

Proceeding Paper

# Design and Development of Attendance and Temperature Recording System: A Smart Companion for the Current VLE Implementation in Malaysian Schools <sup>†</sup>

Nur Suhaili Mansor <sup>1</sup>, Hapini Awang <sup>1,\*</sup>, Ahmed B. Ndanusa <sup>2,3</sup>, Rizatulazri Idris <sup>4</sup>  
and Abdulrazak F. Shahatha Al-Mashhadani <sup>5</sup>

<sup>1</sup> Institute for Advanced and Smart Digital Opportunities (IASDO), School of Computing, Universiti Utara Malaysia, Sintok 06010, Kedah, Malaysia

<sup>2</sup> Department of Computer Science, University of Abuja, Abuja 900105, Nigeria

<sup>3</sup> Department of Engineering and Space Systems, National Space Research and Development Agency, Abuja 900105, Nigeria

<sup>4</sup> Department of Operation, MAP2U SDN BHD, Nilai 71800, Negeri Sembilan, Malaysia

<sup>5</sup> Faculty of Business, Sohar University, Sohar 311, Oman

\* Correspondence: hapini.awang@uum.edu.my

<sup>†</sup> Presented at the International Academic Symposium of Social Science 2022, Kota Bharu, Malaysia, 3 July 2022.

**Abstract:** With the prominence of COVID-19 in our lives and with the looming threat of other pandemics raising in the future, keeping track of everyone's temperature as a clear symptom of the disease is imperative for society moving forward. This applies to schools and other educational institutions too. This study aims to give a comprehensive solution that would be adaptable to multiple institutions. A web application that can run on most consumer devices would let schools keep track of their students' temperature on top of easing the process of tracking class attendance by correlating both. At the moment, Malaysian schools have already implemented several ICT initiatives, including for data management (APDM, SAPS, etc.) as well as for teaching and learning (Virtual Learning Environment). Therefore, the proposed online system for attendance and temperature recording would be a good complement to support the digitalization of Malaysian schools.

**Keywords:** Virtual Learning Environment (VLE); ICT in education; attendance recording system; web-based computing



**Citation:** Mansor, N.S.; Awang, H.; Ndanusa, A.B.; Idris, R.; Shahatha Al-Mashhadani, A.F. Design and Development of Attendance and Temperature Recording System: A Smart Companion for the Current VLE Implementation in Malaysian Schools. *Proceedings* **2022**, *82*, 111. <https://doi.org/10.3390/proceedings2022082111>

Academic Editor: Mohamad Rahimi Mohamad Rosman

Published: 26 October 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Schools throughout the world are adapting to the pandemic situation brought about by the COVID-19 virus. Educational institutions need to make sure that their students are safe in the school environment. One way of doing this, which is comprehensive all over the world and is seen everywhere, is the use of temperature scanners. Temperature is a good indicator that a person is currently infected with a disease. As such, being able to track everyone's temperature is an important aspect of the future of society and every department and institution should not only collaborate with the rest of the country's institutions, but they should also have the right and the tools to manage and track people's temperature to create safer environments. Attendance tracking is almost a natural correlation to the issue of tracking temperature in a school setting [1]. Since most students are registered in the internal school system, and their attendance must be tracked (whether manually or otherwise) in order to receive a passing grade or mark, it is clear that tracking temperature in addition to attendance is the preferred way of doing it. Manually tracking temperatures would be time-consuming and mentally taxing for most educators. Manually calling out a name and marking next to the individual's name is simple enough. However, the addition of manually scanning temperatures and correctly recording them is more complicated. Like

many other educational institutions, SK Tanah Rata is facing these issues and they reached out to create an application that would suit their needs.

Attendance systems in schools have gradually progressed over the years, due to the students’ data being stored in a school database, the wide adoption of mobile devices, and the ease of using an automated system for attendance compared to doing it manually [2]. However, not all schools implement automated attendance systems, especially in institutions for younger students. This is mainly because despite the adoption of mobile devices being great in most areas, many parents do not allow younger children to use mobile devices until they grow older. Temperature, being a key indicator of human illness, plays an essential role in maintaining the safety of all students, educators, and staff members. Manually keeping track of it is a risk of close contact infections. An automated system to keep track of both attendance and temperature would ease many of these problems.

