IT 360 PROJECT

EntriGuard

"Password Strength Tester"

BY :

FERDAWES HAOUALA

MELEK KAHLOUN

HOUDA RABIA

TUNIS BUSINESS SCHOOL

UNIVERSITY OF TUNIS

REPORT

TUNIS BUSINESS SCHOOL

2023-2024

CONTENTS

Introduction

As digital dependence grows, so do the threats to our online security. Passwords remain the frontline defense for user accounts, but weak passwords are easily compromised, leading to devastating consequences like data breaches, identity theft, and financial loss. Thus, creating and remembering strong, unique passwords for various platforms can be a challenge. This is where password strength testers come in to analyze passwords and provide valuable insights into their security.

The Importance of Password Strength Tester:

The Password Strength Tester is a free cybersecurity tool designed to empower users to create strong, secure passwords. It measures the strength of your password using a given criteria such as complexity, length, and randomness. It estimates how much time it would take to crack a given password. To ensure that passwords protect bank accounts, other subscriptions, and internet accounts, use a password tester – it is quick and easy. This project adheres to the best practices outlined by the Open Web Application Security Project (OWASP).

Main Concepts

1.Main Concepts

OWASP: The project of OWASP Application Security Verification Standard (ASVS) focuses on testing technical security controls of web applications as well as giving developers a checklist for safe development. The main goal of the project is to level out the scope and rigor spectrum in the marketplace in terms of Web application security verification using a commercially workable open standard.d.

Password complexity: Password entropy, also known as password strength is a measure of how hard a password can be guessed especially against brute force attacks. Often for first-time users trying to use applications or devices, password complexity relates to safety requirements that evaluate its security level. The more complex the password is the greater protection it will give against guessing and similar attacks.

Strong vs. Weak Passwords: Strong passwords are long and contain a variety of characters (which include capital letters, lowercase letters, numbers, and symbols). Weak passwords could be short, easily predicted, or composed of common words.

Main Concepts

User Flexibility: Subject-Verb Agreement: Some flexibility has to be given to users in enforcing their password rules so that they can make strong and memorable words. Forcing overcomplicated password codes might result in consumers jotting them down or using them for more than one account, which may undermine security as a whole.

Brute-Force Attacks: This hacking technique uses all possible combinations of characters until the password is eventually guessed. However, cracking strong passwords usually takes much more time than using brute-force attacks.

Password Cracking: It encompasses brute force attacks, dictionary attacks (trying out common words and phrases), and social engineering (tricking the user into revealing the password itself); this is thus an all-inclusive term in which different hacking methods can be classified.

Main Concepts

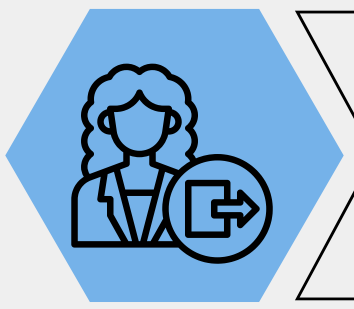
2. Main Components:

Several major components make up the password strength tester and work together to determine how strong a password is:

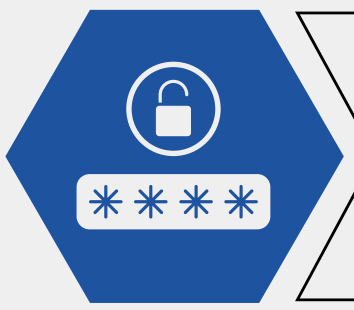
- **User Interface (UI):** With the UI, users type passwords or phrases and have their strengths checked. This part offers explicit directives on how to create strong passwords, checks for some basic rules such as minimum length and complexity, and provides a strength score plus feedback to the user.
- **Database:** It keeps track of frequently used passwords in one place. When a password entered by a user matches any entry in the database, irrespective of whether it satisfies current password requirements or not, the system immediately declines it.
- **A password evaluation algorithm:** The system's core component which defines the strength of a password is referred to as the password evaluation algorithm. Since it should comply with the industry's rules and best practices, this algorithm is expected. To determine how strong a given password is, various aspects ought to be put into consideration by this algorithm; these include length, diversity, and identification among others. It has to give suggestions on ways of bettering the password if required.

