System Design for a Secure Repository for NASA

Table of Contents

Introduction

Data sharing is a core premise of scientific computing, and it is becoming increasingly important as we move toward data-intensive sciences and engineering. This extends beyond scientific outputs to include raw and intermediate data used in the scientific process.

In a report of the International Space Station Independent Safety Task Force (NASA, 2007), a strengthening of the program was recommended by increasing the likelihood of success and mitigating risks to crew safety or health.

As the International Space Station (ISS) (NASA, N.D) is a joint international operation consisting of 16 countries, and there is a need to share data and collaborate to achieve its goals. Therefore, a secure repository is being developed and a design proposal report for the application is included below. The application specifically relates to items 5.9 and 5.10 from the report and due to the tight 3-week time constraint imposed, the application will be built using Agile methods rather than the more traditional Waterfall methods (van Casteren, W, 2017).

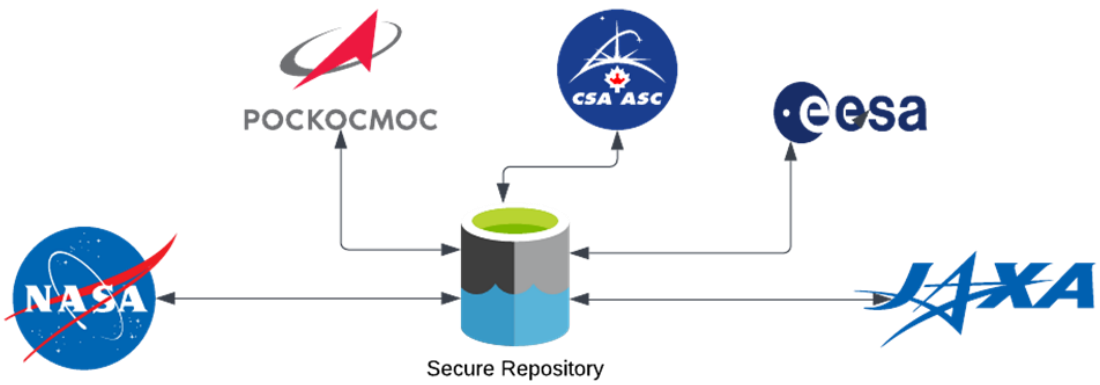

Figure 1: NASA and International Partner Operations Scope

Assumptions, Requirements and Limitations

Assumptions (A) and Limitations (L)	
Network	• **(A)** Sufficient bandwidth routing remote user to and from firewall as IPSec VPN. **(L)** Reduced bandwidth can affect file transfers and cause latency. • **(A)** Firewall is assigned public IP (external port) and communicates with ISS and employees over IPSec VPN tunnel (encryption in transit) and a private IP on its internal port (Mhaskar, Alabbad and Khedri, 2021) • **(A)** Network monitors agent installed on load balancer to send data to Control Centre LAN (Tsochev et al., 2021) • **(A)** Specific service accounts with least privilege access allowing Login and CRUD services in the DMZ network to communicate to Internal LAN resources. • **(A)** Ports: HTTPS (443). **(L)** Additional external applications and services should support this.
CPU	• **(A)** Leverages scalable and redundant cloud architectures (Chieu, Mohindra, Karve and Segal, 2009).
RAM	• **(A)** Leverages scalable and redundant cloud architecture.
Storage	• **(A)** Leverages scalable and redundant cloud architecture. • Web Server / Cloud Functions (containers) / Databases / File
Database	• **(A)** Leverages scalable and redundant cloud architecture. • **(A)** Access and Network Control DMZ communication. • **(A)** Row level security • **(A)** Restricted Views • **(L)** Database performance will be impacted by inefficient query optimisation.
Encryption Algorithms	• Areas of Focus: Data in use, in transit, and at rest (Markandey, Dhamdhere and Gajmal, 2019). • **(A)** Data in Transit - IPSec VPN. • **(A)** Data at Rest – SHA256 / AES 256 (cloud) / on-premise (BitLocker / ceph). • **(A)** Key Management System . • **(L)** Speed and computational overhead will be impacted by the choice of encryption algorithm.

Table 1: Assumptions and Limitations

System Requirements

Requirement	Method
Access type	Local and Remote
Application Server	Nginx (Pallets, N.D.)
Storage location	US, EU
Internet Protocols	IPv4 and IPv6
Processor	Quad core GHz+ CPU (MySQL, N.D.)
RAM	6 GB (MySQL, N.D.)
Storage size	1 TB
Code language	Python 3 (Python, N.D.)
Database	MySQL (MySQL, N.D.)
Encryption Algorithms	SHA256 - Password Hash (Python, N.D.)
	Symmetric encryption - File Encryption (AES 256) (Cryptography, N.D.)

Table 2: System Design requirements

Functionality

As the ISS is responsible for human life as well as billions of dollars' worth of equipment, the application must be built with cardinal levels of security. The proposed application meets security and privacy requirements, of which some are mandated by the European Commission's GDPR (EUR-Lex, 2016). Many further features could be included, such as higher levels of access control granularity, however a prototype model, they are considered out of scope and so Role Based Access Control (RBAC) has been selected.

The system includes the following automatic capabilities:

- Encryption / Decryption
- Account Lockout
- User Notification (of changes)
- Logging

User capabilities include:

- Registration
- Login
- View Files
- Upload
- Download
- Share Files

Administrator capabilities include:

- Login
- Verify User Registration
- Unlock / Disable User
- Update / Delete User Details

Identified Risks and Vulnerabilities

Identified risks are identified in Figures 2A and 2B below, according to the STRIDE methodology. Special consideration is made the most prevalent threats identified by OWASP (2022).

