Design Document
Course: MSc Computer Science
Module: Secure Software Development (Computer Science)
Assignment: Team Project – Design Document
Date: Monday 20th September 2021
Group: 3 (Kieron Holmes, Kikelomo Obayemi, Sergio Zavarce)

1. Background

1.1. The Large Hadron Collider

According to the Guinness World Records (2021), the Large Hadron Collider (LHC) is the world's largest and most complex machine, spanning about 27km of underground tunnel space. The LHC is a high-energy particle accelerator built by the European Organisation for Nuclear Research (CERN) to aid physicists in discovering new particles such as the "Higgs boson" predicted by the standard model in particle physics (Brüning et al., 2012).

1.2. Challenges

With the ongoing upgrades on the LHC set to go live after 2025 (CERN, n.d), it is expected that future experiments by physicists will generate more observation reports, therefore increasing the number of LHC-related digital files. The growing concern here, however, is the theft of digital information. Sharma (2018) described an incident involving the theft of about 8 billion pages of research materials from universities across over 20 countries, research institutions and organisations, including the United nations. Further investigation by cybersecurity experts revealed that the universities targeted were not randomly selected but rather prominent research universities (Sharma, 2018).

In a bid to also address some of the cybersecurity challenges around data theft, The Computer Security Team at CERN has recently conducted an investigation to identify sensitive CERN information that might have been exposed by cyber attackers or obtained for malicious use (Computer Security Team, 2020). Such information includes the organisation's operations and webpages, staff and users' passwords or credit card information. (Computer Security Team, 2020).

To alleviate the aforementioned concerns and position the CERN for safekeeping of future research work, this proposal seeks to develop a secure API for CERN researchers and employees to access and store research materials and miscellaneous documents.

2. Methodology

The Agile software development principles will be used throughout the development of this service. When compared to alternative methodologies, such as waterfall, Agile helps eliminate

the total chances of total project failure (Amit, n.d.). The development will be scheduled into sprints, allowing us to constantly refine and prioritise the items in the upcoming period (Segue Technologies, 2015). Methodologies such as waterfall are extremely inflexible when changing work completed within previous stages (Eby, 2017).

In line with the Agile principles, we will be using Test-Driven Development (TDD) during the development lifecycle of this project. Studies have differing views on the productivity benefits of this method; however, they all agree that this test-first approach often results in better code quality and fewer bugs (Khanam & Ahsan, 2017). Due to the nature of TDD, a test will be created based upon the expected output of a solution (which is expected to fail initially), to which a developer will write some code to make the test pass. This approach ensures that code can be robust and refactorable with minimal risk of breaking functions (IBM, n.d.).

We have chosen to use the GitHub code version control platform due to the extensive workflow library providing test coverage, dependency checking and linting functionalities. In addition, branch policies will be in place to encourage code peer review, ensuring it is well designed and free of obvious errors (Radigan, n.d.).

2.1. Security

To comply with the security requirements of this project, the recommendations of the OWASP API top 10 project will be observed. Awareness of the most common API vulnerabilities is crucial throughout all the development lifecycle stages, which is achieved through the education of everyone involved in development and maintenance (OWASP, 2019). According to OWASP (2019), the most common and critical attack performed on API's is Broken Object Level Authorisation, which leads to breaches, loss or data manipulation. It can also lead to account takeover. To prevent this, every endpoint that can potentially receive an Object ID must check that the user has the privileges to perform the requested action over the object. Table 2.1 describes other significant vulnerabilities and prevention measures.

Vulnerability	Exploit	Prevention
Broken user authentication	Weak password validation; weak expiration date validation; weak encryption or no-encryption for passwords	Check all authentication flows; use standards for authentication; treat password/credential recovery as login endpoints in terms of other authentication validation and protection
Excessive data exposure	Traffic sniffing can expose sensitive data by analysing API responses.	Expose API endpoints only to those valid consumers of this information; review API responses to check if only contains required information; return only required properties instead of entire object converted to string
Lack of Resource & Rate Limiting	Not implementing rate limiting or with a bad implementation, may lead to DoS exploitation, making the API unresponsive	Add server-side validation of parameters; stablish limits for maximum data payloads; use docker to limit computing resources

Table 2.1: Other vulnerabilities to consider for this project (OWASP, 2019)

2.2. Tools/Libraries

When developing the document repository system for CERN researchers, code linting/styling tools will be used to ensure that developers are not required to memorise PEP8 styles and make rule/taste-based decisions themselves (Augustin, 2019). A list of the tools we intend to use has been included in table 2.2.

Code Version Control	Frameworks	Databases	Automated Workflows	Utilities
Github	Django	MySQL	Black	Microsoft Planner
	Django REST		iSort	Visual Studio Code
			CodeCov	
			pyLint	

Table 2.2: Tools selected for this project

2.3. Scope of Work

We will be developing an API that provides the following functionalities:

- User registration.
- Different user roles (CRUD functionality).
- User authentication.
- GDPR Compliance Utilities (Deleting Account, Correcting/Updating Data, Requesting Data).

