Development Team Project

International Space Station: System Design Proposal

MSc Cyber Security

Secure Software Design, Unit 3 Submission

Words 639

[**Introduction**](#_heading=h.p267c9ggr3x8) **2**

[**System Requirements**](#_heading=h.tq1qyxgtgvrb) **2**

[Required](#_heading=h.2v37e3jwekn3) 2

[Nice-To-Have](#_heading=h.hqb7ztmjelbo) 2

[**System Design**](#_heading=h.23lqlbvo9575) **3**

[Image 1.](#_heading=h.r8ok22dna6vh) 3

[Image 2.](#_heading=h.x3fbtg3k801) 4

[Open-source libraries](#_heading=h.6d3z7d3k138w) 4

[**Challenges & Assumptions**](#_heading=h.78txp9zu4gs) **5**

[**Conclusion**](#_heading=h.d47d0wn4c9lk) **5**

[**References**](#_heading=h.4hliqa6awh5u) **5**

## Introduction

The document below provides an overview of a proposed system for the International Space Station (ISS). It includes system requirements based on domain and mandated requirements, a high-level system design and the challenges and assumptions made. This also includes in detail the patterns and approaches to said challenges, and a list of proposed tools and libraries in creating the new system.

## System Requirements

The ISS domain has some key points to consider, including that the data transmitted can be time-sensitive and mission-critical (Jacobson & Hume, 2016). The data is also of a high quantity, and can be sent via various NASA centres (Peters, 2019). It is also known that NASA uses Amazon Web Services to deliver data worldwide (Savaram, 2017). The above have been considered when documenting the below requirements.

### Required

* File upload, download, sharing and search functionality.
* Role-Based Access Control (RBAC)
* GDPR compliant
* Data classification levels in line with government - Top Secret, Secret & Confidential
* Encryption of data at rest and in transit
* Secure user authentication utilising password strength control and secure storage
* Secure Session Management utilizing time-based one-time passwords (TOTP)
* DDoS mitigation
* Logging and monitoring of user actions
* Input validation, including CSRF token
* Secure error handling
* Secure key storage

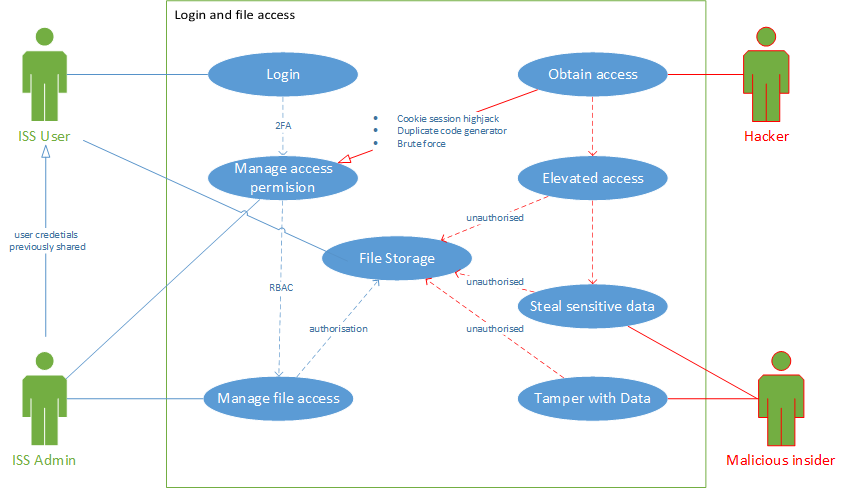
### Nice-To-Have

* Secure password recovery mechanism
* Re-authentication for privileged access
* Two-factor authentication (2FA)

## System Design

The following UML diagrams have been developed to show the system design.

First is an Abuse Case diagram, showing potential threats to the system which will be considered in the design.



### Image 1

The second is a Sequence diagram that lists all steps, including the authentication and multiple authorisation checks necessary for a user to access data.