Main Concepts

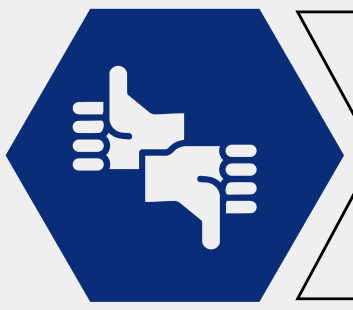
3. Functional Flow:



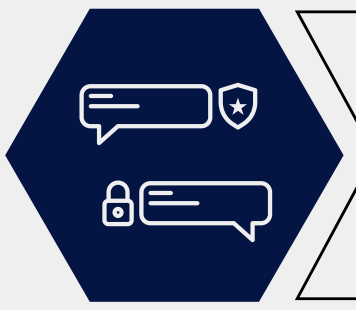
User Input



Password Validation



User Feedback



Additional Features

Main Concepts

3. Functional Flow:

To evaluate the password’s strength and ensure it meets best practices, the Password Strength Tester follows the following process:

1. User Input:

- Users enter their desired password.
- The system should provide clear guidance on creating strong passwords, including length requirements and complexity (mix of uppercase, lowercase, numbers, and symbols).
- Additionally, offering tips for memorability is crucial, as complex passwords can be difficult to remember.

2. Password Validation:

- The system checks the password against essential security standards, like OWASP (Open Web Application Security Project) guidelines.
- This includes minimum length and complexity requirements.
- Furthermore, the system should verify the absence of common words, phrases, or personal information (names, birthdays) to avoid easily guessable passwords.
- Advanced password strength testers might employ techniques like dictionary word checks, common pattern recognition, and verification against known compromised password databases.

Main Concepts

3. User Feedback:

- The system provides clear and concise feedback on password strength.
- This can include a rating system (weak, medium, strong) or a color-coded indicator for better visual representation.

4. Additional Features (Optional):

- Some password strength testers offer:
- Password expiration and reset functionalities to enforce regular password changes.
- Multi-factor authentication (MFA) for added security beyond just password verification.
- Integration with existing systems for streamlined password management.

Overview of the main existing solutions

1. Main Characteristics:

It's important to understand the following features of a good password strength checker:

Checks Password Strength:

1. Length: A good checker verifies the password length. A longer password always takes a longer time to crack through a brute-force attack. All the work of password cracking is done by computer processes. Using long passwords means utilizing more computer power and time. Shorter passwords will be cracked in the blink of an eye. So to find security from brute force attacks, the password length is more important than complexity.

2. Complexity: A good checker checks the complexity of the password. A strong password contains the upper case and lower case letters along with numbers and special symbols. To make passwords more secure and harder to crack it is always better to use a combination of these characters. To summarize, Complexity = Upper Case + Lower Case + Number + Special Symbols ($) + Special Characters (@&#!).

3. Lacks Predictability: A good password strength checker avoids the use of common patterns or dictionary words. Adding birthdays, pet names or personal data tends to weaken the password.

Overview of the main existing solutions

Feedback:

1. Strength Meter: A good password strength checker offers a meter or rating in the form of password strength (weak, medium, strong).
2. Tips and Improvement Suggestions: It also offers additional improvement suggestions if required. For example, if password length increases, different types of characters could be added.

Security:

1. Client Side Processing: The ideal checker runs on your device and doesn't send the password to a server. Therefore, your password will remain private.
2. No Storage: A good password strength checker doesn't store your password in any way.

Additional Features:

1. Password Generation: Some password strength checkers also suggest strong, random passwords instead of verifying the password that the user has entered.
2. Dictionary Check: Some checkers also check the password against the database of commonly leaked passwords.

Overview of the main existing solutions

2. Advantages & Limitations of Password Strength Testers:

Advantages:

- Raise Awareness: Testers educate users on the importance of strong passwords and the dangers of weak ones.
- Immediate Feedback: Users receive real-time feedback on their password strength, guiding them towards more secure options.
- Increased Security: Encouraging strong passwords improves overall account security and reduces the risk of hacking attempts like brute-force attacks.
- Convenience: Testers are readily available online or integrated into password management applications, making them easily accessible.

Limitations:

- Limited Scope: Testers primarily focus on password length and character complexity. They may not account for context-specific risks or advanced hacking techniques.
- Dictionary Reliance: Testers relying solely on dictionary checks may miss novel or unpredictable password variations.
- False Sense of Security: A high password strength rating doesn't guarantee complete security. Other factors like phishing attacks or malware can still compromise accounts.
- Bypassed by Sophisticated Attacks: Testers are less effective against targeted attacks or those exploiting software vulnerabilities.
- Potential Complexity: Complex password creation rules can frustrate users and lead to password reuse across multiple accounts.