3. Design

For this project, a set of recommendations, design patterns and good practices will be implemented to improve security, reliability, readability, and extensibility. The goal is to apply the most important REST API design patterns to make the API easy to consume, ubiquitous in language and friendly to standard HTTP methods. In addition, the OpenAPI standard will be considered while defining security, endpoints, and HTTP status messages in conjunction with the Django framework (Ravindran, 2015).

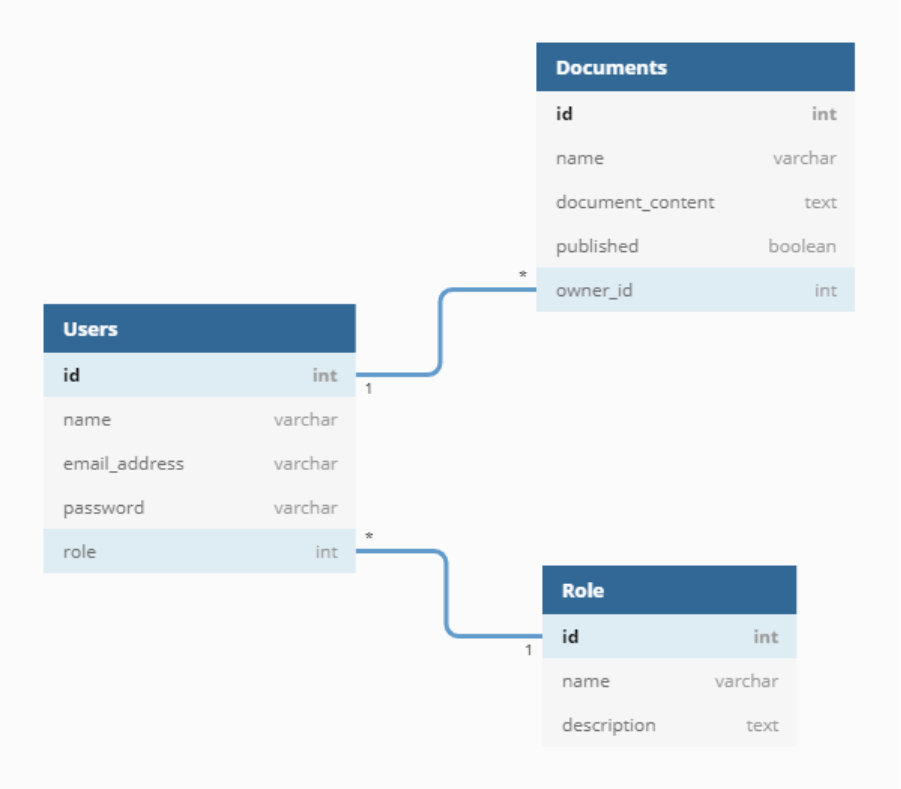
A typical design pattern in REST is to specify the resource as the endpoint name, avoiding actions that might restrict the uses for a specific endpoint. The RESTful API conventions include particular methods for actions, known as "verbs", with the most common being GET, POST, PUT & DELETE. These verbs must respond to the caller using a coherent structure covering all possible scenarios, including bad requests.

3.1. Specifications

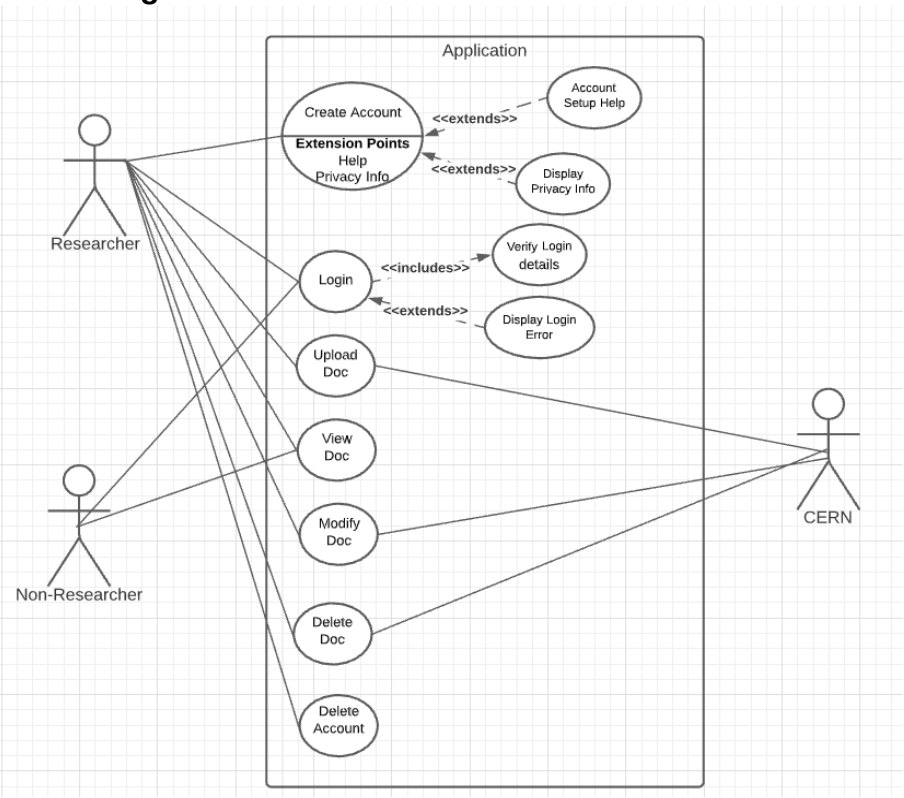
The CERN application will be developed in an operating-system agnostic manner using the Python 3.x programming language, coupled with the Django web framework. In addition, the application will be created using the REST API design patterns discussed within the Design section.

