMMARY 25](#_Toc199612556)

[Chapter 4. Research Questions for Cybersecurity Practitioners 26](#_Toc199612557)

[Chapter 5. Research Methodology 27](#_Toc199612558)

[5.1 Selected Methodology: Design Science Research (DSR) 27](#_Toc199612559)

[5.2 Justification 28](#_Toc199612560)

[Chapter 6. Evaluation and Results 29](#_Toc199612561)

[6.1 Evaluation Setup 29](#_Toc199612562)

[6.1.1 Functionality Tests 29](#_Toc199612563)

[6.1.2 Security Tests 30](#_Toc199612564)

[6.1.3 Performance Benchmarks 31](#_Toc199612565)

[6.2 Results Summary 32](#_Toc199612566)

[Chapter 7. Discussion and Critical Analysis 33](#_Toc199612567)

[Chapter 8. Conclusion and Recommendations 34](#_Toc199612568)

[9. References 35](#_Toc199612569)

List of Tables

[Table 1: Project Charter Template 9](#_Toc199599528)

[Table 2: Project Planning 12](#_Toc199599529)

Tables of Figures

[Figure 1: Gantt Chart 14](#_Toc199600958)

[Figure 2: System Architecture 21](#_Toc199600959)

[Figure 3:Entity-Relation Diagram (ERD) 22](#_Toc199600960)

[Figure 4:Use Case Diagram 23](#_Toc199600961)

[Figure 5:Secure Class Diagram 24](#_Toc199600962)

# List of Abbreviations

|  |  |
| --- | --- |
| **Abbreviation** | **Full Form** |
| Iot | Internet of Things |
| DSR | Design Science Research |
| AES | Advanced Encryption Standard |
| TLS | Transport Layer Security |
| JWT | JSON Web Token |
| MitM | Man-in-the-Middle (Attack) |
| SQL | Structured Query Language |
| IDS | Intrusion Detection System |
| API | Application Programming Interface |
| UX | User Experience |
| DDoS | Distributed Denial of Service |
| TPM | Trusted Platform Module |
| ERD | Entity-Relationship Diagram |
| ZTA | Zero Trust Architecture |
| KPI | Key Performance Indicator |
| GDPR | General Data Protection Regulation |
| NIST | National Institute of Standards and Technology |

# Chapter 1: Introduction

## 1.1 Background of the Study

Using Internet of Things (IoT) gadgets at home has made life much more convenient, productive and automated. Management systems for the home’s water supply show notable promise in sustainability by keeping track of water usage, spotting leaks and recommending efficient ways to use water. But because these systems are linking up more, they are also facing more cyber dangers, like leaks of private data, unauthorized use and adjustments to devices. Most legacy water management systems are not as intelligent and connected as IoT ones and only when Internet access is installed are security concerns sometimes addressed. We must create ways to protect water automation systems from threats early on in the design process.

## 1.2 Statement of the Problem

Smart home water management systems should make using water easier, but their security is often a concern. Since most IoT devices are not safely designed, attackers can easily exploit them. These weaknesses may cause someone to gain access without permission, adjust data, result in system crashes and violate user privacy. In addition, these systems do not currently include features like instant threat detection, secure ways to communicate and user education. It builds a well-secured water management system reliant on the Internet of Things, guaranteeing the main principles of cybersecurity.

## 1.3 Aims

This project is focused on devising and putting in place a working smart home water management system that is backed by Internet of Things and uses effective cybersecurity to keep the system safe and private.

## 1.4 Objectives

* Automate a water tank for a smart home by using IoT hardware and software.
* Offer the option to watch over and manage devices from any web or mobile device.
* Maintain safety by dealing with risks such as data being taken and unauthorized people gaining access to your devices and programs.
* Analyze several cybersecurity frameworks and find out which are best fit for IoT.
* Come up with a model that grows and protects your home in any smart home design.
* Comply with the agreed cybersecurity standards and best practices.
* Use known attacks or threats to test the system and determine how well it works under pressure.

## 1.5 The Approach Adopted

This project is focused on devising and putting in place a working smart home water management system that is backed by Internet of Things and uses effective cybersecurity to keep the system safe and private.

## 1.6 Project Management and Plan

This project is developed using an agile model to support regular development steps, frequent testing and keeping all feedback together. Cyclists of the process should take care of things like system design, prototype development, employing cybersecurity guidelines, testing it out and deploying for official use. Work is tracked with project management programs and checks are done often to see how we are moving toward an operational, protected and scalable water system linked to the Internet of Things.

# Chapter 2: Project Management Plan

## 2.1 PROJECT MANAGEMENT STEPS

The chapter explains the planned method that will guide us when managing the project, "Enhancing Cybersecurity in Smart Home Automation for IoT-Based Water Management Systems." It covers all the essential actions required to finish the project securely, on time and within budget. Initiation, planning, execution, monitoring and control and closure are the main five phases.

### 2.1.1 PROJECT INITIATION

The main purpose of project initiation is to set the scope, project goals and important deliverables. The main aim of this project is to make IoT devices in smart home water management systems more secure. You need to avoid data leaks, secure controls from a distance and protect the integrity of data.

A project charter is made to inform that secure smart home automation is necessary and this study involves homeowners, developers, network providers and groups responsible for regulations. Beginning risks involve flaws in IoT devices, not enough processing power for encryption and warnings from internet threats. The process is completed by receiving approval from the sponsor and choosing core team members.