## 2. Methodology

The study was conducted following the Agile application development. Agile software development refers to a group of software development methodologies based on iterative development, where requirements and solutions evolve through collaboration between self-organizing cross-functional teams. Agile methods or Agile processes generally promote a disciplined project management process that encourages frequent inspection and adaptation, a leadership philosophy that encourages teamwork, self-organization, and accountability, a set of engineering best practices intended to allow for rapid delivery of high-quality software, and a business approach that aligns development with customer needs and company goals. Agile development refers to any development process that is aligned with the concepts of the Agile Manifesto. The manifesto was developed by a group of fourteen leading figures in the software industry and reflects their experience of which approaches do and do not work for software development. Figure 1 illustrates the Agile phases.

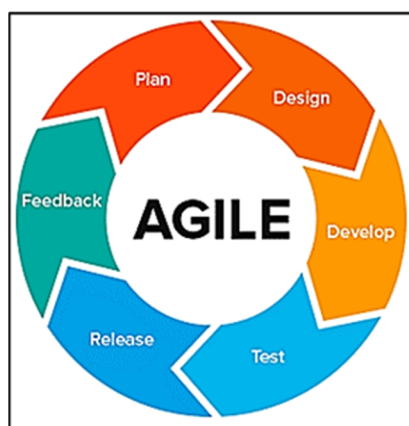


Figure 1. Agile Phases.

### 2.1. Plan

As the Agile application development indicates, the first step was to plan the development of the application, this was done by gathering the requirements (Table 1).

**Table 1.** List of Requirements for Creating and Managing Attendance and Temperature.

Functional Requirements			
Num.	Req. ID	Requirements Description	Priority
1.	<b>TSS_01</b>	<b>Login</b>	
	TSS_01_01	Users shall login to the system by inputting their username and password for authentication.	M
	TSS_01_02	The system can authenticate usernames and passwords.	D
	TSS_01_03	The system can store authentication information in the browser cookies for easy access.	D
	TSS_01_04	Users can request a new password via email in case they forgot it.	O
2.	<b>TSS_02</b>	<b>Manage Users</b>	
	TSS_02_01	Admin shall add users to access the system.	M
	TSS_02_02	Admin can also delete a specific user.	O
	TSS_02_03	Admin can also edit the details of a specific user.	O
	TSS_02_04	The system can verify if the username and password of a new user are acceptable to be added.	D
3.	<b>TSS_03</b>	<b>Manage Students</b>	
	TSS_03_01	Users shall add students to the system.	M
	TSS_03_02	Users can also delete a specific student.	O
	TSS_03_03	Users can also edit the details of a specific student.	O
	TSS_03_04	The system can verify the completion of the details of a new student.	D
4.	<b>TSS_04</b>	<b>Manage Classes</b>	
	TSS_04_01	Admin shall add a new class to the system.	M
	TSS_04_02	Admin can also delete a specific class.	O
	TSS_04_03	Admin can also edit the contents of a class.	O
5.	<b>TSS_05</b>	<b>Manage Attendance</b>	
	TSS_05_01	Admin shall add a student from the system to a class.	M
	TSS_05_02	Admin can also delete a specific student from a class.	O
6.	<b>TSS_06</b>	<b>Manage Temperature</b>	
	TSS_06_01	Users shall append a new value for the temperature to a student of a class.	M
	TSS_06_02	Users can also delete a specific temperature from a student in a class.	O
	TSS_06_03	Users can also edit the temperature of a student in a class.	O
7.	<b>TSS_07</b>	<b>Add Classes</b>	
	TSS_07_01	Teachers shall add a new class to the system.	M
8.	<b>TSS_08</b>	<b>Add Attendance</b>	
	TSS_08_01	Admin shall add a student from the system to a class.	M
9.	<b>TSS_09</b>	<b>Logout</b>	
	TSS_09_01	Users shall log out of the system.	M
	TSS_09_02	The system can force a logout if a certain amount of time passes.	D

### 2.2. Design

**Before Prototype:** The overall design of the application was made using pen and paper, and the main elements, buttons, and links for the users to interact with UML diagrams were created to visualize the elements and features (Figures 2 and 3).

