Pitogo Senior High School
Negros Street, Barangay Pitogo

Feasibility Study in Implementing a Local Web Server Platform (Vitre) that Promotes Real-time Teacher-to-Student Transparency

A research submitted in Partial Fulfillment of the requirements for the course Research Project for Science Technology Engineering and Mathematics track

Submitted by:
Ansay, Sherwin G.
Cordero, Daveren John R.
Nuada, Dave Emmanuel V.
Rivera, Jose Gabrielle A.

12 – STEM – Casimiro del Rosario

Supervised by:
Ms. Joan S. Lanuzga
Research Project Teacher

March 21, 2019

Abstract

The intent of this study is to develop a system that would improve the current day grade disclosure system of schools. In recent years, the way of grade disclosure in most public educational institutions is highly dependent on various factors of chance concerning the teachers. An official system, made to accommodate transparency for the students, has yet to be introduced to the curriculum and so this project aims to resolve that. The project entailed the development and deployment of a local web server platform on the school that would store student grade data that is extracted from teachers' class records and through their desktop companion application. The design also involves mounting routers that are bridged to the main server on crucial locations in the hallways in order to act as access points that would accommodate most of the student body in connecting to the network. The students would have to access their accounts with the use of account details that are provided to them beforehand. Upon connecting, they would be greeted by a website interface that they could use to gain access to their individual records thus getting updated on their progress in class. The feasibility of the project has been proven upon its deployment. It shows through its impact to the students and it shows great promise for the future of grade disclosure such that it benefits both the students as well as the faculty users.

Table of Contents

Abstract **2**
Table of Contents **3**
Acknowledgement **7**
Chapter 1: Introduction **8**
1.1. Motivation 8
1.2. Terms of Definitions 9
Chapter 2: Literature Review **10**
2.1. SPAMAST Mobile Grade Inquiry System 10
2.2. The Motivation Impact of Open Educational Resources Utilization on Physics Learning Using Quipper School App 10
2.3. Faculty Share Data and Improve Students' Learning 10
2.4. Delivering School Transparency in Australia 11
2.5. NILOA Pedagogical Transparency Framework 11
2.6. Online Grading System for Distributed Grading in a Large First Year Project-Based Design Course 13
Chapter 3: Systems Design **14**
3.1. Hardware Systems Design 15
Raspberry Pi 3 Model B+ 15
Accessories 15
Operating System 16
Raspberry Pi as a Web Server 16
Raspberry Pi as a DNS Server 16
3.2. Network Systems Design 17
Category 5-Enhanced Ethernet Cable 17
TP-Link TL-WR840N V6 Router 17
TP-Link TL-SF1008D V11 Network Switch 18
Network Topology and Configuration 18
3.3. Software Systems Design 21
3.3.1. The Web Server Frontend 22
Web Page Design 22
Design Context 22

Color and Art Style 22
Logo Design 23
User Experience Design and Web Pages 24
Login Portal Page 24
Home Page 25
Subject List Page 26
Subject Details Page 27
My Account Page 27
3.3.2. The Web Server Backend 29
Login Authentication 29
JSON Web Token (JWT) 29
Subjects Retrieval 30
Activities Retrieval 30
Database Structure 30
Students Table 30
Teachers Table 31
Classes Table 31
Subjects Table 31
Curricula and Curricula Metadata Table 32
Tracks Table 33
Grades and Grades Metadata Table 33
Analytics Table 34
Tickets Table 35
3.3.3. The Desktop Teacher Application 36
Application Parts 36
Electronic Class Records 37
3.3.4. The Desktop Teacher Application Processes 40
The Patch Cycle 40
The Validation Process 41
Import Manager 41
Record Imports Model 42
Recognizer File 42
Class Record Configuration Model 43
The Parsing Process 45
Previewing a class record 45
Activating a class record 45
Context Manager 46
The Synchronization Process 48
Content Crown 49
Content Body 49
Content Difference 50
Instruction Sets 50

User Interface Update 52
The Patching Process 54
Patch File Payload 55
Grade Data Patch 55
Class Registry Patch 55
The Registration Process 56
User Account Sheets 56
Chapter 4: Implementation 58
4.1. Research Locale and Population 58
4.2. Operational Feasibility 58
4.3. Development Process 58
4.4. Scheduled Deployment 59
4.5. Project Cost 59
Chapter 5: Implementation Barriers 61
5.1 Security 61
Operational Security 61
Information Security 61
Security Mitigation Strategies 62
5.2 Cost 62
Cost Mitigation Strategies 62
5.3 Infrastructure 63
Structural 63
Wireless Connectivity 63
Data Storage 64
Infrastructure Mitigation Strategies 64
5.4 Sustainability 64
Software Sustainability 64
Hardware Sustainability 64
Sustainability Mitigation Strategies 65
5.5 Usability 65
Usability Mitigation Strategies 65
5.6 Sociocultural 65
Sociocultural Mitigation Strategies 65
Chapter 6: Conclusion and Recommendations 67
6.1 Conclusion 67
6.2 Recommendations 67
Features and Design 67
Security 68
Ease of Access 68
Continuity 68
References 69
Appendices 70

List of Profiles 71
List of Figures 72
List of Tables 93

Acknowledgement

This research paper entitled “Feasibility Study in Implementing a Local Web Server Platform (Vitre) that Promotes Real-time Teacher-to-Student Transparency” would not have been made possible without the following people, for which it is dedicated to:

To Ms. Joan Lanuzga, our research adviser, who unconditionally gave her deepest understanding and guidance that helped build this research study.

To Mr. Warren Lucas and Mr. Joseph Recullo, our former research advisers, who gave initial and constructive criticisms which started and influenced the project towards the right direction.

To Mrs. Letecia Beronio II, our project manager, who allowed us to conduct and kickstart the project for the senior high school community.

To Mrs. Gizelle Laud, the former assistant principal, who provided us with guidance, supervision and support in executing the project.

To Mr. Felix Bunagan, the Pitogo High School principal, who enthusiastically supported us in deploying the system.

To Mr. Jomari Peroño, who is the administrator, that guided and supported us in the finance reports and early designs of the project.

To Mr. Roy Cabebe and Mr. Florante Ferrer, our former robotics advisers, for providing us necessary equipment and a place to develop the platform and research.

To our dearest family, classmates and friends, who were there to motivate us up the moment we stumble upon difficulties in the study.

Chapter 1: Introduction

1.1. Motivation

The Department of Education's tradition of grade disclosure is an old mechanic that dates from several centuries ago in line with the classic quarterly system. This same quarterly grading system is implemented in public schools (elementary, junior high school and senior high school) as well as some private schools throughout the Philippines. In relation to this, it has come to the researcher's attention that while the other components of the education system are constantly being criticized and changed, the disclosure of grades have been long tolerated. The original system has seeped deep into the culture of schools that the idea and its method remained timeless or unchanging. This is partially good as a quarterly system of grade disclosure does pose multiple benefits such as order and uniformity. However, unbeknown to most, it has its own fallbacks that impacts the student's academic performance tremendously—fallbacks that were tolerated before due to insufficient technology. For example, it is a widely known fact that students have legal rights to their own records and are also held liable for each of the grading components (activities, written works, examinations etc.) that composes their entire academic assessment. Unfortunately, having the disclosure of grades occur at the end of the semester hinders their awareness to their records throughout the duration of the course. This is a problem since only at the end of the semester would they realize the consequences of their actions making it largely impossible for them to basically fix failing grades. Now, some may argue that teachers do occasionally disclose grades and students may extrapolate their general average by compiling all their activities through that method, but this is not only inefficient, it does not also occur all the time. Teachers are commonly busy people due to the limited time they have in a class session thus they are obliged to manage this time wisely. Because of this, in most cases, they are not willing to compromise their time to disclose recent activities. Nonetheless, as computers began to rise and the contemporary world became more inclined with technology, it is now possible that this system is altered with a more efficient method.

The Vitre system aims to solve these problems by integrating a regular system of grades monitoring where the students can access their own grades in real-time thus promoting student to faculty transparency. Transparency relies on the principle of open information where two complementary parties are well versed upon the data shared between them. In the terms of this study, the student and faculty are those two complementary parties. This is resolved by the researchers upon the development of the local web server platform that the developers call "Vitre". The name "Vitre" is derived from the word "vitreous" which means clear, directly correlating with the transparency theme. This web server collects incoming data which are class records sent by the teachers and store them in an archive. These data are then rendered in a web page that students can access upon logging in with their provided user accounts. The web page consists of a user interface that is catered to presenting academic data to the users, helping them keep track of their progress in class.

1.2. Terms of Definitions

1. Transparency - a term associated with the principle of open information where two complementary parties are well versed upon the data shared between them.
2. Server - a computer which manages access to a centralized resource or service in a network.
3. Client - a computer that is capable of obtaining information and applications from a server.
4. Web Server - a program that uses the Hypertext Transfer Protocol (HTTP) to process, and deliver webpages to its users or clients.
5. Grade Disclosure - An act of providing students information of their own class records.
6. Traditional Grade Disclosure - unaided grade disclosure through simple distribution of test papers and paper works after it has been checked.
7. Local Area Network - a computer network that links devices within a building or group of adjacent buildings.
8. Frontend - the visual elements of a software, application or website.
9. Backend - the logical and core computational logic of a website, software or information system.
10. Data - refers to referenced information within a system.
11. Metadata - a set of data that describes and gives information about other data.
12. Primary Key – a field in one table that uniquely identifies a row.
13. Foreign Key – a field (or collection of fields) in one table that uniquely identifies a row of another table or the same table.
14. MySQL - an open source relational database management system (RDBMS) based on Structured Query Language (SQL).
15. JSON (Javascript Object Notation) - pertains to a common data structure formatting used by the developers.

Chapter 2: Literature Review

2.1. SPAMAST Mobile Grade Inquiry System

Banual and Mamalias (2018) developed a similar system in their campus: Southern Philippine Agri-Business and Marine and Aquatic School of Technology Poblacion, Malita, Davao Occidental. According to them, their study aims to develop a system that allows students to inquire their grades online using android phones or tablets; as well as develop a workable grade inquiry system using an android phone or tablets.

The developed system acts as a mobile version of the current grading system that is being practiced by their campus. Their system enables the users to view grades from smartphones which are required to be connected to the internet. During the application testing, the researchers used sample data to test its functionality. The application connects to the schools database to fetch the student's grade. Based on the result, the application was able to achieve the specified objectives. The application was developed using Android Studio IDE with Java programming language. During the development period, the waterfall model was used since it has the flow on how to properly use and run the application.

2.2. The Motivation Impact of Open Educational Resources Utilization on Physics Learning Using Quipper School App

According to Sulisworo & co. (2017), the Quipper educational platform follows the same process with face-to-face learning. This research was done due to the digital generation that is proficient and devoted to information technology and computer applications. The research focuses on the attention factor, relevance, confidence and satisfaction of the student. The research was a quasi-experimental type of research and uses ANOVA as the statistical treatment. Quipper follows the concept of Open Educational Resources (OER) in which it is freely accessible, openly licensed text in a way that its media and other digital assets are used for educational purposes. The findings on this research indicate a positive impact of the application of the concept of OER in learning to student motivation.

2.3. Faculty Share Data and Improve Students' Learning

According to Wilkelmes (2014), school faculties rarely have opportunities to research their students’ views about how their best learning happens in college or graduate school. Even less common are the means for teachers to gather such information from colleagues on a large scale and distill it into pragmatic insights about teaching practices best suited to their own particular students. The Illinois Initiative on Transparency in Learning and Teaching is a grassroots assessment project doing just that, and it demonstrably enhances students’ learning. The project has two main goals: (1) to promote students’ conscious understanding of how they learn; and (2) to enable faculty to gather, share, and promptly benefit from data about

students' learning by coordinating their efforts across disciplines, institutions, and countries.

2.4. Delivering School Transparency in Australia

According to Organization for Economic Cooperation and Development (OECD, 2012), featuring their project which was conducted on 9 both primary and secondary and both private and public schools. The project was mostly made to cater minor groups such as indigenous people or people with disabilities. The OECD made a website where all the student grades will be disclosed and given feedback by the teachers. The OECD even go out to the limits of monitoring every student demographics and school information to equalize any advantage of a school. The project also includes reward for greatly performing teachers such payment or certification, online diagnostic tools like individualized assessment, feedback for areas of development which includes tools tailored for each student's needs and reward payments to student's school who show the most improvement in a range of areas. The transparency in this project is visible in the grade disclosure in the website which the Organization for Economic Cooperation and Development (OECD) created.

2.5. NILOA Pedagogical Transparency Framework

The National Institute for Learning Outcomes Assessment (NILOA, 2012) has developed a transparency framework to support institutions in sharing evidence and data of student learning. Each section of the framework suggests a component of student learning assessment that may be of interest to specific audiences and outlines opportunities to advance public understanding. With this, the framework provides ideal components of a transparent institutional website in terms of student learning. The framework states that institutions implementing a website dedicated for transparency should take into account the website's purpose, structure, intended audiences, capacity of technology staff and other available resources when reviewing the components. Information on student learning can and should be presented in language that is understandable by specific and multiple audiences, widely available across the website, updated regularly, receptive to feedback, and accessible by multiple web browsers and at various internet speeds.



Figure 1: Model of the NILOA Transparency Framework

A fully transparent website might well contain information about the following six components:

1. Current Assessment Activities
    a. states and include information, like grade and score remarks, on a full range of projects and activities recently completed or currently underway.
2. Assessment Plans
    a. states how student learning will be assessed, the data collection tools and approaches that will be used, and the timeline for implementation.
3. Assessment Resources
    a. states the information about the assessment practices in an institution.
    b. encompasses information provided to faculty and staff to help them understand, develop, implement, communicate, and use evidence of student learning.
4. Student Learning Outcomes Statements
    a. states the expected knowledge, skills, attitudes, competencies, and habits of mind that students are expected to acquire at an institution.
5. Evidence of Student Learning
    a. includes results of assessment activities which is typically presented in graphical form and allows the viewer to infer the general performance of an institution.
6. Use of Student Learning Evidence
    a. represents the extent to which evidence of student learning is used to identify areas where changes in policies and practices may lead to improvement, inform institutional decision-making, problem identification, planning, goal setting, faculty development, course revision, program review, and accountability.

2.6. Online Grading System for Distributed Grading in a Large First Year Project-Based Design Course

Thompson and Ahn (2012) developed a similar system that focuses on developing an online grading system. Based on the results upon the implementation of their system, it is shown that the system has a number of advantages over traditional grading methods (paper-based), including scalability, real-time feedback on the status of grading, the reduced potential for human error in compiling grades, the ability for jury members to grade remotely and to revise their grades after submission, the ability for course administrators to easily review grading results and remove statistical outliers from the score set, the ability to return both provisional and final grades to the course faculty, staff, and students in a timely manner, and the ability to archive and export grading data for future use. Although evidently beneficial, the system presents drawbacks. These drawbacks include the inability to cater all mobile device versions and poor user satisfaction.

Chapter 3: Systems Design

This chapter presents the overall systems design that was done in the development of the project. The chapter has three subchapters, namely: hardware systems design (Chapter 3.1.), network systems design (Chapter 3.2.) and software systems design (Chapter 3.3.).

These systems work hand-in-hand to form the Vitre platform. Within the platform, there are two distinct parties that are involved: the teachers and students. They primarily interact with the system and Figure 3.1. shows the relationship of these groups towards the Vitre platform.

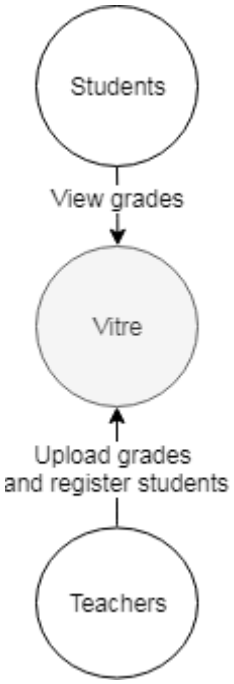


Figure 3.1: Role diagram of the teachers and students towards the Vitre

Students utilize the platform in order to view their grades while the teachers have the ability to upload grades and register students into the system. Moreover, the teachers utilize their copy of the Vitre Desktop Teacher Application to perform and relay these aforementioned actions towards the server. On the other hand, the students use the web browser from their smartphone or laptop to access the web interface of the server.

3.1. Hardware Systems Design

This chapter presents the various hardware utilized for the development of the project. It discusses the hardware specifications and configurations done to harmoniously work within the system.

Raspberry Pi 3 Model B+

The main platform of the Vitre is a budget-sized minicomputer Raspberry Pi 3 Model B+ [9]. The minicomputer functions as the web server and domain name system (DNS) server of the Vitre network. A top view model showcasing the different parts of the minicomputer is shown in Figure 3.1.1. The specifications of the Raspberry Pi is shown in Figure 3.1.2 [10]. These functions are discussed in the latter sections of this chapter.

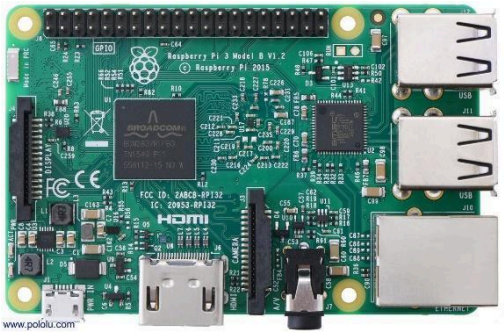


Figure 3.1.1: Top view of the Raspberry Pi 3 Model B+

	Raspberry Pi 3 Model B	Raspberry Pi Zero	Raspberry Pi 2 Model B	Raspberry Pi Model B+
Introduction Date	2/29/2016	11/25/2015	2/2/2015	7/14/2014
SoC	BCM2837	BCM2835	BCM2836	BCM2835
CPU	Quad Cortex A53 @ 1.2GHz	ARM11 @ 1GHz	Quad Cortex A7 @ 900MHz	ARM11 @ 700MHz
Instruction set	ARMv8-A	ARMv6	ARMv7-A	ARMv6
GPU	400MHz VideoCore IV	250MHz VideoCore IV	250MHz VideoCore IV	250MHz VideoCore IV
RAM	1GB SDRAM	512 MB SDRAM	1GB SDRAM	512MB SDRAM
Storage	micro-SD	micro-SD	micro-SD	micro-SD
Ethernet	10/100	none	10/100	10/100
Wireless	802.11n / Bluetooth 4.0	none	none	none
Video Output	HDMI / Composite	HDMI / Composite	HDMI / Composite	HDMI / Composite
Audio Output	HDMI / Headphone	HDMI	HDMI / Headphone	HDMI / Headphone
GPIO	40	40	40	40
Price	$35	$5	$35	$35

Figure 3.1.2: The hardware specifications of the Raspberry Pi 3

Accessories

The minicomputer is equipped with various accessories such as a chassis and heat sink (Figure 3.1.3.). These accessories are installed in order to prolong the hardware lifespan of the Raspberry Pi. The minicomputer is expected to be powered on at least 8 hours a day.



Figure 3.1.3: Raspberry Pi Accessories

Operating System

The Raspbian Stretch Lite (kernel version 4.14) operating system is installed in the SD card for the minicomputer. Raspbian is a Linux Debian-based computer operating system for Raspberry Pi. Since 2015, it has been officially provided by the Raspberry Pi Foundation as the primary operating system for the family of Raspberry Pi single-board computers [11]. The developers ought to install the lite or headless version hence a graphical user interface is not needed and its absence would minimize performance overhead.

Raspberry Pi as a Web Server

For the web server systems of the platform, a LAMP stack is installed in the Raspberry Pi. Furthermore, LAMP is a software bundle and open source Web development platform that uses Linux as the operating system, Apache as the Web server, MySQL as the relational database management system and PHP as the object-oriented scripting language [12].

Raspberry Pi as a DNS Server

In order to resolve offline DNS queries, the BIND (Berkeley Internet Name Domain) 9 DNS software is utilized and configured. It is the most widely used Domain Name System (DNS) software on the Internet [13]. The domain name “shs.vitre.io” is resolved locally through BIND in order to provide convenience for the clients in accessing the network’s main server.

3.2. Network Systems Design

This chapter presents an in-depth overview of the network topology designed for the Vitre platform. It covers the networking peripherals used and the system's overall network design.

Category 5-Enhanced Ethernet Cable

Category 5 Ethernet cable (CAT5E) is the most common cable that is used in connecting wired local area networks. The cable standard provides performance up to 100 MHz and 1 Gbps (Table 3.2.1.) [17]. The developers used the cost-effective Neutron XP CAT5E cable bundle which has a length of 305 meters (Figure 3.2.1.).



Figure 3.2.1: Individual CAT5E cable and Neutron XP Cable Box

Category	Cable Type	Maximum Data Transmission Speed	Maximum Bandwidth
Category 3	UTP	10 Mbps	16 MHz
Category 5	UTP	10/100 Mbps	100 MHz
Category 5 E	UTP	1000 Mbps	100 MHz
Category 6	UTP or STP	1000 Mbps	250 MHz
Category 6 A	STP	10,000 Mbps	500 MHz
Category 7	SSTP	10,000 Mbps	600 MHz

Table 3.2.1: Ethernet cable categories and their specifications

TP-Link TL-WR840N V6 Router

The TL-WR840N router (Figure 3.2.2.) is capable of being a high speed solution that is compatible with IEEE 802.11b/g/n standard [14]. Furthermore, the

router model is based in 802.11n technology and gives users wireless performance at up to 300Mbps [14]. This specific router is utilized as the access point for the users of the Vitre platform to connect to.



Figure 3.2.2: TL-WR840N Router

TP-Link TL-SF1008D V11 Network Switch

The TL-SF1008D (Figure 3.2.3.) is an energy-efficient desktop network switch that provides a convenient way for the developers to expand the Vitre network. The switch has eight Ethernet ports which support auto-MDI/MDIX that enables plug and play functionality [15]. The network switch can process data at a rate of up to 200 Mbps. All three TL-WR840N network routers are connected to this network switch which is also relayed to the Raspberry Pi that hosts the web and DNS server.

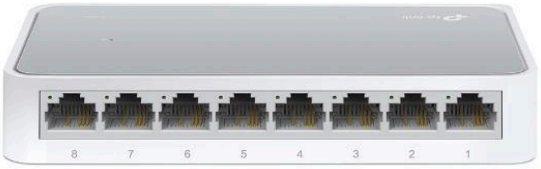


Figure 3.2.3: TL-SF1008D Network Switch

Network Topology and Configuration

The floor map diagram of the network structure is presented in Figure 3.2.4. There are three TL-WR840N routers that are strategically placed throughout the fourth floor hallways. These routers are separately connected with a CAT5E Ethernet cable that relays to the TL-SF1008D network switch that is located in the first senior high school faculty.

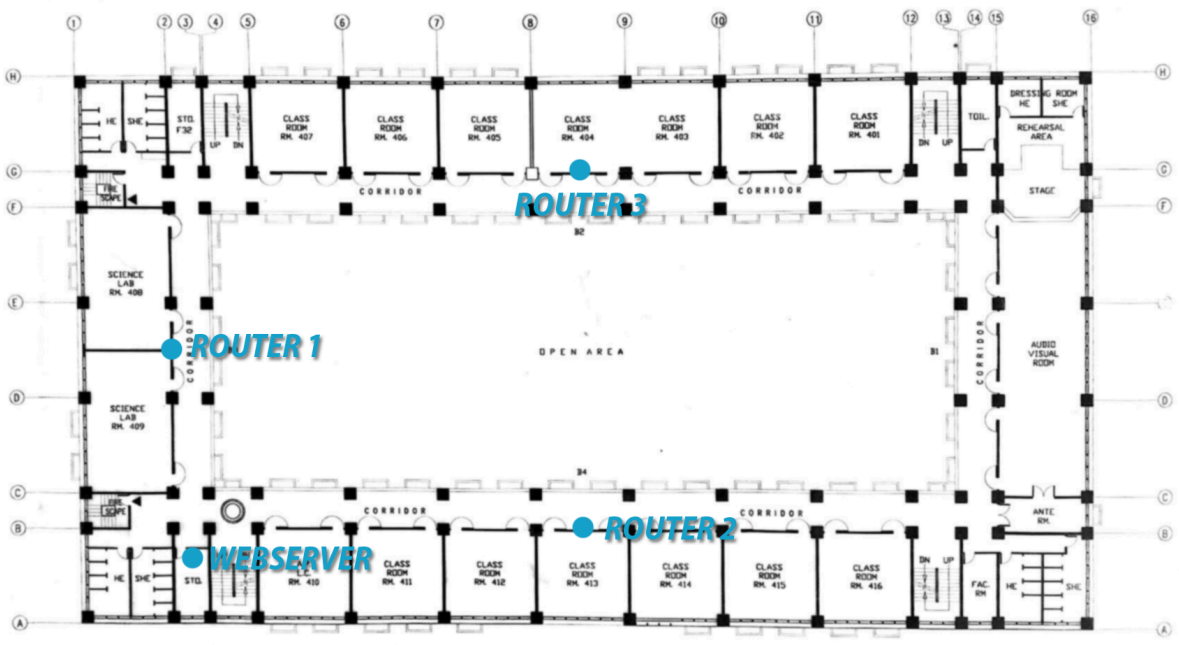


Figure 3.2.4: Network topology diagram of the Vitre network

Moreover, the specific and lower level network topology of the Vitre system is visualized in Figure 3.2.5.