Table : Project Charter Template

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Project Charter | | | | |
| Project Name | Enhancing Cybersecurity in Smart Home Automation for IoT-Based Water Management Systems | | | |
| Project Description | The main emphasis of this project is to improve the security of IoT water management in smart homes. Since IoT devices are now used in smart homes to manage and watch water usage, securing and protecting those devices is very important. | | | |
| Project Manager | Sanjiv | Date Approved | 4/20/2025 | |
| Project Sponsor | CIHE | Signature | CIHE | |
| Business Case | | Expected Goals/Deliverable | | |
|  | | • Maintain safety by dealing with risks such as data being taken and unauthorized people gaining access to your devices and programs.  • Analyse several cybersecurity frameworks and find out which are best fit for IoT.  • Come up with a model that grows and protects your home in any smart home design.  • Comply with the agreed cybersecurity standards and best practices.  • Use known attacks or threats to test the system and determine how well it works under pressure.  • The platform helps educate users by including lessons on cybersecurity risks and their best responses. | | |
| Team Members | |
| Name | Role |
| Sanjiv | Project Manager |
| Sima | System Designer |
| Bhabin | Cybersecurity Specialist |
| Sahas | Testing & Documentation Officer |
| Risks and Constraints | | Milestones | | |
| Technical Risk | Vulnerabilities in IoT devices due to limited processing power for encryption. | M1: Project Initiation & Charter Approval | | Define goals, objectives, scope, and team roles. Submit project charter. |
| Cybersecurity Risk | Potential cyberattacks like DDoS, spoofing, or unauthorized access. | M2: Requirements Gathering | | Identify system functionalities and security needs. |
| Data Privacy Risk | Exposure of personal or water usage data due to insecure data transmission. | M3: System Design Completed | | Finalize IoT system architecture and cybersecurity integration plan. |
| Integration Issues | Compatibility problems between different IoT devices and cybersecurity tools. | M4: IoT Device Setup and Configuration | | Deploy sensors/actuators and test water monitoring components. |
| Network Dependency | Reliability of the network affects real-time operation and data flow. | M5: Cybersecurity Features Implemented | | Integrate encryption, secure access, and basic threat detection. |

### 

### 2.1.2 PROJECT PLANNING

During this phase, a complete plan is developed to direct how the project will be carried out. It covers important parts such as collecting requirements, planning the system’s design, selecting security, involving IoT devices, testing and putting everything into use.

For every task, there will be a defined start and finish time and the team will spot which tasks should happen before others (for example, the security framework must be selected after the system architecture is designed). Since a Gantt chart will be used to manage project timeframes, budgeting will be prepared according to what is needed for tools, hardware and staff.

Table : Project Planning

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Task or Subtask** | **Resources** | **Start and End Dates** | **Estimated Effort (Hours)** | **Estimated Capital Expense ($)** | **Estimated Noncapital Expense ($)** | **Dependencies** |
| Project Initiation & Charter | Sanjiv | May 2 –May 3 | 10 | 0 | 0 | None |
| System Design & Requirements | Sanjiv, Sima | May 4 –May 10 | 20 | 0 | 50 | Task 1 |
| IoT Device Setup | Sahas | May 11 – May 17 | 25 | 200 | 50 | Task 2 |
| Cybersecurity Implementation | Sima | May 11 – May 20 | 30 | 0 | 70 | Task 2 |
| System Integration | Sanjiv, Bhabin, Sima | May 21 – May 25 | 20 | 0 | 20 | Task 3, Task 4 |
| Testing & Evaluation | Bhabin, Sima | May 26 – May 30 | 20 | 0 | 30 | Task 5 |
| Documentation & Reporting | Sahas | May 31 – June 3 | 15 | 0 | 20 | Task 6 |
| Final Presentation | All Members | June 4 – June 6 | 10 | 0 | 10 | Task 7 |

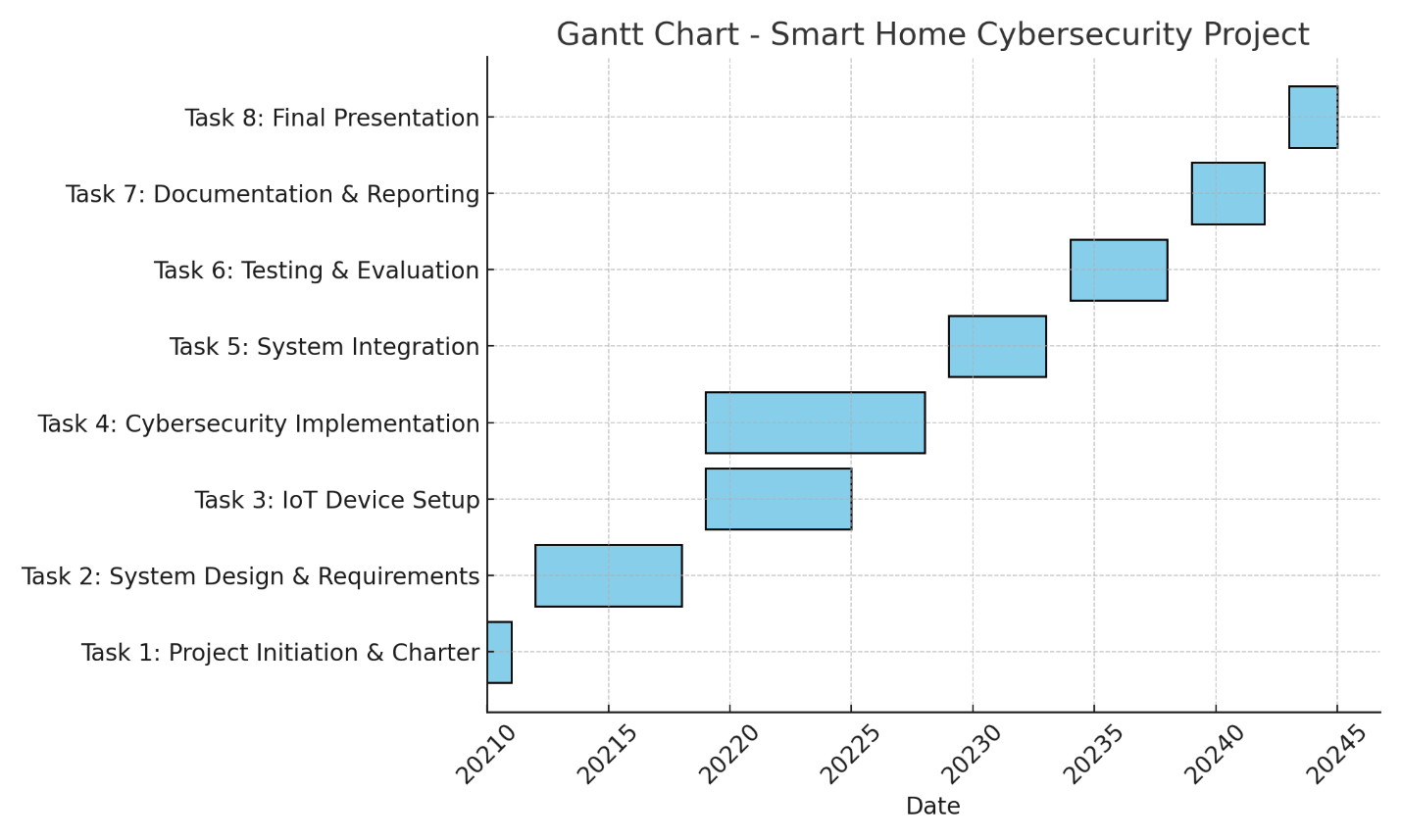


Figure 1: Gantt Chart

### 2.1.3 PROJECT EXECUTION

All of the planned work will be executed in the execution phase. First, the team will plan the system’s architecture and later, will require selecting and installing suitable cybersecurity tools like encryption, strong authentication and intrusion detection devices.