**During Prototyping:** When the prototyping phase started, the design was made using Figma (Figures 4 and 5), an interactive tool that allows the creation of prototypes that are very functional and helps show the connection between functions and parts of the application with one another.

**During Development:** When developing the application, the design elements were created using HTML and CSS and using Bootstrap to create elements such as buttons and clean up the look of the application.

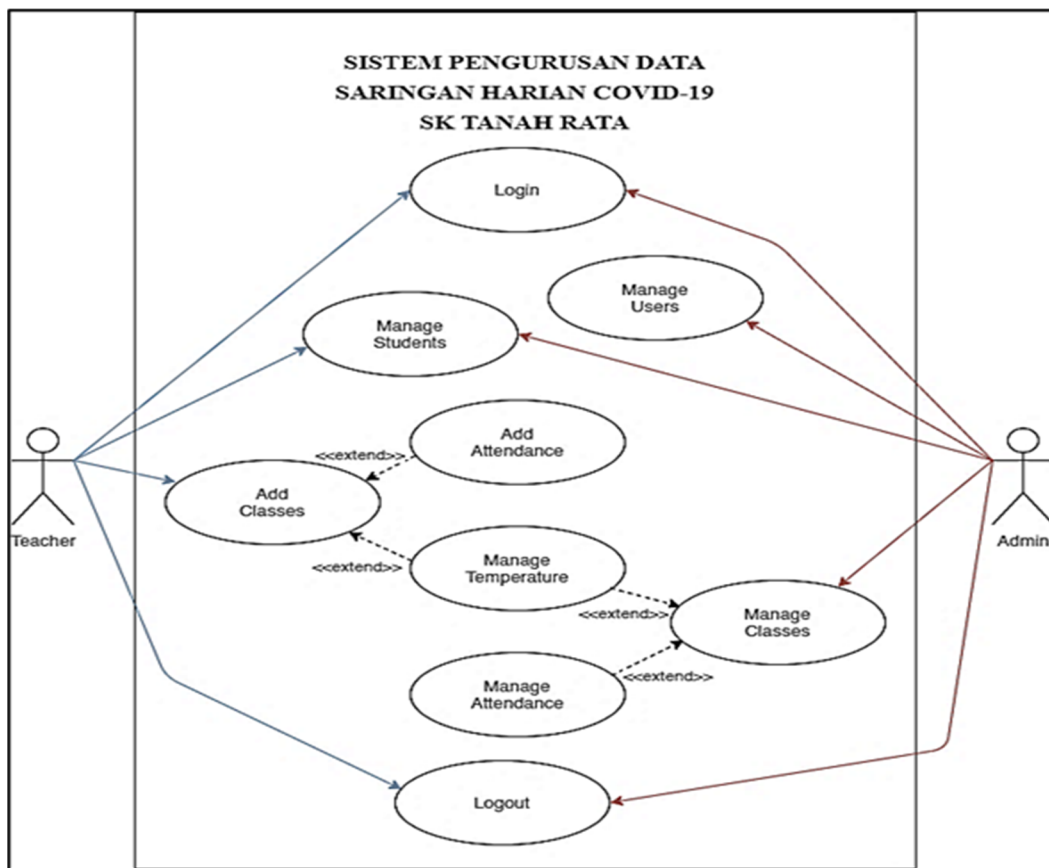


Figure 2. Use Case Diagram.

