Design of Web Application for British University

Prepared by

J110585

Table of Contents

Introduction

This report is focused on the creation and development of a web application as a potential project that can be used by students in universities. In this report many designs are use such as UML Case, Class and Sequence diagrams that shows how users interact with software, how the different aspects of the software interact with each other and their relationships and how it interacts on an architecture level. The technology stack chosen to design and host such an application must prioritize security, considering the sensitivity of academic data. Technologies such as secure cloud services provide robust infrastructure and advanced security features that align with best practices and compliance standards, including GDPR. The use of containers, for instance, can encapsulate specific functionalities, allowing for secure and isolated execution of services. The technology stack chosen to design and host such an application must prioritize security, considering the sensitivity of academic data. Technologies such as secure cloud services provide strong infrastructure and advanced security features that is done with best practices and compliance standards, including GDPR. The use of specific functionalities, allowed for secure and isolated execution of services. This report is also stakeholder's interest and power, secure issues and how they can be solved.

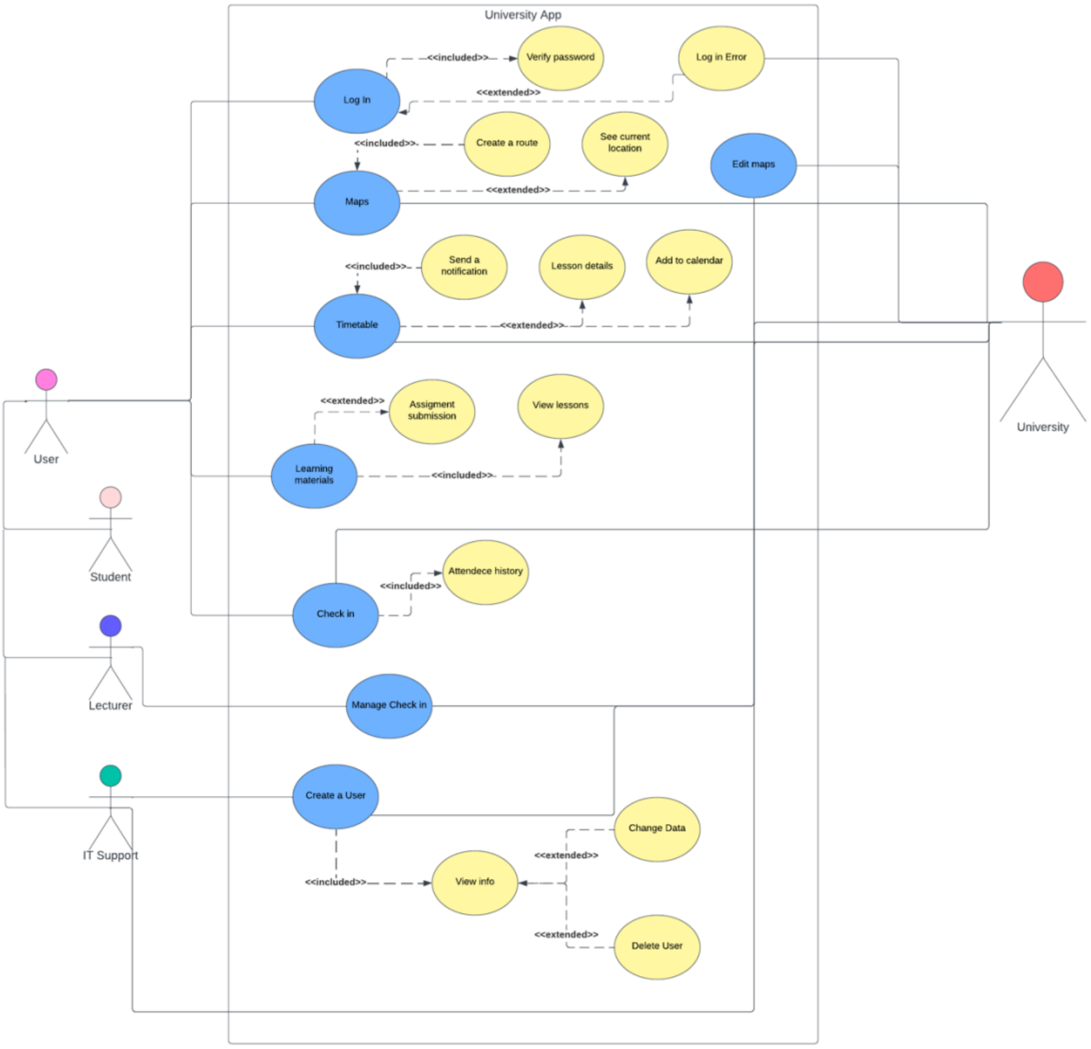
User interaction with software though Use Case Diagram

In this UML Use Case diagram, it is showed how does user interact with this piece of software, but without fully detailed processes. The general User actor can Log in to the system. If it is unsuccessful, it encounters a Log in Error. After authorisation user can interact with Maps to see their current location, create routes, and edit maps. They can also interact with the Timetable to view lesson details, send notifications, and add lessons to the calendar.