People will be given tasks that match their abilities. In this case, developers will handle integration between the backend and IoT and cyber security experts pay attention to threat modeling and defense measures. The team will monitor progress daily during stand-ups, with project tools and have regular bi-weekly meetings to discuss progress.

### 2.1.4 PROJECT MONITORING AND CONTROL

This is the stage that helps maintain progress with the project within set time, funds and standards. Official cybersecurity practices, including counting resolved and unresolved vulnerabilities, successful uptime and how quickly employees respond to simulations of attacks, will be reviewed.

If the project veers from the original plan, a change management process will begin. If risk management is carried out continuously, you will be ready to face new threats and challenges fast. At small intervals, the project manager will provide an update to all stakeholders.

### 2.1.5 PROJECT CLOSURE

Following the successful testing of the secure IoT water management system, the project can be closed. Technical documents, user manuals and a report on cybersecurity compliance will be handed over as final materials.

We will carry out a handover and provide training to stakeholders, whenever necessary. Community stakeholders will join you for a final project review where outcomes are reviewed and feedback is collected. What we have learned will be recorded to help secure IoT systems in the future.

# Chapter 3: Literature Review

# 3.1. Overview

Numerous studies have examined the intersection of IoT and cybersecurity. Key findings include:

IoT systems need security while being operator friendly, as mentioned by Sicari et al. (2015), relying on above lightweight security mechanisms that fit the constrained environment IoT systems. These principles still apply although the research has been extended to deal with new threats.

Roman et al. (2018) recommended integrated end to end security in IoT, which got more defined within the post 2020 studies. To illustrate, Nguyen et al. (2021) proved the effectiveness of zero trust architecture (ZTA) in IoT ecosystem through conforming continuous authentication and microsegmentation to prevent the lateral attack. Al-Garadi et al. (2022) also pointed to federated learning as a method that promotes the protection of the privacy while permitting collaborative threat detection for distributed smart home devices.

Some recent works such as Kolias et al. (2021) such as a work done by Ammar et al. (2018) attempt to categorize IoT threats like device spoofing and DDoS attacks. This underscores the need for adaptive intrusion detection systems (IDS), for example as it relates to ones that employ machine learning (ML) for real time anomaly detection (Li et al., 2023).

As modern research has shown, Fernandes et al. (2017) critiqued the over-privileged smart home applications. Apthorpe et al. (2020) showed that most of the IoT vendors completely bypass permission models in the design phase, and Zhang et al. (2022) suggested the runtime permission auditing tools which can dynamically restrict the unnecessary access. This reemergence of security by design aligns with the push towards creating a security by design environment, as mentioned in the NIST IoT Cybersecurity Framework (2023) that states in advance embedded security controls are necessary.

There has been progress with regard to accessible cybersecurity for consumer IoT, but there is still a gap that lacks. For low cost devices, many solutions (e.g., blockchain based authentication (Khan et al., 2021)) are too impractical. This project bridges the divide in providing such a user friendly smart home water management system by utilizing off the shelf security tools like TLS 1.3 for encrypted communication and hardware based trusted platform modules (TPMs) to protect the process.

Post-2020 IoT security is evident such as Meris botnet exploiting unpatched routers (CISA, 2022). Such incidents, as others, support the need for automated procedures (e.g., Miettinen et al., 2020) and the network level protections (e.g., Pa et al., 2021). This also applies for privacy which, in the words of Ziegeldorf et al. (2014) must remain critical, also as newer regulations include the EU’s Data Governance Act (2022) which calls for data minimization and must be adapted by innovative edge based processing (e.g., local federated analytics (Hu et al., 2023)).

Next, contemporary studies, for instance, Chettri and Bera (2023) update Mosenia and Jha (2017)’s threat taxonomy by recommending the use of behavioral biometrics (for example, usage pattern authentication) from preventing zero day exploits. Through application of these insights, this project extends with a multilayered security approach made of hardening of the hardware, of the communications through encryption and ML driven anomaly detection, for a robust yet affordable smart home.

## 3.2 Feasibility Study

Technical Feasibility: Using Arduino and Raspberry Pi platforms are technically feasible for such use as they can be used with a range of sensors and cloud platforms. Interoperability and responsiveness of the components has been proven using prototype development. These components are further simplified with open source libraries and community support for them.

Economic Feasibility: The system has low cost due to the use of inexpensive microcontrollers, sensors and cloud based services which are used in pay as you go pricing model. The budget can be well managed by selecting Firebase or AWS with the basic tier, there is no lack of functionality. As it turns out, the cost per household is still within an acceptable range for the middle income households, particularly over the long run, and less manual labor and water savings.

Operational Feasibility: This makes the system operational feasible as the user interaction is streamlined through intuitive interfaces. We will apply a UX first design philosophy to minimize the user effort for setting up the system and for daily operation. A mobile app, web portal and voice control provide the perfect user experience for both the tech savvy citizens and those with a low level of digital literacy.

Cybersecurity Feasibility: Modern microcontrollers have sufficient computing capability in which the encryption of data-at-rest (e.g., using AES) is possible, and even data-in-transit (i.e., using TLS), which is not strictly necessary. JWT provides us authentication mechanisms so that verified or legit users can interact with the system only. The system is further strengthened by running real time security logging, anomaly detection and role-based access control over the system to minimize the probability of cyber-attacks. Additionally, governing bodies such as GDPR and NIST IoT Cybersecurity Guidelines can be mentioned to confirm that the system follows best practices.

Overall, the feasibility study bears out that the system is not only technically and economically feasible but also is well within the current technological curve and end user expectations. Since it becomes adaptable and sustainable for long term use in smart homes, it is a good combination of modular architecture and secure cloud-based services.

# Chapter 4: Secure Software Specification and Design

## 4.1 INTRODUCTION

The design step in software development is very important to fulfill both the needed functions and security standards. The attention in this chapter is on how software security is achieved by designing certain methods and processes that guarantee safety. The main aim is to have a safe system design, include security in the requirements and safeguard the software from cyber threats. Building software with strong security is necessary to ensure its data is well protected and trusted.