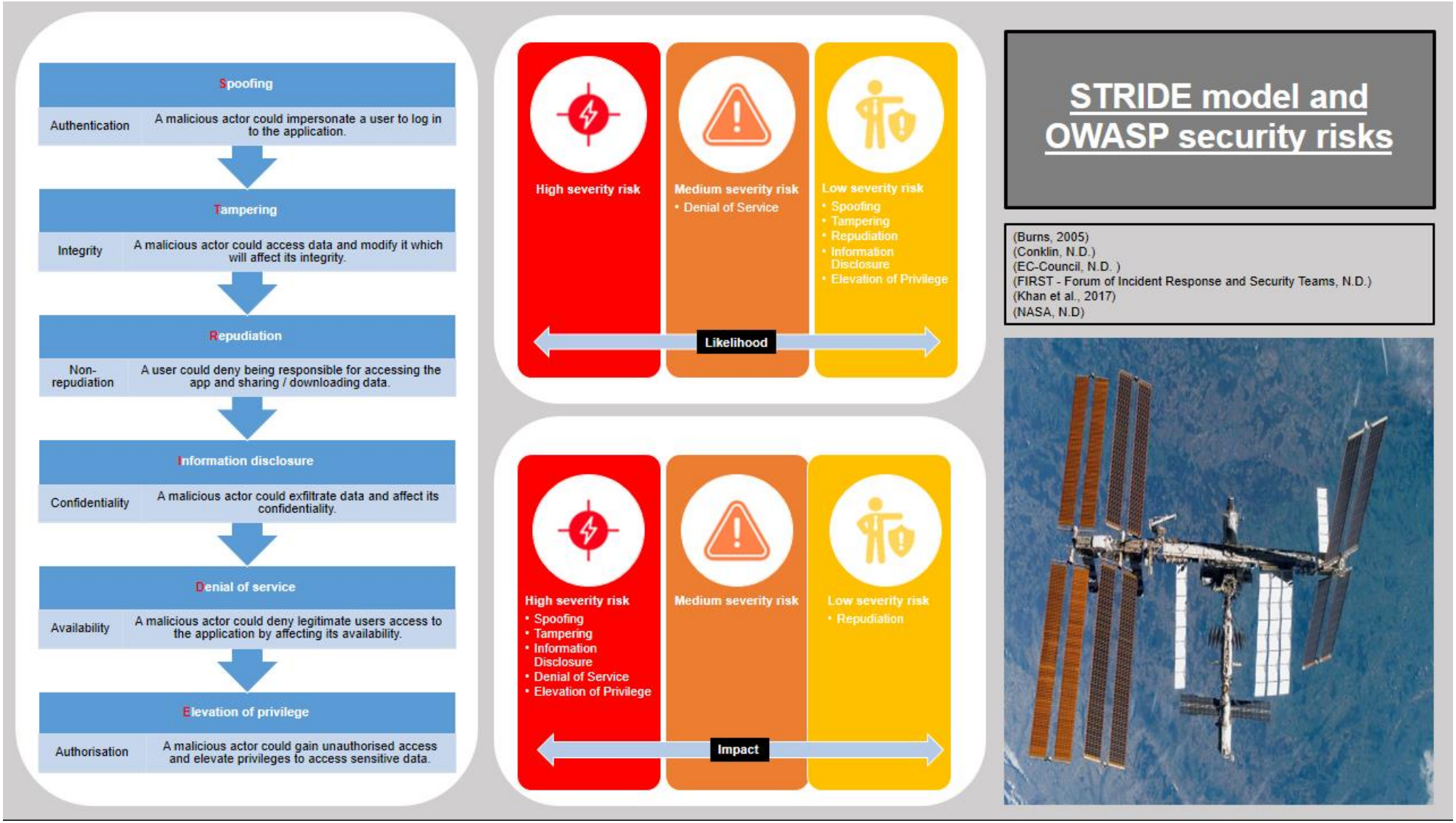

Figure 2A: STRIDE Risks

STRIDE	Threat Example	Mitigations	OWASP Web Application Security Risks
Spoofing	Logging into admin or user account through broken authentication to query database information.	1. Implementation of strong authentication mechanism such as Multi Factor Authentication (MFA).	A2 Broken Authentication
Tampering	Escalation of privilege to tamper with database. Man-in-the-middle attack to manipulate between the app and database. Exploiting vulnerable code to execute remote request and extract sensitive data.	Implementation and enforcement of: 1. Encryption of data at rest and in transit 2. Strong authorisation mechanism to limit access to sensitive data. 3. File hash implementation to validate integrity of data. 4. Safe API 5. Avoid serialisation of sensitive data 6. Only use trusted HTTP requests based on the context in the HTML output.	A1 Injection A4 XML External Entities (XXE) A7 Cross-Site Scripting (XSS)
Repudiation	Malicious user interferes with database information.	1. Implementation of a strong authentication mechanism such as MFA. 2. Ensure logging and integrity controls e.g. append only database tables	A10 Insufficient Logging and Monitoring
Information Disclosure	Malicious user obtains user and/or admin password due to weak encryption.	1. Implementation of strong authentication mechanism such as MFA. 2. Strong authorisation mechanism such as file permission and restriction 3. Encryption of data at rest and in transit. 4. Monitoring and analysis of transmission of data. 5. Store passwords using slated hashing functions e.g. brcypt	A3 Sensitive Data Exposure A6 Security Misconfiguration
Denial of Service	Malicious user such as a privileged insider misconfigures system, modifies code to disrupt operations and communications.	1. Restriction of IP address ranges allowed to access application. Allowed IPs should only be those of ISS and government networks. 2. Geo location restrictions of IP addresses. 3. Implementation of a network firewall to restrict access. 4. Implementation of NIDS/NIPS to detect/block intrusion.	A3 Sensitive Data Exposure A6 Security Misconfiguration A9 Using components with Known Vulnerabilities
Elevation of Privilege	All users given the same privileges and all resources open for access.	Implementation and enforcement of: 1. A strong authentication mechanism such as MFA. 2. A strong authorisation and access restriction. 3. An effective data classification and segregation. 4. Monitoring of file access and operations. 5. Deny by default unless public resources.	A6 Broken Access Control A6 Security Misconfiguration A9 Using components with Known Vulnerabilities