Student actor inherits all from User, but also can access to Learning Materials to view resources, check in to the lessons, engage with attendance history and mange assignments.

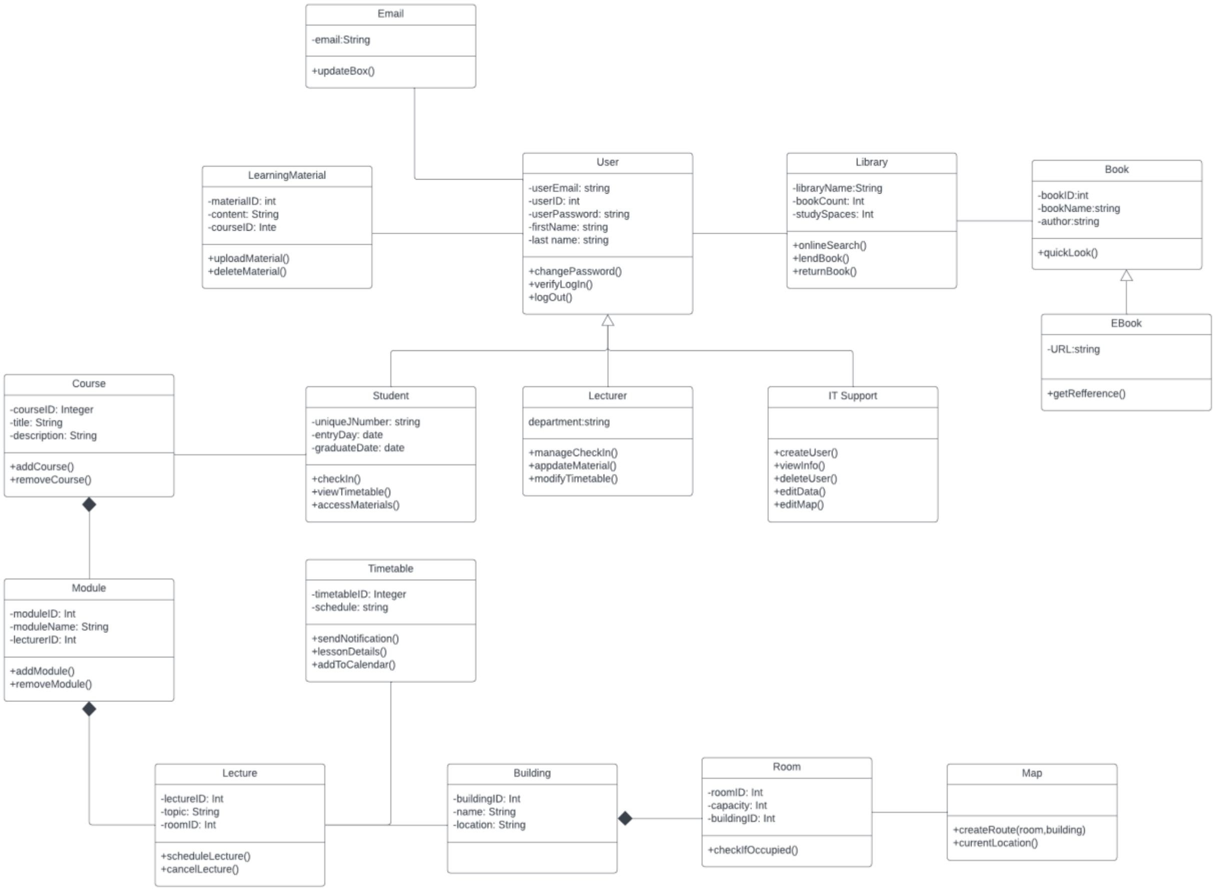
Lecturers can additionally manage student's check-in for a lesson.

IT Support can create a user view all information and change user details.