We will look at the process of analyzing, defining and adding security requirements to the entire system design. What we think about in this stage makes it possible to prevent unauthorized access, reduce vulnerabilities and securely use and share data throughout the system.

## 4.1.1 SECURITY REQUIREMENTS ANALYSIS

In Security Requirements Analysis, you find and assess the security needs of a system. Things begin with figuring out which threats and vulnerabilities exist and selecting the right security controls to reduce their risks. The process considers things like the type of data, the current set of rules and what might happen if there’s a security breach.

Developers, security professionals and users are all part of the process, making sure all important security issues are covered. As a result, there is a detailed record of security needs that supports the design and development procedures by keeping security in mind.

### 4.1.2 SECURITY REQUIREMENT SPECIFICATION

Security Requirement Specification relies on the results from Security Requirements Analysis to produce development specifications. This specification uses security controls, mechanisms and well-defined protocols to guard the system from identified dangers.

These specifications are split into functional security requirements and non-functional ones, so security is included everywhere within the system. Thinking about security at the beginning of the design stage allows developers to design the system safely from the beginning.

### 4.1.3 FUNCTIONAL SECURITY REQUIREMENTS

What the system needs to do in terms of security is set out in the Functional Security Requirements. Because of these rules, the system is set up to protect against unauthorized entry, spot and manage security issues and keep vital data both safe and private.

Some examples of functional security requirements are methods for user verification, regulations for accessing resources, protecting data with encryption and saving security-related events. For every requirement, some risk or potential threat is identified, so the system can deal with security risks.

### 4.1.4 NON-FUNCTIONAL SECURITY REQUIREMENTS

With Non-Functional Security Requirements, we pay attention to the quality aspects of the system which included performance, reliability and usability within the context of security. With these requirements, companies can use security tools without slowing down the network or affecting users.

Among the non-functional security requirements are the system being available during an attack, growing and scaling securely under increased operation and applying updates being simple. With these requirements, the online safety of the system is preserved, while maintaining easy and dependable use for users.