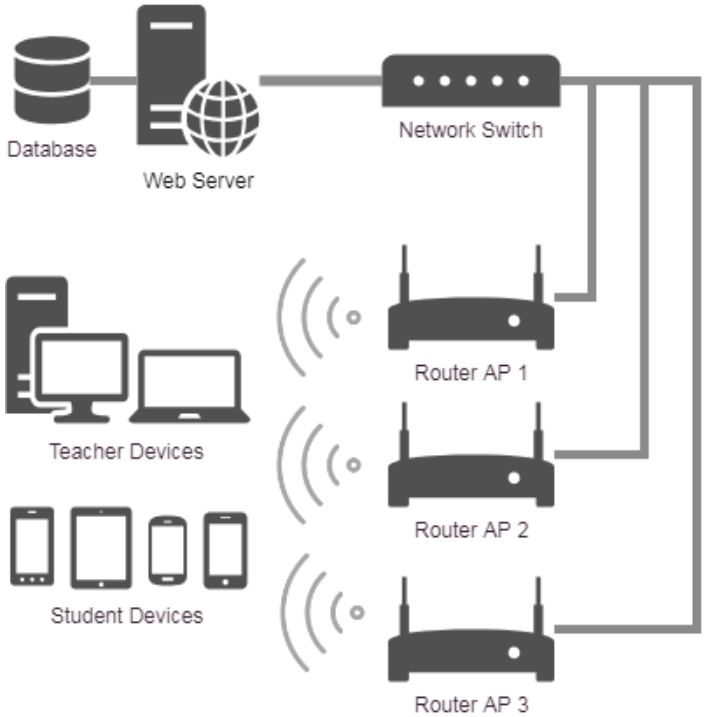


Figure 3.2.5: Network topology diagram of the Vitre network

Router Access Point 1, 2 and 3 connect to the Raspberry Pi which acts both as the web server and DNS resolver service. The configuration of these services are shown in Table 3.2.3.

Raspberry Pi Service	IP Address
Apache Web Server	192.168.0.1
BIND 9 DNS Resolver	192.168.0.1

Table 3.2.3: IP addresses of each service hosted within the Raspberry Pi

The developers manually configured each router to reference and forward the IP addresses of these services. Furthermore, the routers' DHCP service is configured to distribute IP addresses that are mutually exclusive to one another. This is done in order to prevent IP address collisions within the network when a client transitions to a nearer router. The reservation range for each associated router is shown in Table 3.2.4.

Router Access Point	IP Address	DHCP Reservation Range
AP 1 (Laboratory)	192.168.0.2	192.168.0.10 to 192.168.0.81
AP 2 (Grade 11 Hallway)	192.168.0.3	192.168.82 to 192.168.0.163
AP 3 (Grade 12 Hallway)	192.168.0.4	192.168.0.164 to 192.168.0.255

Table 3.2.4: DHCP Reservation Table for each Router Access Point

Furthermore, each router is configured to have separate wireless channels to prevent signal overlapping. This allows any connected client to easily transition and connect to the closest Vitre access point as if it is a single network signal. The wireless channel assigned to each router is presented in Table 3.2.2. Figure 3.2.4. shows the signal overlapping exhibited in each wireless channel [16]. It also presents the evidence of no signal overlapping in channels 1, 6 and 11.

Access Point	Wireless Channel
AP 1 (Laboratory)	1
AP 2 (Grade 11 Hallway)	6
AP 3 (Grade 12 Hallway)	11

Table 3.2.5: Access points and their wireless channels

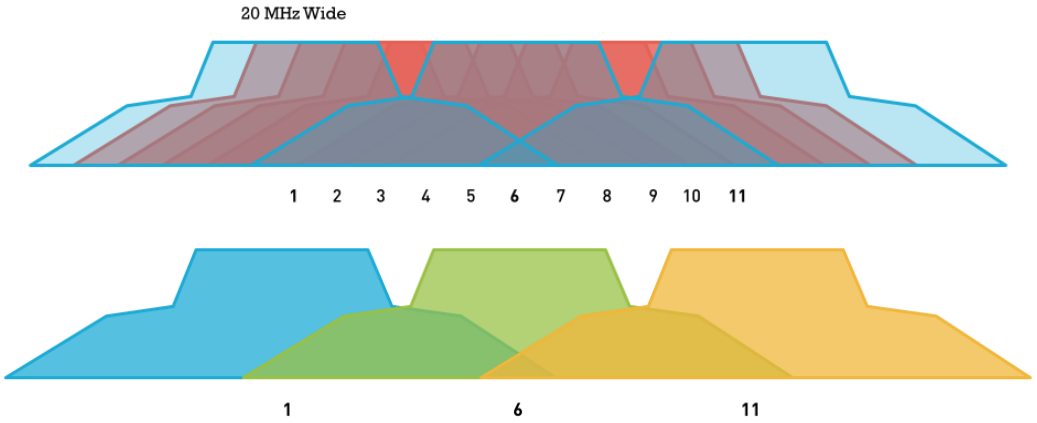


Figure 3.2.4: Channels 1, 6 and 11 showing no signal overlapping within the 2.4 GHz spectrum

3.3. Software Systems Design

The Vitre is separated into two distinct dimensions which forms the system as a whole. These sections are the web server systems that are discussed in Chapter 3.3.1. and desktop teacher applications (DTA) that are discussed in Chapter 3.3.2.

The first dimension of the project was focused on web systems development that included the frontend and backend design for the end-user. The frontend of the Vitre web server systems were primarily developed with Bootstrap 4 and Fontawesome. Backend systems were built with the foundation of PHP 7, Javascript, JSON, JQuery, AJAX, and Chart.js. As mentioned in Chapter 3.1, the web server platform uses Apache as its HTTP server and MySQL as its relational database management system.

The second dimension revolved around the development of a desktop client (dubbed as the Vitre Desktop Teacher Application or DTA) that allows the teachers' to interface with the Vitre web server. The client application was made in the Java programming language. The specific version utilized is Java Development Kit 8 Update 112. The graphical user interface is designed with the JavaFX 2.0, ControlsFX and Gluon Scene Builder. There are also myriad libraries utilized to develop the desktop application, namely:

- ❏ Apache Commons IO 2.4
- ❏ Apache Commons Lang 3.7
- ❏ Apache POI 3.17
- ❏ Apache POI-OOXML 3.17
- ❏ Dropbox Core SDK 3.0.9
- ❏ Fontawesome FX 8.15
- ❏ HTTP Core 4.3.3
- ❏ HTTP Client 4.3.3
- ❏ Hashids 1.0.3
- ❏ Java JWT 3.4.0
- ❏ JSON Simple 1.1.1
- ❏ ZJSONPatch 0.4.4

3.3.1. The Web Server Frontend

This chapter presents an in-depth overview of the web server design. It presents the interface that the students use.

Web Page Design

The webpage was designed to cater to student grade record disclosure. This comes with the thought that the students or the users would be able to view their grades through the webserver and their reactions could either be positive or negative depending on the results. This part of the system design documentation briefly discusses the design decisions that took place in forming the graphical content of the webpage as well as the content it includes in both a technical, functional and aesthetic standpoint in order to influence the students' reactions.

Design Context

In order to create an appropriate design for the system, the developers first identified the context that the system is aiming to address. In the case of the Vitre systems, the main goal is to deploy a campus wide local network that mainly promotes student to faculty transparency. The system would serve as the bridge of which will connect those two parties in order to efficiently deliver information between one another. In addition, the developers are promoting the use of more web servers in the community with the creation of the Vitre especially from an automation standpoint.

Color and Art Style

As most creations, the design started by forming the overall theme of the web server. This includes the color as well as the art style that is featured in the final product. Color is the property possessed by an object of producing different sensations on the eye as a result of the way it reflects or emits light. (Oxford Dictionaries, 2019). It is perceived through the reflection and absorption of light which contains varying wavelengths of electromagnetic waves. In the case of mobile devices, these colors are created in an additive color space with the use of the RGB (Red Green and Blue) color model.

Initially, the developers saw through a blue based design as most educational sites emphasize. This common theme is due to the color blue interpreted mentally as calming, refreshing and intelligence according to modern color theory studies. However, blue light and its negative effects result in long term eye strain due to it being conjured from a high frequency electromagnetic wave. This is backed up by the movement, Prevent Blindness, which also mentions in their article called "Blue Light and Your Eyes" that blue light may cause retinal damage and accelerated eye degeneration. Therefore, in order to mitigate this as well as continually fit the design context of the system, a contemporary color of yellow was added as an accent color for blue. This presents a more lively feeling in the design. In addition, this makes the system more welcoming in presenting the results of student grades as blue also tends to influence focused behavior which could negatively impact students whenever they receive lower scores.

The developers tried to find a shade of blue that is not too saturated thus straining the eyes while also being not too dark that it would stray attention. For that, the shade of blue with the hex value of #438196 was chosen. As for the shade of yellow, the developers made sure to not use a bright shade of yellow as it would dominate the shade of blue hence they settled with the shade of yellow with the hex value of #F0C766. Furthermore, the four other analogous shades of these colors were used for variety (Figure 3.3.1.1.).

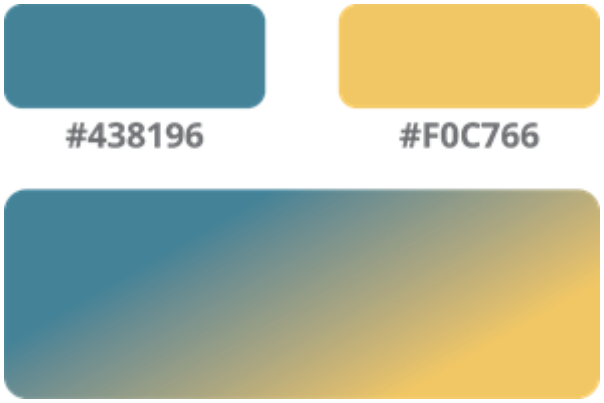


Figure 3.3.1.1: Main color scheme of the Vitre Project

For the art style of the system, the developers went for a flat and cartoony design in order to match the atmosphere that the color yellow presents.

Logo Design

The logo (Figure 3.3.1.2.) is the strongest graphical element that could be found on any creative project. It is the brief unifying element that provides the viewer the gist of the project. After a lot of thought, the developers chose to feature a padlock in order to depict student security in monitoring their grades as well as their progress their class. Since transparency exist, there is no need to worry in terms of the instability of their grades as they themselves know the outcome based on the readings presented to them by the system.



Figure 3.3.1.2: Vitre Logo

The uniform color of the logo is a design choice. It allows the element to be colorized into other colors hence allowing it to be placed on any graphic design as it could easily be converted into a contrasting graphic element.

As a contrasting choice, the developers made the webpage icon (Figure 3.3.1.3) with the yellow of the Vitre color scheme. Its counterpart is the Desktop Teacher Application icon which is based around the blue of the Vitre Color scheme. The DTA is an application that would be referenced later on this study. The structure of the icon is rounded which echoes the style that the system is going for. The gradient on the base of the icon emphasizes its roundness. In the middle of the base is the logo of the system, colorized with a brighter and contrasting shade of the Vitre yellow.



Figure 3.3.1.3: Vitre webpage icon

User Experience Design and Web Pages

The website of the system is divided into four main webpages: the home page, subject page, account page, news page. These pages are discussed and presented individually in this section.

Login Portal Page

Aside from the 4 main pages, the login page is also a key factor in designing the website. The login page is the first impression that is presented to the users which is analogically the page that welcomes them to use the system thus it was deeply thought over. This is also where, as the name suggest, the login fields are located. the login fields are the input text boxes that allows users to type in their account details in order to access their Vitre account. In the latest design, the developers opted with using an introductory welcoming art (Figure 3.3.1.4.) that lies on top of the login fields.



Figure 3.3.1.4: The introductory welcoming art

Straying from the round trend that have been seen so far, the welcoming graphic art is sharp and dynamic which presents a sense of movement and evolution. This aims to illuminate the users to the versatility and possibilities of the system. The page also features a remember me option that prolongs their session time expiration to 1 hour. This page is seen in Figure 3.3.1.5.

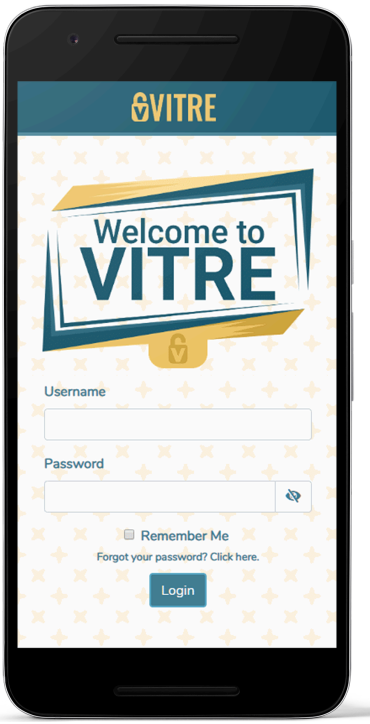


Figure 3.3.1.5: Login portal

Home Page

After logging in, the first page that the user lands in is the home page. The home page is the primary page that the students view after logging in. The homepage has had the most revisions throughout the Vitre development as the space available in the web server is limited as the screen of most phones are occupied with the status bar as well as the browser navigation bar. All graphic art ideas on the homepage was scrapped to benefit functionality and so in the latest design, the home page simply has a list of all the recent activities that have been uploaded to the system which is associated to the user.

The students can navigate through the website with the use of the navigation bar. It allows them to access the home page, my account page, my subjects page, and the logout button. The Home page generally showcases their recent activities from all the subjects that recently uploaded. Each activity card presents a summary of information about it such as the associated subject, description and resulting score. Upon clicking on an activity card, the student is redirected to the activity's subject page. The home page and navigation bar is presented in Figure 3.3.1.6.

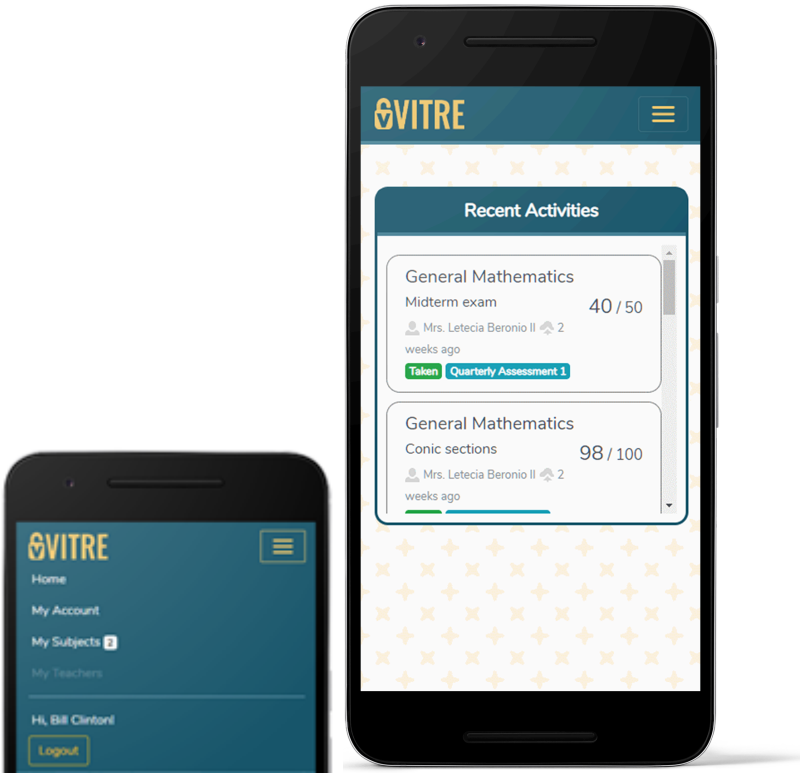


Figure 3.3.1.6: The Home Page and Navigation Bar

Subject List Page

The “My Subjects” page presents a list of subjects that the student is enrolled in. The list features a search input field that allows the student to quickly find the subject he or she wants to access. Upon clicking a subject card, it redirects the student to its associated and individual subject page. This page is seen in Figure 3.3.1.7.

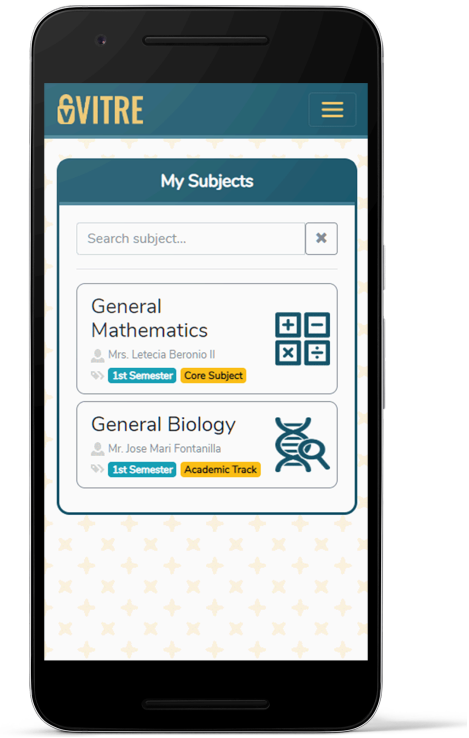


Figure 3.3.1.7: My Subjects page

Subject Details Page

The subject details page shows wholesome information about a student's progress for a particular subject. The page presents each uploaded activity per grading component, namely: written work, performance tasks and quarterly assessment. This page is seen in Figure 3.3.1.8.

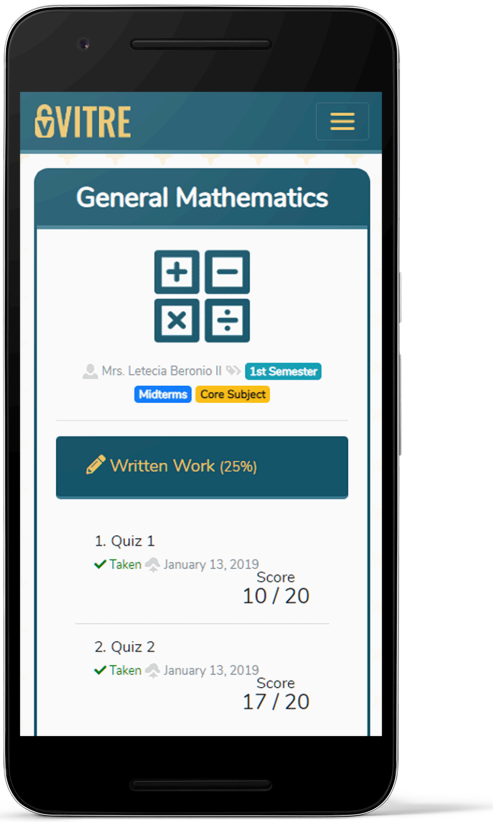


Figure 3.3.1.8: Subjects page for General Mathematics

My Account Page

The "My Account" page is dedicated for configuring a student's user account. It allows the student to modify their account username and password. It also allows them to submit a name correction request; in case of spelling errors in their name. Moreover, it holds the feature of configuring various tweaks such as enabling animations and visual effects. This page is seen in Figure 3.3.1.9.

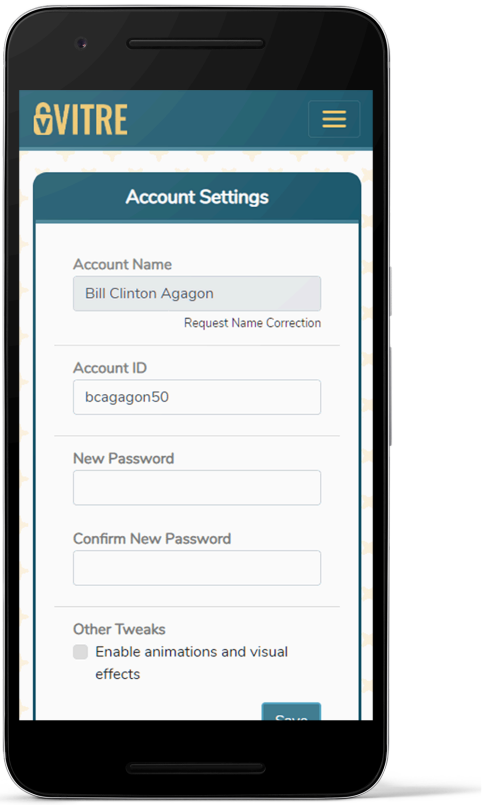


Figure 3.3.1.9: My Account page

3.3.2. The Web Server Backend

This chapter presents the design of the webserver's backend processes. It covers the primary and underlying processes occurring within the server-side. It also presents the MySQL database structure and its wholesome entity relationship diagram (EDR) in Figure 3.2.2.13.

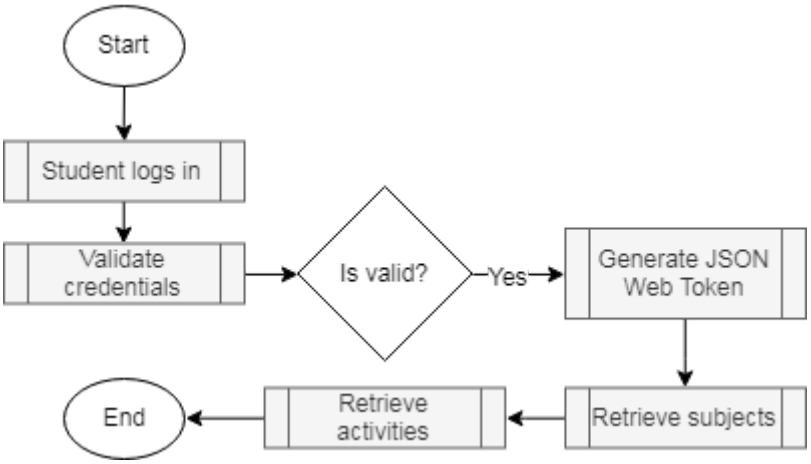


Figure 3.3.2.1: Backend processes visualized as the student logs in to the web server

Login Authentication

As the student submits his or her logic credentials, the server validates it. The server checks its MySQL database for the existence of the user and validates the password. It validates the password by generating a SHA-256 hash of the current input and then comparing it to the database stored password hash. When the validation passes, it proceeds to generate a JSON Web Token.

JSON Web Token (JWT)

JWT is a token that is stored on a student's browser upon successful login and authentication. The token is a JSON body that contains commonly needed information such as the student's server ID and associated subjects. Table 3.3.2.1. presents the interpretation of each data that is stored on a JWT.

Data	Interpretation
Username	The username of the student
First name	The first name of the student
Last name	The last name of the student
Student ID	The registered ID of the student in the server
Class ID	The registered ID of the student's associated class or section in the server
Curriculum ID	The registered ID of the student's associated curriculum in the server

Session Time In	The student's login time in milliseconds
Session Expiration	The student's login expiration time before automatically logging out from the server

Table 3.3.2.1: Data stored in a JSON Web Token

Subjects Retrieval

The server references the curriculum ID within a student's JWT. It queries the server's database, retrieves every subject associated with the curriculum ID and then stores the garnered subjects into a list.

Activities Retrieval

The server references the student ID for retrieving his or her activities. It searches the "Grades" table in the database and retrieves all the grade data associated with the student. It then queries the "Grades Metadata" table for providing descriptions in each activity. Furthermore, the database structure is elaborated and described in the next section.

Database Structure

Students Table

The students data table contains a collection of all student information that is registered within the system. Figure 3.3.2.2. shows the structure of this table.

Students	
PK	Student ID
FK	Class ID
	Student Index
	User Name
	First Name
	Last Name
	Gender

Figure 3.2.2.2: Students table structure

The student ID column refers to a unique identifier that serves as the primary key and reference point in gathering data from other tables (such as grade data) for a particular student. The class ID column is a foreign key in which it is the unique ID of a class or section that this student belongs to; this key points to the Classes table. The student index column refers to the ordinal number (when alphabetically arranged) of the student in his or her class.

Teachers Table

The teachers data table contains a collection of all teacher information that is registered within the system. Figure 3.2.2.3. shows the structure of this table.