Burns, S. F. (2005). Threat modeling: A process to ensure application security.*GIAC security essentials certification (GSEC) practical assignment*
Conklin, L. (N.D.). OWASP Threat Modeling Process. Available from: https://owasp.org/wwwcommunity/Threat_Modeling_Process [Accessed 22 May 2022].
EC-Council. (N.D.). Threat Modeling | Importance of Threat Modeling. Available from: https://www.eccouncil.org/threatmodeling/ [Accessed 21 May 2022].
First - Forum of Incident Response and Security Teams. (N.D.). Threat Modelling Cyber Threat Intelligence SIG Curriculum. Available from: https://www.first.org/global/sigs/cti/curriculum/threat-modelling [Accessed 21 May 2022].
Khan, R., Mclaughlin, K., Laverty, D. & Sezer, S. STRIDE-based threat modelling for cyber-physical systems. 2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe), 26-29 Sept. 2017. 1-6.

Figure 2B: STRIDE Risks

Tools and Code Libraries

Tools and libraries for the prototype are listed below, and Tables 3A and 3B justifying the selection can be found in the appendix.

Tools:

- Python (Code language)
- MySQL (Database)
- Locust (Load tester)
- PyCharm (Editor)

Code Libraries:

Cryptographic:

1. Hashlib (Password encryption)
2. Cryptography (File encryption)
3. Fernet (Key generation)

Webapp:

1. Flask (Web framework)
2. Flask-login (Session management)
3. Flask-JWT (Web Token)
4. Werkzeug (WSGI utilities)
5. Getpass (Hiding password)
6. Smtplib (SMTP client)

Database connector:

1. Flask-sqlalchemy (SQL toolkit)
2. Pymysql (MySQL client)

Testing tool:

1. Pylint (Syntax checker)
2. Flake8 (Syntax checker)
3. Pytest (Function tester)

Diagrams

Finally, figures 2 to 7 describe interactions between the application and its users:

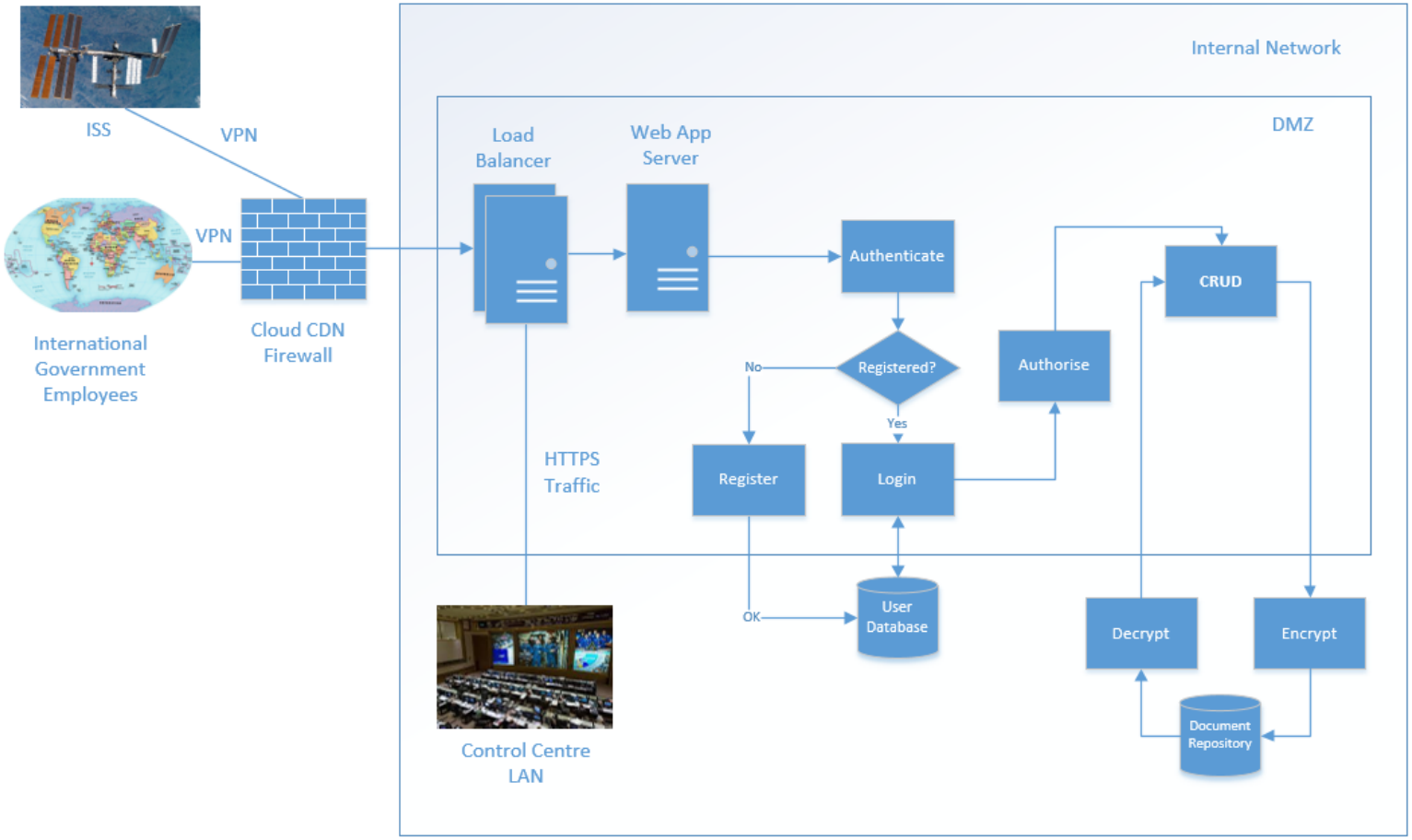

Figure 3: System Architecture (NASA, N.D)

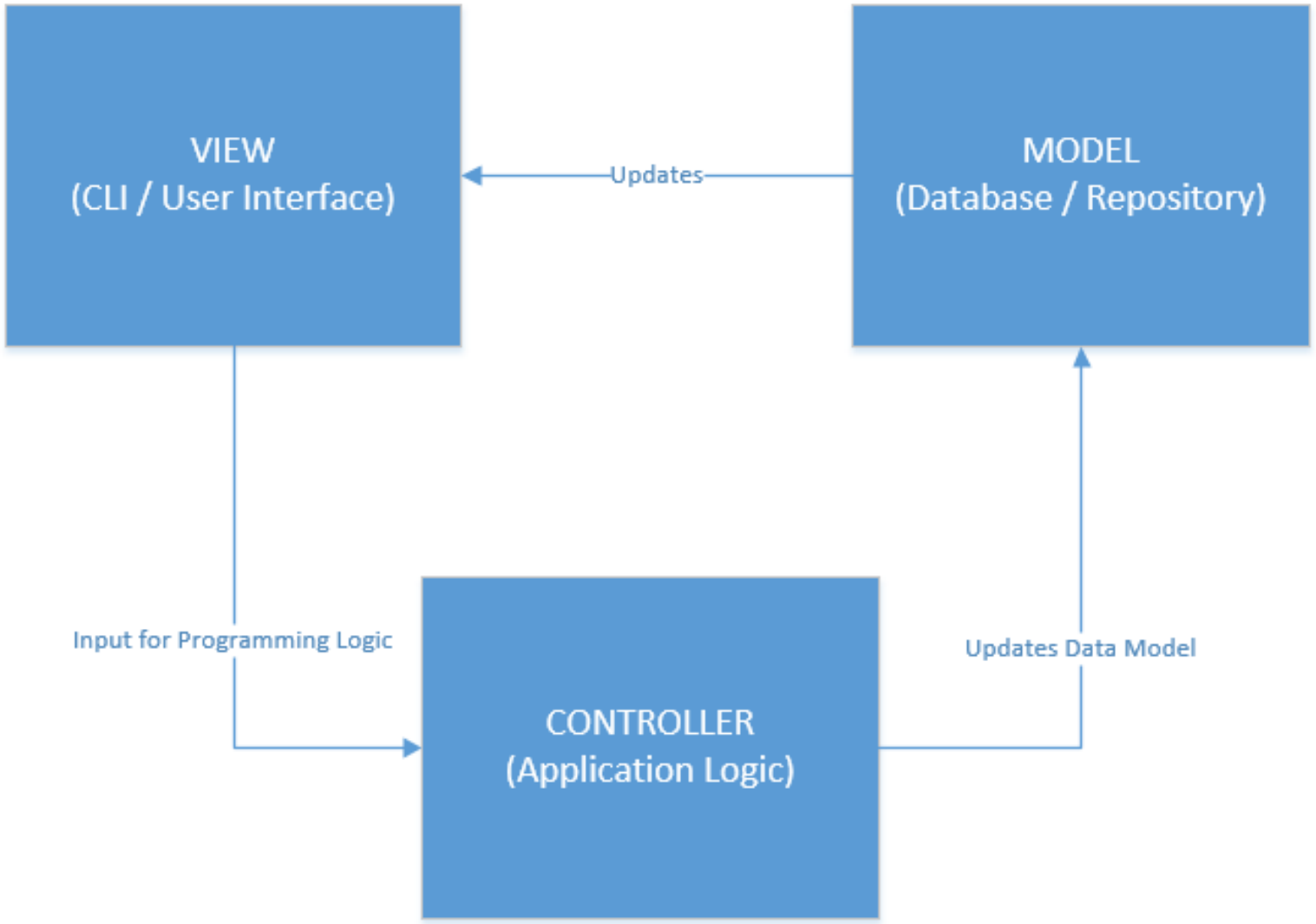

Figure 4: Software Architecture – Model, View, Controller (MVC) (University of Nebraska-Lincoln, N.D.)