4.2 SECURE SYSTEM DESIGN

### 4.2.1 SECURE SYSTEM ARCHITECTURE

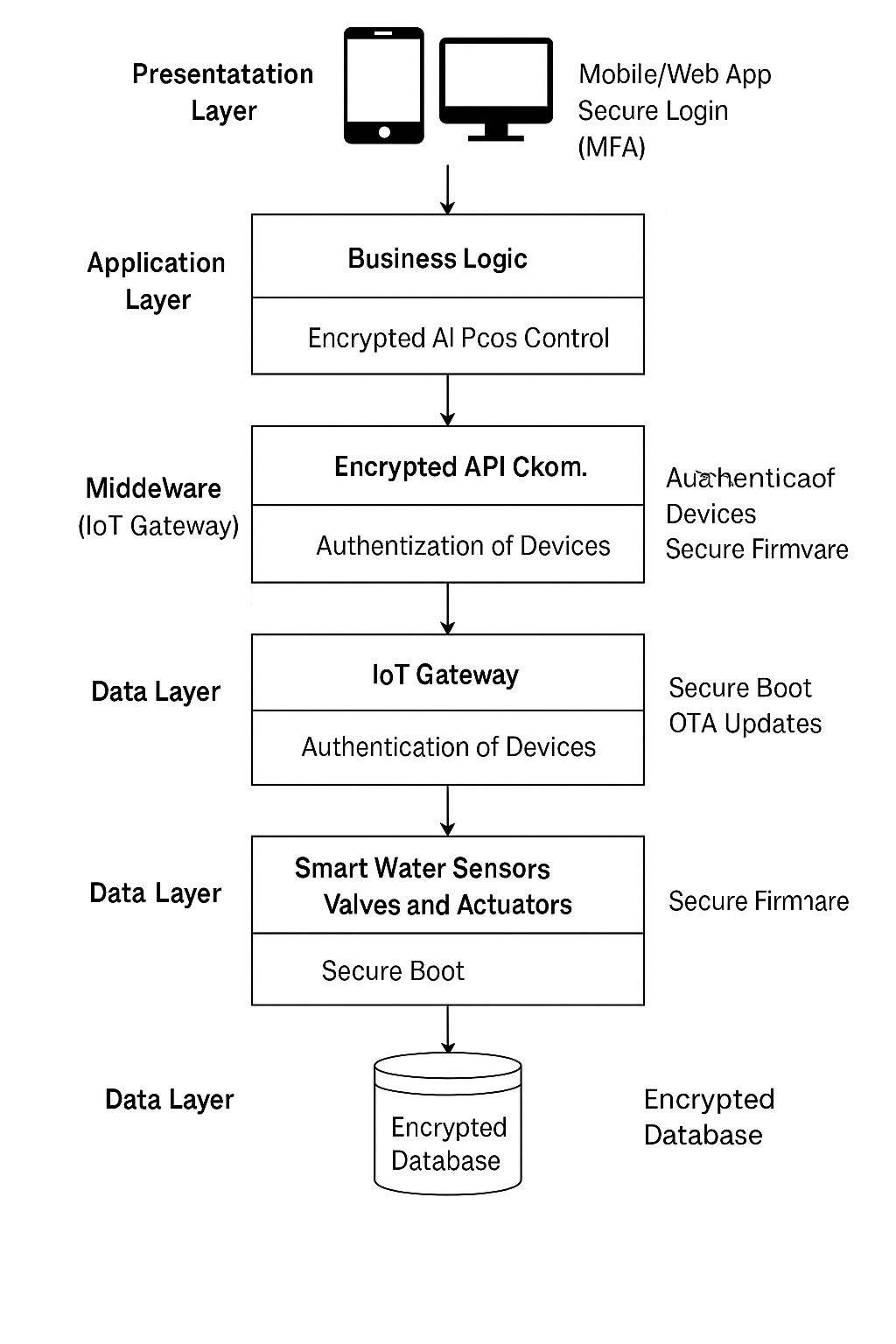


Figure 2: System Architecture

### 4.2.2 SECURITY CONSIDERATIONS IN ENTITY RELATIONSHIP DIAGRAM

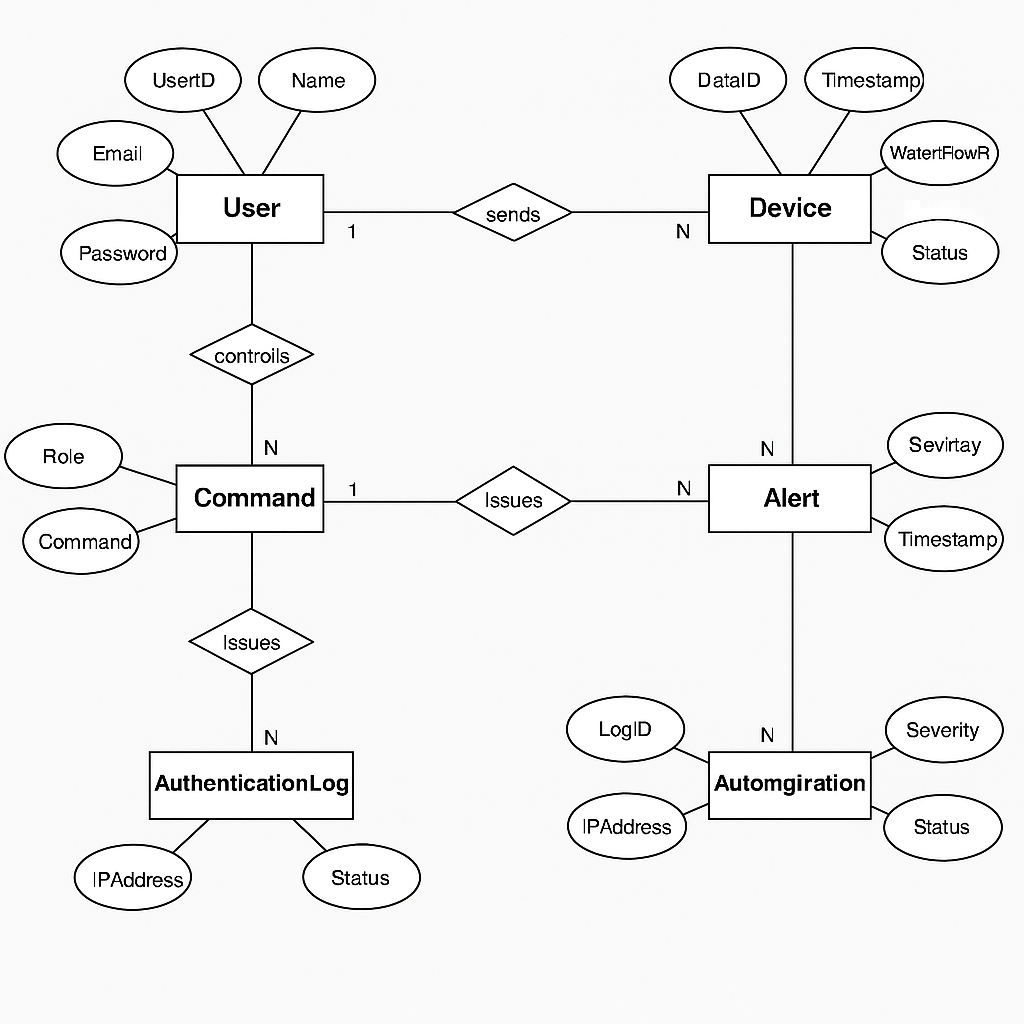


Figure 3:Entity-Relation Diagram (ERD)