Interaction with software using UML Class diagram

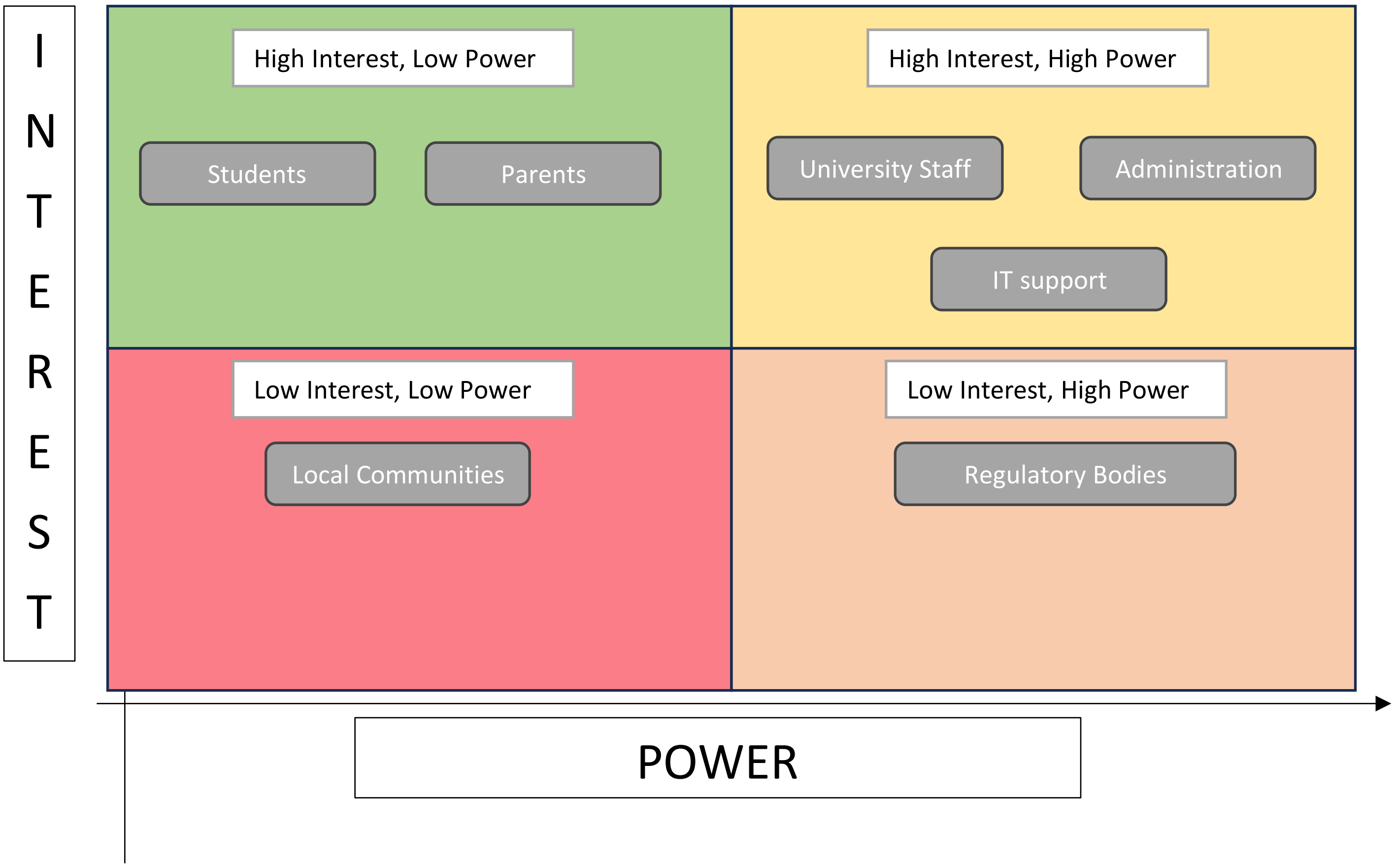
User class is the central class with attributes for user information and methods used for account management. It is a class from which user roles such as Student, Lecturer, and IT Support inherit attributes and methods. Student class inherits from User and includes attributes applicable to students. There methods for academic interactions like checking in to lessons and accessing materials. Lecturer class also inherits from User, with added functionality to manage check-ins and modify timetables.IT Support class inherits from User and includes methods for updating user accounts and system information, helping to maintain the system. Learning Material class relates to Course and Module, learning materials are associated with specific courses and modules. It includes methods to upload and delete data, also lecturers will interact with this class. Course and Module classes contain information about courses and modules, used by students and lecturers to manage course content. Timetable class attached to the Student and Lecturer classes, it would manage scheduling information, allowing users to add lessons to calendars and send notifications. Library and Book are connected to the User and include methods for searching the library catalogue. The eBook subclass indicates, electronic books, with attributes and methods through specific content. Building and Room are physical existing classes, these might interact with the Timetable and Map classes to manage room assignments and provide location information. Map includes methods that suggest functionality for route creation and location services within the campus. Email an utility class used across the system for sending notifications and updates, connected to the User. All the arrows between classes represent relationships, which can be inheritance or association. Black rectangle with line represents composition relationships. (Sommerville, 2019)