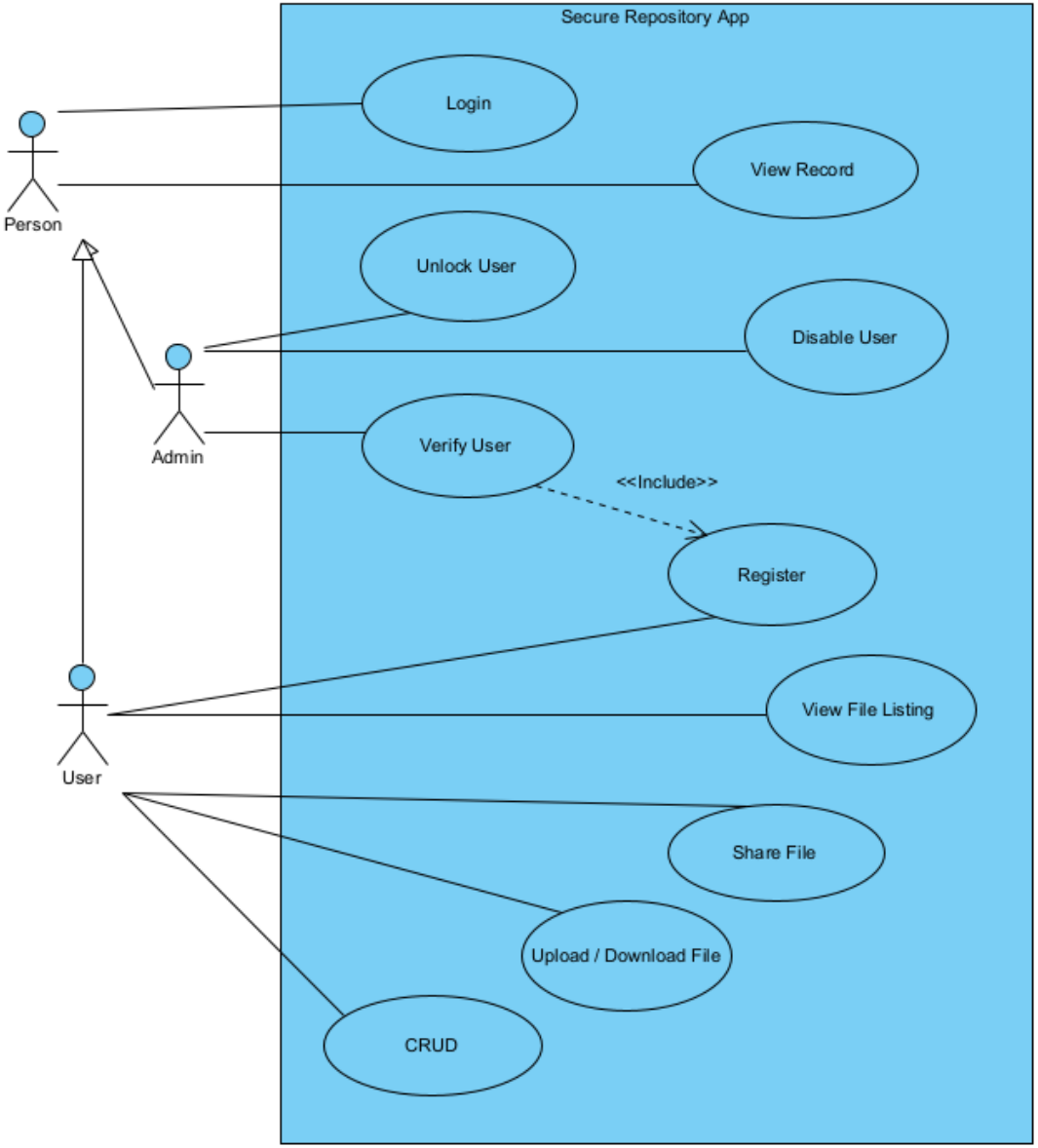

Figure 5: Use Case Diagram

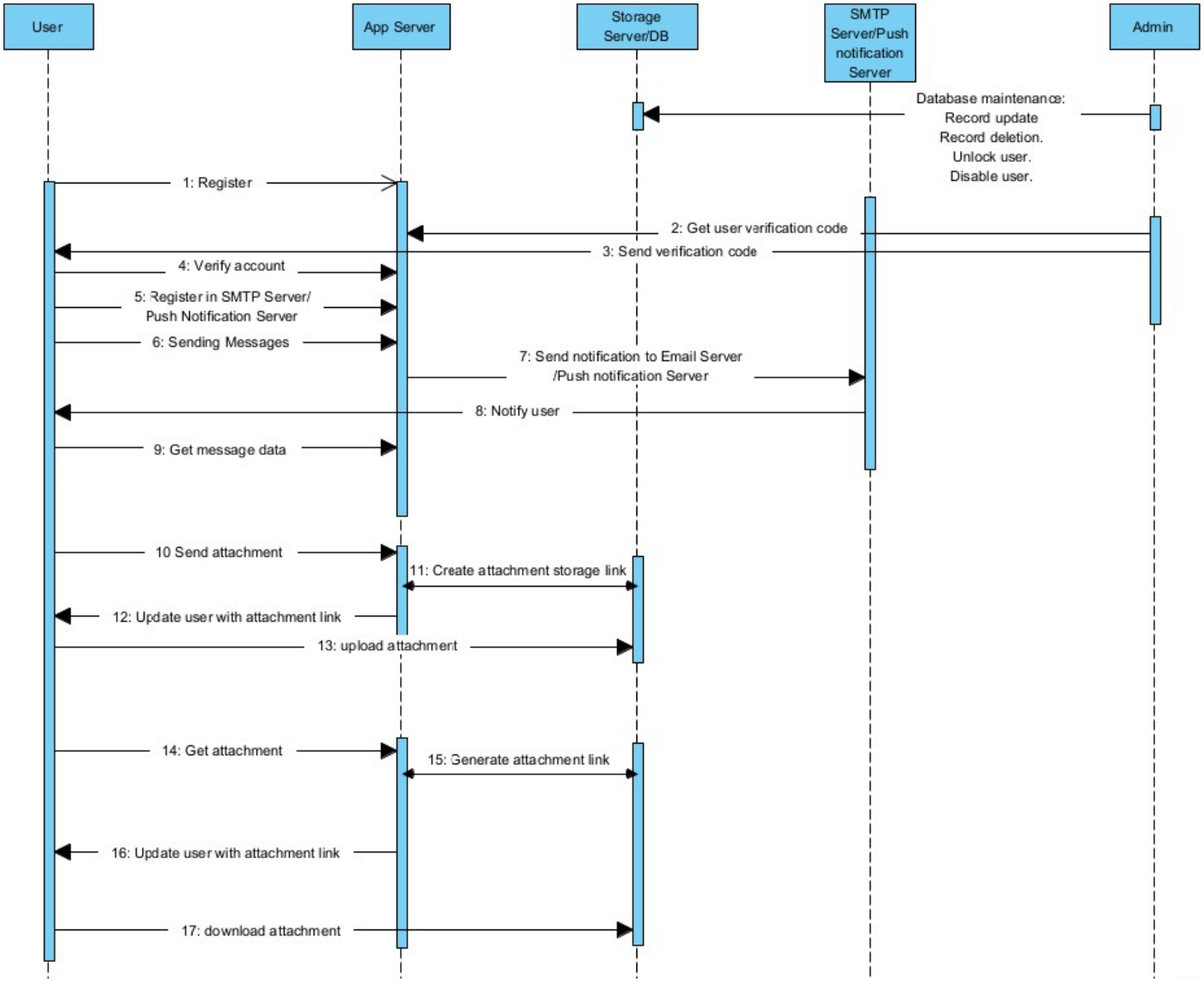

Figure 6: Sequence Diagram

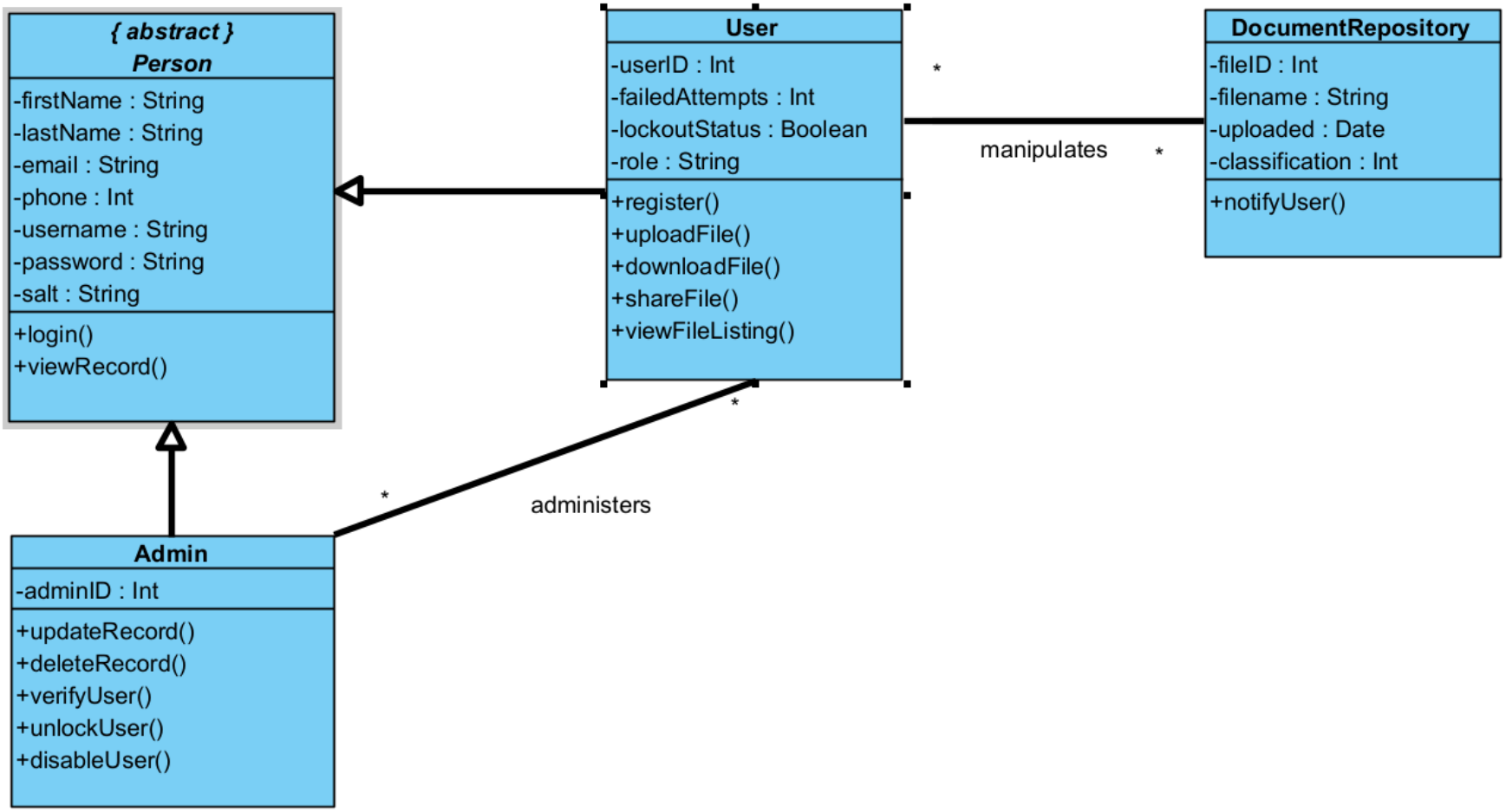

Figure 7: Class Diagram

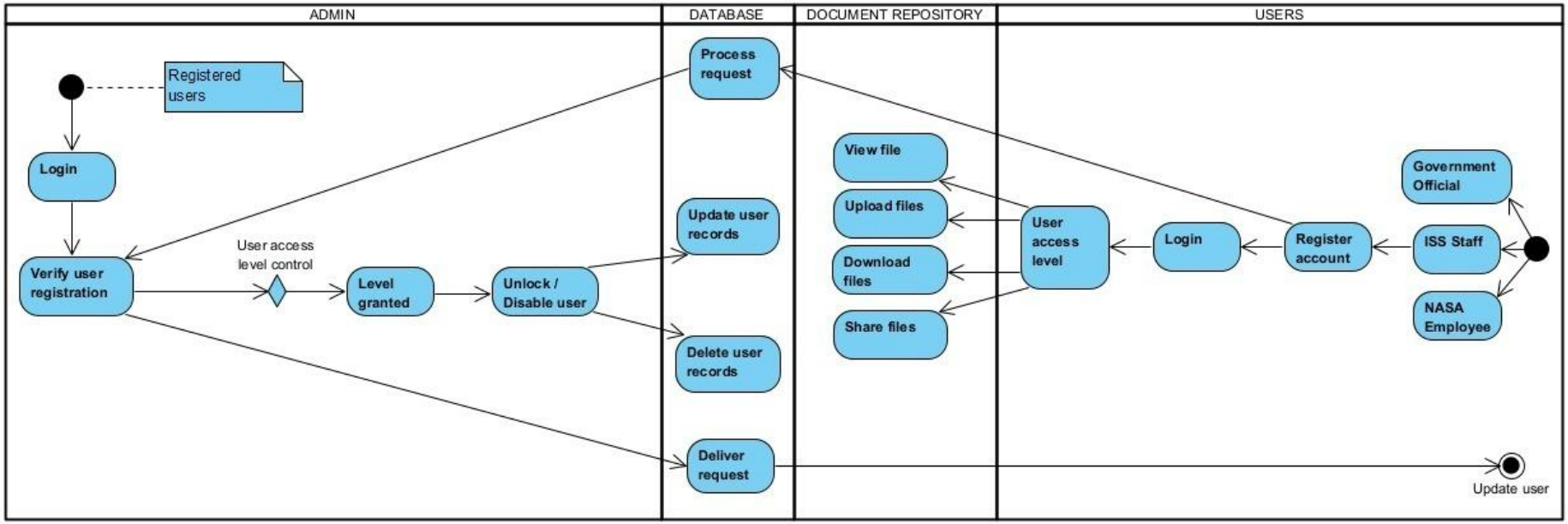

Figure 8: Activity Diagram

References

Chieu, T. et al. (December 1, 2009) Dynamic Scaling of Web Applications in a Virtualized Cloud Computing Environment. *2009 IEEE International Conference on e-Business Engineering (ICEBE)*. Available from: https://ieeexplore.ieee.org/document/5342101 [Accessed 20 May 2022].

Cryptography. (N.D.) Fernet (symmetric encryption). *The Recipes Layer*. Available from: https://cryptography.io/en/latest/fernet/ [Accessed 21 May 2022].

EUR-Lex. (2016) Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). Available from: EUR-Lex - 02016R0679-20160504 - EN - EUR-Lex (europa.eu) [Accessed 20 May 2022].

Flask-Login. (N.D.) How it Works. *Latest*. Available from: https://flask-login.readthedocs.io/en/latest/ [Accessed 21 May 2022].