Overview of the main existing solutions

3. Advantages & Limitations of Existing Solutions:

There exist primarily two categories of password strength checkers: online tools and built-in checkers:

Online Password Strength Checkers:

These platforms, accessible via websites or web applications, allow users to input their passwords for strength assessment. However, caution is advised due to potential security risks associated with certain online checkers.
Key attributes of reliable online checkers include:

- Client-side processing to maintain password confidentiality.
- Absence of password storage to mitigate security risks.

Examples:
Some reputable online password strength checkers include LastPass, Bitwarden, How Secure Is My Password, and Kaspersky Password Strength Meter.

Advantages:

- Accessibility from any device with internet access.
- Potential for advanced features like password generation and leak checking.
- Wide range of options catering to diverse user needs.

Limitations:

- Security risks associated with potential data interception or tracking.
- Varying accuracy in strength assessment.
- Privacy concerns related to data collection practices.

Overview of the main existing solutions

Built-in Password Checkers:

Many websites and applications integrate built-in password strength checkers that evaluate passwords as users input them. While generally more secure than online alternatives, built-in checkers may lack certain functionalities. Important features to look for in built-in checkers include:

- Strength meter for visual feedback on password strength.
- Recommendations for enhancing password security.

Examples:

Examples of platforms offering built-in password checkers include web browsers, password manager applications, and various online registration forms.

Advantages:

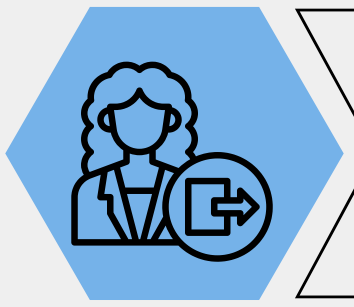
- Enhanced security by avoiding password transmission for analysis.
- Seamless integration into website and application interfaces.
- Mitigation of privacy risks associated with third-party interception.

Limitations:

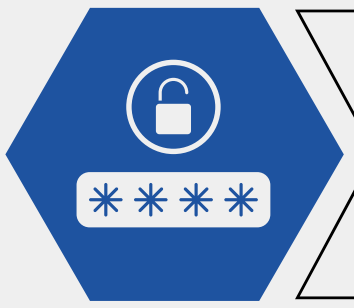
- Limited feature set compared to online counterparts.
- Basic strength assessment may lack advanced techniques like dictionary checks.
- Dependency on platform support for built-in checkers.

3rd deliverable

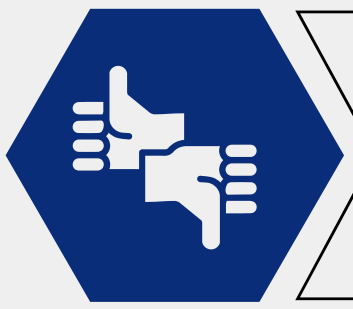
contents



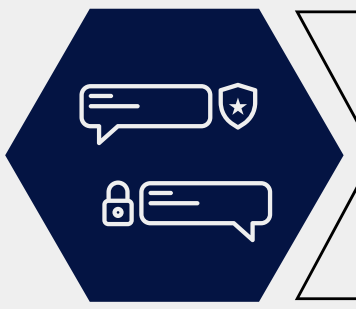
The design



The components



The exchanged messages ...



Working flows

The design

Secure Password Strength Tester: A Two-Part Solution

This Password Strength Tester solution leverages a secure two-component architecture:

By combining a user-friendly front-end with a secure and comprehensive back-end, this Password Strength Tester solution empowers users to create strong passwords, significantly enhancing overall account security.

1. User-Friendly Web Interface:

- **This web application acts as the user's entry point. Accessible through a standard web browser, it provides a clean and intuitive interface for users to:**
  - **Enter their desired password.**
  - **Receive immediate feedback on its strength.**

The interface communicates securely with the back-end system using an API (Application Programming Interface), ensuring data privacy.