Stakeholder analysis and level of interest

Effective management of stakeholder involves identifying the needs and expectations of each group. Key people in the process require more attention and involvement in the process, where groups with less power or interest need different approaches such as regular updates or consultations or even questionaries to ensure their needs are fulfilled and that they have all information. High Interest, Low Power: Students are benefit from the services that university provides. While they have a high interest in education, their direct power is low. They should consult to ensure their needs and feedback are considered. Parents are like students; parents have high interest but limited power to make changes. High Interest, High Power: University Staff members have significant power to influence university operations and a high interest in its' success. Administration group has both the interest and the power to make decisions and should be involved in the planning and proceeding projects. IT Support is essential to the university systems, IT support has a huge influence on the university. Low Interest, Low Power: Local Communities especially their power and interest might be low, it is important to monitor this group and keep them informed to have good public relations and community involvement. Low Interest, High Power: Regulatory Bodies have the power to enforce regulations and standards but might have a low interest in process. (Lloyd, 2014)

Mendelow's Matrix for Stakeholders



High Interest, Low Power

Students

Parents

High Interest, High Power

University Staff

Administration

IT support

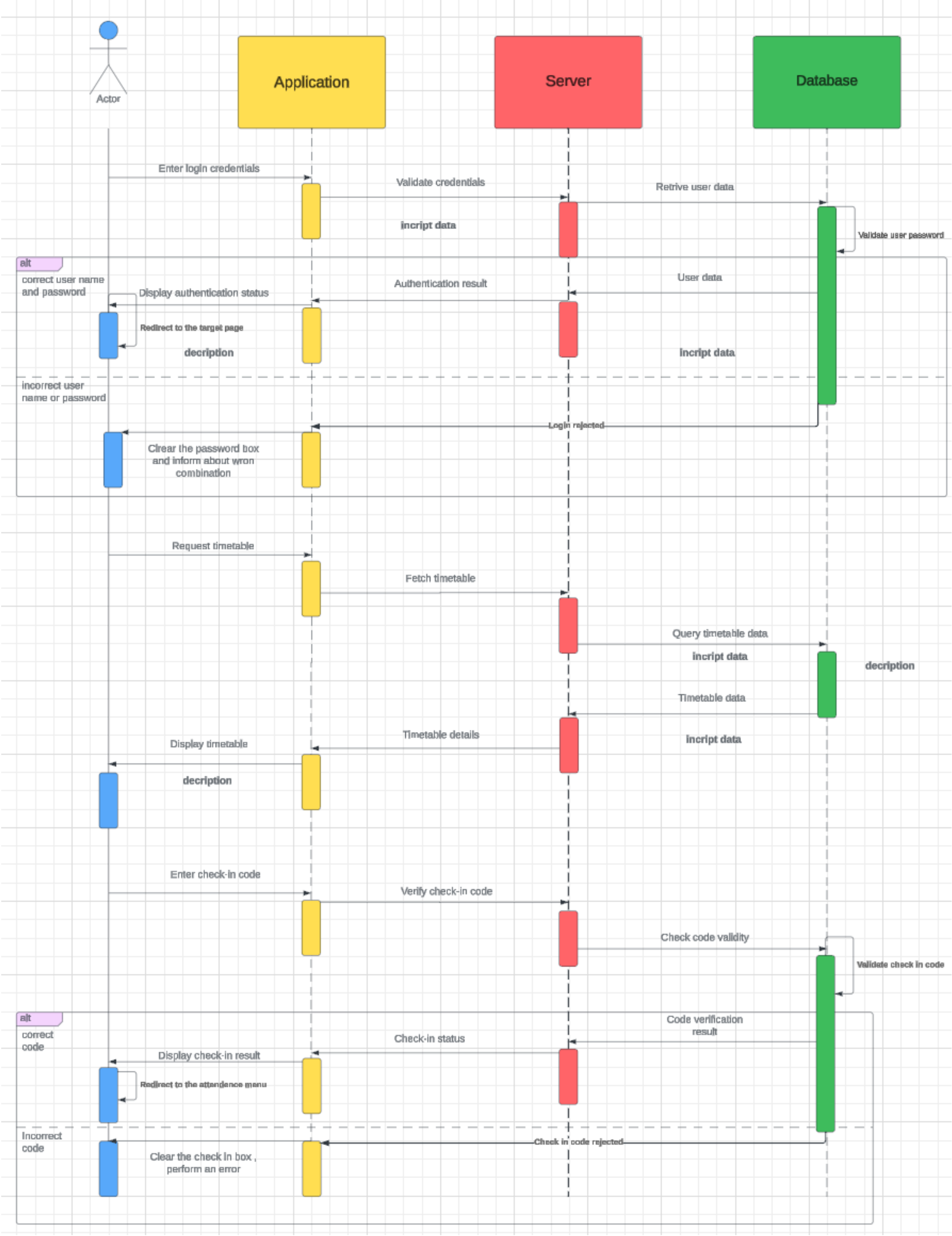
Low Interest, Low Power

Local Communities

Low Interest, High Power

Regulatory Bodies

UML Sequence Diagram and interaction on architecture level



Security for web application

It will include isolating and securing the database, encoding data, enforcing data validation tools, managing access control, logging and system auditing, authentication. To make this application secure it is needed to minimalize data by collecting only data that is necessary for the specified purposes as was shown in GDPR. Encryption is pivotal in this case as implementing strong encryption for data personal information protection is always needed. (Lloyd, 2014) It is needed to make sure that there is no unauthorised access from individuals, and they don't have an access to specific sets of data. Regular updates of software will help to keep application up to date with latest security to protect from hacker attacks. Also documented and tested plan for responses on data breaches is essential and required by GDPR and other data protection laws. (Brewer, 2013).

Risk Table

Hazard	**Who Might Be Harmed and How?**	**Existing Controls**	**Further Action Required**	**Responsible Party**	**Risk Measure**