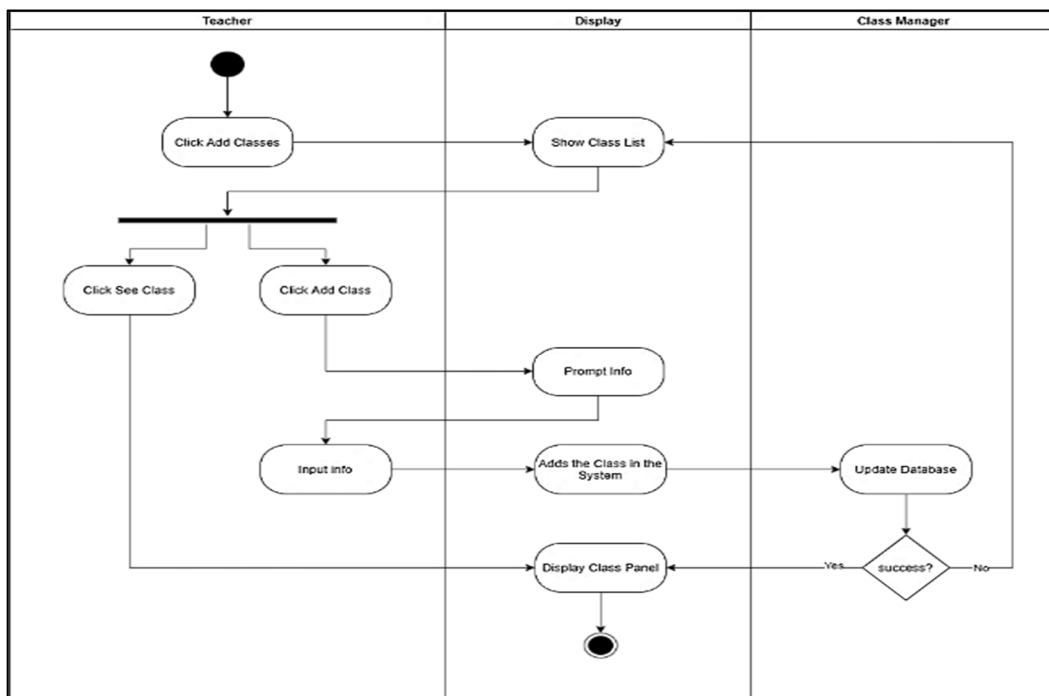


Figure 3. Activity Diagram for Add Classes Use Case.