Teachers	
PK	Teacher ID
	Teacher Chunk Key
	Teacher Name
	Honorific
	Last Login Date

Figure 3.2.2.3: Teachers table structure

The teacher ID column is a primary key that serves as the unique identifier for a registered teacher. The teacher chunk key column refers to the associated SHA-256 hash signature that identifies a teacher as discussed in the synchronization process (See Chapter C, Synchronization Process). The last login date column is a timestamp in milliseconds that contains the last login date recorded by the server.

Classes Table

The classes data table contains all registered classes or sections within the system. Figure 3.2.2.4. shows the structure of this table.

Classes	
PK	Class ID
FK	Curriculum ID
FK	Adviser ID
	Class Chunk Key
	Class Name
	Population
	Room Number

Figure 3.2.2.4: Classes table structure

The class ID column is a primary key that serves as a unique identifier for a registered class or section. The curriculum ID column is a foreign key that pertains to the curriculum or collection of subjects that the class follows; this key points to the Curricula table. The adviser ID column is a foreign key that pertains to the adviser of the class; this key points to the Teachers table. The class chunk key column refers to a SHA-256 hash signature that identifies a class in the synchronization process.

Subjects Table

The classes data table contains all registered subjects within the system. Figure 3.2.2.5. shows the structure of this table.

Subjects	
PK	Subject ID
FK	Instructor ID
FK	Track ID
	Subject Chunk Key
	Subject Name
	Semester
	Icon Name

Figure 3.2.2.5: Subjects table structure

The subject ID column is a primary key that serves as a unique identifier for a registered subject. The instructor ID column is a foreign key that pertains to the teacher that teaches on a subject; this key points to the Teachers table. The track ID column is a foreign key that refers to the track a subject classifies into; this key points to the Tracks table. The subject chunk key column is a SHA-256 hash signature that identifies a subject in the synchronization process. The semester column pertains to the semester that this subject entails to. The icon name refers to the file name that the server references in displaying for the subject list page.

Curricula and Curricula Metadata Table

The curricula data table contains all registered curricula within the system. Within the senior high school system of Pitogo, there are three recognized and registered curriculums, namely: the STEM, HUMSS and ABM curriculum. Figure 3.2.2.6. shows the structure of this table.

Curricula	
PK	Curriculum ID
FK	Curriculum Metadata ID
FK	Subject ID

Figure 3.2.2.6: Curricula table structure

The curriculum ID column is a primary key that refers to a unique identifier for a curriculum. The curriculum metadata ID column is a foreign key that points to the Curricula metadata table whereas it describes it. Figure 3.2.2.7. shows the structure of the metadata table.

Curricula Metadata	
PK	Curriculum Metadata ID
FK	Curriculum Name

Figure 3.2.2.7: Curricula metadata table structure

Tracks Table

The tracks data table contains all registered tracks within the system. Figure 3.2.2.8. shows the structure of this table.

Tracks	
PK	Track ID
	Track Name
	WW Weight
	PT Weight
	QA Weight

Figure 3.2.2.8: Tracks table structure

The track ID column is a primary key that uniquely identifies a track. WW weight pertains to the weighted percentage (as a floating data point) of written works. In consonance, PT weight pertains to the weighted percentage of performance tasks while QA weight pertains to weighted percentage of quarterly assessment grading component.

Grades and Grades Metadata Table

The grades data table contains all uploaded grade data within the system. The table is a cumulative collection of all grade data from all subjects and sections. Figure 3.2.2.9. shows the structure of this table.

Grades	
PK	Grade ID
FK	Student ID
FK	Grade Metadata ID
	Grade

Figure 3.2.2.9: Grades table structure

The grade ID column is a primary key that uniquely identifies a grade data point. The student ID column is a foreign key that refers to the owner of this grade; this key points to the Students table. The grade column refers to the actual grade value or score garnered by a student. The grade metadata ID column is a foreign key that references the associated metadata of this grade data point; this key points to the Grades metadata table whereas it supplements and describes the grade data further. The structure of the metadata table is shown in Figure 3.2.2.10.

Grades Metadata	
PK	Grade Metadata ID
FK	Subject ID
	Item Index
	Component Index
	Component
	Upload Date
	Description
	Highest Possible Score

Figure 3.2.2.10: Grades metadata table structure

Grades metadata ID column refers to a primary key (as String text data point) that serves as a unique identifier for a grade metadata. The metadata table is the cached version of a sent patch context (See Chapter C, Parsing). The item index column refers to grade data's ordinal column number from its original class record. The component index refers to the grade data's origin and zero-based component index as shown in Table 3.2.2.2.

Grading Component	Index
Written Work	0
Performance Tasks	1
Quarterly Assessment	2

Table 3.2.2.2: Associated indices of each grading component

The component column refers to a component name that this grade data originated from. The upload data column refers to a timestamp in milliseconds that entails to when the grade data was patched or uploaded. The description column is a short text that describes what activity transpired for this particular grade data. The highest possible score (HPS) column is derived from the class record's HPS columns.

Analytics Table

The analytics data table refers to a collection of statistical data that aids the developer's further research. At the moment, usage analytics are able to be reliably monitored. Figure 3.2.2.11. shows the structure of the analytics table.

Analytics	
FK	Student ID
	Time In
	Time Out
	Session Time

Figure 3.2.2.11: Analytics table structure

The student ID column is a foreign key that shows ownership of an analytical data point; this key points to the Students table. The time in column is a millisecond timestamp that records a student's log in activity. The time out column is a millisecond timestamp that records a student's log out activity. The session time column is computed by subtracting the time in and time out columns, respectively.

Tickets Table

The tickets data table refers to all submitted support tickets within the system. The developers implemented the table in the essence of maintaining the quality of the software system. Figure 3.2.2.12. shows the structure of the tickets table.

Tickets	
PK	Ticket ID
	Ticket Owner
	Ticket State
	Ticket Subject
	Ticket Description
	Ticket Stamp

Figure 3.2.2.12: Tickets table structure

The ticket ID is a primary key that uniquely identifies a ticket. The ticket owner refers to the name of the ticket's submitter. The ticket state column refers to the status of the ticket which could be resolved, unresolved or unchecked. The ticket subject and description describes the issue or concern regarding the system.

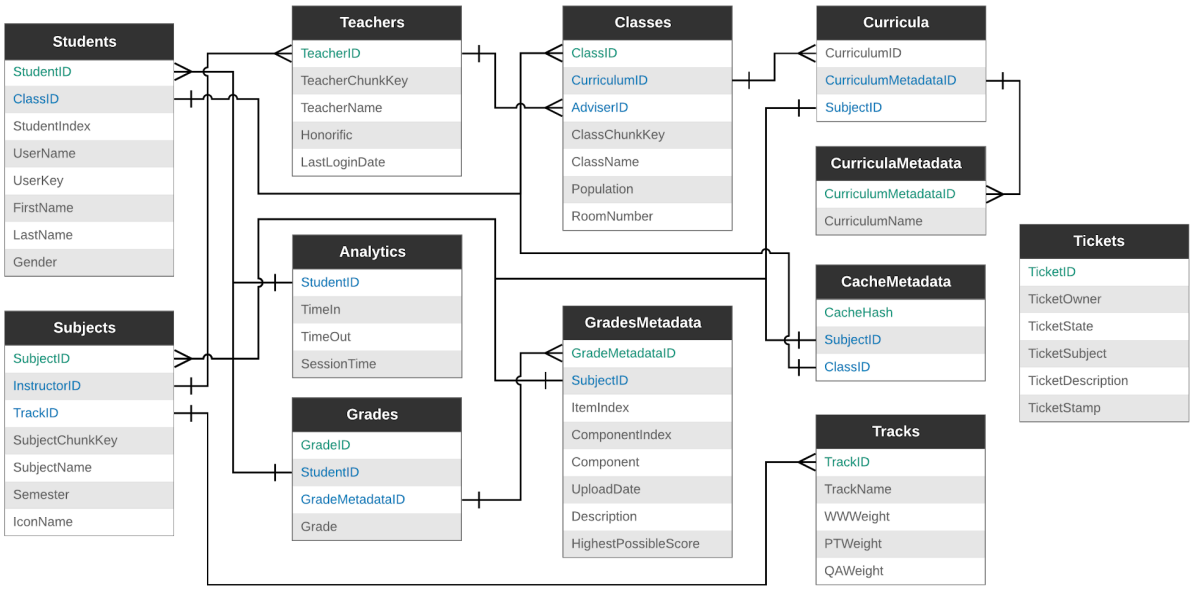


Figure 3.2.2.13: Entity relationship diagram of the Vitre SQL database

3.3.3. The Desktop Teacher Application

This chapter presents an in-depth overview of the system design of the Vitre Desktop Teacher Application software. It tackles upon its graphical user interface and back-end design.

The Vitre Desktop Teacher Application (DTA) is a program that the teachers mainly utilize to interface with the Vitre server. It is equipped with many tools necessary for related tasks such as reading class records, registering students and sending grade data for the server. It also features an auto-updater system that is used for maintaining the application up-to-date. Its graphical user interface is presented in Figure 3.3.3.1.

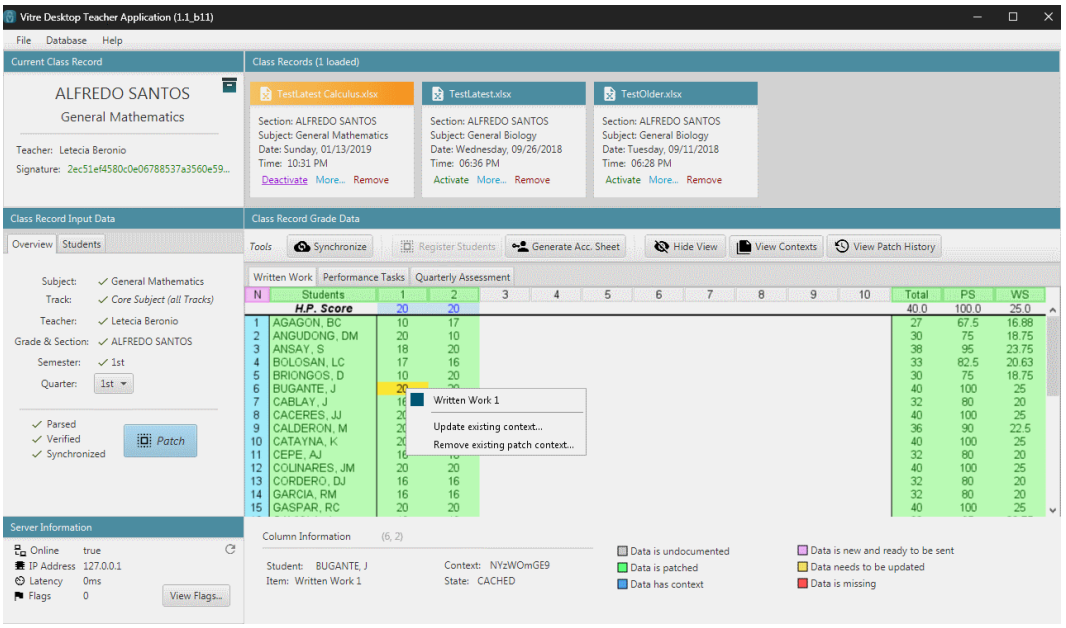


Figure 3.3.3.1: The graphical user interface of the DTA

Application Parts

The DTA contains parts that the user and the system interacts with. The labeled parts are shown in Figure 3.3.3.2.

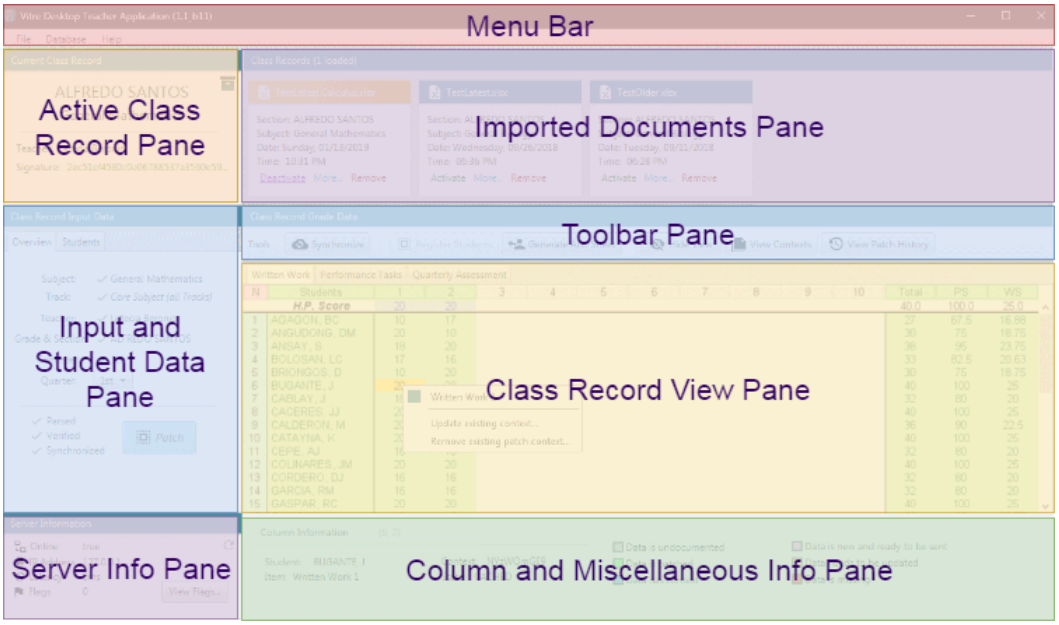


Figure 3.3.3.2: The graphical user interface of the DTA

The parts of the DTA user interface are elaborated and described:

(1) Menu Bar – contains the File, Database and Help menus.
(2) Active Class Record Pane – contains information about the currently loaded class record into the system. It presents the class record's associated section, subject, teacher and digital signature.
(3) Imported Documents Pane – contains the imported class records. Each import is called a Document Pane that could be activated and removed.
(4) Input and Student Data Pane – contains parsed information from the class record's input data fields; as well as the computed student list.
(5) Toolbar Pane – contains the actions and tools that the user uses in manipulating the class record.
(6) Class Record View Pane – contains the simplified view of a class record that allows the user to see the current synchronization state and patch contexts.
(7) Server Info Pane – contains information about the server's state; as well as current flag or warning reports.
(8) Column and Miscellaneous Info Pane – contains information about a highlighted grade column and interpretation about different cell colors.

Electronic Class Records

The Department of Education (DepEd) provided electronic class records in which public school teachers utilize for computing and organizing their students' grade data [3]. The DTA, by default, primarily utilizes these e-class records for its processes. Although the system could support many other class record types (See Chapter C), the developers ought to mainly support this e-class record since majority of the Pitogo Senior High School faculty uses it. The DepEd senior high school e-class record was released in 2015 [3]. This specific e-class record is included and pre-configured by default in the DTA. The following figures (Figure 3.3.3.3. to 3.3.3.5.) present sheets shown within Microsoft Excel as well as its intended uses.

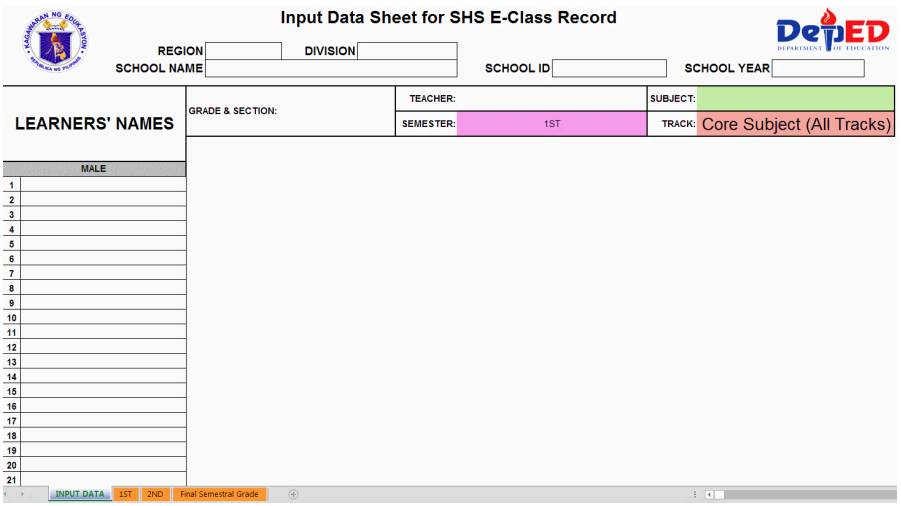


Figure 3.3.3.3: Input data sheet of an e-class record as viewed in Microsoft Excel

The input data sheet is the first page in the e-class record (Figure 3.3.3.3.). This is where the teachers supply basic information about this record such as the subject and section it pertains to. It also contains a list of student names; separated by a male and female list. The fields inputted, specifically the teacher, subject and section, would serve as the basis in identifying and authorizing a record towards the server.

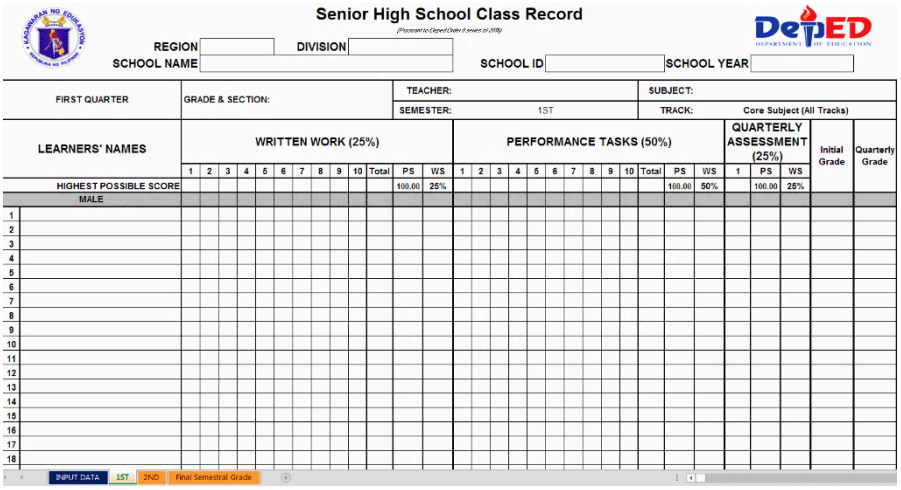
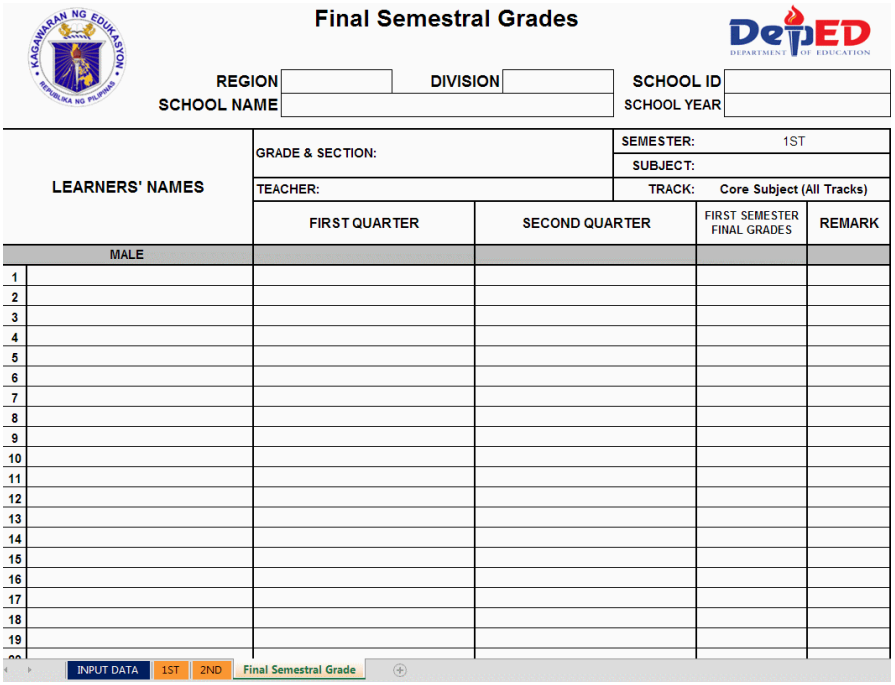


Figure 3.3.3.4: Semester grade sheet of an e-class record as viewed in Microsoft Excel

The semester grade data sheet is the second and third page in the e-class record. Each grade data sheet accommodates the midterm and final terms respectively. This is where the DTA collects all student grade data through parsing and class record activation (See Chapter 3.3.4.).



Final Semestral Grades

REGION | DIVISION | SCHOOL ID
SCHOOL NAME | SCHOOL YEAR

LEARNERS' NAMES	GRADE & SECTION:	SEMESTER: 1ST
	TEACHER:	SUBJECT: / TRACK: Core Subject (All Tracks)

		FIRST QUARTER	SECOND QUARTER	FIRST SEMESTER FINAL GRADES	REMARK
	MALE				
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					

INPUT DATA | 1ST | 2ND | Final Semestral Grade

Figure 3.3.3.5: Final semester grade sheet of e-class record as viewed in Microsoft Excel

The final semester grade sheet is the fourth page in the e-class record (Figure 3.3.3.5.). This page contains a final and cumulative grade average that each student garnered. It also interprets these final grades with a certain remark. The DTA formerly included this sheet in collecting grade data but was excluded due to the nature of grade transmutations being volatile.

3.3.4. The Desktop Teacher Application Processes

The Patch Cycle

The main function of the DTA is to properly and automatically parse class records in order to be sent to the Vitre server. The patch cycle (Figure 3.3.4.1) is a repeated task within the DTA where it describes how the system processes a class record. Every imported class record undergoes the same cycle. Each predefined process is briefly discussed in the latter part of this chapter.

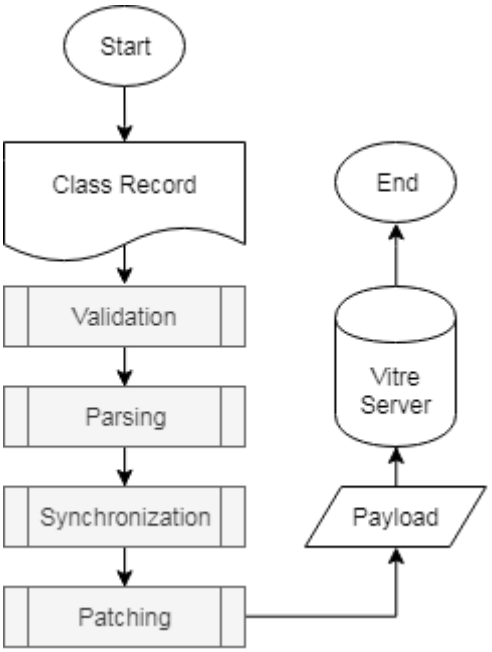


Figure 3.3.4.1: The upload process methodology

The cycle starts when the user imports a class record document to the system. The imported document is validated to make sure that it is appropriate and structured as a class record.

After validation, the document is fully parsed and read into the system. The user sees a simplified version of the record through the Record View Pane.

Afterwards, the user attempts to synchronize the class record with the lastly uploaded class record cache within the server. It infers the differences between the record versions and generates a patch file containing the instructions, signatures and content body of it. This patch file is encrypted and sent to the server via a HTTP POST request. The server then entertains the patch file where it commits each instruction to its database where the student can now view his or her associated grade.

The Validation Process

Before any document gets imported to the DTA system, all documents undergo a validation process. This is done to prevent complications and unwanted errors that may arise from using improper document formats. The validation process is attributed to the purpose of the import manager.

Import Manager

The import manager inspects every imported document before it is duly added to the system. There are two ways to import a document to the system. One is to use the drag-and-drop feature and another is to open the file system dialog. It has two criterions in assessing whether an imported document is valid:

(1) The document should be a Microsoft Excel document.

(2) The document should be following a known class record format.

