**NUA**

Student Portal

Joshua Leivers | 6033

28275

Contents

[1 Analysis 2](#_Toc14424574)

[1.1 Introduction 2](#_Toc14424575)

[1.2 Existing Systems 2](#_Toc14424576)

[1.2.1 Moodle 2](#_Toc14424577)

[1.2.2 SIMS 3](#_Toc14424578)

[1.3 End Users 4](#_Toc14424579)

[1.4 Objectives 4](#_Toc14424580)

[1.5 Constraints and Limitations 6](#_Toc14424581)

[1.6 Proposed Solution 6](#_Toc14424582)

[1.7 Evidence of Analysis 6](#_Toc14424583)

[1.7.1 Interviews 6](#_Toc14424584)

[1.7.2 Questionnaires 7](#_Toc14424585)

[1.7.3 Online Research 9](#_Toc14424586)

# 1 Analysis

## 1.1 Introduction

A student portal is an online gateway and system for providing students of a school, college or university with access to important academic information, resources and statistics (such as grades, attendance, announcements, etc.), software for download and use, links to other academic websites, and useful webpages.

Such a portal would usually use an existing login system/database, as well as linking in with the registration/academic records of the school, but this may conflict with GDPR or the Data Protection Act.

The system behind a student portal would use a database of connected tables, a web application and a script/program of some kind to either link into existing academic databases/information, or to export information from the existing records and import them into the student portal’s database.

## 1.2 Existing Systems

NUAST currently uses Moodle, SIMS, Office 365 and a website. SIMS and Office 365, as well as the school network/computers, are connected to shared resources with the Nova Education Trust, which the academy is a part of, meaning the trust’s policies apply.

Except the website, which only provides generic information and forms to fill in, and does not require logins for anything, all of the school’s systems share a login system. This login system is Microsoft’s Active Directory, and while it is technically possible for my project to use this system for users to log in, the school/trust’s policies do not allow this.

### 1.2.1 Moodle

Moodle allows teachers to set up “courses” which they can add students to, where the teachers can upload links and files for students in the course to download. This covers part of what a student portal should do, and so my project should focus more on the other aspects of it.



Figure .2.1.1 Moodle Course Overview

Moodle also allows teachers to add announcements, news and events, however the implementation of this is not effective, as each individual course has an announcements section and users are not notified in any way if there are new announcements.



Figure 1.2.1.2 Moodle Course Example

### 1.2.2 SIMS

SIMS is the registration system used by the academy, which records student attendance, detentions and academic information – including student timetables. For anything relating to these statistics and information, my project will probably need to link into or work with SIMS in some way. Staff mainly use PARS for interacting with SIMS, but SIMS is the system that controls everything and that my project would need to interact with.



Figure 1.2.2.1 SIMS Interface

SIMS is primarily an information management system, based on a large database containing student information. My project ideally would link with this database, rather than interact with SIMS – as this would easily allow my system to access up-to-date info such as a student’s attendance without taxing resources. However, SIMS’ database operation is subject to change between updates, and is deliberately not easily accessible. Instead, SIMS is able to automatically generate reports, so I may be able to either schedule reports to be exported, which can then be accessed by my system, or set up a script that does this instead.



Figure 1.2.2.2 SIMS Report Options

Linking into SIMS would require high standards of data protection and cyber security, as confidential information is stored in SIMS, and my system should be secure and only allow users access to information they are allowed to access.

The school/trust is currently planning on changing to a different information management system next year, which likely will mean a different database structure and report system. This would require certain parts of my project to be changed if the school implements it in the future, specifically anything that requires access to the database or SIMS reports, but the existing features would almost certainly be available within the new system, and would only need small query changes and/or new report scheduling (as well as report handling).

## 1.3 End Users

There are varying end users for this project. The main end user would be the students, who the majority of systems within the project would be aimed towards. However, there would be other end users, these being teachers and other school and trust staff. These users would be able to access different tools and parts of the application depending on their roles.

For example, a computer science teacher who is also part of the safeguarding team would be able to edit and enter grades for computer science students, and would also be able to access tools available to the safeguarding team (e.g. if students can leave an anonymous message, safeguarding staff could reveal who left the message when necessary), but would not be able to view their own attendance, as it would not be recorded.