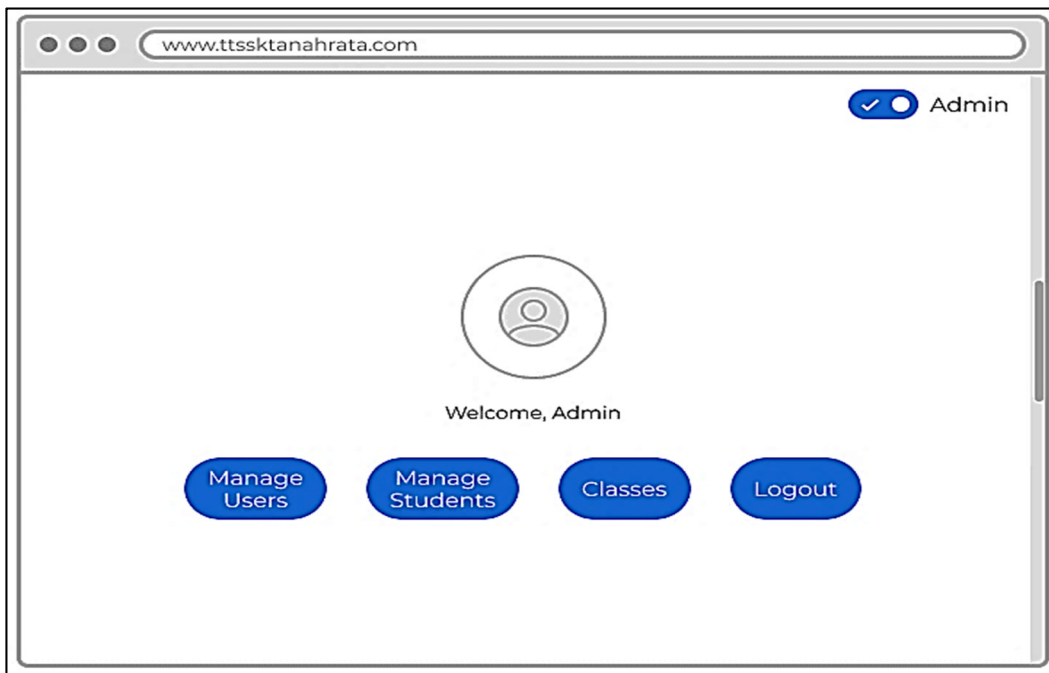


Figure 4. Landing Page Using Figma.

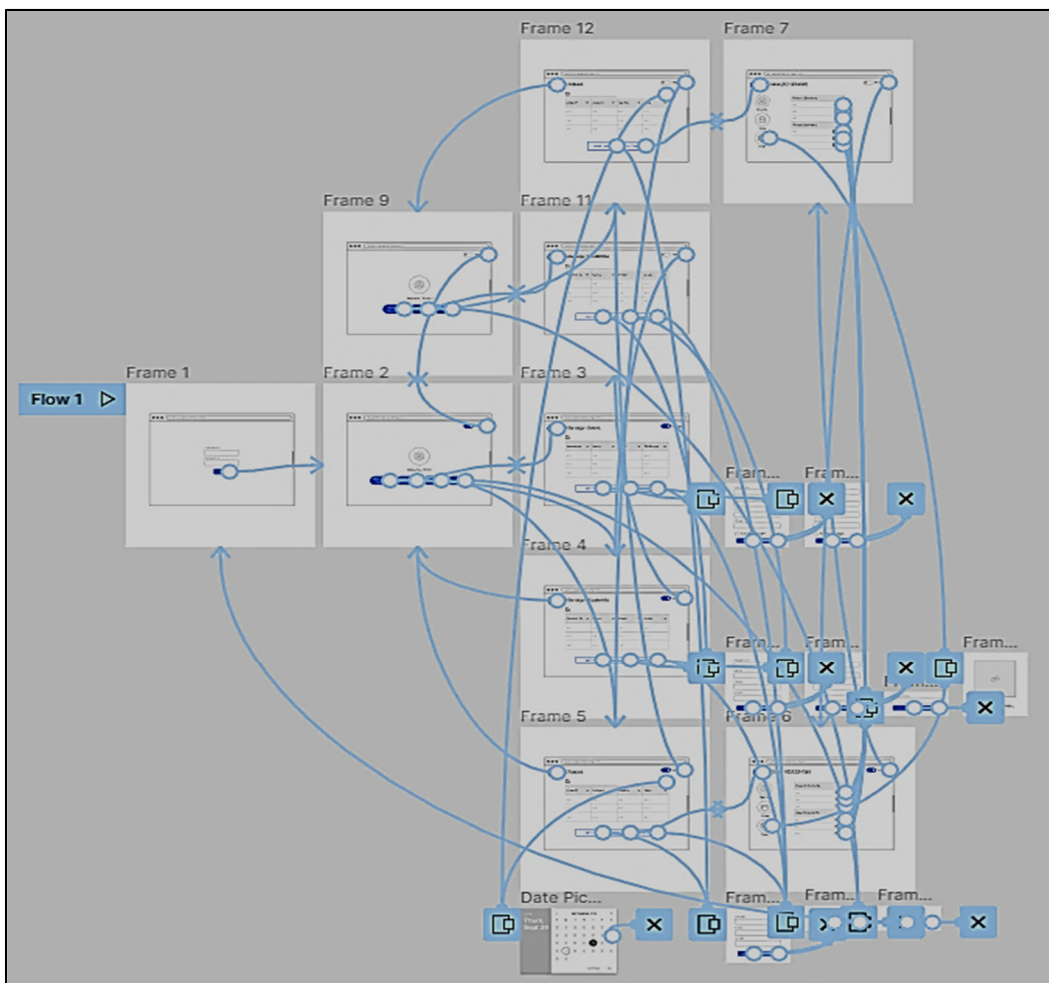


Figure 5. Prototype Interactions in Figma.