3.2. Diagrams

3.2.1. Entity Relationship Diagram:



3.2.2. Use Case Diagram:



4. Assumptions

When developing the CERN document repository, we have made the following assumptions:

- **Overall Network Security:** The CERN network and infrastructure are secure and protected against malicious users wanting to carry out attacks from elsewhere in the business (Cisco, n.d.).
- **Social Engineering Vulnerabilities:** The staff are well trained and resistant to Social Engineering attacks.

5. Ethical Considerations

We are committed to the privacy and protection of user information; therefore, our proposed solution will be developed using the Privacy by Design (PbD) principles introduced by the GDPR to ensure that privacy is engineered into the design process (Privacypolicies, n.d).

6. References

Amit. (n.d.) Why Agile Project Management Is Better Than Waterfall?. Available from: https://www.orangescrum.com/blog/why-agile-project-management-is-better-than-waterfall.html [Accessed 19th September 2021].

Augustin, A. (2019) DEP 0008: Formatting Code with Black. Available from: https://github.com/django/deps/blob/main/accepted/0008-black.rst [Accessed 19th September 2021].

Brüning, O., Burkhardt, H. and Myers, S. (2012). The large hadron collider. Progress in Particle and Nuclear Physics, Elsevier, 67(3):705-734.

CERN (n.d) Computing: A central role in the fulfilment of CERN's mission. Available from: https://home.cern/science/computing [Accessed 06 September, 2021].

Cisco. (n.d.) What is network security?. Available from: https://www.cisco.com/c/en_uk/products/security/what-is-network-security.html [Accessed 19th September 2021].

Eby, K. (2017) What's the Difference? Agile vs Scrum vs Waterfall vs Kanban. Available from: https://www.smartsheet.com/agile-vs-scrum-vs-waterfall-vs-kanban [Accessed 19th September 2021].

Guinness World Records (2021) Largest Machine. Available from: https://www.guinnessworldrecords.com/world-records/103591-largest-machine [Accessed 06 September, 2021].

IBM. (n.d.) Test-driven development. Available from: https://www.ibm.com/garage/method/practices/code/practice_test_driven_development/ [Accessed 19th September 2021].

Khanam, Z. Ahsan, M. (2017) Evaluating the Effectiveness of Test Driven Development: Advantages and Pitfalls. *International Journal of Applied Engineering Research* 12(18):7705-7716. Available from: http://www.ripublication.com/ijaer17/ijaerv12n18_81.pdf [Accessed 19th September 2021].

Radigan, D. (n.d.) Why code reviews matter (and actually save time!). Available from: https://www.atlassian.com/agile/software-development/code-reviews [Accessed 19th September 2021].

OWASP (2019) OWASP API Security Project. Available from: https://owasp.org/www-project-api-security/ [Accessed 19th September 2021].

Privacypolicies (n.d) Implementing Privacy by Design Available from: https://www.privacypolicies.com/blog/privacy-by-design/ [Accessed 19th September 2021].

Ravindran, A. (2015). *Django Design Patterns and Best Practices*. Packt Publishing Ltd.

Segue Technologies. (2015) 8 Benefits of Agile Software Development. Available from: https://www.seguetech.com/8-benefits-of-agile-software-development/ [Accessed 19th September 2021].

Sharma (2018) Hundreds of Universities Targeted in Global Data Steal.*The University World News.* Available from: https://www.universityworldnews.com/post.php?story=20180327132912436 [Accessed 06 September, 2021].

The Computer Security Team (2020) Computer Security: Digital Stolen Goods of CERN? Available from: https://home.cern/news/news/computing/computer-security-digital-stolen-goods-cern [Accessed 06 September, 2021].