**Software Requirements Specification Summary**

**Requirements:**

- The system must provide students with the ability to view a comprehensive list of all their enrolled modules.

- Students must be able to search for modules using criteria such as course codes, title, or department

- Students must be able to register and deregister for modules.

- Administrators should be able to manage modules, student lectures, and other system settings within the application through an admin dashboard.

- Prospective students can submit applications for student status. Administrators are responsible for reviewing these applications and approving or denying them accordingly.

**Security Requirements:**

- Implement secure authentication mechanisms (e.g., password hashing, encryption) to protect user credentials.

- Implement a password hashing abstraction layer to ensure the system can adapt to future password requirements while supporting legacy passwords, in accordance with security by design principles. (Find reference for this)

- Encrypt sensitive data (e.g., user information, module registrations) both in transit and at rest.

- Utilize the AES/CBC/PKCS5Padding encryption algorithm with a 128-bit key to implement column-level encryption for sensitive data (names, contact details etc) stored in the database. (Find reference for this)

- Implement role-based access control (RBAC) to restrict access to system functionalities based on user roles [ADMIN/MODULE\_ADMIN/STUDENT]. (find reference for this)

- Use HTTPS protocol to encrypt data transmission between the client and the server to prevent data interception and tampering.

- Implement input validation mechanisms to prevent common security vulnerabilities such as SQL injection and cross-site scripting (XSS) attacks. (find reference for this)

- Implement logging mechanisms to track user activities and system events for audit and forensic analysis. (NGINX)

- Implement micro-segmentation within Docker containers to enhance network security by isolating and segmenting communication between containers (find reference for this)

**Technologies:**

* Vue.js (frontend framework)
* Nginx (proxy)
* Spring Boot (
* Postgres/Neon
* Docker
* AWS EC2