2. Robust Back-End Processing:

- **This server-side application forms the core of the solution. It handles the following critical tasks:**
- **Processing user-entered passwords.**
- **Evaluating password strength against a set of predefined rules. These rules typically include minimum length requirements, character complexity (uppercase, lowercase, numbers, symbols), and avoiding common patterns.**
- **Utilizing external services for enhanced security:**
- **Third-party password dictionaries: Checking entered passwords against known weak or compromised passwords.**
- **Blacklist services: Verifying passwords are not on lists of leaked credentials.**

Based on this comprehensive analysis, the back-end generates a strength rating for the user's password. This rating can be displayed as a color indicator (e.g., green for strong, red for weak) or a textual score (e.g., "Strong", "Weak").

The components

Behind the Scenes of the Strong Password

A password strength tester is a valuable tool for creating secure passwords. Here's a breakdown of the key components working together to evaluate the password's strength:

the exchanged data process

Ensuring Strong Passwords: A Secure Communication Flow

The Password Strength Tester employs a secure communication flow between the user interface and the back-end system to evaluate password strength:

front end

User Input

Users enter their desired password into a user-friendly front-end interface.

API REQUEST.

HTTPS FOR SECURE COMMUNICATION

back end

This ensures the password doesn't travel in plain text across the network

Back-End Validation

The back-end system receives the password and performs several validation checks:

Predefined Rules: The password is assessed against established criteria like minimum length and character complexity requirements (uppercase, lowercase, numbers, symbols).

Test Cases: Advanced validation might involve checks for common patterns, dictionary words, and variations to identify potentially weak passwords.

External Verification
Third-Party Password Dictionaries

The back-end system analyzes the password based on these checks

RETURNS A STRENGTH RATING

easy visual feedback about password strength & the system might provide specific suggestions for improvement

The working flow

A- Functional requirements :

Strong Passwords for Enhanced Security

This system prioritizes user account security by implementing robust password creation procedures:

- **User Input: Users provide essential information during registration:**
  - **Full Name**
  - **Email Address (for communication and potential password reset)**
  - **Birthdate (optional for some systems, consider security implications)**
  - **Password (the focus of security checks)**
- **Dictionary Check: The system safeguards against easily compromised passwords by verifying if the chosen password exists in a dictionary or known weak password database. If a match is found, the system immediately rejects the password and prompts the user to create a stronger one.**
- **OWASP Standards Compliance: The system enforces password complexity based on OWASP best practices:**
  - **Minimum length of 8 characters (increased length strengthens security)**
  - **Inclusion of a combination of uppercase, lowercase letters, numbers, and special characters (improves resistance to brute-force attacks)**
  - **Exclusion of predictable patterns (e.g., keyboard sequences, birthdays)**
  - **Uniqueness (prevents password reuse across multiple accounts)**

- **Password Strength Estimation (Optional):**
  - **Advanced systems might estimate the time it would take to crack the password using different algorithms. This provides users with a sense of their password's robustness.**
- **User Feedback: Clear and informative feedback is crucial. The system should indicate the strength of the password using a rating system (weak, medium, strong) or a color-coded indicator. This helps users understand if their chosen password meets security requirements.**
- **Password Strength Report (Optional):**
  - **Some systems might provide a detailed report outlining the password's strengths and weaknesses based on the various checks performed. This can guide users towards creating even stronger passwords in the future.**