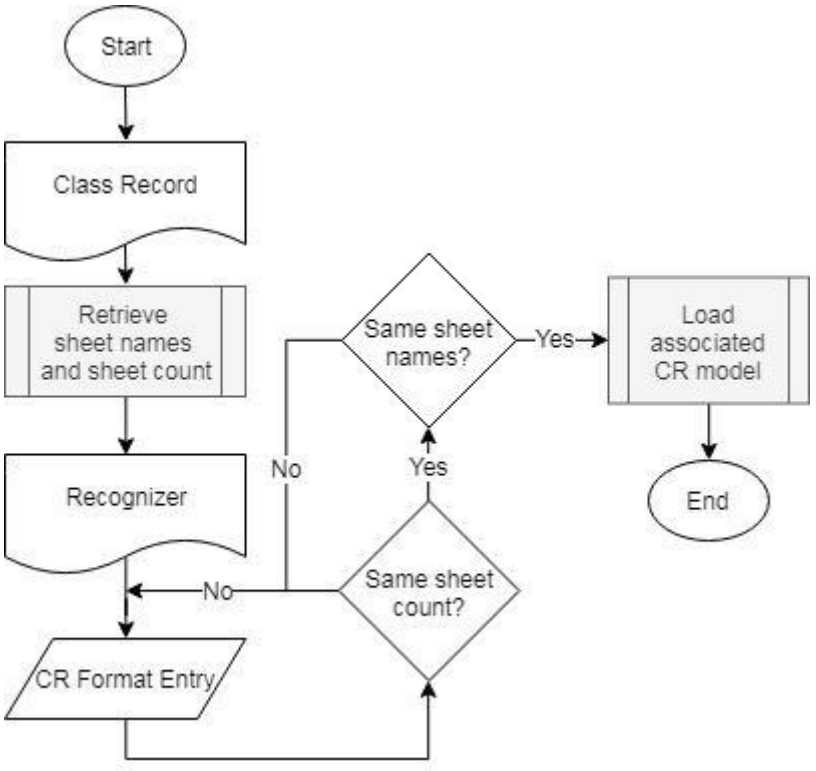


Figure 3.3.4.2: The selection process in identifying a class record format

This process is further visualized in Figure 3.3.4.2. It does this by acquiring the total number and names of each sheet in the spreadsheet. It uses this data for the recognizer files which contain known class record formats. With this, it will find its applicable class record model by comparing its data to all the listed formats. It will loop through all the known formats and once it finds the closest format, it then imports it to the system along with the class record model that it will reference.

Specifically, the import manager selects the appropriate format for the imported class record by checking whether its sheet count is equal to the compared format's "sheet_count" parameter; as well as whether its sheet names are equal to the format's "sheets" text array.

Record Imports Model

The record imports model is a JSON file (Figure 3.3.4.3.) that contains data about imported class record documents within the DTA. It is read beforehand by the system in order to automatically add known documents to the DTA imported document list at startup.

imported.json

```
   "TestLatest.xlsx":{
      "schema":"ECRDeped2015",
      "path": "D:\\Users\\Bob\\Desktop\\TestRecord.xlsx",
      "context": "D:\\Users\\Bob\\.vitre\\bin\\contexts\\TestRecord",
      "index": 1
   },
```

Figure 3.3.4.3: Example referenced entry of an imported document

Recognizer File

The recognizer file contains known class record formats. In the course of development, the developers manually registered common class record templates. The DTA references this file in checking whether the imported Excel document files are valid and actually following a class record format. Figure 3.3.4.4. shows the contents of a recognizer file.

classrecords.recog

```
[ECRDeped2015]
config_path = local:ecr_deped2015.crcm
sheet_count = 5
sheets = "INPUT DATA, 1ST, 2ND, Final Semestral Grade,
Helper(IMPORTANT!)"
```

Figure 3.3.4.4: Example contents of a recognizer file named classrecords.recog

In each registered format, an entry is declared with an encapsulated block that would be the name of the entry. In figure F, the “ECRDeped2015” format contains three parameters, namely: “config_path”, “sheet_count” and “sheets”. The interpretation of these parameters are shown in Table 3.3.4.1.

Parameter	Description
config_path	The path of the class record configuration model that this entry is associated with. This parameter has two supported modes: ● local - the configuration file is within the same directory. This field requires the file name only. ● absolute - the configuration file is within another directory. This field requires an absolute file path.
sheet_count	The number of total sheets in this Excel document.
Sheets	A text array containing the names of each Excel sheet; including its hidden sheets.

Table 3.3.4.1: Configuration parameters of a recognizer file

Class Record Configuration Model

The class record configuration model file (CRCM) is a configurable mapped representation of a class record's contents. It is fundamentally referenced for locating cells and then parsing the record into a simplified data structure.

The CRCM file follows the A1 reference style, the same style Microsoft Excel uses, for positioning and locating spreadsheet cells. Moreover, the style refers to columns as letters (A through IV, for a total of 256 columns), and rows as numbers (1 through 65,536) [R].

The default CRCM (Figure 3.3.4.5.), included in all Vitre Desktop Teacher Application distributions, maps the commonly used 2015 DepEd electronic class record (See Chapter C).

ecr_deped2015.crcm

```
[S0]
# Input Data Sheet
region = G4
division = O4
schoolName = G5
schoolID = X5
schoolYear = AG5
gradeSection = K7
teacher = S7
semester = S8
subject = AE7
track = AE8
maleStudents = B13 [ 62
femaleStudents = B64 [ 113

[S1]
# GRADING SHEET
# Proper Names
writtenWork = F9
performanceTask = S9
quarterlyAssessment = AF9

# Configuration
~sheetQuarter1 = 1
~sheetQuarter2 = 2
~rowMale = 13
~rowFemale = 64

# Input Columns
~colWrittenWork = F > R
~colPerformanceTask = S > AE
~colQuarterlyAssessment = AF > AH
~colAll = AI > AJ
~colOffset = 5

# Score Columns
~scoreColCountWrittenWork = 3
~scoreColCountPerformanceTask = 3
~scoreColCountQuarterlyAssessment = 2
~scoreColCountAll = 0

~rowAtleastBeforeBlank = 10
~enableConfig = 0
```

Figure 3.3.4.5: An example of a CRCM file for the 2015 DepEd e-class record

Most of the CRCM file maps cell coordinates and manually-set values for certain variables. The defined values use functions that work altogether in informing the DTA where and how to extract cell values from a class record. The functions' syntax and interpretations are listed in Table 3.3.4.2.

Syntax	Interpretation
[Sx]	**Sheet instantiation block** ▪ Where x is the zero-based sheet number of the class record. ▪ All configuration settings after this block would gather data from the set sheet page number.
#	**Comment instantiation** ▪ The coordinate file would ignore this line and is commonly given to provide documentation to the user.
~	**Key-Value Pair instantiation** ▪ This is used to set "key → value" variables; such as single-valued variables. ▪ It is acquired later by the program for defining properties such as the threshold for identifying empty columns or rows.
x [y	**Vertical or row scan operation** ▪ Where x is the starting row and y is the ending row to read cell data from. ▪ This is used to tell the program to read cell values on a class record by row on a single column.
x] y	**Horizontal or column scan operation** ▪ Where x is the starting column and y is the ending column to read cell data from. ▪ This is used to tell the program to read cell values on a class record by column on a single row.
x > y	**Range or "to" scan operation** ▪ This is used to tell the program to read cell values from either a column or row range. ▪ Both x and y values should be on the same axis (column → column or row → row).

Table 3.3.4.2: Parameters, syntax and functions of a CRCM file

The Parsing Process

Parsing is a process that involves extracting data from a document and converting it to a simplified form that could be understood and processed by the DTA. After successful importing of a document, it is then parsed and read by the DTA.

The application uses the Java library named Apache POI [1] (version 3.9) for processing and interfacing Microsoft Office formats, such as Word, PowerPoint and Excel. The Apache POI library specifically allowed the DTA to process Microsoft Excel documents into usable and programmable information. With this library, it allowed the developers to retrieve information from each individual cell in an Excel document. In addition, POI-HSSF [2] (version 3.17) is a specific subcomponent and extension of the POI library that allows parsing of XSSF-based [2] (.xlsx and .xls formats) Microsoft Excel document files. This was an introduced format by 2007 and above Microsoft Excel versions which is also the format of an electronic class record.

There are two sub-processes involved in parsing a class record: (1) previewing and (2) activating.

Previewing a class record

This sub-process is a subtle type of parsing whereas it entails upon retrieving only basic information from a class record; such as information about its teacher, subject and class names. Retrieved data from previewing a record is supplied to each Document Pane that would inform the user which class record is currently activated and could be activated (Figure 3.3.4.6.).

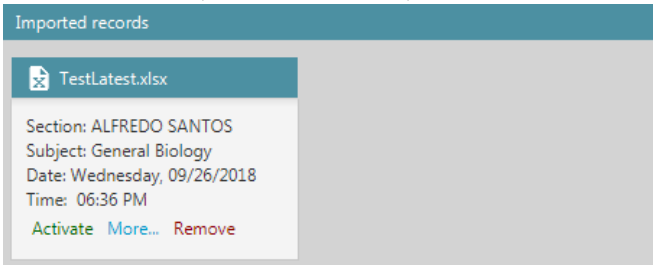


Figure 3.3.4.6: Previewed data shown in the Document Pane

Activating a class record

This sub-process involves fully parsing a document whereas the entirety of the class record is read. Activation occurs as the user clicks on the activation button located on each Document Pane. In the activation process, there are multiple steps involved:

(1) It extracts data from the Input Data sheet for acquiring basic information from the record such as its teacher, section and subject.
(2) It collects each individual student name from the record's student list.
(3) It extracts the grades from each grading component, namely: written works, performance tasks and quarterly assessment.
(4) It initializes the context manager of the record.
(5) It renders and interprets the collected data into the DTA's Class Record View Pane (Figure 3.3.4.7.).

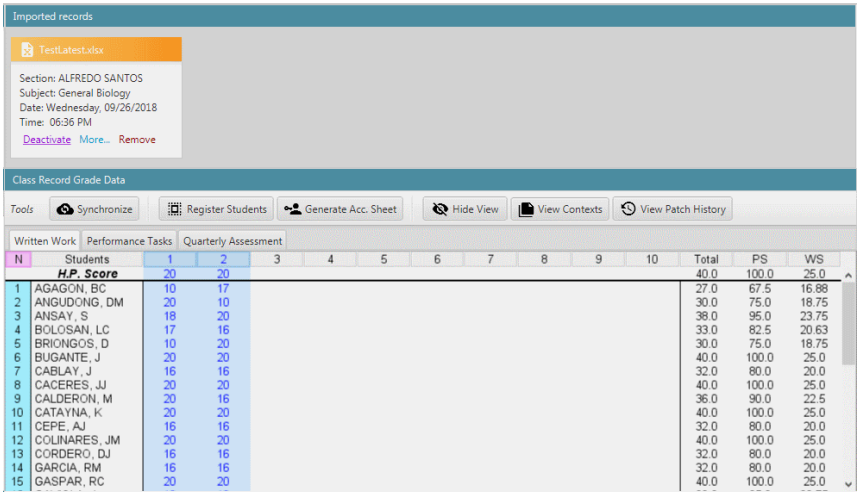


Figure 3.3.4.7: Activated class record shown in the Class Record View Pane

After class record activation, the user is granted with the capability to create patch contexts through the use of the Context Manager.

Context Manager

The context manager handles data about describing each grade column. This allows the students to know what activity transpired from a certain grade they received. Each imported class record has their own context manager within the system. The user defines a context through the Patch Context Settings window (Figure 3.3.4.8. to 3.3.4.9.).

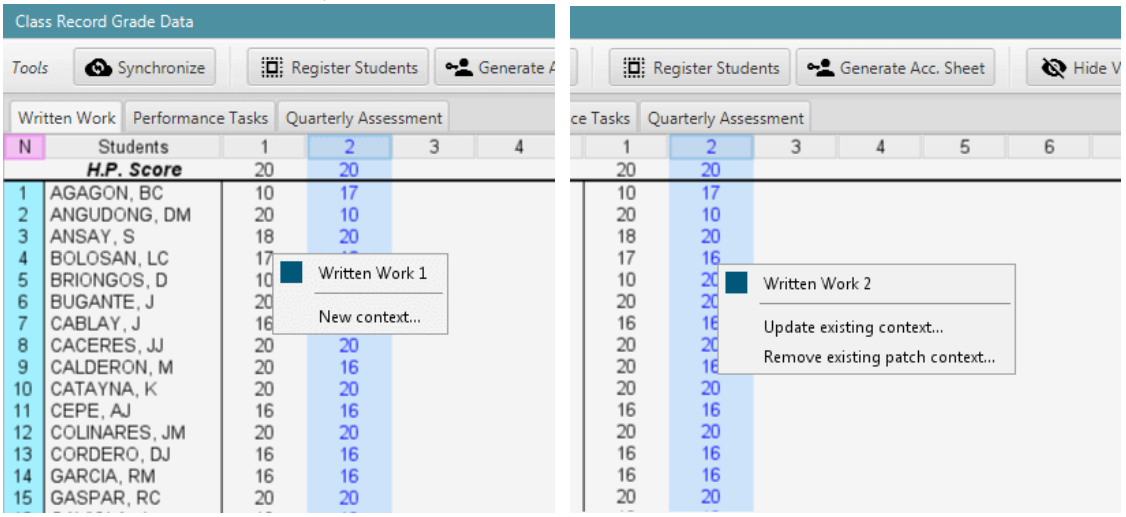


Figure 3.3.4.8: Context actions menus, triggered by right-clicking the view pane.

The user creates a patch context for a certain grade column by right-clicking on it. It is presented with various options such as creating, updating and removing a patch context. Grade columns with available contexts are highlighted as blue within the view pane. The latter section discusses the interpretation of each cell rendering.

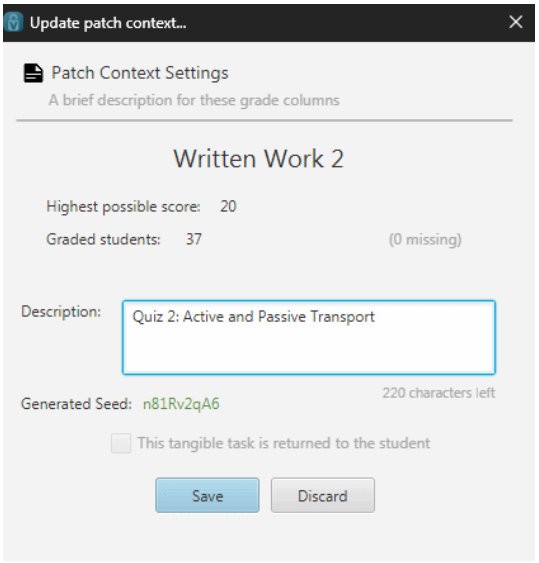


Figure 3.3.4.9: Patch context settings window

Multiple types of data are stored in a single patch context. The context manager prepares these data for retrieval and referencing when the record is reactivated once again. These data include: (1) indices, (2) unique hash identifier, (3) column's grading component index and item order, (4) highest possible score and (5) description, and (6) creation timestamp in milliseconds. An example of a context data body, which is formatted in JSON, is presented in Figure 3.3.4.10.

```
"context": {
    "DElyJQW6e": {
            "componentIndex": 0,
            "component": "Written Work 2",
            "cap": 20,
            "itemIndex": 1,
            "contextIndex": 0,
            "description": "Quiz 2: Active and Pass...",
            "stamp": 1547389956293
        },
```

Figure 3.3.4.10: Stored data within a patch context.

The Synchronization Process

Synchronization is a process that involves communicating with the server in order to know the current state and authenticity of a class record. There are four sub-processes involved in synchronization: (1) network ID retrieval and content crown creation, (2) content body creation, (3) content differentiation and (4) user interface update.

The first sub-process in synchronization is retrieving the record's registered network IDs as visualized in Figure 3.3.4.11.

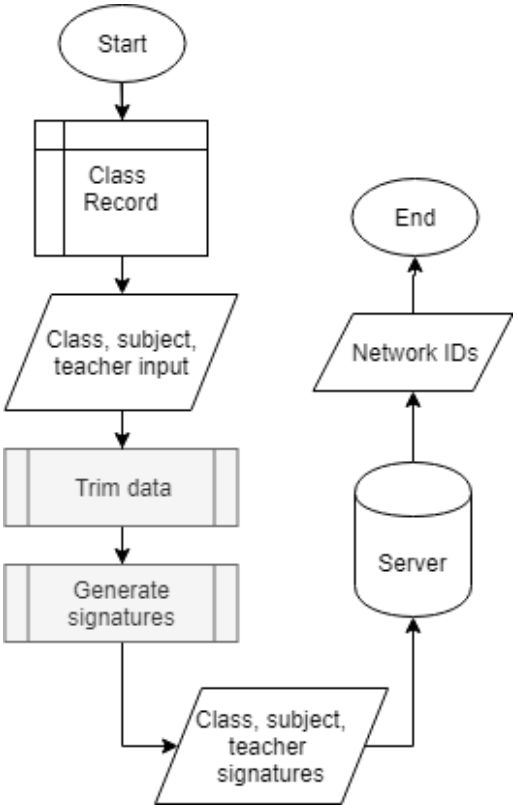


Figure 3.3.4.11: Acquiring network IDs from the server.

Before retrieving the network IDs, the class, subject and teachers input data are reduced and formatted through a Regex string expression. The expression functions by removing all non-alphanumeric characters from a text or string data. These trimmed inputs are referred to as "chunks" and used to generate a SHA-256 digital signature hash for each input. The results of each string manipulation are shown in Table 3.3.4.3.

Input String Data	Trimmed String Data	Hashed String Data (Truncated)
12 – STEM	12stem	2F16F7B2AEF36DAF52F8952FBB99B…
General Biology 2	generalbiology2	E135FDEE536D1BF2A93A10C125B77…
Jose Mari Fontanilla	josemarifontanilla	649C53070E73B6CDFA7008F6794C6…

Table 3.3.4.3: Results of manipulating the string data from each operation.

After generating the hashes, it communicates with the server for acquiring its registered network IDs. The class, subject and teacher signatures are then sent to the server where it attempts to find matching signatures within its database and reply with found registered IDs. The acquired network IDs are then used to construct a content crown.

Content Crown

The content crown is a part of a content body which contains all authentication data needed such as the class record's network identifiers, signatures, chunks, and raw data from the class, subject and teacher input fields. The record's primary SHA-256 digital signature is generated from each retrieved network ID (Figure 3.3.4.12). The signature serves as the key in accessing its cache files stored in the server and the unique identifier of a class record.

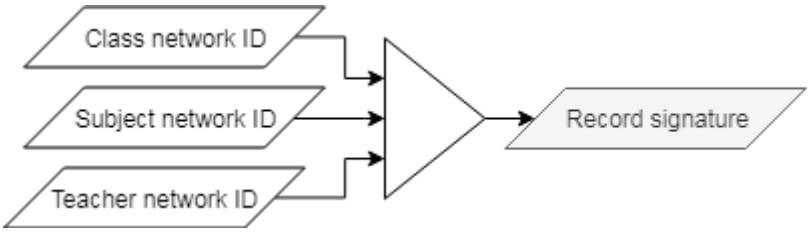


Figure 3.3.4.12: Record signature generation.

Content Body

The second sub-process in synchronization is the creation and preparation of a content body file. After crown creation, the crown, patch contexts and grade data of the record is attached to the content body. The content body refers to a JSON formatted file that encapsulates the actual data and metadata needed to be transferred to the server. Figure 3.3.4.13. shows the whole composition of a content body JSON file.

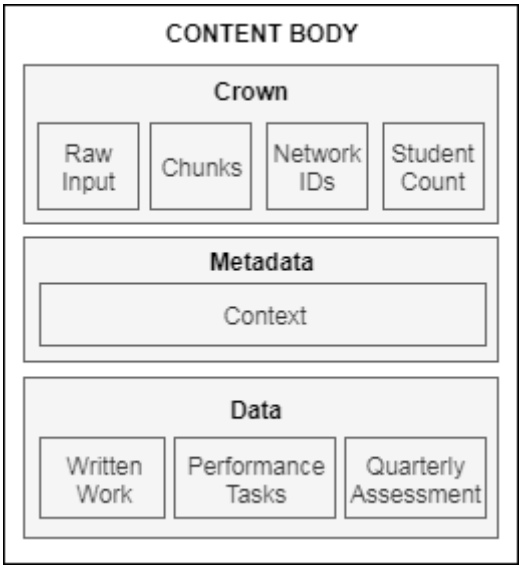


Figure 3.3.4.13: Composition of a content body.

There are three partitions of a content body: (1) crown, (2) metadata and (3) data. The crown partition refers to the class record's authentication data. The metadata partition comprises the patch contexts defined for the class record. The

data partition comprises all the read grade data from each grading component in the class record.

Content Difference

The third sub-process in synchronization is comparing the user's current class record content to the server's last cached class record content (Figure 3.3.4.14.).

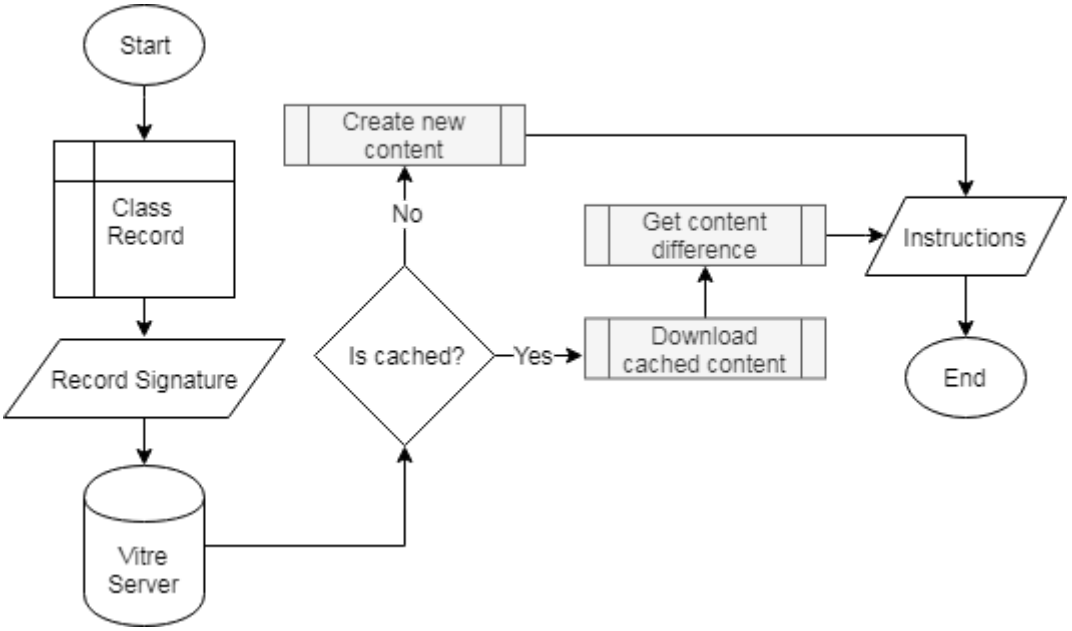


Figure 3.3.4.14: Patch difference generation.

The DTA attempts to download an existing cached content file of its record from the server using the currently active record signature. The difference, if any, between these data models is inferred in order to generate a set of instructions needed to make some changes to the currently cached data model. In the case of no downloaded cache files to compare with, it creates a full instruction set describing the current state of the record. These set of instructions are inserted to the payload in completing the patch request. The next section discusses the specific structure and formatting of these instruction sets.

Instruction Sets

Each instruction follows a strict format that the server expects and directly reads. This format contains 7 parameters. The parameters are separated by a comma (,) while each instruction is separated by a semicolon (;). An example of an instruction set is presented in Figure 3.3.4.15. and visualized in Figure 3.3.4.16.

```
1   ADD, ITEM, 1, 0, 0, 10;
2   SET, ITEM, 1, 0, 1, 10;
```

Figure 3.3.4.15: An example of a two series of instructions.

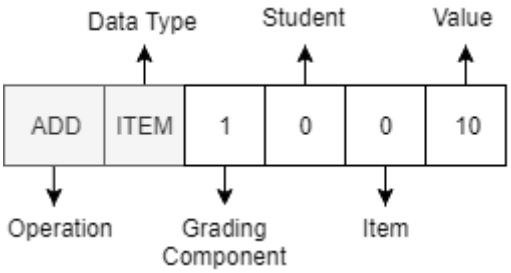


Figure 3.3.4.16: An instruction and its visualized parameters

In Figure 3.3.4.16, line 1 pertains to adding a new data point for the Written Work component. Moreover, it belongs the 2nd student of the class who scored 10 points for the first written work activity. Line 2 pertains to updating an existing data point in the server with a new value. Table 3.3.4.4. elaborately describes each specific parameter in a patch instruction.