### Image 2

### Open-Source Libraries

* Python
* Flask
* SQLAlchemy
* Werkzeug
* Jinja
* Hashlib
* Sqlite
* ItsDangerous
* MarkupSafe
* HTML
* CSS
* Visual Studio Code
* GitHub

Python 3 will be used to create the application due to its versatility, automatic memory management and abundant documentation (Pillai, 2017). The intention is to use Flask as it employs a lightweight Model Template View (MTV) architecture. The team will also ensure the cryptographic hashes are checked during download to reduce the risks associated with using open source libraries.

The key stages of the Software Development Lifecycle (SDLC) will be followed to ensure a quality product is produced. In addition to employing secure coding techniques, such as input validation (Centre for Secure Design, 2014), security will be shifted left through security testing the code early and often (Mansfield-Devine, 2018). This approach will reduce the need for rework. Furthermore, functional testing will be employed to ensure the product works as required to fulfil intended use cases (Pillai, 2017).

A combination of The Open Group Architecture Framework (TOGAF) and the Sherwood Applied Business Security Architecture (SABSA) will be used to also provide a holistic enterprise architecture that covers risk and opportunity-driven security architecture alongside business strategy (The Open Group, 2011).

## Challenges & Assumptions

The main challenge faced is the short timeline. To resolve, an adaptive project management approach is required (Abrahamsson, 2003). The team has chosen to adopt Secure Scrum, based on the flexible agile framework, Scrum, but has considerations for security activities built-in (Pohl & Hof, 2015).

The main assumption made is that Flask is better suited for the creation of this solution. Whilst Django was considered, Flask creates opportunities to extend functionality using plugins, including SQL Alchemy. Whilst this application is intended to be monolithic to meet requirements, Flask would ensure flexibility to adapt this model to a microservices architecture allowing for scalability.

## Conclusion

The document provides an overview of the proposed system for the ISS, including a list of system requirements, a high-level system design via multiple UML diagrams, potential challenges faced and assumptions made. It also includes a comprehensive list of the patterns and approaches taken, and a list of tools and libraries to be used. All will be utilised in the creation of said system for the ISS.

## 

## References

Abrahamsson, P., Warsta, J., Siponen, M. & Ronkainen, J. (2003) New Directions on Agile Methods: A Comparative Analysis. Proceedings of the 25th International Conference on Software Engineering (ICSE '03) 1(1): 244 - 254.

Centre for Secure Design (2014) *Avoiding the top 10 software security design flaws.* IEEE Computer Society.

Jacobson, A. & Hume, A. (2016) NASA communications network to double space station data rates. Available at:<https://phys.org/pdf400853634.pdf> [Accessed 24 August 2021].

Mansfield-Devine, S. (2018) DevOps: finding room for security, *Network Security* 2018(7): 15-20.

Peters, D. (2019) Data Rate Increase on the International Space Station Supports Future Exploration. Available at:<https://www.nasa.gov/feature/goddard/2019/data-rate-increase-on-the-international-space-station-supports-future-exploration> [Accessed 24 August 2021].

Pillai, A. (2017) *Software Architecture with Python*. Birmingham, UK. Packt Publishing Ltd.

Pohl, C. & Hof, H. (2015) *Secure Scrum: Development of Secure Software with Scrum*, in Proc. of the 9th International Conference on Emerging Security Information, Systems and Technologies.

Savaram, R. (2017) What Is NASA Doing With Big Data? Available at:<http://highscalability.com/blog/2017/7/5/what-is-nasa-doing-with-big-data-check-this-out.html> [Accessed 24 August 2021].

The Open Group (2011) TOGAF and SABSA Integration: How SABSA and TOGAF complement each other to create better architectures. Available at: <https://publications.opengroup.org/w117> [Accessed 26 August 2021].

Wells, S. (2020) Protecting ISS. Available at:<https://aerospaceamerica.aiaa.org/features/protecting-iss/> [Accessed 25 August 2021].