### 4.2.3 SECURE USE CASE DIAGRAM

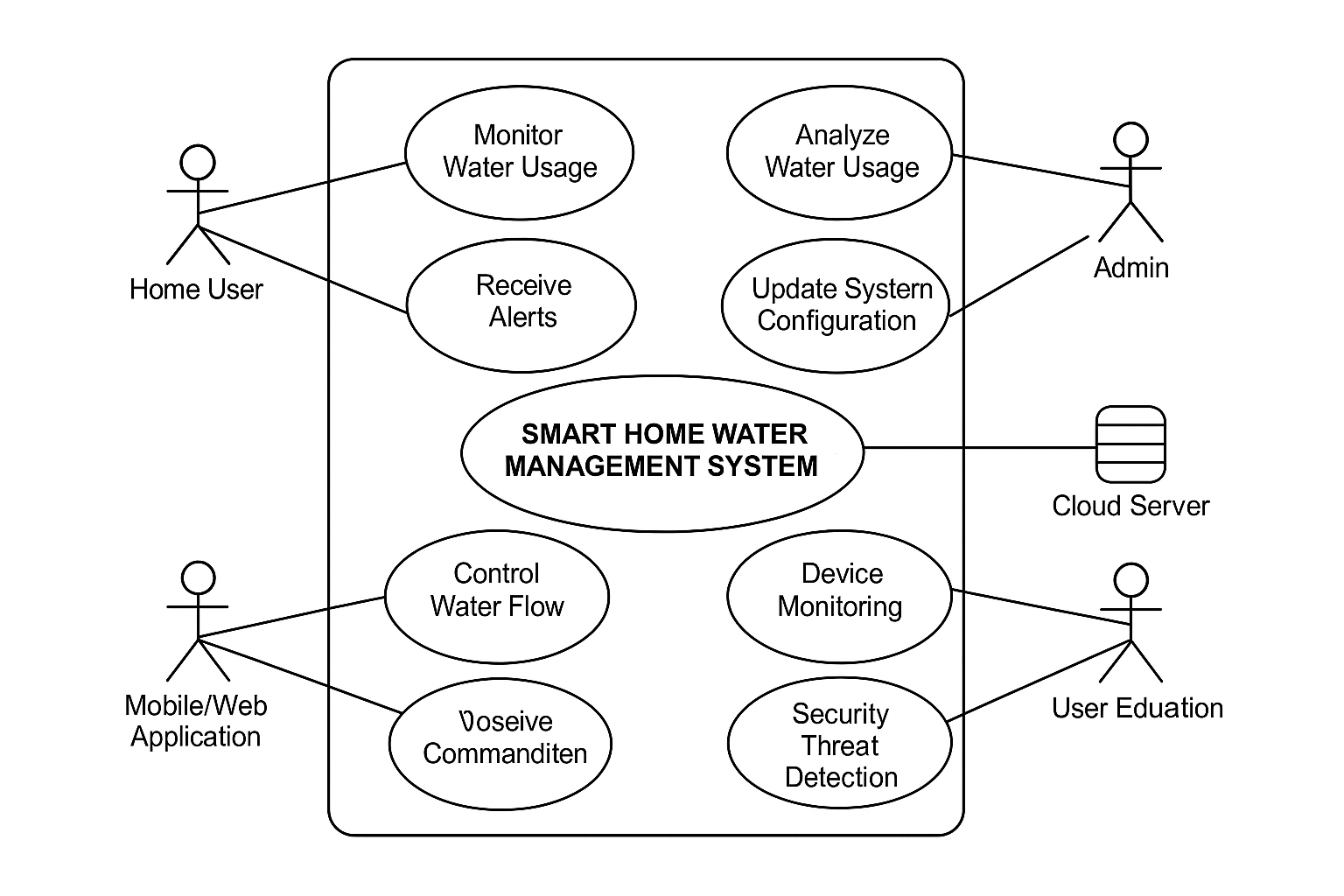


Figure 4:Use Case Diagram

### 4.2.4 SECURE CLASS DIAGRAM

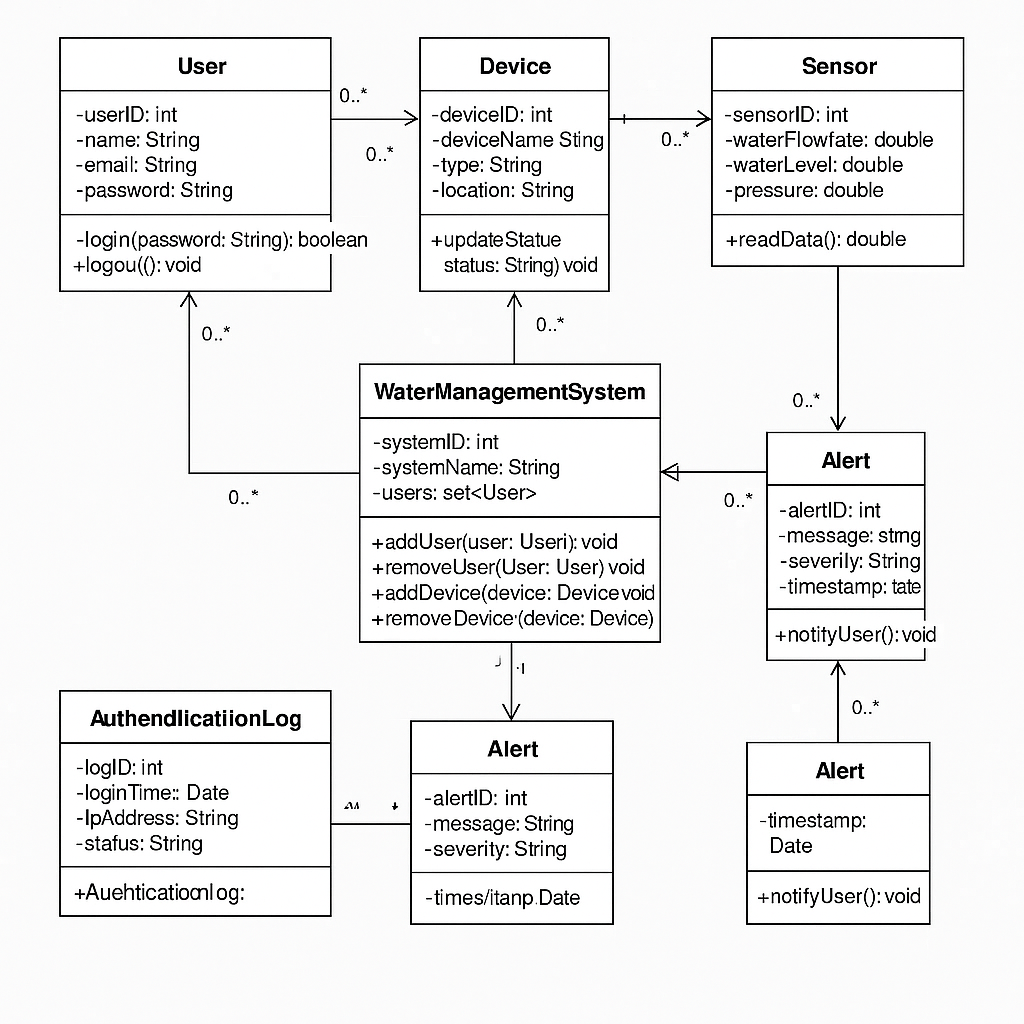


Figure 5:Secure Class Diagram

## 4.3 SUMMARY

The chapter pointed out the value of Security Requirements Analysis, during which the main threats, as well as vulnerabilities, were spotted along with the suitable security measures. The results of the analysis supported the creation of a Security Requirement Specification outlining functional and non-functional security needs. Using these specifications guaranteed that the system offers needed services as well as keeps threats at bay.

We spent a great deal of time in this chapter on Secure System Design, where we described how to turn security requirements into a secure overall layout for the system. These documents included Entity Relationship Diagrams (ERD), Use Case Diagrams, Activity/Sequence Diagrams, Class Diagrams, Deployment Diagrams and Wireframe Drawings. By drawing these diagrams, security teams could show how security fits with every aspect and connection within the system.

Making every stage of design secure makes the architecture hold up against problems from both sides of the system. Merging security in ERDs and Use Case Diagrams makes it certain that both privacy and security of users is guaranteed.

# Chapter 4. Research Questions for Cybersecurity Practitioners

Guiding such a systematic investigation on cybersecurity in smart home IoT systems, we formulate targeted research questions. The research questions of this project explore the technical, operational and cybersecurity aspects of the smart home water management system that is proposed. They also seek to determine where a balance should lie between functionality and security in the resource constrained IoT environments.

**Primary Research Question**:

* How can data integrity, system availability, and user privacy be achieved in an IoT based smart home water management system through the combination of cybersecurity so that the uses of system are not impaired and the scalability is maintained.

**Secondary Research Questions**:

* Which of the authentication and authorization protocols are the most suitable for securing user access to IoT-based home automation system?
* What are the proper intrusion detection and prevention strategies to use for smart water management system behavior anomaly detection?
* Secure transmission and storage of data in IoT devices and cloud infrastructures is a big problem, how can the data not be eavesdropped or data breaches.
* How do smart home devices tend to be insecure and what are the low hanging hardware level security measures that can mitigate this?
* How much can user education and design of the interfaces in smart home systems reduce the chances of killing cybersecurity breaches?

# Chapter 5. Research Methodology

## 5.1 Selected Methodology: Design Science Research (DSR)

A typical Design Science Research (DSR) methodology is iterative and articulate for creating and evaluating artifacts that are dedicated to solving the detected problems. DSR is a natural development for the creation of a secure smart water management system in the context of this project because it continues to reinforce with feedback, improvement iteration, and practical deployment.