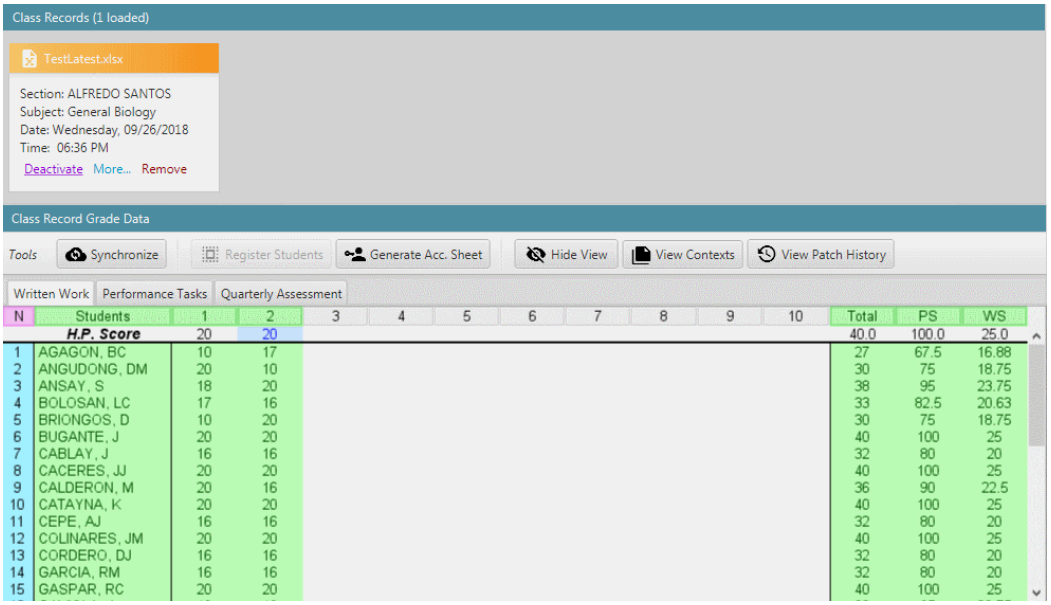
Parameter	Value
1	**Patch Operation** ▪ It refers to the specified operation that the server would do with this data point. There are four supported operations: ▪ ADD is an addition operation that informs the server to insert this new data to its database. ▪ REM is a removal operation that informs the server to remove this new data from its database. ▪ SET is an update operation that informs the server to replace an existing data point with a new value on the server's database. ▪ NONE is an operation that does nothing. It also implies that this data point is up-to-date and does not have any changes needed.
2	**Data Type** ▪ It refers to the type of data this instruction pertains to. There are two types of data: ▪ The data type "ITEM" refers to an individual grade data point for a single student. ▪ The data type "AGGREGATE" refers to a cumulative grade data point that could either be the Total Score, Percentage Score, and Weighted Score of a grading component.
3	**Grading Component Index** ▪ It refers to the grading component this instruction targets. There are three values that refer to different grading components: ▪ Value 0 refers to Written Work component. ▪ Value 1 refers to Performance Tasks component. ▪ Value 2 refers to Quarterly Assessment component.
4	**Student Index** ▪ It is a zero-based index of a student which this

	instruction targets. ▪ The instruction informs the server to set the owner of the data point to this index.
5	**Item Index** ▪ It is a zero-based index of an item which this instruction targets. ▪ The instruction informs the server to set the item of the data point to this index.
6	**Value** ▪ It refers to the value that this data point carries. ▪ It is typically the raw grade score from the class record.

Table 3.3.4.4: Parameters and values of a single instruction

User Interface Update

After the instruction set is generated, the fourth sub-process occurs which is updating the graphical user interface. The DTA renders the difference to the Class Record View Pane (Figure 3.3.4.17.) in order to notify the user about the class record's current status to the server.



N	Students	1	2	3	4	5	6	7	8	9	10	Total	PS	WS
	H.P. Score	20	20									40.0	100.0	25.0
1	AGAGON, BC	10	17									27	67.5	16.88
2	ANGUDONG, DM	20	10									30	75	18.75
3	ANSAY, S	18	20									38	95	23.75
4	BOLOSAN, LC	17	16									33	82.5	20.63
5	BRIONGOS, D	10	20									30	75	18.75
6	BUGANTE, J	20	20									40	100	25
7	CABLAY, J	16	16									32	80	20
8	CACERES, JJ	20	20									40	100	25
9	CALDERON, M	20	16									36	90	22.5
10	CATAYNA, K	20	20									40	100	25
11	CEPE, AJ	16	16									32	80	20
12	COLINARES, JM	20	20									40	100	25
13	CORDERO, DJ	16	16									32	80	20
14	GARCIA, RM	16	16									32	80	20
15	GASPAR, RC	20	20									40	100	25

Figure 3.3.4.17: The color of each table cell changes after synchronization

It does this by updating each individual cell in the view pane with respect to the contents of the generated instruction set. Moreover, the patch operation in each instruction stylizes its associated cell. For instance, all "NONE" operations would color its associated table cells as green. All operations and their associated cell styling are presented in Table 3.3.4.5.

Cell Styling	Interpretation	Associated Operation
	Data is unknown. No prior synchronization has occurred.	-

	Data is synchronized and found in the server's database.	NONE
	Data has patch context.	-
	Data is newly recognized by the server and ready to be added.	ADD
	Data is about to be replaced with a new value.	SET
MISS	Data is missing locally in the class record or in the server.	-

Table 3.3.4.5: Cell renders and their interpretations.

The Patching Process

Patching is an elaborate process that focuses on reshaping synchronized class record data into a concise and encrypted payload that is sent to the Vitre server. The patching process is based upon the RFC 5789 PATCH Method for Hypertext Transfer Protocol (HTTP) [8]. This method tackles upon partial resource modification over HTTP. The user patches a record by clicking on the Patch button at the Class Record Input Data pane (Figure 3.3.4.18.).

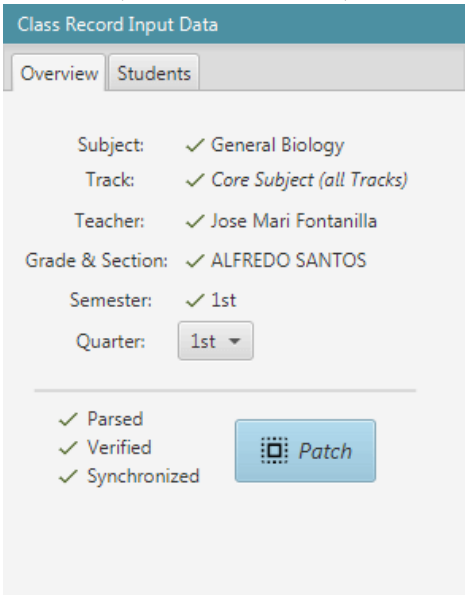


Figure 3.3.4.18: The patch button located on the view pane

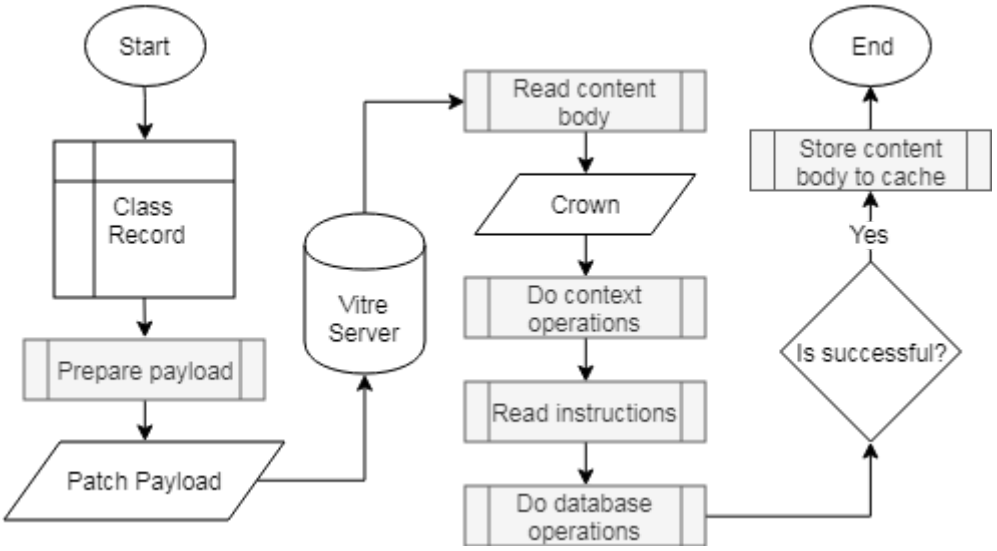


Figure 3.3.4.19: The patching process

After synchronization, the patch manager prepares a payload for the server. The contents of a patch payload is discussed in Section 4.1. The server reads the content body and extracts the crown partition for supplying the needed data in processing each patch context. The servers then reads the instruction set; containing a series of changes needed to be done. Each individual instruction is interpreted and queried to the server's MySQL database. The developers ought to apply specific changes to the database rather than replacing it as a whole. This was decided in order to lessen security risks, latency and bandwidth. The content body of

a payload is archived and copied within the server's file system for eventual synchronization to other class records.

Patch File Payload

The payload sent by the DTA contains three partitions which are visualized in Figure 3.3.4.20. Each partition is encrypted before being included in a payload.

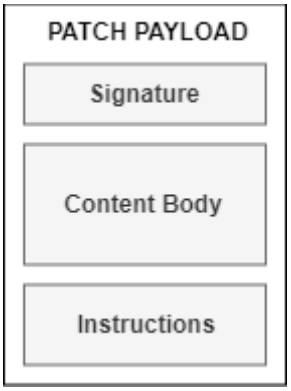


Figure 3.3.4.20: Partitions of a patch payload

The signature contains the record's unique identifier. The content body contains the data and metadata as discussed in Chapter C. The instructions contain the changes needed to be applied and interpreted by the server. Moreover, there are two types of patch requests that carry a different type of payload: (1) grade data patch and (2) class registry patch.

Grade Data Patch

A grade data patch encapsulates all grade information for each individual student into a payload content body. The patch is frequently sent to the server for inserting grade data to its database.

Class Registry Patch

A class registry patch encapsulates student registry data, such as account information, full name and indices, into a payload content body. It is primarily sent for registering new students to the server. Conventionally, class advisers send this patch request in order to register their advisory class.

The Registration Process

As mentioned, registration of a particular class section is done by its class adviser's DTA software. The adviser of a class accesses the registration window (Figure 3.3.4.21.) to register his or her students. The window asks the synchronize the record first and then designate which file directory the account sheet would be generated into.

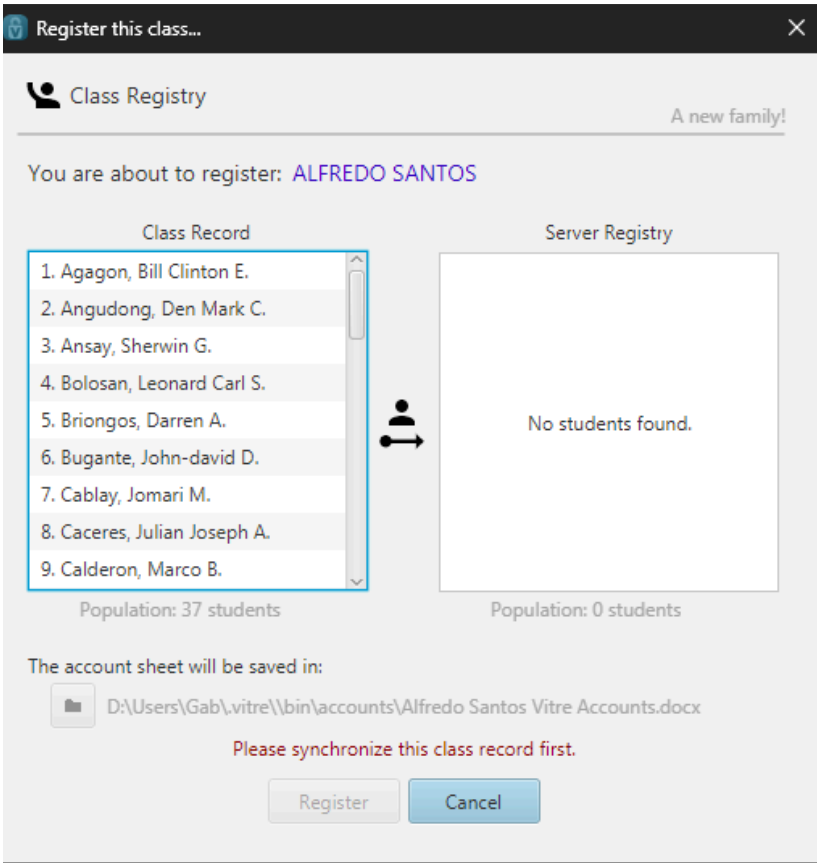


Figure 3.3.4.21: Registration window pane shown

User Account Sheets

In designing the user account system, initial usernames and passwords were generated by the DTA. These account details are used to create a Microsoft Word document that serves as an account sheet for distribution. A generated account sheet is presented in Figure 3.3.4.22.

Vitre Account Credentials **Agagon, BC** *user: bcagagon80* *pass: gakevede44*	Vitre Account Credentials **Angudong, DM** *user: dmangudong0* *pass: nobamebi16*	Vitre Account Credentials **Ansay, S** *user: sansay71* *pass: ragetaye36*	Vitre Account Credentials **Bolosan, LC** *user: lcbolosan55* *pass: jikahisi54*
Vitre Account Credentials **Briongos, D** *user: dbriongos91* *pass: pawifaci55*	Vitre Account Credentials **Bugante, J** *user: jbugante41* *pass: valahore71*	Vitre Account Credentials **Cablay, J** *user: jcablay10* *pass: gevamimi85*	Vitre Account Credentials **Caceres, JJ** *user: jjcaceres71* *pass: pivamoce51*
Vitre Account Credentials **Calderon, M** *user: mcalderon20* *pass: fekeyego30*	Vitre Account Credentials **Catayna, K** *user: kcatayna43* *pass: hicoyele05*	Vitre Account Credentials **Cepe, AJ** *user: ajcepe83* *pass: denereke62*	Vitre Account Credentials **Colinares, JM** *user: jmcolinares0* *pass: sopesado31*
Vitre Account Credentials **Cordero, DJ** *user: djcordero29* *pass: sofiyiso53*	Vitre Account Credentials **Garcia, RM** *user: rmgarcia55* *pass: pasevoyo68*	Vitre Account Credentials **Gaspar, RC** *user: rcgaspar7* *pass: hibepoki64*	Vitre Account Credentials **Gaviola, J** *user: jgaviola79* *pass: yisodibi41*
Vitre Account Credentials **Hindap, JA** *user: jahindap91* *pass: nejenewi03*	Vitre Account Credentials **Jugarap, JP** *user: jpjugarap98* *pass: levasehi78*	Vitre Account Credentials **Lopez, M** *user: mlopez74* *pass: meyeneme64*	Vitre Account Credentials **Mostrales, R** *user: rmostrales19* *pass: nihiyifi77*
Vitre Account Credentials **Nuada, DE** *user: denuada7* *pass: wibepima55*	Vitre Account Credentials **Olimberio, A** *user: aolimberio32* *pass: fahimene01*	Vitre Account Credentials **Peralta, G** *user: gperalta95* *pass: dogitato65*	Vitre Account Credentials **Rivera, JG** *user: jgrivera56* *pass: bonekave12*

Figure 3.3.4.22: An account sheet viewed in Microsoft Word

In Figure 3.3.4.22, account details are populated to each table cell. These details contain a pseudo-generated username and password for a student. The username is created with the initials, last name of a student and appended by two numbers; while the password is a randomly generated and human-readable word that is suffixed by two numbers.

Chapter 4: Implementation

4.1. Research Locale and Population

The platform was installed and conducted at the senior high school faculty of Pitogo Senior High School located in Barangay Pitogo, Makati City, Metro Manila. This particular location was chosen as the target users of the Vitre are those who are situated in a public school wherein stable internet access is scarce thus correlating to the reason of it being a local server. The developers of the Vitre are responsible for facilitating the operations of the system as well as managing its deployment with the aid if the teacher faculty whom will be the users that will ultimately define the progress of the system as the grade data is highly dependent upon their upload schedule. Furthermore, additional help was obtained from authoritative figures such as the school principal who guided the proposal and its execution, the custodian who responsible for the organization of technical employees and the IT employee of the school which are all going to be future consumers of the system as well.

4.2. Operational Feasibility

The thought of user experience, more commonly known in the graphic industry as user experience (UX) design, was discussed considerately in the development of the project. The developers ought to make the system self-sustainable and user friendly as much as possible in order for the success of its implementation in the following years as well as its eventual expansion. In truth to this, the graphical user interface of the Vitre was designed in a style that is moderately minimal thus presenting an authentic guidance to its usage while still retaining some control over advance functionality such as in the case of the DTA.

As for the hardware, the planned out schematic for the deployment of the system network were also optimized in order to accommodate for all the senior high school students of the school. Since the classrooms in the top floor of the campus are located on the two lengthwise hallways, individual routers were placed there along with the hallway that keeps the laboratories which sums up to a total of three routers whose range would reach all the students without the need of them exiting their classrooms. This design was executed due to sufficient additional funding from the school however the system could be narrowed down to only the raspberry pi in a small room and yet remain feasible despite of the decrease in cost.

4.3. Development Process

The project was nearing the end of its development in terms of the main software functionality when the developers started to coordinate with the school to obtain funding to proceed with the development. The developers first initiated a proposal by inquiring Mr. Jose Mari Fontanilla, one of the senior high school teachers in the faculty of Pitogo High School, concerning coordinating and pitching with the school for the project. This is where information concerning the nature of project proposals in the school was obtained as well as the right personal to approach in order to address the proposal. In addition, Mr. Fontanilla relayed the

information to the head of the faculty to which the developers consequently approached in order to request for a brief presentation to a teacher's faculty meeting. With the help of the former school assistant principal, Mrs. Gizelle Laud, noted by the school principal, Mr. Felix Bunagan and the project manager, Mrs. Letecia Beronio II, a presentation was organized at the end of one of the teacher meetings wherein the gist of the project was explained. The criticisms and suggestions that were brought upon in that meeting were taken into consideration and so the system was polished before the main proposal directly to the school principal which consequently resulted into an approval.

The approved project gave way for the budget proposal of the developers wherein a finalized total budget of 8,300 pesos was accepted. The details of the budget are elaborated in the later portion of this chapter. The materials were then purchased online and were documented for a budget report. After arrival, the materials were tested for any faults or malfunctions which could result in a returned order. The software was then integrated into the acquired hardware and eventually, the whole network. This integration is done by distributing copies of the DTA software to each associated teacher. They were given a subtle background and debriefing in how to use the system.The webserver and its hardware was also deployed in a smaller scale proceeding that, which then led to its eventual publication.

4.4. Scheduled Deployment

The development process of the system lasted for more than a year and is still undergoing further refinements and updates as time progressed. The project has been set to be deployed around the 18th to the 23rd of February. After a few months of implementation, the developers would then pursue a larger deployment with either expanding to the other levels of the school or to other educational institutes, depending upon if another proposal would bring the aid of the Department of Education.

4.5. Project Cost

The hardware mentioned earlier in this study were referenced online, specifically in Lazada, in order to acquire the predicted cost of all the materials. The materials and the orders made in the online shop are enumerated below in Figure 4.5.1.

Order #214425299490425

Omni Universal Outlet Extension Cord 3 Gang with Switch 1.83 Meter Cord Length	349.75 php
Strontium Nitro SRN16GTFU1R 16GB Micro SD Card (Speed up t0 65MB/s)	233.00 php
TP-Link TL-WR840N 300Mbps Wireless N Router (Qty:3)	(Failed Delivery)
TYLEX X-C11 1M Data Charging Micro USB Cable for Android	89.00 php
TP-LINK TL-SF1008D 8Port 10/100mbps Switch	429.00 php
JBtek Raspberry Pi Heatsink Copper and Aluminum	119.75 php
Raspberry Pi 3 Model B Plus Starter Kit with NOOBS	2,999.75 php

Subtotal	4,220.25 php
Shipping Fee	150.00 php
Shipping Fee Discount	-95.00 php
TOTAL	**4,525.25 php**

Order #215720711290425

Neutron 305m Cat5E UTP Cable	1,250.00 php

Subtotal	1,250.00 php
Shipping Fee	50.00 php
TOTAL	**1,300.00 php**

Order #216616747990425

TP-Link TL-WR840N 300Mbps Wireless N Router (Qty:3)	2,685.00 php

Subtotal	2,685.00 php
Shipping Fee	155.00 php
Voucher	200.00 php
TOTAL	**2,640.00 php**

Total Budget	8,500.00 php
Total Expenses	8,215.25 php
Money in hand	284.75 php

Total Expenses: 8,215.25 php

Figure 4.5.1: Budget report entailing the project expenses

Chapter 5: Implementation Barriers

5.1 Security

Upon developing the system, several security measures have been implemented in order to preserve the privacy and anonymity of the users. This portion of the chapter elaborates upon such precautions as well as the mitigation strategies that have been developed to combat them.

Operational Security

The operational security of the system concerns both physical and software factors. The system contains installed hardware such as the routers, network switch and the Raspberry Pi that are exposed to physical tampering. This is more emphasized in the case of the three deployed network routers which are situated in dense student areas.

The system is also in risk of bad management especially since it is in early stages of school implementation. For example, the lines that power the system are dependent on outside outlets rather than being integrated to one of the school's breakers. This means that powering the system is reliant upon a facilitator that will switch it on and off when needed which is unlikely for servers. Risk of bad management is also evident on the software side of the system. The webserver and the DTA work hand-in-hand when operating the system. In times that an error or a notice appears, such as when a student simply forgets their dedicated password to the system, the need for administrators to do simple fixes are gravely needed.

The networking of the system is also not free from vulnerabilities that hackers could exploit. In fact, the server is not SSL certified which makes it susceptible to hackers capturing the network traffic and obtaining the transmitted data.

Information Security

The dissemination of information from the web server is targeted towards the specific users. That is a process in which the system still controls however the responsibility is then transferred to the student once the data has been delivered which means that the sharing of information is now dependent upon the students' actions. In the case the student accesses an account in front of someone eavesdropping, his private information could be leaked which strays the project from its original intention of protecting student information by creating an alternative to the traditional disclosure of grades.

The reliability of the system presenting grades is also crucial in its operation. The system is still exposed to bugs and malfunctions that could potentially alter the grades presented by the webserver. This could not only false

information but also discourage the use of the system all together by both the teachers and the students.

Security Mitigation Strategies

In order to mitigate the physical tampering of the system, the developers coordinated with the senior high school faculty to give precautions to students concerning the piece of hardware that is mounted in both the faculty and the hallways. Furthermore, the teachers on the faculty are imposing a strict rule in letting students stay or even enter the faculty thus mitigating the risk.

To combat the bad management that is at risk in maintaining the system, a manual concerning complete documentation and guidance of the system was created by the developers in order to serve guidance to the teachers and administrators facilitating them. This allows them to fix common problems that the system may face and address student problems in its consumption. Also, seminars are to be held in the school to inform students about the details in using the system as well as maintaining it for continual use.

The developers have thoroughly considered the penetration potential when hacking the system and have done several security precautions to mitigate this. It is also difficult to penetrate the system without leaving the trace as it is situated in a local web server so the use of proxychains and DNS changers are not applicable. The presence of security cameras in school also contributes to the extra security of capturing the attacker whenever the system is threatened. Furthermore, the fact that it is simply an archive of grade data renders any attack to the web server useless as it is simply a copy of the official grade records.

5.2 Cost

The cost of the system determines the rate of its expansion. Initially, the system is advertised as a cost-efficient and better alternative to online education portals so the budget proposed to the school was limited to only but the essentials. Not to mention, since the developers weren't able to experiment with various technologies of broadcasting an access point and running the system so the materials used might not be optimized. With the total funding that was provided for the project, the developers were only able to accommodate the senior high school students and the plan for setting up the same system to the junior high school has been moved to future advances.

Cost Mitigation Strategies

Though being a major barrier, the total funding for the system was cleverly distributed to only the essential hardware that was needed. The facilities of the school were also utilized to lessen the weight of mundane things such as paper works in the funding. Rather than allocating funds for printing, the faculty

offered theirs instead to aid in the development of the project. The setting for development was also provided by the school which allowed the developers to be confined in one place upon development which is not possible otherwise. Furthermore, the idea of deploying a prototypical system was considered and so the cost was rather placed into a less prioritized factor in creating the system as the most significant thing to focus on is the feasibility of the system in implementing it to the school.

5.3 Infrastructure

Structural

There were several structural disadvantages that were addressed when the system was under development. One of the major disadvantages are the two faculties being situated on opposite ends of the hallways which doubled the amount of effort in terms of briefing the teachers, deploying hardware as well as the distribution of the desktop teacher's application. Another one is the lack of reliable information concerning the dimensions of the school hallway which delayed the deployment of the three routers as the LAN cables needed to get routed on the rooftop still have to be processed and converged with Ethernet ports.

In addition, since the local school that was chosen to deploy the project did not have a student lounge, the deployment plan for the system was altered to accommodate for the students while they are inside their classrooms instead which also spiked up the cost for the project. Also, the system could have accommodated for more people with less materials if not for the one sided hallways of the school. If in the case that the hallways had two alternating opposite rooms on each side, more classrooms could've been reached with per access point.

