`

Ass2-Module 5- Computer Systems (2024-2025)
Project

UNIVERSITY OF TWENTE.

Requirement Analysis Phase 1

[Security by Design Checklist]

Project Name: WasteWatch	**Team Members:** Maarten van Dort, Rein de Boer, Alexandr Gidikov, Ivan Mandev, Darius Luca, Danil Badarev
Team ID: 25	**Mentor(s):** Dany Shalhoub, Vivan Biju

`

Security Policy	Confidentiality, Integrity, and Availability					
Security Requirements	**Security mechanisms (List down for your application)**	**Remarks on why you considered these requirements. (in a brief)**	**Supplement requirements for your application (user story/Abuse case)**	**Risk identification/Threat Assessment (at least one risk identification/abuse case)**	**Appropriate Security Controls**	**Tick ✓if you have applied the given security controls as suggested in the left column**
Authentication	Email and Password validation	To properly control access to the list of trashcans, their information and their hardware, we need some kind of identification system. We chose email and password validation because it's easy to use and implement.	The system verifies that there is no default password in use. To get access to the control dashboard, authentication is required. As a user, I can enter my email and password to access the application. Abuse case: As an attacker, I can log into an account by brute-forcing passwords.	Passwords that are shorter than 6 characters and do not contain letters, numbers and at least one special character are considered vulnerable. When more than 3 login attempts fail, it is seen as an attack on the user credentials.	Weak passwords are not allowed, a password should be 6 chars or longer, contain letters, numbers and at least one special character. More than 3 failed login attempts will result in a login timeout for the account.	✓

`

			The system ensures that user input is properly sanitized. To prevent malicious users reading or modifying the database, the entered login credentials should be sanitized. Abuse case: As an attacker, I can inject special characters to extract sensitive information from the database.	Writing specific malicious text as username or password may cause SQL injection, causing the access to the database to be exposed.	All user input should be properly sanitized before it is processed or stored.	✓
Authorization	Role-based access	We have 2 roles: admin and user. An admin is a person, most likely from a cleaning staff, who has the following privileges: 1) open the lid to take out the trash. 2) access the hardware (for example to change the power bank).	As a user, I can view the level of trash that we have in the product. As a user I want to be able to open up the trash and use it. Abuse case: As an attacker I would attempt to bypass the web application to access the trash can and tamper with its hardware.	Unauthorized Access: An attacker may attempt to use privilege escalation to access the trashcan's lock mechanism. Weak Role Management: Improper role assignment could lead to unauthorized users getting access to trashcan controls.	Secure session management: User sessions timeout after inactivity, reducing the risk of unauthorized access if the cleaning staff forgets to log out. Logging and alerts: Log every access attempt (successful or unsuccessful) and alert the admis when unauthorized access is attempted.	✓
Audit	Protection of Log files	Log files are crucial for auditing system access and actions. Protecting these files ensures that sensitive data, such	User story: As a system admin I want the log files to be securely stored and protected from unauthorized access, ensuring accountability for user's actions.	Risk: Tampering or deleting log files to hide malicious activities, such as unauthorized access or privilege escalation.	Log access control: ensure that log files cannot be altered or deleted by using write-only storage or tamper-proof logging mechanism.	✓

`

		as login attempts and system activity, cannot be tampered with, providing an accurate audit trail for troubleshooting, incident investigation and compliance.	Abuse Case: As an attacker I would attempt to tamper the files to hide malicious activities through accessing the auditing system.		Encryption of logs: Encrypt log files to ensure that even if they are accessed, the contents cannot be easily interpreted.	
	Backup files	Backup files are very important, since we have profiles which give access to the functionality and in case of losing profile data, the system could be broken, because nobody will be able to access particular features. To prevent this, we use the backup files to have all the necessary information copied and loaded in case of its loss.	User story: As a system admin I would want to have a replacement for my files in case of power outage or files that were compromised. Abuse Case: As an attacker I would tamper with the files in the database so that the modified version would remain in the database.	Risk: Unauthorized access to backup files could allow an attacker to retore compromised data or seal confidential system information.	Encryption of backup files: Ensure all backup files are encrypted both in transit and at rest to prevent unauthorized access. Secure backup storage: Store backup files in a secure, off-site location to prevent physical theft or tampering.	✓
	Temporary files	Temporary files and software/database licenses often contain sensitive information and configurations.	User story: As an admin I want to ensure that all temporary files are securely managed, and that software and database licenses are protected.	Risk: temporary files could be exploited to extract sensitive data or software/database licenses could be stolen or used beyond their legal	Automatic deletion of temporary files: Implement a policy to automatically delete or securely wipe temporary	✓