### 2.3. Development

The application was developed using the Ruby language, and the Ruby on Rails application framework written in Ruby [3–5]. Ruby on Rails is a server-side model–view–controller (MVC) framework, that provides default structures for a database, a web service, and web pages (Figure 6). It encourages and facilitates the use of web standards such as JSON or XML for data transfer and HTML, CSS, and JavaScript for user interfacing. In addition to MVC, Rails emphasizes the use of other well-known software engineering patterns and paradigms, including convention over configuration (CoC), do not repeat yourself (DRY), and the active record pattern [6,7]. Active record is a key component in the development of this application.

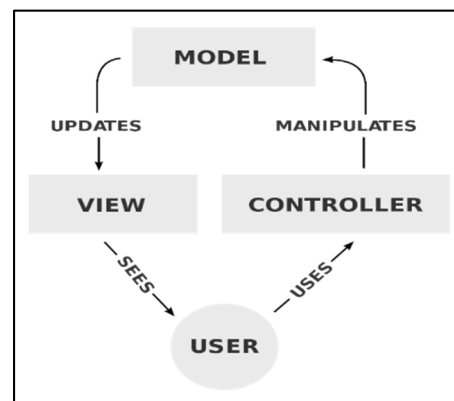


Figure 6. MVC Model.

All of the development processes can be tracked on GitHub, where the open-source code of the application resides, and all the comments with information regarding the improvements can be seen. The first step was to install Ruby, and all the dependencies needed by Rails in the main computer, this included packages such as yarn and node.js. Once Ruby was installed, gems (Ruby development packages) could be installed, followed by Rails. Once Rails was installed in the system, we generated a repository using Rails, which created all the directories and files needed to run a barebones application. Most of the code was written using Visual Studio Code, an open-source editor from Microsoft. The code was run using Microsoft PowerShell. Version control was implemented using Git, and a remote repository was created on GitHub to mirror all the changes made to the application public for interested parties to see.

We utilized the Ruby Gem 'devise' for user authentication. We created routes, models, controllers, and views with all the CRUD actions for each of the elements we wanted to integrate: users, students, classes, and attendance. The database was handled by sqlite3 in the development environment, however, we planned on using PostgreSQL for the production environment. To create the models and migrate them into the database, Rails implemented what is known as migration files, these files were created to generate the tables, columns, and relationships of the different elements of the application. These migration files were meant to be run only once and any updates to a table had to be done by creating a new migration file. Once the migrations were complete, we started to write the code for every CRUD action in the system. It was an iterative process in which we tested each feature, identified the errors, attempted to fix them, made appropriate changes, uploaded the code, tested the system, and so on. There are still things to be improved. The Entity-Relationship Diagram (ERD) of the different resources of the system is presented in Figure 7.