Data Breach	Students and staff through loss of personal data	Encrypted data storage, regular security training	Implement multi-factor authentication	IT Security Team	High
System Downtime	Students and faculty via disruption to learning and teaching	Redundant system architecture, regular backups	Establish a more robust incident response plan	IT Operations Manager	Medium
Non-compliance with GDPR	University could face penalties; users' data rights could be infringed	GDPR training for staff, data protection policies in place	Conduct a GDPR compliance audit	Data Protection Officer	Normal
Inadequate User Training	Users could misuse the system leading to errors	User manuals, periodic training sessions	Develop an ongoing training program	User Training Coordinator	High
Software Bugs	Students and faculty could experience disruptions	Use of automated testing, regular updates	Increase code review frequency	Development Team Lead	High
Scalability Issues	Students might experience slow system response times during peak usage	Cloud-based services with auto-scaling	Perform load testing and enhance resources	System Architect	Medium

Conclusion

In conclusion, this report has successfully showed the design and development of a software web application aimed at improving the university experience for students. Through UML diagrams, we have visualized user interactions and software relationships, performing a user-centred design. By focusing on a security of given technology, using GDPR standards, the importance of safety of sensitive academic data is demonstrated. Through Mendelow's Matrix all the stakeholders need by their interest and power in project is fully performed in stakeholder analysis, so all their requirements will be included. The comprehensive security, including data encryption and access control done with all GDPR standards, is pivotal in this project and is implemented against potential cyber threats. All risks are needed to be managed as it is a potential affect for the user experience and for their data. Therefore, our team is doing their best to reduce bugs and make improvements. This project is an excellent representation of future university app for students which will help them to fulfil all their needs and will be suitable for all kinds of users.

References

Sommerville, I. (2016). Software Engineering (10th ed.). Essex, England: Pearson Education.

Sommerville, I. (2019). Engineering Software Products (1st ed.). Hoboken, NJ: Pearson.

Pressman, R. S. (2015). Software Engineering: A Practitioner's Approach. New York: McGraw-Hill Education.

Lloyd, I. J. (2014). Information Technology Law. Oxford: Oxford University Press.

Ransome, J. F., & Misra, S. (2014). Core Software Security: Security at the Source. CRC Press.

Brewer, D. (2013). An Introduction to ISO/IEC 27001:2013. BSI British Standards Institution.