`

		Protecting these ensures that temporary data is not exploited and licenses are used in compliance with legal and regulatory requirements like GDPR and software licensing agreements.	Abuse case: As an attacker I want to exploit weaknesses in temporary files to gain access to the sensitive data or misuse data licenses, which will violate legal agreements.	scope, leading to agreement violations.	files after use to avoid data leakage.	
	Database licenses (Legal aspect)	To avoid legal complications, it is important to ensure that any database used in the project complies with the appropriate licensing terms. Non-compliance with license agreements can result in legal consequences	User Story: As a system admin, I want to have the database and its licensing legally compliant to avoid improper use of software or shutdowns. Abuse Case: As an attacker I want to use illegal software to gain unauthorized access or cause the system to shut down.	Risk: If we use a proprietary database, but mistakenly assume that it's open source, it could lead to licensing fees.	We must ensure compliance with license terms by limiting database access to authorized users. To mitigate these risks, tools like multi-factor authentication, and password management systems should be implemented.	✓
	Processing of personal identifiable information on the devices (GDPR policies)	User's personal identifiable information is private and therefore should be secure. In order not to violate the rights of the User we should proceed	User Story: As a user I want my personal identifiable information to be safe and used in a way, which doesn't violate my rights	Risk: Unauthorized access to PII could result in data breaches, identity theft, or other privacy violations, leading to fines and reputational damage under GDPR.	Regular audits and monitoring: Conduct regular audits to ensure compliance with GDPR and monitor for any unauthorized access or breaches of PII.	✓

`

		with his private information very carefully and in accordance with GDPR policies.	Abuse Case: As an attacker I want to steal, access or manipulate data stored in the database. (Violation of GDPR)			
Team members' reviewed:	Luca Darius - Yes Maarten van Dort - Yes Ivan Mandev - Yes Alexandr Gidikov - Yes Danil Badarev - Yes Rein de Boer - Yes					

`

Requirement Analysis Document Template

1. Introduction

1.1. **Purpose**:

The project, namely the Trashcan Fullness Detection System, is selected for the following reasons:

- *This can digitally monitor how full trashcans in an area are from a remote location. There is no longer a need to physically check every trashcan individually.*
- *This can improve trash collection quality, as the cleaning company will know remotely that a trashcan needs emptying, and act on that. No more full trashcans for multiple days.*
- *This can improve trash collection efficiency, as the cleaning company will be able to adapt their route to what trashcans are full, resulting in less time spent and less emissions.*
- *This can be embedded broadly for any place with many trashcans, such as companies, universities and city parks.*

1.2. **Limitations of the current system (If any)**:

There are only a very limited number of implementations that are similar to the system we are planning to create. Most notably, there is a broad pilot in Bratislava in Slovakia, but there are no results or anything about it shared, so we cannot know its limitations. One difference between that pilot and our system will be the targeted audience/users: the pilot is focussed on use in public,

1.3. **Intended Audience**

- *Any user of the trashcans will benefit from this system, as they can relatively safely assume that a trashcan is empty or will be emptied soon. However, they will not be able to empty trashcans themselves or do any managing actions on the system (they are only authorised to see the locations of trashcans)*
- *The cleaning company will have access to the system and can use this to monitor and access the trashcans.*

`

1.4. **Define** **SMART** **Goals**:

The goals for the project WasteWatch are as follows:

Specific (What)	Measurable (Up to)	Attainable (How)	Relevant (Why)	Time-bound (when)
*1. To improve the **system's efficiency** by having a user-friendly web interface.*	*The system will be accessible by web application allowing the users to manage the trashcan/s.*	*Web application will allow the users to see the status of the trashcan/s, this way they will know when to dispose the trash.*	*Being able to see if a trashcan is full can save a lot of time and for example if there are a lot of trashcans this web app will be very useful for the users.*	*To finish this the task between Week 6 and Week 7.*
*2. To improve the **productivity** of the system by adding sensors such as door and window sensors, motion sensors, light sensors, temperature sensors, etc. for controlling the devices.*	*To evaluate how many times was the product cleaned/emptied, after the system noticed it was full(had status -FULL)*	*To have a separate method of finding out and retain information about when it gets cleaned*	*So that people get to throw trash without inconvenience*	*To finish task in between Week 4 and Week 5*
*3. To improve the **quality** of the system…*	*To maintain how often the trashcan is used*	*Based on the last opened time we can calculate with a formula overall quality of the trash (if people don't use it often, either the quality of the trashcan is bad or the location is bad which both lead to lack in the other fields such as productivity)*	*To ensure whether the system up to standards at all times.*	*To finish this task between Week 4 and Week 5.*
*4. To improve the **security***	*The trashcan will not*	*We will do that with latches*	*This is important because this prevents*	*To finish the task*

`

of the system…	*be able to be disposed of the trash by a person that is not part of the staff. Only a person that is responsible for disposing of the trash will be able to access the trashcan and the top part will unlock.*	*that will lock the top part and will be able to unlock through the web app, only if the person can login.*	*pollution around the trashcan.*	*between week 7-8*
5. To ensure the availability of the system using depth sensors	*Depth sensors will be used to measure how full the trashcan is and report the status to the web application.*	*The system will provide real-time percentage readings of the trashcan's fill level, notifying users when it reaches 80% or more.*	*The Raspberry Pi will process sensor data at regular intervals and send updates to the web application.*	*Accurate trash level monitoring helps with optimizing waste management and prevents overflow, ensuring a cleaner environment.*

`

2. Scope:

Stack:

- ***Software:***
  - *Python for sensor control, user authentication, and backend logic.*
  - *Web technologies (HTML, CSS, JavaScript) for the front-end web interface.*
  - *A Python-based web framework.*
  - *Secure login system for users to access trashcan status and control latches.*
  - *Algorithms to handle sensor data and manage the locking/unlocking mechanism based on user authentication.*
- ***Hardware:***
  - *Raspberry Pi 5 for controlling the system.*
  - *Ultrasonic or IR sensors for detecting user proximity and triggering the automatic opening of the lid.*
  - *Depth sensors to measure how full the trashcan is.*
  - *Latch system for locking/unlocking access to authorized users only.*
  - *Wi-Fi module to connect the Raspberry Pi to the web application.*
  - *Power supply to ensure continuous operation.*
- ***Interfaces:***
  - *Internet access via Wi-Fi to connect the Raspberry Pi to the web server.*
  - *Web-based application accessible by users with credentials, providing control over the trashcan and real-time monitoring of the fill level.*

Limitations:

- *The project is limited to controlling only the trashcan's opening mechanism and monitoring its trash level.*
- *The system can manage a finite number of trashcans through the web interface, depending on the available hardware resources.*
- *The depth measurement system will only work accurately within certain sensor limitations and may require calibration for different types of trash.*
- *Physical security of the trashcan is limited by the strength and reliability of the latch system.*
- *Internet connectivity is essential for the full functionality of the system; if the connection is lost, users will not be able to access the web interface remotely.*

Constraints:

- *The duration of the project may be affected by the availability of hardware components, especially sensors and security latches.*
- *Ensuring reliable Wi-Fi connectivity in outdoor or difficult-to-reach locations can be a technical challenge.*
- *Developing a secure login system that prevents unauthorized access to the web application will require rigorous testing.*

`

- *Battery or external power sources may be needed if the trashcan is to be used in locations without easy access to electrical outlets.*

3. **Product features:**

A. Functional requirements:

- *[H] The system should be connected to the website and database with user accounts in order to send notifications.*
- *[H] The system should determine at least 3 levels of how full it is (for example <85%, >85%&<98%, >98%).*
- *[H] The system should block the servicing lid to limit access to the trash bag and hardware, unless unlocked by the web app.*
- *[M] The system should block the lid if it reaches the third level.*
- *[M] The system should control the lid of the trashcan, i.e. open it when someone approaches the sensor and close it when no one is nearby.*
- *[M] The system should have a slot for bags, which opens using the app in order to simplify the process of garbage removal.*
- *[L] The system should have an air freshener installed, which removes the unpleasant smell.*
- *[L] The system should turn the light on when the lid opens in order to find the trashcan even in the middle of night.*

B. Nonfunctional requirements:

- *[H] The system should react to a person's appearance within 10 cm radius and 2 seconds of time. (i.e. open the lid).*
- *[H] The system should accurately inform the user about its fullness (i.e. 90% of the time the notification should be relevant)*
- *[M] The system should only react on a person and not be triggered by random event.*
- *[M] The system should send a reminder to empty it in less than 2 minutes after it is full.*
- *[M] The system should be energy efficient and work on a single charge for two weeks.*

C. Security requirements:

- *[H] The system should allow user authentication on the website.*
- *[M] Hardware inside the trashcan should not be easily accessible. (i.e. to take the trash out you should use the app)*

4. **Conclusion:**

`

Based on the requirements we listed, this smart trashcan system aims to provide a scalable solution for managing and servicing trash cans. It combines depth sensors, latches and raspberry pi's to realize a network of trash cans that can be controlled from a web application.

The integration of Python for the backend and GPIO control coupled with modern web technologies for the web application allows for a flexible and easy to use product. One of the critical security design choices is the login system for the web application. It secures the locking and unlocking of trash cans by activating a latch, making sure only authorized people can empty the bin and get access to the hardware that will be placed on the inside of the trash can.
The use of depth sensors allows the system to monitor the amount of trash in each trash can, allowing a warning to be displayed when a trash can should be emptied. It also allows to run further analysis of the usage and efficiency of the trash can in a specific location.

The system relies on internet connectivity which allows it to be deployed in a wide area. This system also enables remote management which further improves the scalability of the system.

5. **Reference**: List the existing literature (documents/articles/blogs/research papers) references you have considered for finalizing the project idea.
   - *No specific documents or articles, as we used ChatGPT with many different prompts for the broad ideas, then worked things out since then.*

Regarding existing projects:

- *A pilot regarding waste sensors, executed in Bratislava: https://sensoneo.com/trash-bins-monitoring-in-city-centre/*
- *https://www.astron.si/en/smart-city/garbage-container-fullness-sensor/*

6. **Usage of AI tools:** State whether AI tools were used for this assignment, If yes, which AI tools are used, describe how and why you used them; alternatively, state clearly that AI tools were not used.

   Our team has used Generative AI to help brainstorm ideas. We did not copy any (large) sections of generated text.

[1]**Note:** ***The security requirements should be mapped with the SBD requirement analysis (phase 1) checklist. You are free to write the security requirements in the form of a user story/abuse case.***