Wireless Connectivity

The only thing that limited the systems in terms of wireless connectivity was the poor wall penetration of the access point radio signals. According to penetration testing that was done in order to identify the range of the routers when placed on the hallway, the connectivity drops off either ends of the lengthy hallways efficiently with the wireless adapter grasping the network decently. Unfortunately, it was concluded that the walls are the greatest factor that hinders the signal. Nonetheless, there were several factors that could have affected the testing such as the lowered placement of the router as opposed to the planned location which was a lot higher and could have resulted in a higher range and penetrating ability.

Data Storage

The data storage of a system which is only 8gb worth of flash memory may also result in some hindrances in its runtime as well as errors such that the storage is filled up. Granted, the data flowing through the system only composed of a few bytes of integers and strings but it is still prone to reaching its full capacity

Infrastructure Mitigation Strategies

To address the structural disadvantages of the system, the developers modified the initial deployment idea to better provide service to the school. With that, additional budget was required such as the addition of more routers which was fortunately provided by the school. On the other hand, the wireless connectivity literal barrier which is the school walls have been considered in mounting the routers and thus the signal that it emits logically would've penetrated the walls or if not could at least reach half of every room on that hallway given that it is placed on the middle of the hallways.

To avoid the storage from reaching its full capacity, the developers treated it with a software approach wherein the data in the system is cleansed on a regular basis, preferably annually or per semester, by the facilitators. This process was also simplified to the benefit of the person assigned to this task.

5.4 Sustainability

Software Sustainability

The software of the system should be constantly given updates for the welfare of the project and its continuity. Over time, software quality naturally degrades as it is prone to bugs, errors and several complications due to prolonged use and bypassing failsafes set by the developers. Reliability and overall quality of the software is greatly accompanied by maintaining development and its continuous improvement.

Hardware Sustainability

The system is deployed with hardware exposed to several environmental factors that could deteriorate them significantly over time. This is less evident to the materials that are situated inside the teacher's faculty such as the raspberry pi and the network switch since they are given the luxury of having a cool atmosphere which decreases the tendency to overheat unlike the routers that are mounted on the hallways. However, all of them are still in risk of intentional and unintentional tampering.

Furthermore, sustaining the system would need the assigned facilitators to perform mundane tasks such as turning the system on and off and occasionally debugging it in times of need. This poses a great hindrance in

such that they forget and or are unable to fix problems that occur throughout its deployment.

Sustainability Mitigation Strategies

For the software, the developers equipped it with an auto-updating system. This auto-updating system maintains the quality of the software by providing it up-to-date fixes and revisions. It ensures that the software, both the DTA and web server, would be free of errors or bugs.

5.5 Usability

The usability of the system is crucial in its development. Usability refers to the ability of an individual unfamiliar with a system to use it with little or no training. The DTA specifically requires intervention and briefing on how to use it as it is complicated at first glance.

A system, no matter how useful or beneficial, is useless if it accumulates no users. It would never evolve without the feedback and it is eventually abandoned. In the case of the Vitre, the students and the teachers are the main consumers and the system ultimately helps to aid them in their tasks and processes in school. It would be a great hindrance in development if one of these two groups of people refuse to use the Vitre.

Usability Mitigation Strategies

The simplistic and well thought out user experience design of the system mitigates this barrier and allows the users to easily maneuver throughout the system's features with little knowledge. Furthermore, manuals and other print mediums have been used to disseminate information on how to utilize the system.

5.6 Sociocultural

Sociocultural barriers refer to implementation difficulties that arise because of current norms and cultures existing within the target community; particularly, the teachers. It is evident in Pitogo Senior High School that most of its teachers record their students' grades at least a week before a term ends. This in itself nullifies the essence of the platform as a whole, if left practiced. This also holds true for the students who are considered as the main consumers of the project. The existence of the system will give them a dilemma or "culture shock" concerning whether or not they will integrate it into their daily schedule.

Sociocultural Mitigation Strategies

The developers have thought of ways to introduce the newly developed culture to greatly benefit both parties, the system and the teachers. This was done in the form of marketing strategies that advocated for the beneficiaries. For

example, teacher and student seminars alike are set in place to provide resolving information about the concept of practicing transparency that may arise from using the system. Furthermore, the use of brochures, manuals as well as other empowering tools are alongside the development of the system. It is there where it is emphasized that the consumption of the system would aid in not only the overall awareness of students in class as well as simplify redundant tasks that teachers have to go through for each semester. This would result in greater academic performance as well as an elevated level of education if used correctly.

Chapter 6: Conclusion and Recommendations

6.1 Conclusion

The development of the system went through several refinements that allowed it to be fully functional by the time of deployment. After deployment, the Vitre platform is welcomed as part of the Pitogo Senior High School academe. It proved to be very efficient in displaying student information as well as aiding teachers in disclosing the student records and so far there has not been any issues in using it. Although promising, the system still comprises several imperfections that pose certain vulnerabilities that would be resolved in the course of its future development, considerably outside the bounds of this feasibility paper. In terms of the newly integrated culture it presents, the participants of the platform effortlessly transitioned to the use of it thanks to the marketing precautions that took place to inform the benefits of its consumption however it seems that upon further observation that they struggled mostly upon its technical aspects such as the processes or procedure that involves the uploading grades to the web server. This will be addressed further as the system is given more time to operate inside the school. Nonetheless, a demonstration was held to present its use to the public and was successful. The system in its entirety was well received by the students who were involved.

6.2 Recommendations

Features and Design

- Implement a News page in coordination with the school's organizations to notify and inform students about certain events or announcements that are needed to be propagated.
- Implement efficient web server based communication system within the teachers and the students which could also give way to the creation of classroom tools such as one that poses anonymous questions in class discussion or one that allows the creation of class polls to which the teacher may benefit to more effectively discuss the lesson
- Implement an open research tool that allows the broadcasting of research surveys to a mass number of respondents with ease. This would ease the weight of methodologies in student-made research. Admins may also use this technology to conduct polls on things such as teacher evaluation
- Integrate charts and graphs to the web server to provide the students a brief interpretation of their progress in class.
- The platform itself could spearhead to being an open platform whereas students could develop their programs and coexist within the existing Vitre network. This would open opportunities to encourage the usage and improvement of the system as a whole.
- Remodel and reorganize the database structure. The developers identified limitations of the current database model within the platform. Particularly, a

one-to-one relationship exists for assigning a teacher to a subject in the database. This limitation is problematic hence it is not uncommon for multiple teachers to be assigned to a single subject.
- Create gamified elements in order to create a linear economic model that would generate more incentives in participating on web server based activities

Security
- Develop a more robust authentication system and improve overall security. There are a lot of vulnerabilities presented, especially within an offline network. The developers compensated this by implementing methodology-based encryption within the network. Although implemented, the current way is still highly susceptible to attacks unlike a commercial-grade TLS or SSL encryption protocol.
- Adopt the use of Hypertext Transfer Protocol Secure (HTTPS) in sending grade data from the teachers. In suggestion, this would be done by manually installing SSL certificates in each associated teacher's computer. This would add another layer of security.
- A robust framework such as Laravel could be potentially adopted in order to eliminate most vulnerabilities that are present in the web stack.
- In the case of student authentication, two-factor authentication should be practiced for much more reliable security. For instance, the use of a student's school ID information such as its embedded QR code could be potentially implemented.

Ease of Access
- Upgrade the network infrastructure to accommodate a larger coverage area such as considering the use of repeaters rather than routers for efficiency in terms of expansion.

Continuity
- A follow-up study should be conducted that determines the effectiveness of the platform in invoking grade transparency through the platform.
- Every school year, a new set of developers from the senior high school community for the platform should be founded and identified. This ensures that the platform would still be spontaneously developed and improved; albeit the absence of its original developers.

References

[1] Apache Software Foundation, “Apache POI – the Java API for Microsoft Documents,” [Online]. Available: https://poi.apache.org/

[2] Apache Software Foundation, “Apache POI – Component Overview,” [Online]. Available: https://poi.apache.org/components

[3] Department of Education, “E-Class Record Templates,” [Online]. Available: http://www.deped.gov.ph/resources/downloads/e-class-record-templates

[4] FX Experience. “ControlsFX,” [Online]. Available: https://github.com/controlsfx/controlsfx

[5] The PHP Group. “PHP 7.3.1,” [Online]. Available: https://secure.php.net/downloads.php

[6] Bootstrap. “Bootstrap 4,” [Online]. Available: https://getbootstrap.com/

[7] Gluon HQ. “Scene Builder – Gluon,” [Online]. Available: https://gluonhq.com/products/scene-builder/

[8] Dusseault & Snell. “PATCH Method for HTTP,” [Online]. Available: https://tools.ietf.org/html/rfc5789

[9] The Raspberry Pi Foundation. “Raspberry Pi,” [Online]. Available: https://www.raspberrypi.org/

[10] Brian Benchoff. “Introducing the Raspberry Pi 3,” [Online]. Available: https://hackaday.com/2016/02/28/introducing-the-raspberry-pi-3/

[11] The Raspberry Pi Foundation. “Raspbian,” [Online]. Available: https://www.raspberrypi.org/downloads/raspbian/

[12] Search Enterprise Linux. “LAMP (Linux, Apache, MySQL, PHP),” [Online]. Available: https://whatis.techtarget.com/definition/LAMP-Linux-Apache-MySQL-PHP

[13] Internet Systems Consortium. “BIND - The most widely used Name Server Software,” [Online]. Available: https://www.isc.org/downloads/bind/

[14] TP-Link Technologies Company. “TL-WR840N,” [Online]. Available: https://www.tp-link.com/in/products/details/cat-9_TL-WR840N.html

[15] TP-Link Technologies Company. “TL-SF1008D,” [Online]. Available: https://www.tp-link.com/us/products/details/cat-5582_TL-SF1008D.html

[16] Metageek LLC. “Why Channels 1, 6 and 11?” [Online]. Available: https://www.metageek.com/training/resources/why-channels-1-6-11.html

[17] Sridhar T. "Layer 2 and Layer 3 Switch Evolution," [Online]. Available: https://www.cisco.com/c/en/us/about/press/internet-protocol-journal/back-issues/table-contents-19/switch-evolution.html

Appendices

List of Profiles

Sherwin G. Ansay

STEM Strand, With Honors
Hardware Systems Designer



Daveren John R. Cordero

STEM Strand, With Honors
Graphical User Interface
and User Experience Designer



Dave Emmanuel V. Nuada

STEM Strand
Hardware Systems Designer



Jose Gabrielle A. Rivera

STEM Strand, With High Honors
Software Systems Designer



List of Figures

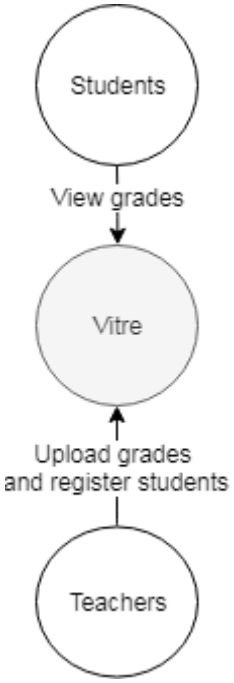


Figure 3.1: Role diagram of the teachers and students towards the Vitre

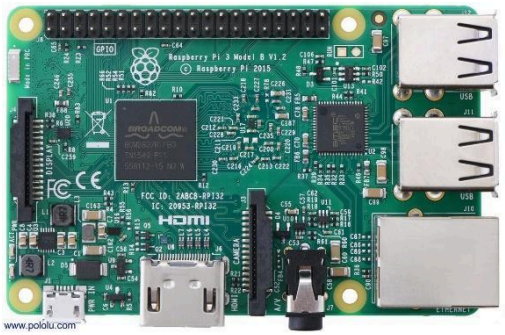


Figure 3.1.1: Top view of the Raspberry Pi 3 Model B+

	Raspberry Pi 3 Model B	Raspberry Pi Zero	Raspberry Pi 2 Model B	Raspberry Pi Model B+
Introduction Date	2/29/2016	11/25/2015	2/2/2015	7/14/2014
SoC	BCM2837	BCM2835	BCM2836	BCM2835
CPU	Quad Cortex A53 @ 1.2GHz	ARM11 @ 1GHz	Quad Cortex A7 @ 900MHz	ARM11 @ 700MHz
Instruction set	ARMv8-A	ARMv6	ARMv7-A	ARMv6
GPU	400MHz VideoCore IV	250MHz VideoCore IV	250MHz VideoCore IV	250MHz VideoCore IV
RAM	1GB SDRAM	512 MB SDRAM	1GB SDRAM	512MB SDRAM
Storage	micro-SD	micro-SD	micro-SD	micro-SD
Ethernet	10/100	none	10/100	10/100
Wireless	802.11n / Bluetooth 4.0	none	none	none
Video Output	HDMI / Composite	HDMI / Composite	HDMI / Composite	HDMI / Composite
Audio Output	HDMI / Headphone	HDMI	HDMI / Headphone	HDMI / Headphone
GPIO	40	40	40	40
Price	$35	$5	$35	$35

Figure 3.1.2: The hardware specifications of the Raspberry Pi 3



Figure 3.1.3: Raspberry Pi Accessories



Figure 3.2.1: Individual CAT5E cable and Neutron XP Cable Box



Figure 3.2.2: TL-WR840N Router

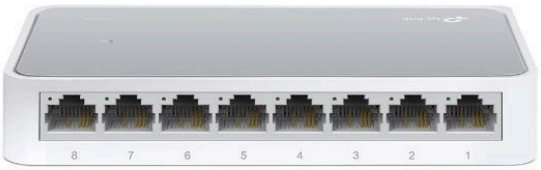


Figure 3.2.3: TL-SF1008D Network Switch

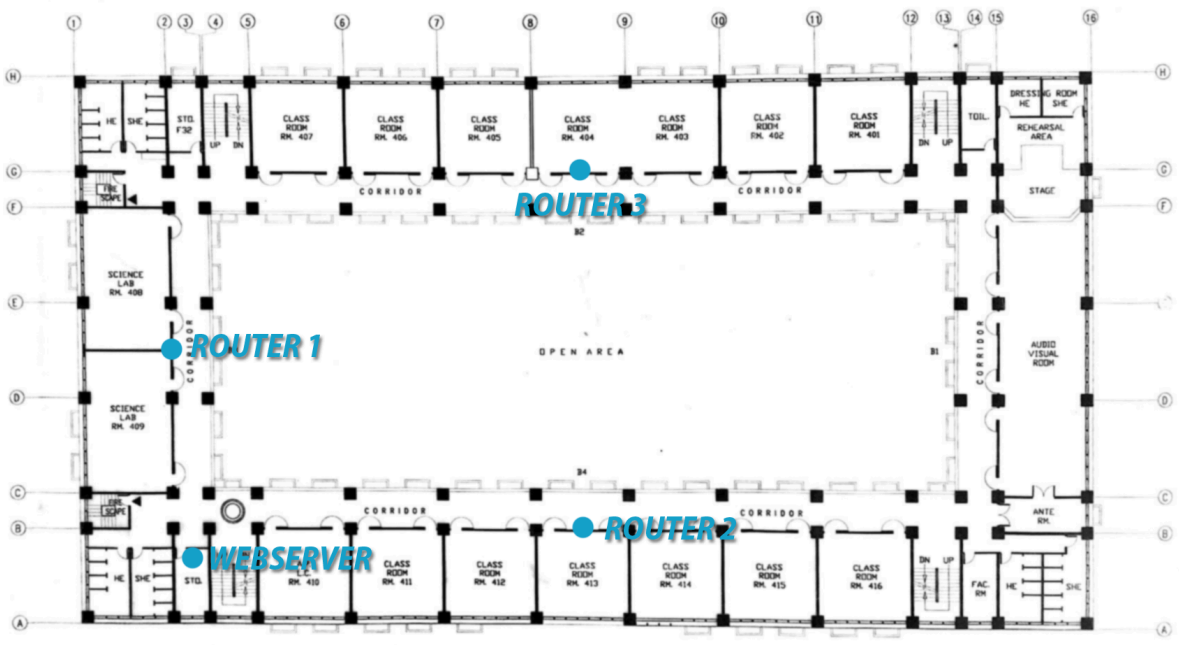


Figure 3.2.4: Network topology diagram of the Vitre network

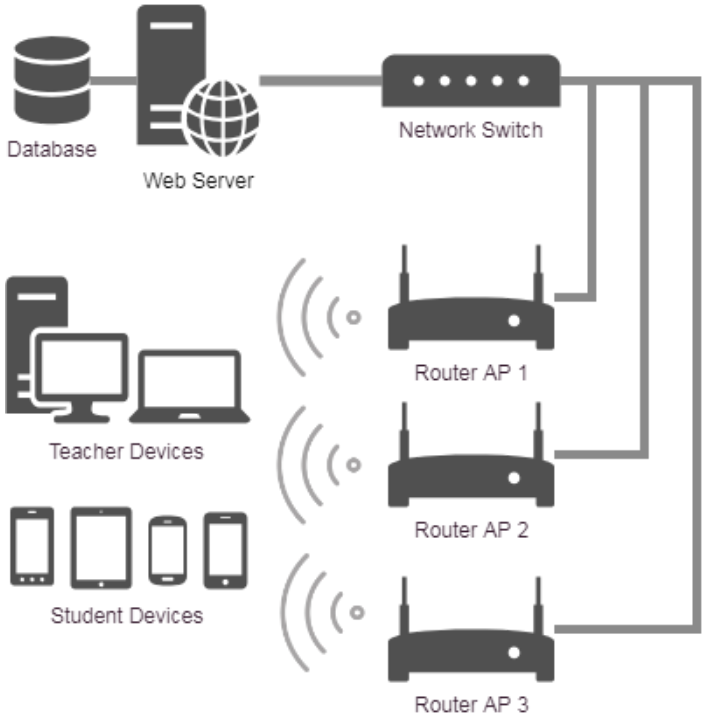
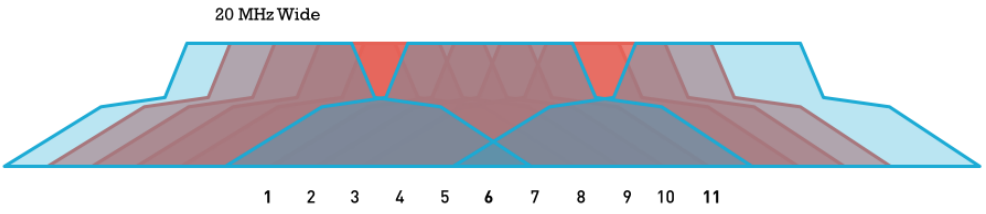


Figure 3.2.5: Network topology diagram of the Vitre network



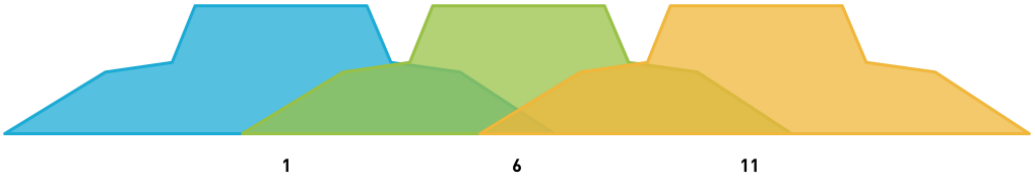


Figure 3.2.4: Channels 1, 6 and 11 showing no signal overlapping within the 2.4 GHz spectrum

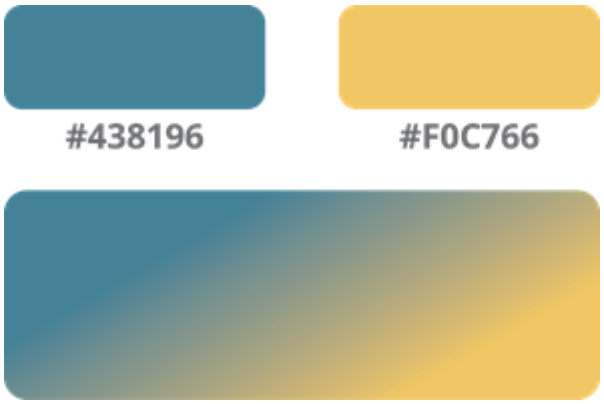


Figure 3.3.1.1: Main color scheme of the Vitre Project



Figure 3.3.1.2: Vitre Logo



Figure 3.3.1.3: Vitre webpage icon



Figure 3.3.1.4: The introductory welcoming art



Figure 3.3.1.5: Login portal

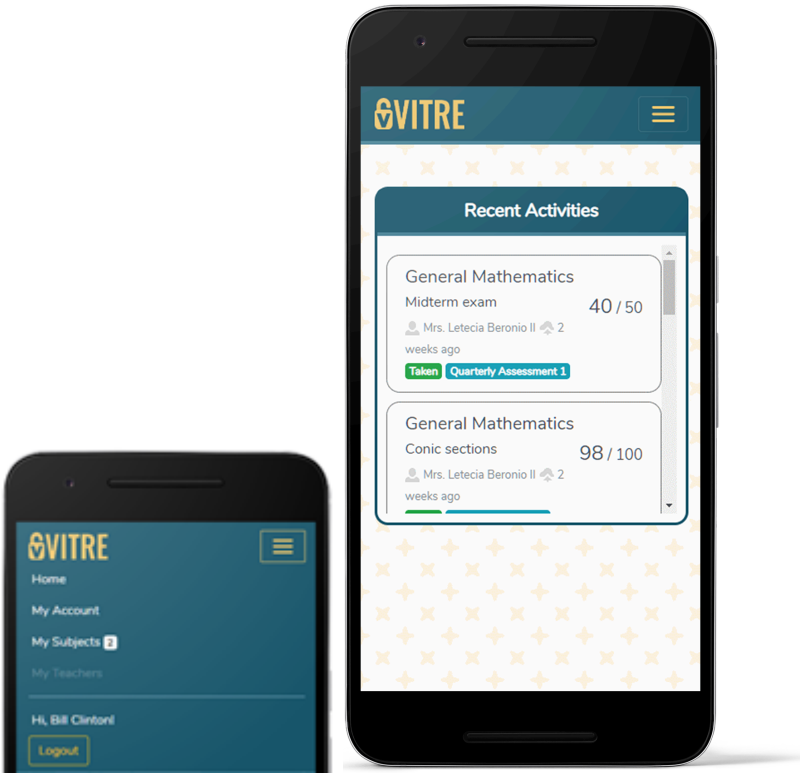


Figure 3.3.1.6: The Home Page and Navigation Bar

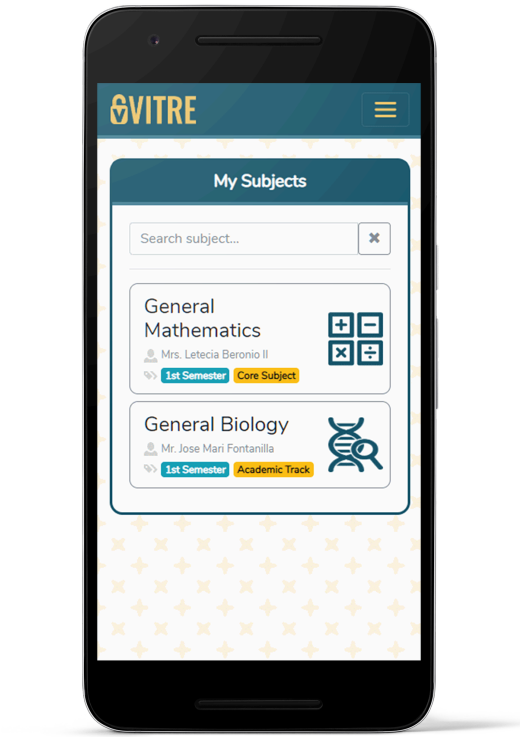


Figure 3.3.1.7: My Subjects page

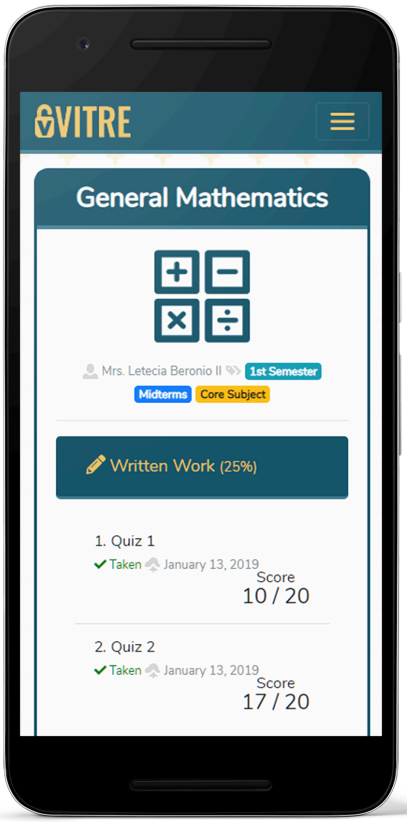


Figure 3.3.1.8: Subjects page for General Mathematics

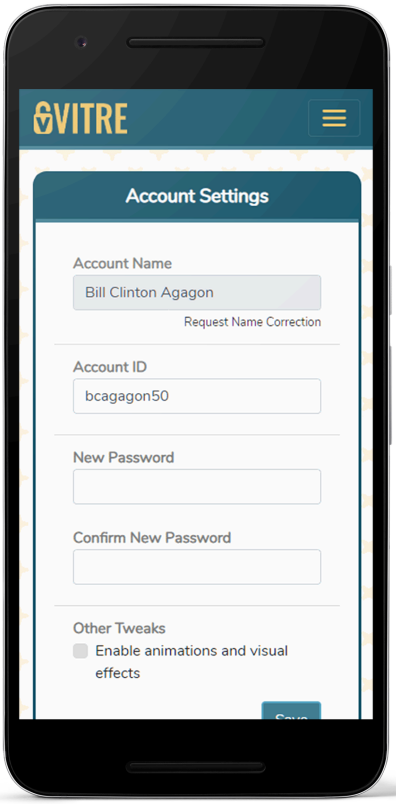


Figure 3.3.1.9: My Account page

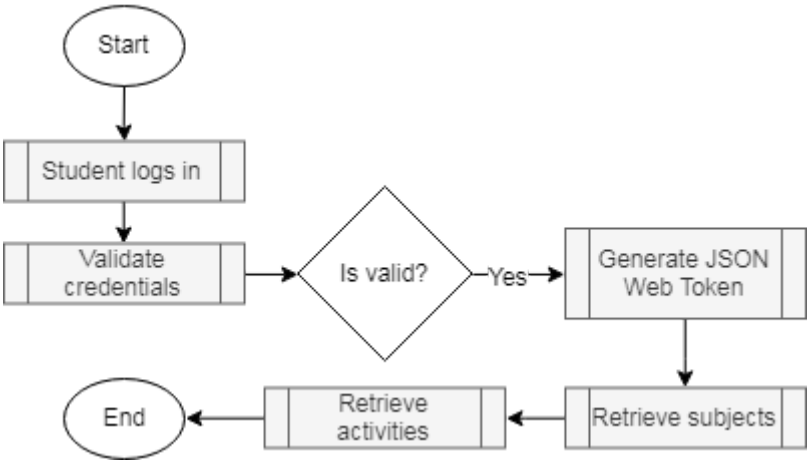


Figure 3.3.2.1: Backend processes visualized as the student logs in to the web server