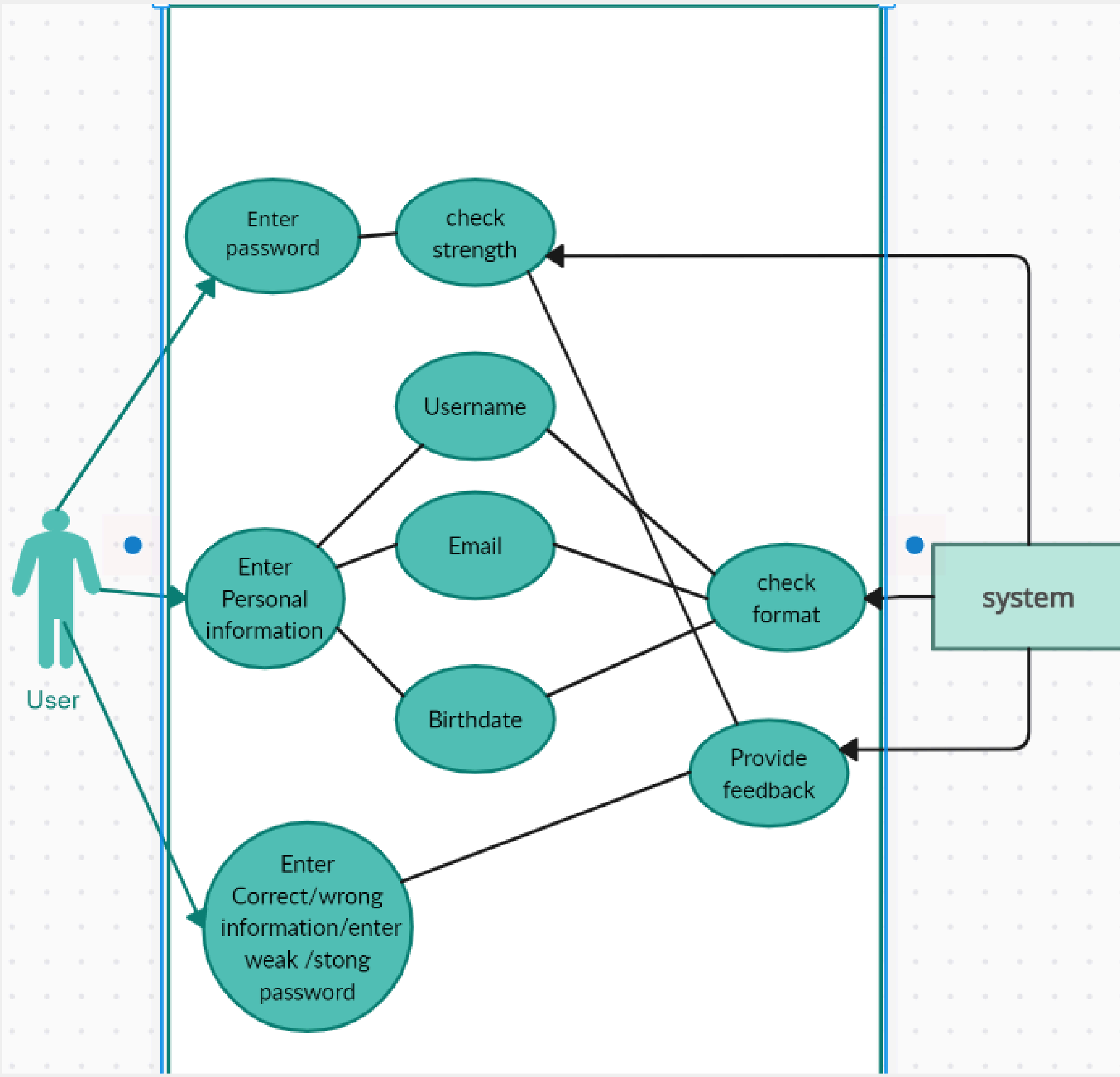
B- Non-functional requirement :

User Access and Security:

- **The system will be accessible through a web interface, allowing users to interact with it from any device with a web browser.**
- **The user interface (UI) will be designed for intuitiveness and ease-of-use, requiring minimal to no training for users to navigate and complete tasks.**
- **Security is paramount. The system will provide a secure environment for users and ensure that no confidential user information is ever stored. This includes sensitive data like passwords, financial information, or personally identifiable details (PII).**
- **Compliance is crucial. The system will be designed and operated to comply with all relevant laws and regulations, including data protection laws like GDPR (General Data Protection Regulation) and CCPA (California Consumer Privacy Act). Additionally, it will adhere to ethical requirements related to user data handling, such as transparency and user control over their information.**

Important Information Added:

- **Accessibility: Specifying web-based access emphasizes user convenience and potential for remote interaction.**
- **Data Security: Highlighting "no storage of confidential user information" strengthens the security focus and user privacy considerations.**
- **Compliance: Adding specific examples of data protection laws (GDPR, CCPA) clarifies compliance expectations.**
- **Ethical Requirements: Mentioning ethical considerations regarding user data handling reflects a commitment to responsible data management practices.**


Enter password
check strength
Username
Email
Enter Personal information
check format
system
Birthdate
Provide feedback
User
Enter Correct/wrong information/enter weak /stong password

Tools and Development Phases:

A- Development Phases :

1. **Requirement Analysis & Design:** Document all functional and non-functional requirements. Design system architecture and user interface mockups.
2. **Back-end Development:** Use Python and Flask to develop the core application logic, password validation routines, and integrations with external services.
3. **Front-end Development:** Utilize HTML, JavaScript and CSS to build a user-friendly web interface for user registration, password creation, and feedback on password strength. Integrate password strength meters and optional detailed reports.
4. **Testing and Deployment:** Conduct thorough unit testing, integration testing, and security testing before deployment. Deploy the application to a secure web server.

