Case study: Implementing an attendance logging web service

Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David
1822 - 2022

Patrick Atkinson (1811357), BSc (Hons) Computing (Software Engineering)

1. Introduction

University of Wales Trinity Saint David (UWTSD) uses an Excel-based Attendance Monitoring Systems (AMS) on the Welsh campuses. The process for collecting attendance information is onerous, with lecturers having to complete paper sign in sheets that must then be transcribed into the appropriate spreadsheets. This approach is not only prone to user error, but also reduces the amount of time lecturers are able to dedicate to students.

2. Aims and objectives

It was decided that a better solution was required for attendance logging, with the onus for logging attendance being moved away from the lecturers towards the students, allowing lecturers to dedicated more time to the students. Therefore the aim of the project was to conduct a case study to implement an improved approach for attendance logging at UWTSD. The objectives to meet this were to:

- Evaluate the available approaches for logging attendance, including prior and current implementations, and those from other institutions.
- Determine the system requirements and implement a new solution that will be compared to the existing approach.
- Critically evaluate the solution against the requirements and the existing approach and assess its viability.

3. Research

Different attendance logging approaches were researched, highlighting that many required additional hardware, thereby increasing cost and deployment overheads. In contrast, the software-based solutions only had development costs to contend with. This informed the choice of using a QR code system for attendance logging.

Research into Application Programming Interface (API) testing explored the areas of testing that impact APIs in particular. This helped inform the testing approach that was chosen for the project. The different approaches to securing APIs were also considered and this informed the option chosen for the development.

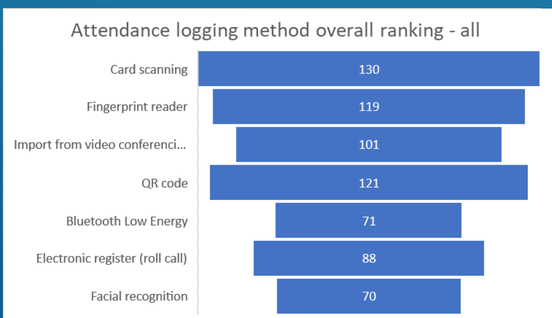

Figure 1 – Results from the staff survey which showed a preference towards card scanning and QR codes.

4. Requirements

There were two main approaches to gathering the requirements: a survey shared with teaching and administrative staff about the existing system, and discussion with department leaders.

The survey helped steer the approach for how to log the attendance. Participants were asked to rank some proposed attendance logging approaches to see if there what preferences were on a new approach as (see Figure 1). The discussion with department leaders defined some of the key resources limitations of the project, such as costings, security and infrastructure considerations (see Figure 2 for the final infrastructure).

A user story map (see Figure 3) was created to summarise the collected requirements. This allowed them to be prioritised by need and grouped by process.

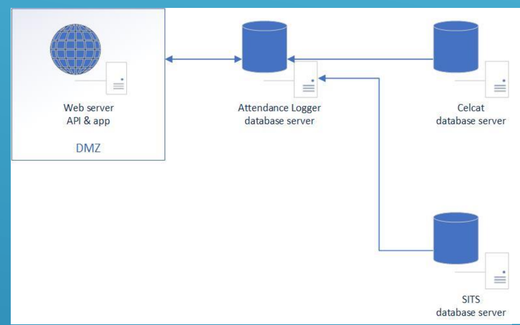

Figure 2 – The infrastructure of the attendance logging system.

5. Implementation

The system was built in two parts: an API and a web app for exercising the API. Both were built using .NET 6 as the development platform and C# as the language. The API was configured to provide data for various attendance related resources (e.g., timetable data, staff names, etc) as well as provide a means for creating sessions and logging data. The web app provided an interface for setting up the sessions and logging the attendance both as a lecturer and a student.

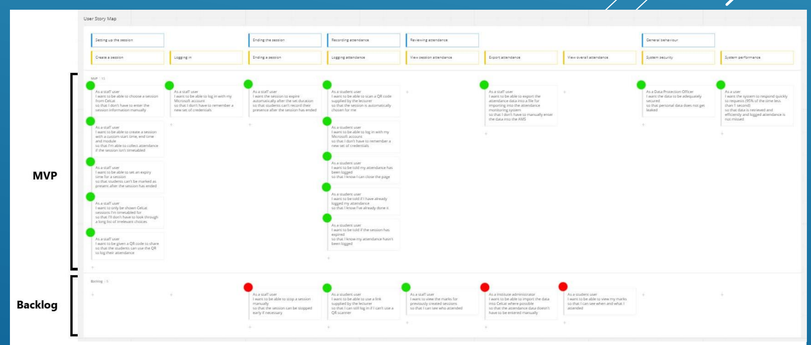

Figure 3 – The user story map, with the completed and incomplete requirements indicated with green and red dots respectively. Those requirements in the Backlog section were not compulsory for the system to function.

6. Testing

Testing was split into functional, non-functional and user testing.

The functional testing used Postman as the tool. 67 tests with over 200 assertions were created, testing areas such as response time, response status code and response schema. These test were designed to ensure the API met the requirements.

The non-functional testing conducted was performance testing in the form of load and stress tests. The load testing checked the system's performance when exposed to its maximum expected load, and the stress tests were designed to test the system to breaking point. Artillery was the testing tool. The system performed well in both cases, with it exceeding the required performance metrics. Figure 4 shows an example of the response times logged during one of the load tests.

The pilot tested the usability of the web interface. Feedback from the users was positive, with attendance being logged quickly by the users involved.

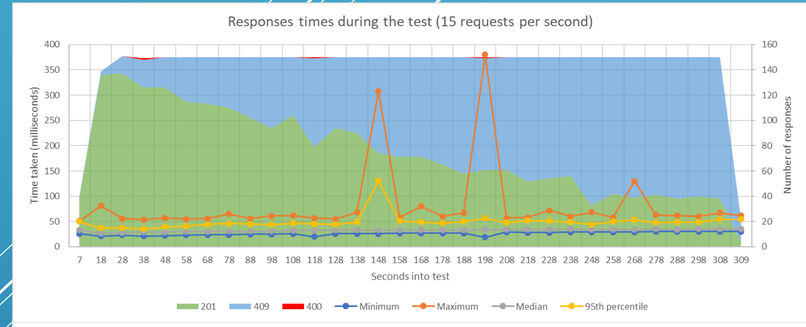

Figure 4 – Response times logged during the load tests done as part of the non-functional testing.

7. Conclusions and recommendation

The system met the aim as an improved attendance logging system was created and successfully piloted. The impact on lecturers was greatly reduced without negatively impacting the students. The maintenance overhead of the new system is much lower than the existing AMS and the data generated is much more easily used for reporting and integrations with other systems. The system met all the requirements to satisfy what constituted the Minimum Viable Product (MVP), with some additional requirements also being met (see Figure 3).

There is potential in the future for the pilot to be extended to gather a greater range of feedback. The data could also be synchronised to the student portal (MyDay) and a reporting dashboard set up for tracking attendance and engagement. The fact that the system uses an API means it can be more easily extended with alternative logging approaches that use the same underlying service.