GeeksforGeeks. (Jan 13, 2021) Encrypt and Decrypt Files using Python. *Python*. Available from: https://www.geeksforgeeks.org/encrypt-and-decrypt-files-using-python/ [Accessed 21 May 2022].

JetBrains. (N.D.) PyCharm Features. *Features*. Available from: https://www.jetbrains.com/pycharm/ [Accessed 21 May 2022].

Locust. (N.D.) What is Locust?. *Locust Documentation*. Available from: https://docs.locust.io/en/stable/ [Accessed 21 May 2022].

Markandey, A. et al. (March 28, 2019) Data Access Security in Cloud Computing: A Review. *2018 International Conference on Computing, Power and Communication Technologies (GUCON)*. Available from: https://ieeexplore.ieee.org/document/8675033 [Accessed 22 May 2022].

Mhaskar, N. et al. (2021) *Computers & Security: A Formal Approach to Network Segmentation*. ScienceDirect. Available from: https://doi.org/10.1016/j.cose.2020.102162 [Accessed 22 May 2022].

MySQL. (N.D.) General Information. MySQL 8.0. *Reference Manual*. Available from: https://dev.mysql.com/doc/refman/8.0/en/introduction.html [Accessed 21 May 2022].

MySQL. (N.D.) Hardware Requirements. Installing and Launching MySQL Workbench. Available from: http://download.nust.na/pub6/mysql/doc/workbench/en/wb-requirements-hardware.html [Accessed 20 May 2022].

NASA. (N.D.) International Space Station. Available from: https://www.nasa.gov/mission_pages/station/main/index.html [Accessed 18 May 2022].

NASA. (2007) Final Report of the International Space Station Independent Safety Task Force. Available from: https://www.nasa.gov/pdf/170368main_IIST_%20Final%20Report.pdf [Accessed 20 May 2022].

OWASP. (2022) OWASP Top Ten. Available from: https://owasp.org/www-project-top-ten [Accessed 23 May 2022].

Pallets. (N.D.) Documentation. *Werkzeug*. Available from: https://werkzeug.palletsprojects.com/en/2.1.x/ [Accessed 21 May 2022].