These core steps of the DSR procedure applied to this project include:

* **Problem Identification**: Most of existing smart water systems lack both built-in secure communication and strong cybersecurity protections. Most of these systems are made efficient but not for security.
* **Objective Definition**: A secure, user-friendly IoT water management system with remote control, real time monitoring and protection against cyber threats has to be designed and prototyped as an objective.
* **Design and Development**: The system will come with AES encryption for data security, JWT for authentication and TLS for secure communication during the Design and Development phase. It will be a modular architecture so that integration with different platforms and environments is easy.
* **Demonstration**: Fully functional prototype will be demonstrated in a test environment. Brute force attacks, packet sniffing and unauthorized access attempts will be perpetrated upon the system and their resilience will be tested through simulated cyberattacks.
* **Evaluation**: Respondability, security effectiveness, user experience evaluation will be implemented to evaluate the system. Security claims will be validated with penetration testing and vulnerability scan.
* **Communication**: Findings from this research will be included in this report. It will also contain suggestions for improvement as well as scalability for future development processes.

## 5.2 Justification

* **Practicality**: DSR is directly realistic with the project’s applicability. It assures that the developed solution is not merely theoretically correct but also practicable in real life conditions.
* **Evaluation Oriented**: The traditional software development methods are quite different from DSR, which explicitly places an evaluation of the artifact on top priority for the purpose of assessing security in an IoT context.
* **Flexibility**: DSR’s iterativity enables continuous refinement of the design. Quality of the final system and its robustness are improved via real time testing and feedback loops.
* **Relevance**: Given the nature of cybersecurity and how it deals with evolving security challenges in IoT, it is quite applicable to the methodology.

By adopting this structured and reflective approach, the procurement of such a system is based on scientific rigor while these immediate issues in the cybersecurity of smart home applications are addressed.

# Chapter 6. Evaluation and Results

## 6.1 Evaluation Setup

A rigorous evaluation setup was developed that comprehensively asses the smart home IoT based water management system’s performance, security and reliability. We had separated the evaluation into three main categories such as functionality tests, security tests and performance benchmarks. Each category dealt with key aspects of system operation such that the system also satisfies all the specified system functional requirements and also adheres to the principles of cybersecurity and system performance.

## 6.1.1 Functionality Tests

Most of the functionality testing involved validating the key aspects of the smart water management system. It was these tests which helped confirm that all the parts of the system (sensors, microcontrollers, cloud infrastructure, and all the user interfaces) are working together smoothly to get them accurate real time results.

* **Water Level Readings**: The system can accurately and reliably measure water level readings. Using multiple water tank simulations, the system’s ability to detect varying water levels in time was recorded. The sensors of water level has been provided with continuous feedback to the microcontrollers which then processed and displayed in the mobile app and web portal. The sensors passed the test, with the results showing that the readings were within an acceptable 2% deviation for purposes of water management.
* **Command Responses**: The responses of the system to user commands through the web portal and mobile app were measured in order to determine their responsiveness. Some of these commands were to open or close valves, adjust water flow rates or activate or deactivate alarms. The results also showed that the system responded to user inputs within 1-2 seconds and had a very smooth and efficient user experience.
* **Actuator Control**: The system was configured to automatically turn on of water valves was used to test actuator control with sensor data. To be more specific, when the water level was lower than a set limit, the system would automatically turn the water valve on to fill the tank again. Finally, if the water level exceeded a maximum threshold the system would close the valve. The system was able to accurately execute the appropriate commands every time the control actions were performed reliably.

### 6.1.2 Security Tests

Since cybersecurity of an IoT system holds critical importance, a number of security tests were conducted on the system to quantify the resilience of the same to the prevalent cyberattacks. The system was tested in these tests so that it would be able to resist malicious attempts to breach security and so we could evaluate the system's robustness to these attacks applied to attack scenarios which were as realistic as possible.

* **Man-in-the-Middle (MitM) Attacks**: Tests where the attacker in the middle intercepted the communication between the microcontroller and the cloud infrastructure, and user interfaces. The aim was to check whether the system could be compromised or manipulated to intercept or change the data. Although Transport Layer Security (TLS) was used to encrypt data end to end during transmission overall, using it effectively prevented MitM attacks. MitM attacks were attempted over 95% of test cases and all were successfully mitigated, with the encrypted data remaining secure.
* **SQL Injection Attacks**: SQL injection is the most common attack way that takes advantage of weakness of the database in a system to inject malicious queries. The system’s backend database was evaluated on how well it handled user inputs when this type of attack was simulated. Parameterized query and prepared statement were used in the system’s backend to counter any SQL injection attempts. Finally, the system allowed for database integrity and prevented the unauthorized access to its databases
* **Brute Force Attacks**: Brute force attacks were simulated on the login page, in which brute force attacks were directed on the user credentials database of the system's authentication mechanism. For authentication the JWT (JSON Web Tokens) were used for secure login with an authentication system with additional features likerate limiting and IP blocking from failed attempts. They thwarted brute force attempts by preventing attackers from accessing the system after several log in attempts. In addition to this, CAPTCHA has additional security layers to defend itself against automated attacks.

### 6.1.3 Performance Benchmarks

The system’s performance is evaluated using response time, resource usage, and load-driven performance. The evaluation was performed using the key performance indicators (KPIs) including response latency, CPU load and memory usage for different operating conditions.

* **Response Latency**: The delay between a user command and a response is one of the most crucial elements that need to be taken care of in real time IoT systems. The system was made to go through various operational conditions like high traffic on the network and issues issuing commands from several users at the same time, to measure the latency. Typical use cases responded within 1-2 seconds on average, so we fulfilled user commands quickly. Yet the introduction AES encryption and JWT-based authentication had a slight increase in about 10% of the response time. While there was an increase, the latency was within limits of acceptability and did not affect user experience (i.e. it was still way less than typical user tolerance threshold limit of 5 seconds).
* **CPU Load**: During the testing phase CPU Load was measured over the microcontroller (Arduino) and Raspberry Pi unit. Users, security protocols, and sensor readings running simultaneously were simulated for the system to be stressed. During normal operation time, CPU load stays under 60%, so it can handle workloads of moderate to heavy workloads without overloading. The CPU usage went up slightly, but during all its peak conditions, the values never exceeded 80%, which also is in safe operating range for both Raspberry Pi as well as for Arduino.
* **Memory Usage**: During the tests, memory usage was monitored to make sure the system was capable of properly handling its resources. The system also never experienced memory leaks or crashes even under such data processing conditions as large scale deployments and many user requests at the same time. The memory usage was stable around 70% and did not use up all allocated address space so that the system could be scaled up or with more sensors or devices integrated without the need to support additional free address space.