Students	
PK	Student ID
FK	Class ID
	Student Index
	User Name
	First Name
	Last Name
	Gender

Figure 3.2.2.2: Students table structure

Teachers	
PK	Teacher ID
	Teacher Chunk Key
	Teacher Name
	Honorific
	Last Login Date

Figure 3.2.2.3: Teachers table structure

Classes	
PK	Class ID
FK	Curriculum ID
FK	Adviser ID
	Class Chunk Key
	Class Name
	Population
	Room Number

Figure 3.2.2.4: Classes table structure

Subjects	
PK	Subject ID
FK	Instructor ID
FK	Track ID
	Subject Chunk Key
	Subject Name
	Semester
	Icon Name

Figure 3.2.2.5: Subjects table structure

Curricula	
PK	Curriculum ID
FK	Curriculum Metadata ID
FK	Subject ID

Figure 3.2.2.6: Curricula table structure

Curricula Metadata	
PK	Curriculum Metadata ID
FK	Curriculum Name

Figure 3.2.2.7: Curricula metadata table structure

Tracks	
PK	Track ID
	Track Name
	WW Weight
	PT Weight
	QA Weight

Figure 3.2.2.8: Tracks table structure

Grades	
PK	Grade ID
FK	Student ID
FK	Grade Metadata ID
	Grade

Figure 3.2.2.9: Grades table structure

Grades Metadata	
PK	Grade Metadata ID
FK	Subject ID
	Item Index
	Component Index
	Component
	Upload Date
	Description
	Highest Possible Score

Figure 3.2.2.10: Grades metadata table structure

Analytics	
FK	Student ID
	Time In
	Time Out
	Session Time

Figure 3.2.2.11: Analytics table structure

Tickets	
PK	Ticket ID
	Ticket Owner
	Ticket State
	Ticket Subject
	Ticket Description
	Ticket Stamp

Figure 3.2.2.12: Tickets table structure

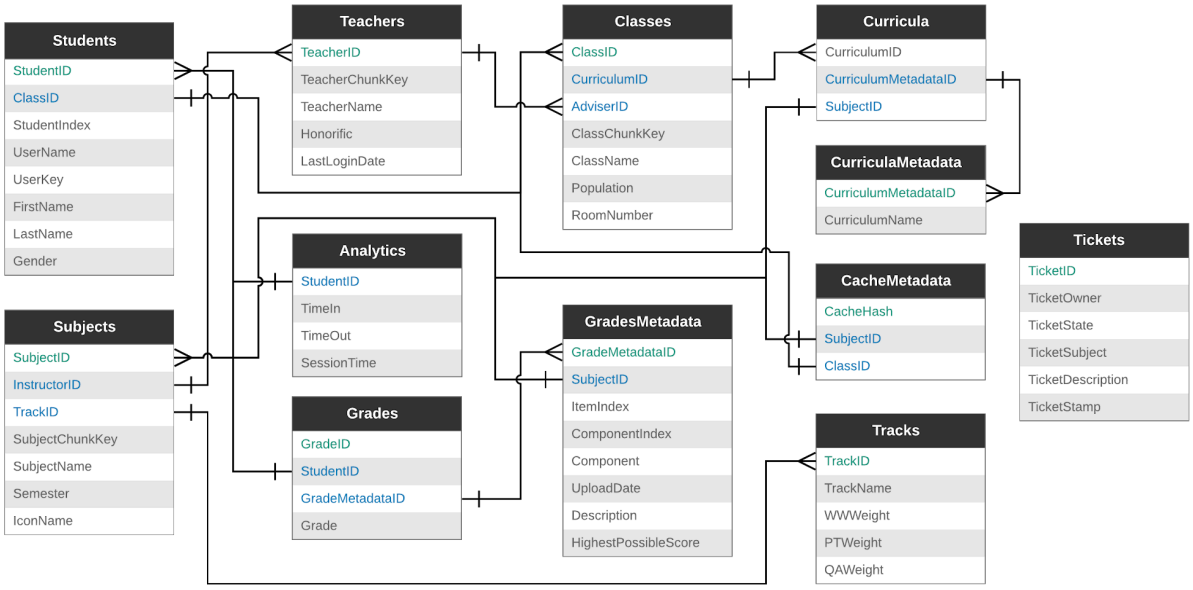


Figure 3.2.2.13: Entity relationship diagram of the Vitre SQL database

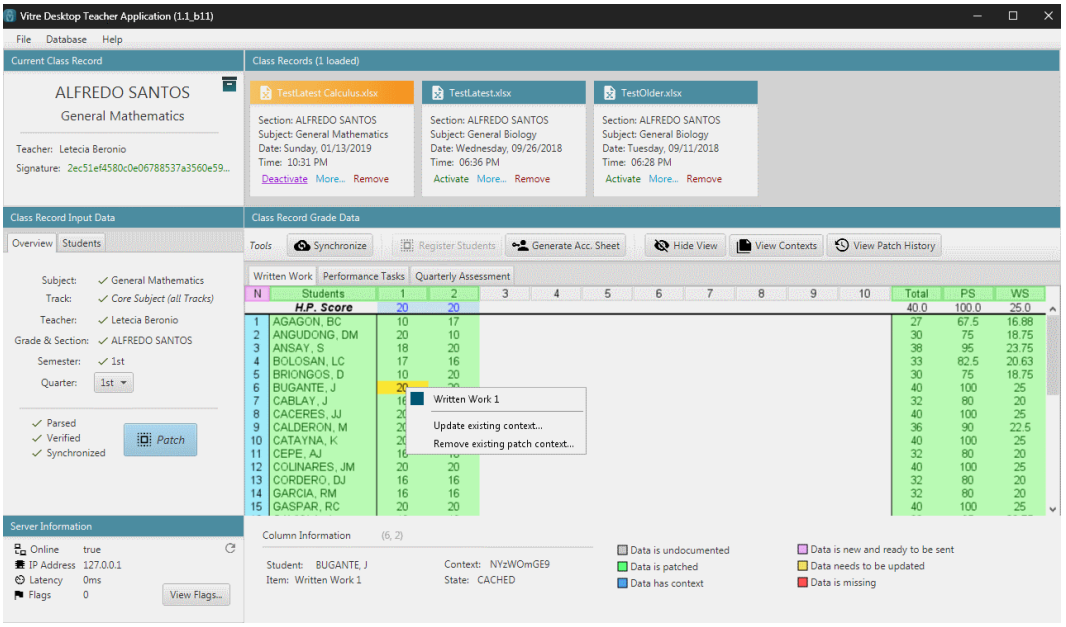


Figure 3.3.3.1: The graphical user interface of the DTA

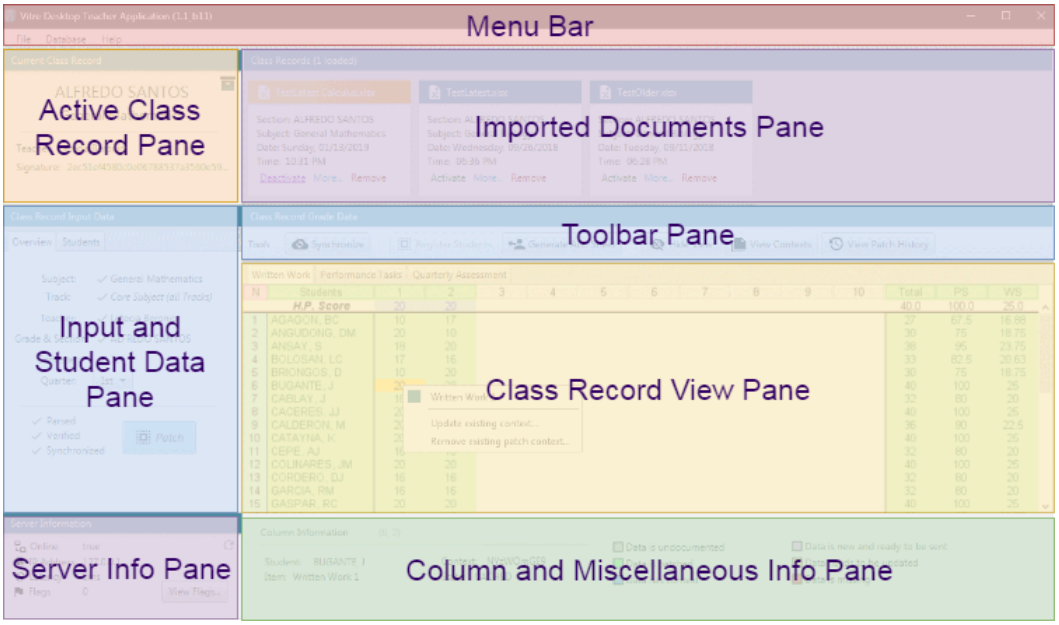


Figure 3.3.3.2: The graphical user interface of the DTA

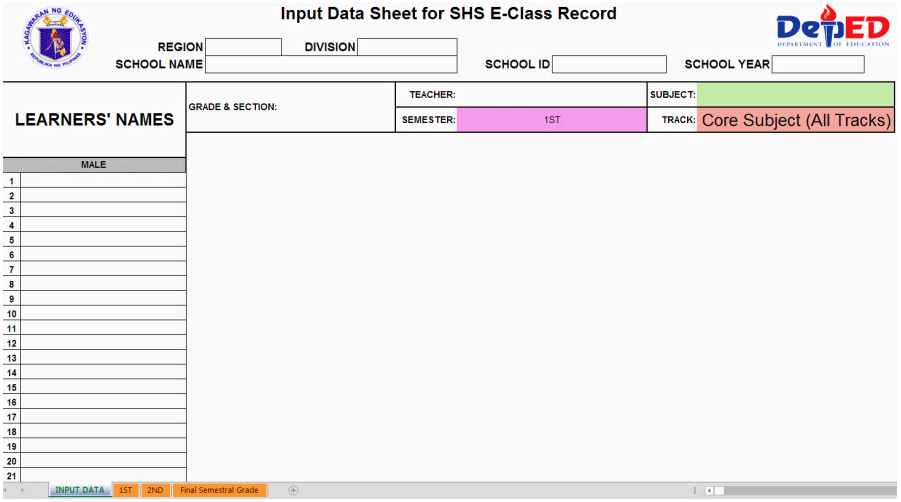


Figure 3.3.3.3: Input data sheet of an e-class record as viewed in Microsoft Excel

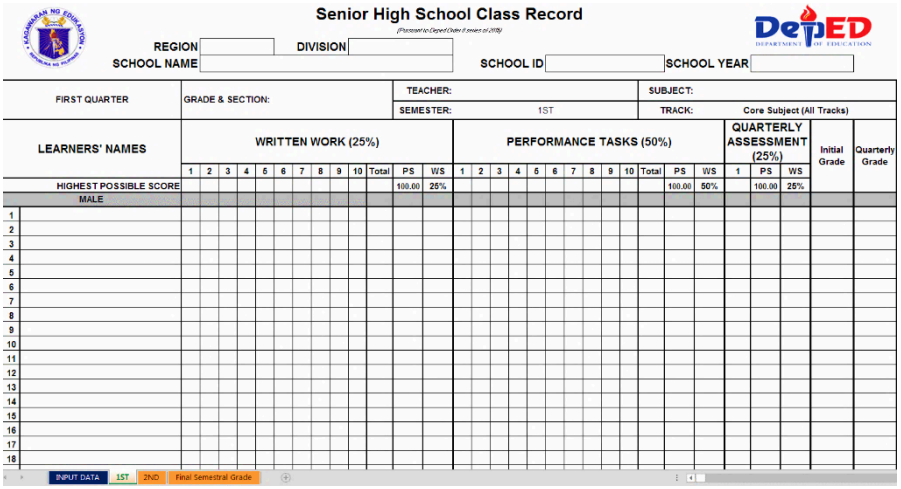
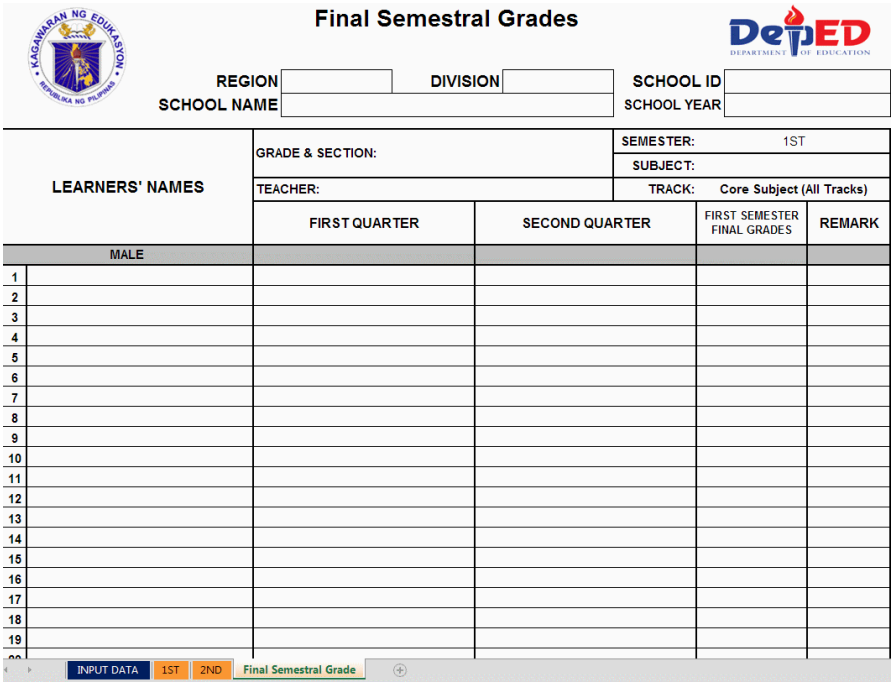


Figure 3.3.3.4: Semester grade sheet of an e-class record as viewed in Microsoft Excel



Final Semestral Grades

REGION | DIVISION | SCHOOL ID
SCHOOL NAME | SCHOOL YEAR

LEARNERS' NAMES	GRADE & SECTION:		SEMESTER: 1ST	
	TEACHER:		TRACK: Core Subject (All Tracks)	
	FIRST QUARTER	SECOND QUARTER	FIRST SEMESTER FINAL GRADES	REMARK
MALE				
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				
19				

INPUT DATA | 1ST | 2ND | Final Semestral Grade

Figure 3.3.3.5: Final semester grade sheet of e-class record as viewed in Microsoft Excel

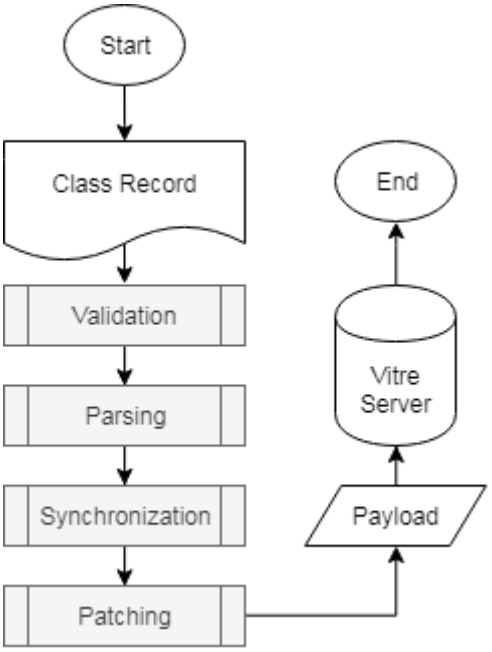


Figure 3.3.4.1: The upload process methodology

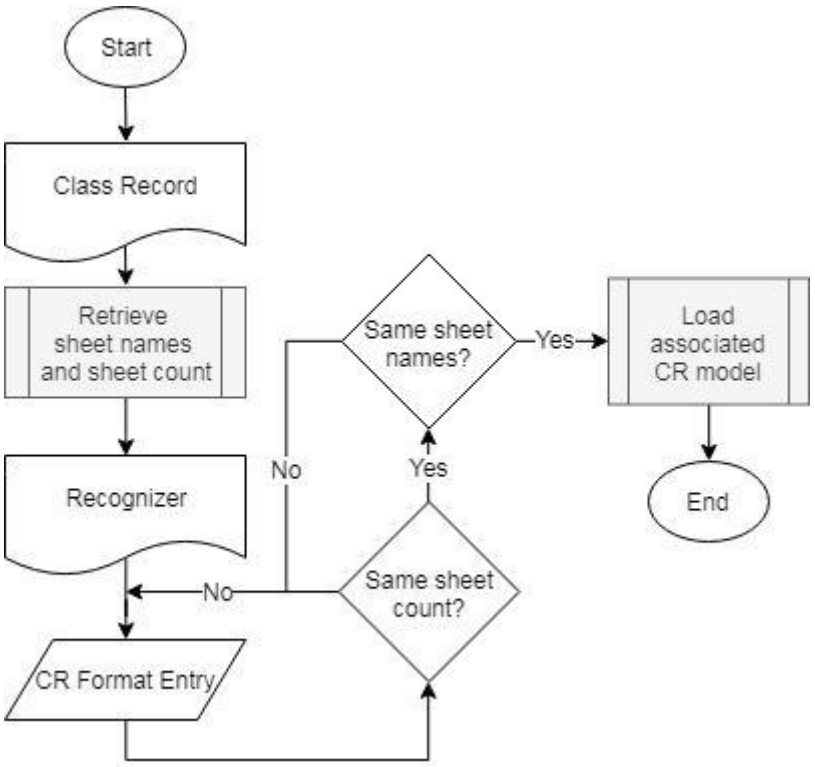


Figure 3.3.4.2: The selection process in identifying a class record format

imported.json

```
   "TestLatest.xlsx":{
      "schema":"ECRDeped2015",
      "path": "D:\\Users\\Bob\\Desktop\\TestRecord.xlsx",
      "context": "D:\\Users\\Bob\\.vitre\\bin\\contexts\\TestRecord",
      "index": 1
   },
```

Figure 3.3.4.3: Example referenced entry of an imported document

classrecords.recog

```
[ECRDeped2015]
config_path = local:ecr_deped2015.crcm
sheet_count = 5
sheets = "INPUT DATA, 1ST, 2ND, Final Semestral Grade,
Helper(IMPORTANT!)"
```

Figure 3.3.4.4: Example contents of a recognizer file named classrecords.recog

ecr_deped2015.crcm

```
[S0]
# Input Data Sheet
region = G4
division = O4
schoolName = G5
schoolID = X5
schoolYear = AG5
gradeSection = K7
teacher = S7
semester = S8
subject = AE7
track = AE8
maleStudents = B13 [ 62
femaleStudents = B64 [ 113

[S1]
# GRADING SHEET
# Proper Names
writtenWork = F9
performanceTask = S9
quarterlyAssessment = AF9

# Configuration
~sheetQuarter1 = 1
~sheetQuarter2 = 2
~rowMale = 13
~rowFemale = 64

# Input Columns
~colWrittenWork = F > R
~colPerformanceTask = S > AE
~colQuarterlyAssessment = AF > AH
~colAll = AI > AJ
~colOffset = 5

# Score Columns
```

```
~scoreColCountWrittenWork = 3
~scoreColCountPerformanceTask = 3
~scoreColCountQuarterlyAssessment = 2
~scoreColCountAll = 0

~rowAtleastBeforeBlank = 10
~enableConfig = 0
```

Figure 3.3.4.5: An example of a CRCM file for the 2015 DepEd e-class record

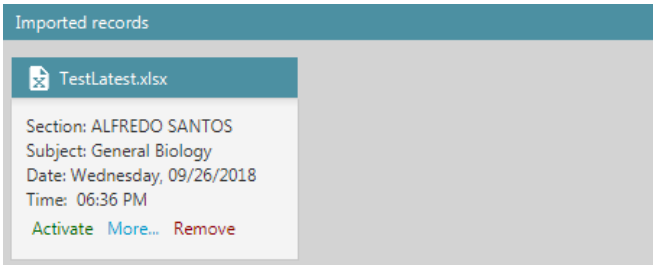


Figure 3.3.4.6: Previewed data shown in the Document Pane

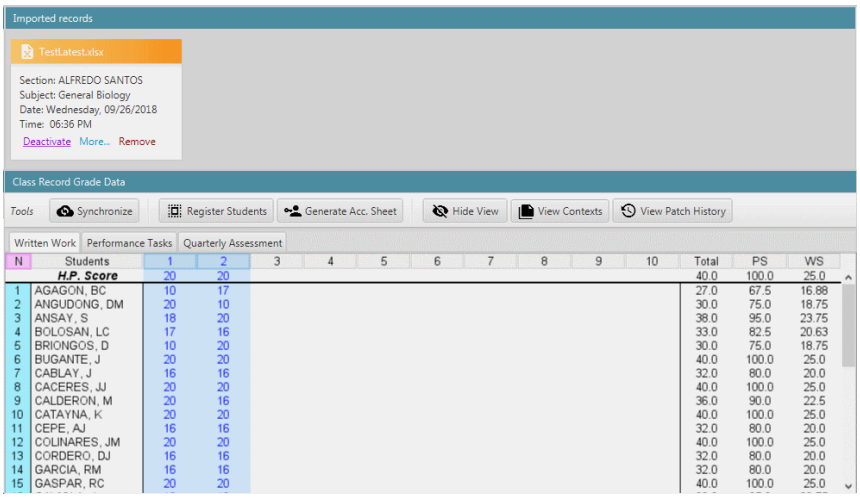


Figure 3.3.4.7: Activated class record shown in the Class Record View Pane

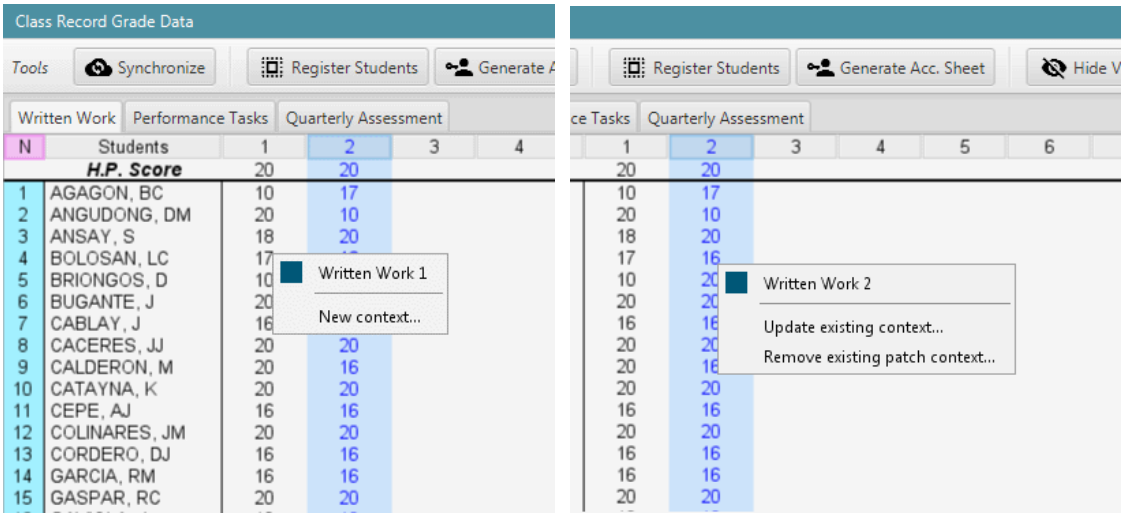


Figure 3.3.4.8: Context actions menus, triggered by right-clicking the view pane.