Pallets. (N.D.) Welcome to Flask. *Contents*. Available from: https://flask.palletsprojects.com/en/2.1.x/ [Accessed 21 May 2022].

PyCQA. (May 20, 2022) Pylint documentation. *Latest*. Available from: https://pylint.pycqa.org/en/latest/ [Accessed 21 May 2022].

PyCQA. (N.D.) Flake8: Your Tool For Style Guide Enforcement. *Docs*. Available from: https://flake8.pycqa.org/en/latest/ [Accessed 21 May 2022].

PyPI. (Jan 9, 2021) PyMySQL 1.0.2. *Project*. Available from: https://pypi.org/project/PyMySQL/ [Accessed 21 May 2022].

Pytest. (N.D.) Pytest: helps you write better programs. *Home*. Available from: https://docs.pytest.org/en/7.1.x/ [Accessed 21 May 2022].

Python. (N.D.) About. *Python*. Available from: https://www.python.org/about/ [Accessed 21 May 2022].

Python. (N.D.) Getpass — Portable password input. *Library*. Available from: https://docs.python.org/3/library/getpass.html [Accessed 21 May 2022].

Python. (N.D.) Hashlib — Secure hashes and message digests. *Library*. Available from: https://docs.python.org/3/library/hashlib.html [Accessed 21 May 2022].

Python. (N.D.) Smtplib — SMTP protocol client. *Internet Protocols and Support*. Available from: https://docs.python.org/3/library/smtplib.html [Accessed 21 May 2022].

Pythonhosted. (N.D.) Documentation. *Flask-JWT*. Available from: https://pythonhosted.org/Flask-JWT/ [Accessed 21 May 2022].

SQLAlchemy. (N.D.) SQLAlchemy's Philosophy. *Home*. Available from: https://www.sqlalchemy.org/ [Accessed 21 May 2022].

Tsochev, G. et al. (December 15, 2021) Analysis of Threats to a University Network Using Open Source Technologies. *2021 International Conference Automatics and Informatics (ICAI)*. Available from: https://ieeexplore.ieee.org/document/9639729 [Accessed 20 May 2022].

University of Nebraska-Lincoln. (N.D.) Tiered Architecture and MVC. Available from: https://its.unl.edu/bestpractices/tiered-architecture-and-mvc [Accessed 22 May 2022].

V Casteren, W. (2017) The Waterfall Model and Agile Methodologies :

A comparison by project characteristics. Available from:

https://moam.info/the-waterfall-model-and-agile-methodologies-a-_5b87344a097c4770628b474d.html [Accessed 20 May 2022].

WhiteSource. (N.D.) What Are The Most Secure Programming Languages?. *Research Reports*. Available from: https://www.whitesourcesoftware.com/most-secure-programming-languages/ [Accessed 21 May 2022].

Appendix

Tools	Name	Purpose	Justification
Misc.	Python	Code language	User-friendly programming language Code reuse via libraries The viable choice for quick prototyping in web development Support garbage collection (Python, N.D.)
	MySQL	Database	ANSI SQL-compliant Structured Query Language (SQL) database Minimal operating storage space Sufficient for storing passwords and data for the web app Simple to use and set up (MySQL, N.D.)
	Locust	Load tester	Popular Load testing and user behaviour simulation program written in Python Faster and easier to configure Leaner and does not require bulky configuration files (Locust, N.D.)
	PyCharm	Editor	PyCharm is designed for the Python programming environment Git integration Clearly debug properties (JetBrains, N.D.)

Table 3A: Software Justification

Libraries Type	Name	Purpose	Justification
Cryptographic	Hashlib	Password encryption	Support hash algorithms One way hashing method with SHA-256 and salt (Python, N.D.)
	Cryptography	File encryption	Support symmetric algorithm Data encryption and decryption (GeeksforGeeks, Jan 13, 2021)
	Fernet	Key generation	Methods for generating keys Converting plaintext to ciphertext (Cryptography, N.D.)
Webapp	Flask	Web framework	A flexible framework for developers to demonstrate the functions of the prototype

			Micro-based framework and is simple to extend (Pallets, N.D.)
	Flask-login	Session management	Able to save the active user's session Assist in preventing cookie thieves from stealing sessions Allow logged-in users to see the content (Flask-Login, N.D.)
	Flask-jwt	Web Token	Protecting routes to transmit data Revoking and block listing token Cross-Site Request Forgery Prevention protection (Pythonhosted, N.D.)
	Werkzeug	WSGI utilities	Interacts with headers, query args, form data, files, and cookies Capture variables from URLs HTTP utilities are provided (Pallets, N.D.)
	Getpass	Hiding password	Without echoing the password May delivered warning when password entry is echoed (Python, N.D.)
	Smtplib	SMTP client	Support secure connection Send out mail for Multi Factor Authentication (Python, N.D.)
Database connector	Flask-sqlalchemy	SQL toolkit	Modified into a simple Pythonic language for database access Automate unnecessary processes to avoid SQL injection (SQLAlchemy, N.D.)
	Pymysql	MySQL client	Pure-Python MySQL client library (PyPI, Jan 9, 2021)
Testing tool	Pylint	Syntax checker	Checks for errors and sticks to the code standard Analyses code without executing it (PyCQA, May 20, 2022)
	Flake8	Syntax checker	Checks for errors and sticks to the code standard Simpler, quicker, and false positives are lowered (PyCQA, N.D.)
	Pytest	Function tester	Develop simple, understandable tests Never been linked to a security vulnerability (Pytest, N.D.)

Table 3B: Code Libraries Justification