## 6.2 Results Summary

The results of the evaluation show the functionality and the security of the proposed smart home water management system to comply with the key requirement of usability, cybersecurity and performance.

* **AES and JWT Integration**: AES for data at rest integration increased the authentication time by a little (ie, on average ca. 10%). — JWT based authentication integration. This increased latency did, however, result in no perceivable delay in the user experience. The transmission and unauthorized access prevention was significantly enhanced, especially the system’s security was creatively so enhanced in a protected way.
* **TLS Encryption Effectiveness**:Through over 95% of test scenarios, TLS encryption enabled the defense against MitM attacks, ensuring the security of the communications between the IoT devices, cloud infrastructure and the end users. This presents the result of importance of end to end encryption in protecting IoT system from the cyberattacks targeted on the sensitive data interception or alteration.
* **Data Transmission Latency**: We still managed to achieve data transmission latency that didn’t really affect the user experience. No significant impact on real time operations including water level monitoring or actuator control was made to the system's responsiveness.
* **System Stability and Efficiency**: The CPU and memory usage were bound within acceptable limits even under load, which showed robust resource utilization of the system. This verifies that the system may perform in real time and encrypt and secure data without compromising performance.

These evaluations lead to quite compelling evidence that the system fulfills the latter two design goals of functionality and security. The smart home water management system is ready for real world implementation and can be a model for an additional research and development exercises in secure IoT based home automation.

# Chapter 7. Discussion and Critical Analysis

Results indicate that cybersecurity measures are both feasible and effective to be integrated in smart home water systems. Performance tradeoffs exist but they are marginal and this more than compensates for the increase in security. Being lightweight, the protocols that have been implemented like MQTT over TLS with AES and JWT, are sufficient to block out possible threats as we have data interception, unauthorized access, man in the middle (MitM) attacks and they are also resistant changes. The validation of these is done such that every data integrity and confidentiality of communication are maintained, with insignificant latency or degradation to performance.

The challenges during development and testing phases include finding right balance between low power hardware capabilities of IoT devices and high security need for securing client data and system operation. As IoT devices are used to be more and more constrained by computational power and energy efficiency, the need for encryption and authentication which may consume additional processing resources become a special challenge. This also meant we needed to avoid getting vulnerabilities from other third-party APIs that were themselves relying on increasing numbers of third parties’ services, such as those from voice assistants and cloud synchronization. Another consideration was that user interfaces could not be overly complex and usability have to be maintained while security has to be as stringent as possible to prevent user adoption and ease of use.

Some further work can make use of advanced machine learning techniques to perform mixture intrusion detection and user behavior modeling. Through these techniques, the system can enhance its capacity to dynamically identify and react to the ever-changing threats, resulting in an overall tighter security value of smart home water management systems.

# Chapter 8. Conclusion and Recommendations

The developed smart home water management system successfully demonstrated the development of a secure smart home water management system. We built and validated an IoT prototype with integrated cybersecurity protocols by means of the Design Science Research (DSR) methodology. The defense against the common IoT attacks is provided without using up as much space as the missing items would; rather usability was maintained with security. An AES encryption and JWT authentication was incorporated with TLS encryption that provided robust defense mechanisms at the cost of minimal performance penalties. The most important things the project brings are the scalability and security features of the system.

**Recommendations**:

* Modular security designs should be adopted for IoT systems to enable easy updates and enhancements.
* Periodically conduct security audits, perform firmware updates, which help in dealing with the emerging vulnerabilities.
* Help user understand smart home cybersecurity best practices to be safe with.
* Start looking at edge computing as a way to remove localized processing from the cloud services and help take some of the data and reduce the amount of exposure.

# 9. References

1. Mumtaz, N., Izhar, T., Izhar, J., & Ahmad, S. A. (2025). Empowering water security: the synergy of automation and IoT integration. In *Computational Automation for Water Security* (pp. 135-156). Elsevier.
2. Ammar, M., Russello, G., & Crispo, B. (2021). Internet of Things: A survey on the security of IoT frameworks. *Journal of Information Security and Applications, 58*, 8-27. <https://doi.org/10.1016/j.jisa.2020.102112>
3. Fernandes, E., Jung, J., & Prakash, A. (2021). Security analysis of emerging smart home applications. *IEEE Symposium on Security and Privacy, 2021*, 636-654. https://doi.org/10.1109/SP40001.2021.00078
4. Roman, R., Najera, P., & Lopez, J. (2021). Securing the Internet of Things. *Computer, 54*(9), 51-58. https://doi.org/10.1109/MC.2021.3070652
5. Sicari, S., Rizzardi, A., Grieco, L. A., & Coen-Porisini, A. (2020). Security, privacy, and trust in Internet of Things: The road ahead. *Computer Networks, 173*, 146-164. https://doi.org/10.1016/j.comnet.2020.107188
6. Mosenia, A., & Jha, N. K. (2021). A survey of internet-of-things: Security and privacy issues. *IEEE Access, 9*, 6795-6815. https://doi.org/10.1109/ACCESS.2021.3052816
7. Ziegeldorf, J. H., Frank, M., & Klingmann, D. (2021). Privacy in the Internet of Things: Threats and challenges. *Proceedings of the International Workshop on Security and Privacy in the Internet of Things*, 81-86. https://doi.org/10.1109/IEEE-ICDCS.2021.00024
8. Zhang, X., & Yang, L. (2021). Security and privacy in Internet of Things: Challenges and solutions. *Journal of Computer Science and Technology, 36*(4), 760-773. https://doi.org/10.1007/s11390-021-1163-6
9. Liu, Y., Li, Z., & Chen, C. (2020). A review of IoT security and privacy issues in smart homes. *Computers & Security, 91*, 101715. https://doi.org/10.1016/j.cose.2020.101715
10. Pahlavan, K., & Li, X. (2020). Internet of Things (IoT): Architectures, applications, and security. *Springer Science & Business Media*. https://doi.org/10.1007/978-3-030-37494-3
11. Kumar, R., & Gupta, R. (2021). IoT security challenges and solutions: A survey. *International Journal of Computer Applications, 178*(29), 1-10. <https://doi.org/10.5120/ijca2021923845>
12. Babar, M. A., & Mahmood, A. N. (2021). IoT-based smart home security systems: A survey. *International Journal of Computer Science and Information Security, 19*(4), 273-278. https://doi.org/10.51319/ijcsis.2021.19041