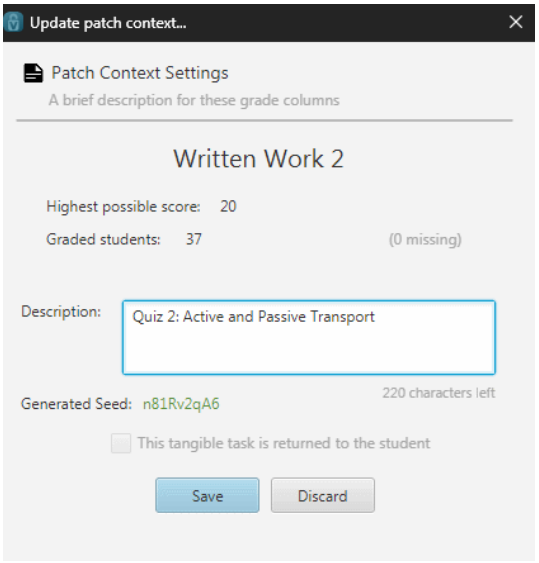


Figure 3.3.4.9: Patch context settings window

```
"context": {
    "DElyJQW6e": {
            "componentIndex": 0,
            "component": "Written Work 2",
            "cap": 20,
            "itemIndex": 1,
            "contextIndex": 0,
            "description": "Quiz 2: Active and Pass...",
            "stamp": 1547389956293
        },
```

Figure 3.3.4.10: Stored data within a patch context.

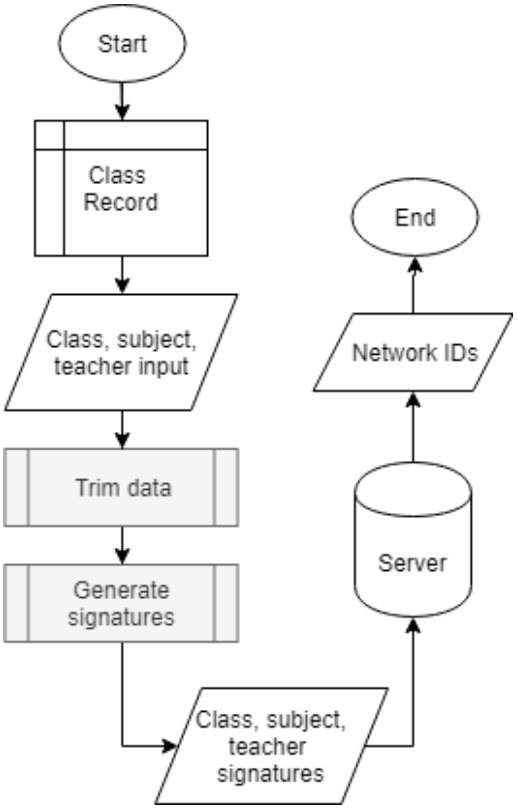


Figure 3.3.4.11: Acquiring network IDs from the server.

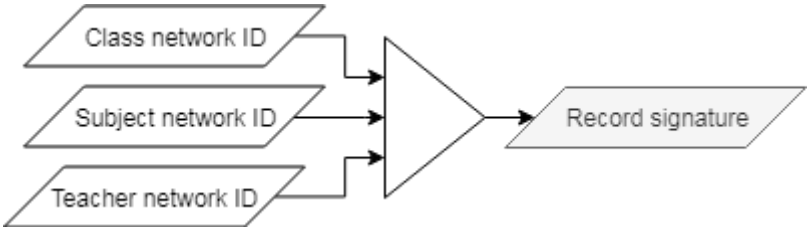


Figure 3.3.4.12: Record signature generation.

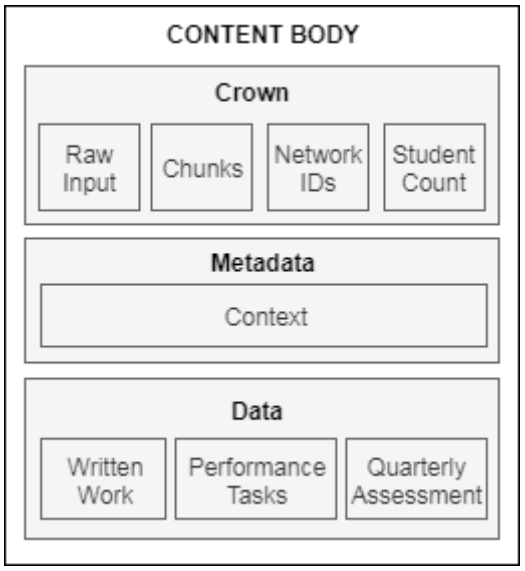


Figure 3.3.4.13: Composition of a content body.

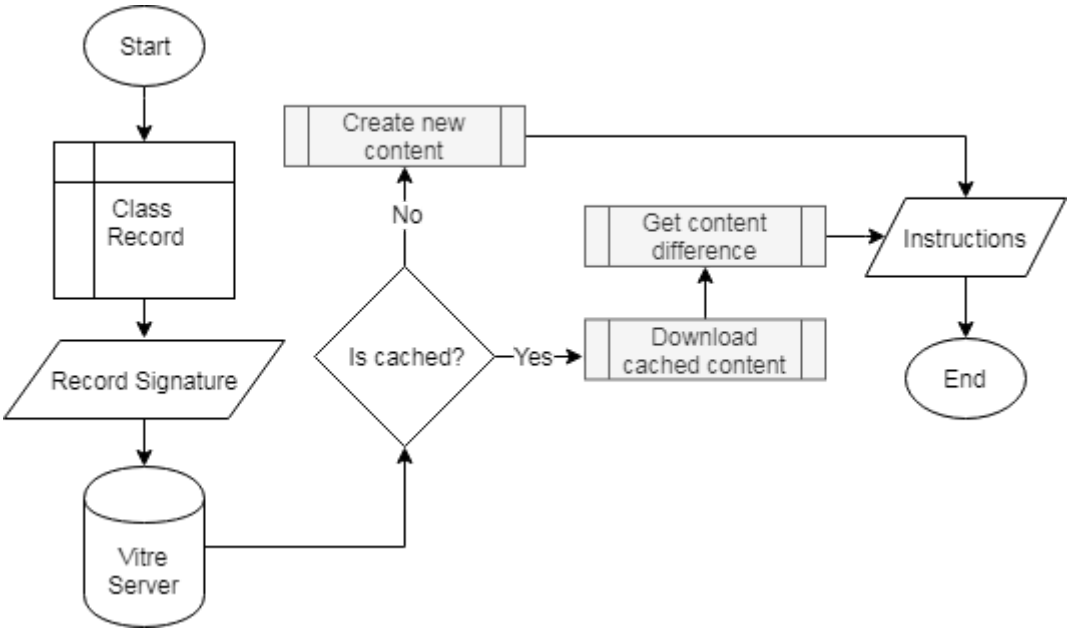


Figure 3.3.4.14: Patch difference generation.

```
1  ADD, ITEM, 1, 0, 0, 10;
2  SET, ITEM, 1, 0, 1, 10;
```
Figure 3.3.4.15: An example of a two series of instructions.

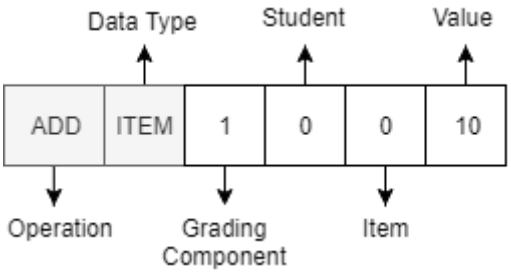
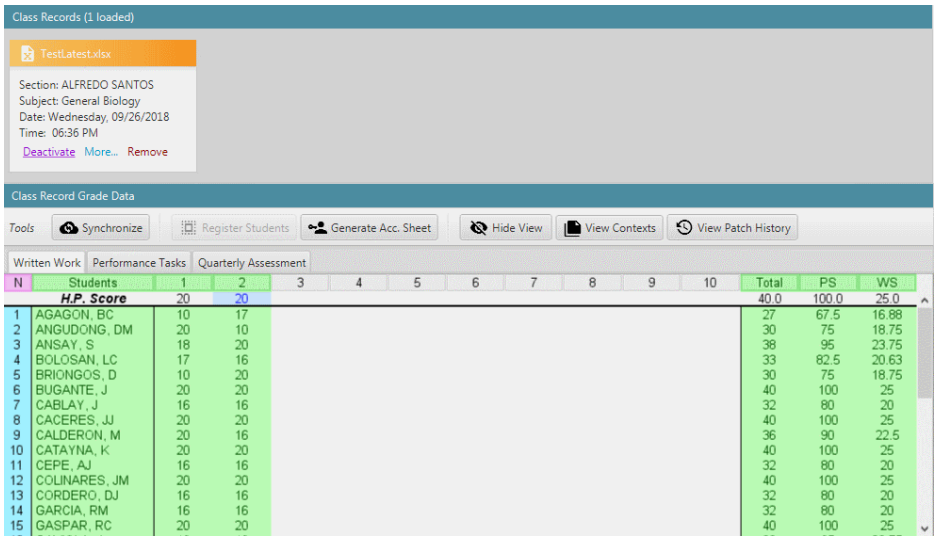


Figure 3.3.4.16: An instruction and its visualized parameters



N	Students	1	2	3	4	5	6	7	8	9	10	Total	PS	WS
	H.P. Score	20	20									40.0	100.0	25.0
1	AGAGON, BC	10	17									27	67.5	16.88
2	ANGUDONG, DM	20	10									30	75	18.75
3	ANSAY, S	18	20									38	95	23.75
4	BOLOSAN, LC	17	16									33	82.5	20.63
5	BRIONGOS, D	10	20									30	75	18.75
6	BUGANTE, J	20	20									40	100	25
7	CABLAY, J	16	16									32	80	20
8	CACERES, JJ	20	20									40	100	25
9	CALDERON, M	20	16									36	90	22.5
10	CATAYNA, K	20	20									40	100	25
11	CEPE, AJ	16	16									32	80	20
12	COLINARES, JM	20	20									40	100	25
13	CORDERO, DJ	16	16									32	80	20
14	GARCIA, RM	16	16									32	80	20
15	GASPAR, RC	20	20									40	100	25

Figure 3.3.4.17: The color of each table cell changes after synchronization

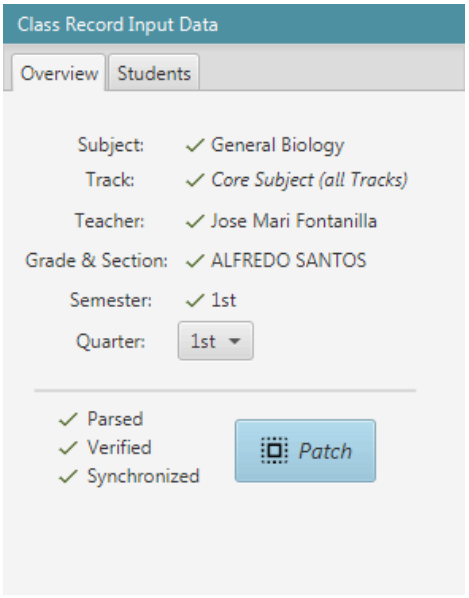


Figure 3.3.4.18: The patch button located on the view pane

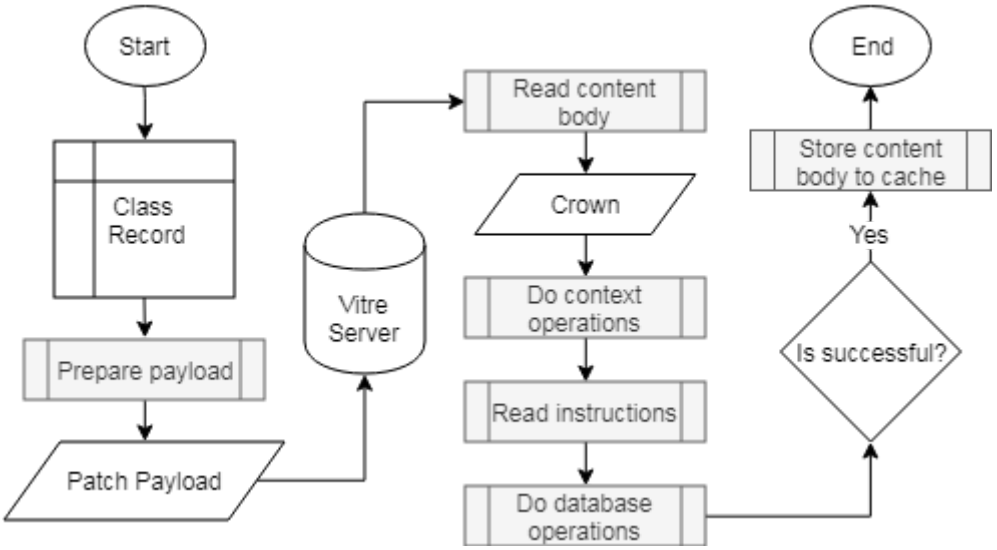


Figure 3.3.4.19: The patching process

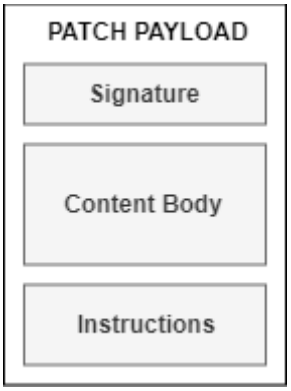


Figure 3.3.4.20: Partitions of a patch payload

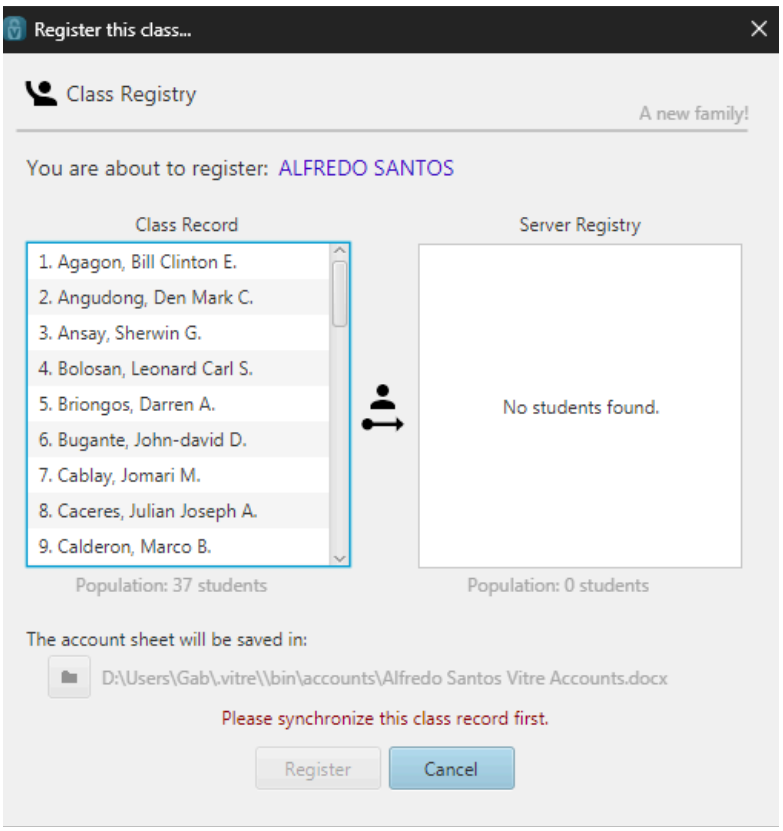


Figure 3.3.4.21: Registration window pane shown

Vitre Account Credentials **Agagon, BC** *user: bcagagon80* *pass: gakevede44*	Vitre Account Credentials **Angudong, DM** *user: dmangudong0* *pass: nobamebi16*	Vitre Account Credentials **Ansay, S** *user: sansay71* *pass: ragetaye36*	Vitre Account Credentials **Bolosan, LC** *user: lcbolosan55* *pass: jikahisi54*
Vitre Account Credentials **Briongos, D** *user: dbriongos91* *pass: pawifaci55*	Vitre Account Credentials **Bugante, J** *user: jbugante41* *pass: valahore71*	Vitre Account Credentials **Cablay, J** *user: jcablay10* *pass: gevamimi85*	Vitre Account Credentials **Caceres, JJ** *user: jjcaceres71* *pass: pivamoce51*
Vitre Account Credentials **Calderon, M** *user: mcalderon20* *pass: fekeyego30*	Vitre Account Credentials **Catayna, K** *user: kcatayna43* *pass: hicoyele05*	Vitre Account Credentials **Cepe, AJ** *user: ajcepe83* *pass: denereke62*	Vitre Account Credentials **Colinares, JM** *user: jmcolinares0* *pass: sopesado31*
Vitre Account Credentials **Cordero, DJ** *user: djcordero29* *pass: sofiyiso53*	Vitre Account Credentials **Garcia, RM** *user: rmgarcia55* *pass: pasevoyo68*	Vitre Account Credentials **Gaspar, RC** *user: rcgaspar7* *pass: hibepoki64*	Vitre Account Credentials **Gaviola, J** *user: jgaviola79* *pass: yisodibi41*
Vitre Account Credentials **Hindap, JA** *user: jahindap91* *pass: nejenewi03*	Vitre Account Credentials **Jugarap, JP** *user: jpjugarap98* *pass: levasehi78*	Vitre Account Credentials **Lopez, M** *user: mlopez74* *pass: meyeneme64*	Vitre Account Credentials **Mostrales, R** *user: rmostrales19* *pass: nihiyifi77*
Vitre Account Credentials **Nuada, DE** *user: denuada7* *pass: wibepima55*	Vitre Account Credentials **Olimberio, A** *user: aolimberio32* *pass: fahimene01*	Vitre Account Credentials **Peralta, G** *user: gperalta95* *pass: dogitato65*	Vitre Account Credentials **Rivera, JG** *user: jgrivera56* *pass: bonekave12*

Figure 3.3.4.22: An account sheet viewed in Microsoft Word

List of Tables

Category	Cable Type	Maximum Data Transmission Speed	Maximum Bandwidth
Category 3	UTP	10 Mbps	16 MHz
Category 5	UTP	10/100 Mbps	100 MHz
Category 5 E	UTP	1000 Mbps	100 MHz
Category 6	UTP or STP	1000 Mbps	250 MHz
Category 6 A	STP	10,000 Mbps	500 MHz
Category 7	SSTP	10,000 Mbps	600 MHz

Table 3.2.1: Ethernet cable categories and their specifications

Raspberry Pi Service	IP Address
Apache Web Server	192.168.0.1
BIND 9 DNS Resolver	192.168.0.1

Table 3.2.3: IP addresses of each service hosted within the Raspberry Pi

Router Access Point	IP Address	DHCP Reservation Range
AP 1 (Laboratory)	192.168.0.2	192.168.0.10 to 192.168.0.81
AP 2 (Grade 11 Hallway)	192.168.0.3	192.168.82 to 192.168.0.163
AP 3 (Grade 12 Hallway)	192.168.0.4	192.168.0.164 to 192.168.0.255

Table 3.2.4: DHCP Reservation Table for each Router Access Point

Access Point	Wireless Channel
AP 1 (Laboratory)	1
AP 2 (Grade 11 Hallway)	6
AP 3 (Grade 12 Hallway)	11

Table 3.2.5: Access points and their wireless channels

Data	Interpretation
Username	The username of the student
First name	The first name of the student
Last name	The last name of the student
Student ID	The registered ID of the student in the server
Class ID	The registered ID of the student's associated class or section in the server
Curriculum ID	The registered ID of the student's associated curriculum in the server
Session Time In	The student's login time in milliseconds
Session Expiration	The student's login expiration time before automatically logging out from the server

Table 3.3.2.1: Data stored in a JSON Web Token

Grading Component	Index
Written Work	0
Performance Tasks	1
Quarterly Assessment	2

Table 3.2.2.2: Associated indices of each grading component

Parameter	Description
config_path	The path of the class record configuration model that this entry is associated with. This parameter has two supported modes: ● local - the configuration file is within the same directory. This field requires the file name only. ● absolute - the configuration file is within another directory. This field requires an absolute file path.
sheet_count	The number of total sheets in this Excel document.
Sheets	A text array containing the names of each Excel sheet; including its hidden sheets.

Table 3.3.4.1: Configuration parameters of a recognizer file

Syntax	Interpretation
[Sx]	**Sheet instantiation block** ▪ Where x is the zero-based sheet number of the class record. ▪ All configuration settings after this block would gather data from the set sheet page number.
#	**Comment instantiation** ▪ The coordinate file would ignore this line and is commonly given to provide documentation to the user.
~	**Key-Value Pair instantiation** ▪ This is used to set "key → value" variables; such as single-valued variables. ▪ It is acquired later by the program for defining properties such as the threshold for identifying empty columns or rows.
x [y	**Vertical or row scan operation** ▪ Where x is the starting row and y is the ending row to read cell data from. ▪ This is used to tell the program to read cell values on a class record by row on a single column.
x] y	**Horizontal or column scan operation** ▪ Where x is the starting column and y is the ending column to read cell data from. ▪ This is used to tell the program to read cell values on a class record by column on a single row.
x > y	**Range or "to" scan operation** ▪ This is used to tell the program to read cell values from either a column or row range. ▪ Both x and y values should be on the same axis (column → column or row → row).

Table 3.3.4.2: Parameters, syntax and functions of a CRCM file

Input String Data	Trimmed String Data	Hashed String Data (Omitted)
12 – STEM	12stem	2F16F7B2AEF36DAF52F8952FBB99B…
General Biology 2	generalbiology2	E135FDEE536D1BF2A93A10C125B77…
Jose Mari Fontanilla	josemarifontanilla	649C53070E73B6CDFA7008F6794C6…

Table 3.3.4.3: Results of manipulating the string data from each operation.

Parameter	Value
1	**Patch Operation** ▪ It refers to the specified operation that the server would do with this data point. There are four supported operations: ▪ ADD is an addition operation that informs the server to insert this new data to its database. ▪ REM is a removal operation that informs the server to remove this new data from its database. ▪ SET is an update operation that informs the server to replace an existing data point with a new value on the server's database. ▪ NONE is an operation that does nothing. It also implies that this data point is up-to-date and does not have any changes needed.
2	**Data Type** ▪ It refers to the type of data this instruction pertains to. There are two types of data: ▪ The data type "ITEM" refers to an individual grade data point for a single student. ▪ The data type "AGGREGATE" refers to a cumulative grade data point that could either be the Total Score, Percentage Score, and Weighted Score of a grading component.
3	**Grading Component Index** ▪ It refers to the grading component this instruction targets. There are three values that refer to different grading components: ▪ Value 0 refers to Written Work component. ▪ Value 1 refers to Performance Tasks component. ▪ Value 2 refers to Quarterly Assessment component.
4	**Student Index** ▪ It is a zero-based index of a student which this instruction targets. ▪ The instruction informs the server to set the owner of the data point to this index.
5	**Item Index** ▪ It is a zero-based index of an item which this instruction targets. ▪ The instruction informs the server to set the item of the data point to this index.
6	**Value** ▪ It refers to the value that this data point carries. ▪ It is typically the raw grade score from the class record.

Table 3.3.4.4: Parameters and values of a single instruction

Cell Styling	Interpretation	Associated Operation
	Data is unknown. No prior synchronization has occurred.	-
	Data is synchronized and found in the server's database.	NONE
	Data has patch context.	-
	Data is newly recognized by the server and ready to be added.	ADD
	Data is about to be replaced with a new value.	SET
MISS	Data is missing locally in the class record or in the server.	-

Table 3.3.4.5: Cell renders and their interpretations.