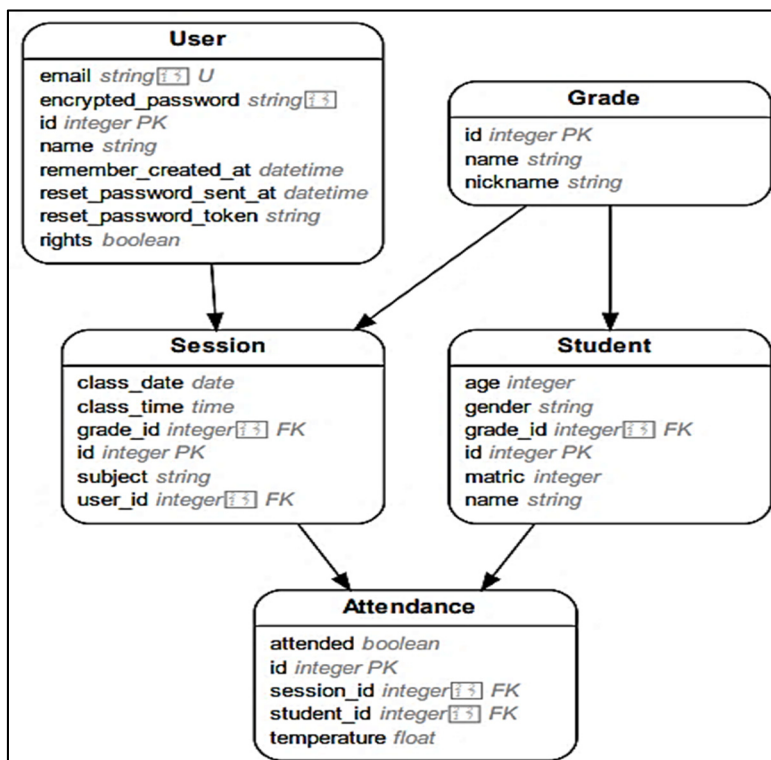


Figure 7. Entity Relationship Diagram of the Different Resources.

### 2.4. Testing

Testing was done in Google Chrome, using localhost:3000 as the root address of the application. We used Google Chrome to test features such as JavaScript, HTML, and CSS. The testing processes are presented in Figure 8.

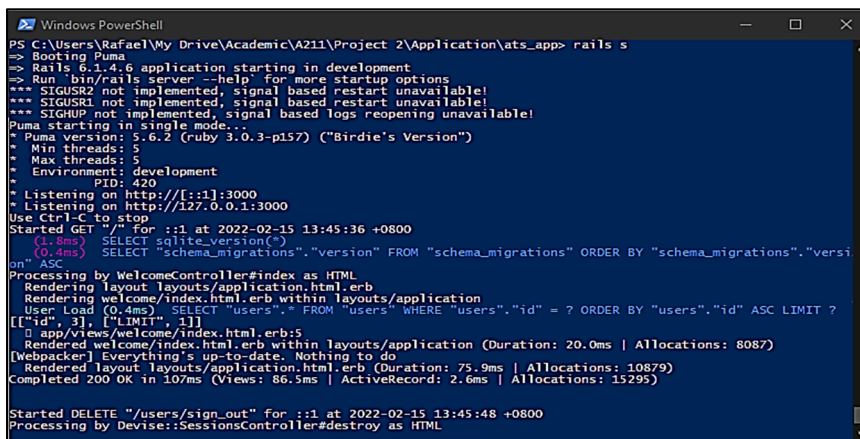


Figure 8. Server running on Windows via Microsoft PowerShell.

### 2.5. Deployment

Deployment of the application was done on Heroku, a subsidiary of Salesforce. Heroku is a cloud platform as a service (PaaS) supporting several programming languages [8]. Applications that are run on Heroku typically have a unique domain used to route HTTP requests to the correct application container or dyno. Each of the dynos is spread across a “dyno grid” which consists of several servers. Heroku’s Git server handles application repository pushes from the permitted user. All Heroku services are hosted on Amazon’s EC2 cloud-computing platform. A Heroku account was created, a Heroku application was initiated, and the code was pushed to the Heroku git, which then ran the code on their Ruby

server. Migration files were run and any updates to the software were carried over Git push. The application is currently online and can be found on <https://rafael-ats.herokuapp.com/> (accessed on 25 March 2021) An example of the system's interfaces is shown in Figure 9.

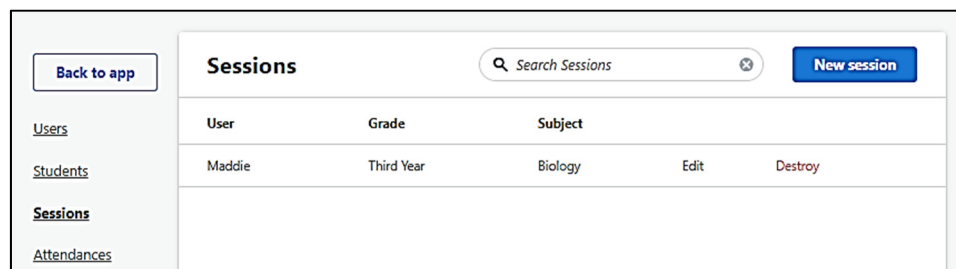


Figure 9. Application Dashboard Running on Heroku.