B-Tools:

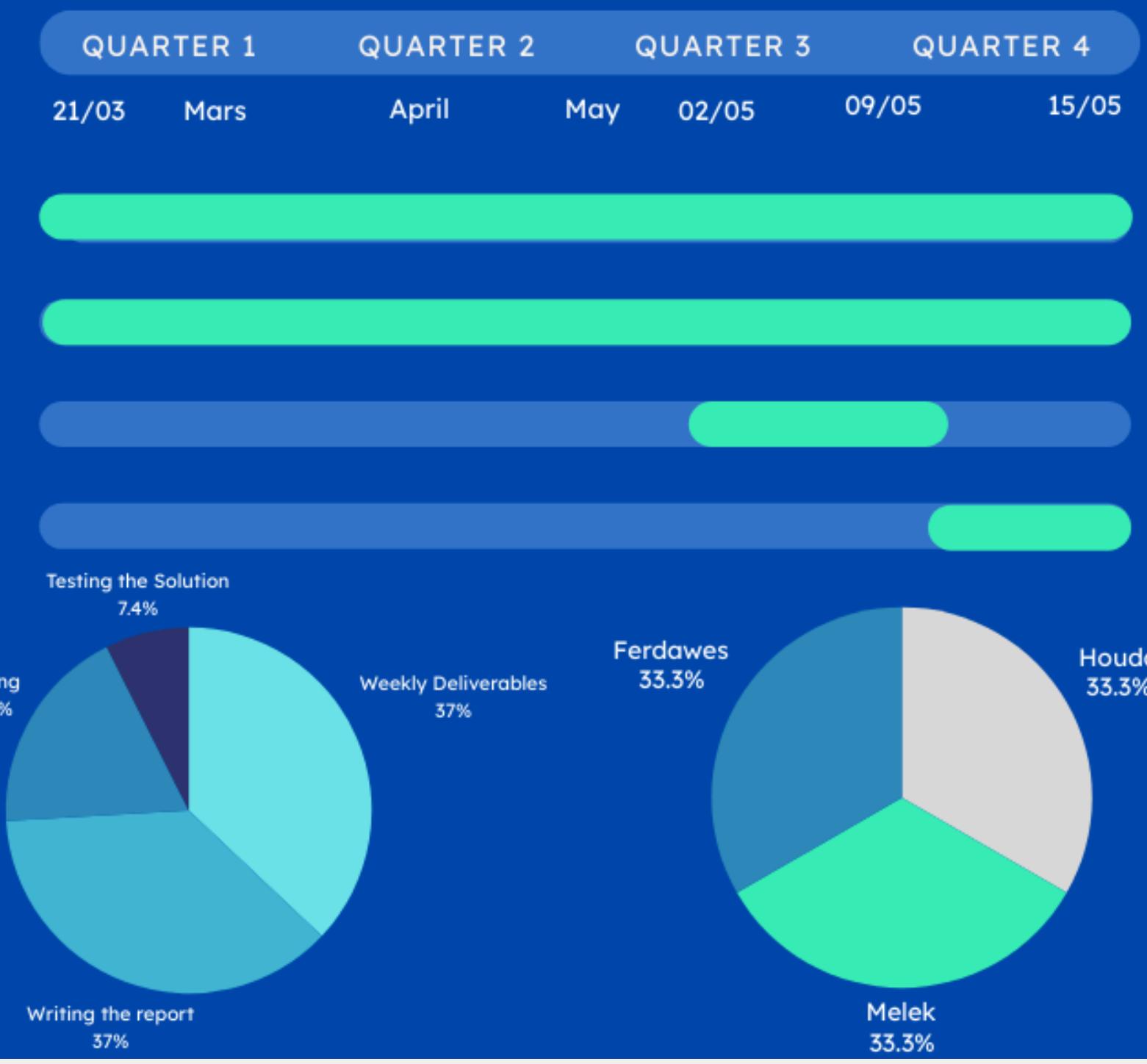
- **Programming Language:** Python (back-end development)
- **Web Framework:** Flask (back-end development)
- **Front-end Framework:** HTML/JavaScript/CSS (user interface development)

Project’s Timeline

EntriGuard

Gantt Chart

PROCESS



Executable Version of the Solution

Click Here

https://github.com/MelekKahloun/EntriGuard