## 1.4 Objectives

1. Allow students to access their own attendance information, timetables and grades from SIMS, up-to-date within a reasonable time frame. This information should be exported from SIMS as reports automatically every 24 hours and transferred to appropriate databases within my system.
2. Keep high standards of data protection, allowing students to access only their own or generic information that is appropriate for their use. No student or staff member should be able to access information or data that they are not otherwise able to access through existing systems or are already entitled to be able to access but without systems to do so.
3. Allow students to receive notifications and announcements through the web app from appropriate teachers and staff.
4. Allow teachers and staff to send notifications and announcements through the web app to appropriate students.
5. Each student, teacher and staff member that is allowed access to the system is to have their own account, with permissions tailored to their roles that allow them access to appropriate tools and information within the system.
6. Accounts within the system are to be set up by an account creation system in which new users enter their school login username/email, and their school email is sent a confirmation email for a new account. Upon confirming this new account, users are prompted to set a secure password for the account.
7. Login information and other appropriate inputs are to be handled with appropriate protection. Passwords must be salted and then hashed using a secure hashing algorithm, and other information is to be handled using appropriate hashing, encryption or other secure handling.
8. Each account created is to have a randomly generated salt - of appropriate length to be secure in its use - assigned to it and used for password salting.
9. Users must be required to re-enter their password to access secure settings such as changing their password or other account security options.
10. Users must be able to add two-factor authentication to better secure their accounts. This is an option that users can enable or disable.
11. Student users must be able to anonymously or non-anonymously report issues and using the system.
12. Student users must be able to write reviews and/or rate teachers, subjects, facilities and other aspects of the school. Where appropriate, students must be able to submit these anonymously as an option.
13. Where students are provided anonymity in reports, reviews, ratings, and other parts of the system, only safeguarding staff members must be able to reveal the identity of the student, in the event of a comment causing or revealing a safeguarding issue where this identity is required for further investigation or action. Staff with these permissions attempting to access the identities of anonymous students must be prompted to confirm their reasons for doing so and to enter a reason, that will be stored within the system, as well as to confirm their password to be able to access these identities.
14. A record must be stored of all the times a staff member has accessed an anonymous student’s identity, from which report/review/etc. it was accessed, which staff member did so, when, and what reason they gave. These records must be available to staff investigating inappropriate access.
15. Appropriate staff members should be able to suspend or alter accounts in ways appropriate to the staff member’s role, requiring password confirmation, and being able to target users appropriate to their relationship. For example, the principal should be able to suspend any account, and must give a reason – this reason would be stored for records, and shown to the user attempting to access the account. An IT technician would be able to suspend accounts of a lower level than them, as well as being able to do more things related to their technical role, and be able to force a password reset on any account.
16. Users unable to access their accounts will be able to trigger a password reset request, sent to their school email for confirmation.
17. When a password reset request is triggered, an email will be sent to the account user’s school email. This email will contain a link, which will lead the user to a password reset prompt. The email will also allow the user to flag suspicious reset requests to the appropriate staff to handle the situation.
18. When a user creates their account, the system should be able to automatically identify the user’s roles and add permissions, and if they are a student, identify them from within SIMS exports.
19. Teachers must be able to set homework using the system to a class, selection of classes, student or selection of students, providing information about what the homework is, as well as providing when the homework is due.
20. Students must be able to view homework set by teachers on the system, including which teacher set the homework, when it was set, when it is due, what the homework is, and, if set to their class rather than them specifically, which class they are in that was set the homework.
21. Students must be able to mark the homework they are set as complete, in progress, cannot complete or other options, and add notes to the set homework.

## 1.5 Constraints and Limitations

* Access to student information must comply with data protection legislation (mainly GDPR and Data Protection Act)
* There is no readily available SIMS API and accessing the database directly is not possible due to updates changing its structure
* NUAST/Nova Education Trust’s policies may not allow access to some tools, software and/or information

## 1.6 Proposed Solution

A web app written in Python using Flask that uses an SQL database to store login, security, settings and other information uploaded to the app. This web app would use a secure login system to be able to access any tools or information. In addition to this, scripts and/or scheduled tasks would be set up to export student information from SIMS every 24 hours to the app’s database for students to access. The app would allow students to rate/review teachers, subjects, facilities and the school, and to report issues. Both of these have optional anonymity, except for safeguarding issues, in which appropriate staff can revoke this anonymity.

## 1.7 Evidence of Analysis

### 1.7.1 Interviews

#### 1.7.1.1 IT Technician

**Q.** What does the school use for a login system?

**A.** Microsoft’s Active Directory

**Q.** Would it be possible to link my student portal with the school’s login system?

**A.** No. Keep everything separate from the school system.

**Q.** When is the school changing away from SIMS?

**A.** Next year

**Q.** Would it be possible to set up a system on the school network where reports from SIMS are scheduled to be exported every 24 hours, converted to CSV using CommandReporter, then uploads the CSV file to my database?

**A.** <ASK WHEN AVAILABLE>

#### 1.7.1.2 Head of Computer Science

<SIMS info>

### 1.7.2 Questionnaires

#### 1.7.2.1 Students’ Opinions on Usefulness











### 1.7.3 Online Research

#### 1.7.3.1 SIMS

From doing research online, I found that SIMS’ database structure tends to change between updates, and so hooking directly into the database in my app would likely only work for a short time, before a new update changes the database structure and breaks any queries within the changed tables.

In general, the suggested way to access information from SIMS is to export reports (this can be scheduled using Task Scheduler), have them converted to CSV files by CommandReporter, then import them into the system that the user wants the data in.

<http://www.edugeek.net/forums/mis-systems/63879-sims-database-structure.html>

<https://blogleedrury.files.wordpress.com/2019/03/command-reporter-user-guide.pdf>