### 3. Conclusions and Future Work

In line with the Malaysian vision to digitalize its education, the proposed online system for attendance and temperature recording would support the existing implementation of VLE [9–11]

Towards a future of limited close contact, constant health checks, and close dependency on technology, we must adjust to the new paradigms in order to continue living fulfilling lives and producing products and services on par or better than what we had before. Applications such as the one described in this paper will aid in the improvement of different institutions and sectors in the country and help provide better services and education. There are still many ways in which to improve applications such as this one. For example, the use of face recognition, geolocation and long-range temperature devices, would remove the middleman and provide a truly automated attendance system with a temperature infrastructure.

**Author Contributions:** Conceptualization, N.S.M. and H.A.; methodology, A.B.N. and R.I.; testing, A.F.S.A.-M. and N.S.M.; formal analysis, H.A.; investigation, N.S.M.; resources, R.I.; writing—original draft preparation, N.S.M. and H.A.; writing—review and editing, A.B.N. and A.F.S.A.-M. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Acknowledgments:** This research is supported by the Universiti Utara Malaysia College Grant: S/O Code: 14826.

**Conflicts of Interest:** The authors declare no conflict of interest.

### References

- Patel, A.; Joseph, A.; Survase, S.; Nair, R. Smart Student Attendance System Using QR Code. In Proceedings of the 2nd International Conference on Advances in Science & Technology (ICAST), Mumbai, India, 8–9 April 2019; pp. 1–4.
- Wei, X.; Manori, A.; Devnath, N.; Pasi, N.; Kumar, V. QR Code Based Smart Attendance System. *Int. J. Smart Bus. Technol.* **2017**, *5*, 1–10. [CrossRef]
- Hartl, M. *Ruby on Rails Tutorial: Learn Web Development with Rails*, 3rd ed.; Addison-Wesley: Boston, MA, USA, 2015.
- Tate, B.A.; Hibbs, C. *Ruby on Rails: Up and Running*; O'Reilly Media: Farnham, UK, 2008.
- Tate, B.A.; Hibbs, C. *Ruby on Rails: Up and Running: Lightning-Fast Web Development*, 1st ed.; O'Reilly Media: Sebastopol, CA, USA, 2006.
- Bächle, M.; Kirchberg, P. Ruby on Rails. *IEEE Softw.* **2007**, *24*, 105–108. [CrossRef]
- Ruby, S.; Copeland, D.B.; Thomas, D. Agile Web Development with Rails 6. In *Pragmatic Programmers*; Pragmatic Bookshelf: Raleigh, NC, USA, 2020.
- Richardson, L.; Ruby, S. *RESTful Web Services*; O'Reilly Media: Sebastopol, CA, USA, 2008.



9. Awang, H.; Zahurin, M.A.; Wan, R.S.O. Measuring Virtual Learning Environment Success from the Teacher's Perspective: Scale Development and Validation. In Proceedings of the 3rd International Conference on Applied Science and Technology, Penang, Malaysia, 10–12 April 2018; pp. 1–6.
10. Awang, H.; Zahurin, M.A.; Wan, R.S.O. The Moderating Effect of Workload in Determining the Continuous Usage of Virtual Learning Environment amongst School Teachers. In Proceedings of the Pacific Asia Conference on Information Systems, Yokohama, Japan, 26–30 June 2018; p. 51.
11. Awang, H.; Wan, R.S.O.; Zahurin, M.A. A Conceptual Model to Evaluate Virtual Learning Environment among Malaysian Teachers. *J. Telecommun. Electron. Comput. Eng.* **2018**, *10*, 59–63.