FlexPay - Payment Application

Nisarg Bhalavat[1], Benita Rego[2], Nolita Rego[3], Deep Virag Savla[4]

[1,2,3,4] *Department of Computer Science and Engineering, School of Engineering, Santa Clara University*

[1]nbhalavat@scu.edu

[2]brego@scu.edu

[3]nrego@scu.edu

[4]dsavla@scu.edu

***Abstract* — The system provides a user-friendly front-end interface that allows customers to easily initiate transactions and a robust back-end system that handles the processing and settlement of funds. The server-side application is the main FlexPay Server. The server is connected to Autonomous Database (ADB) available on Oracle Cloud Infrastructure (OCI). Once the server is running, it checks the status of ADB and if that is currently not available, then restarts the provisioned ADB. The system will also include fraud prevention capabilities by using multi-factor authentication (MFA) to ensure the security of customers' information and transactions. MFA will be implemented using Microsoft Authenticator. Microsoft Authenticator provides an offline TOTP service which is better and safer than traditional sms-based MFA. The project will be over the cloud and the scope includes the development of the payment system (backend and UI), testing, and deployment. The users can store their profiles for faster payments and easy recognition. Overall, the FlexPay Payment project aims to create a payment system that is reliable, secure, and easy to use for both customers and merchants, while meeting all industry standards.**

***Keywords* — payment, mobile application, flex points, oracle cloud infrastructure, flexpay, cloud computing, data protection, student payment system**

I. Introduction

Digital methods for different payments and transactions have been closely related domains of study. In the present state, cashless payments have become an important part of our life. With the rapid pace of life, more and more people are switching to digital modes. The prevalence of various third-party integrated applications, including Zelle, Venmo, and others, has been observed to significantly impact an individual's transaction management practices. Thus, there is significant importance to make digital payments/transfers better, not just with cashless payments, but also provides security and data protection. Traditional ways of making payments are mainly used in public transportations, or to pay minimal amounts in the stores. However, traditional methods are actually reactive, which are usually time-consuming, difficult to handle, and hysteretic. With the popularity of digital payment modes, people are used to sharing their daily activities and major money transfers on third-party integrated applications, making it less safe for user's data. Security and data protection has become one of the biggest challenges while using such payment modes, which leads to money phishing, scamming and other malicious attacks on users.

This application is similar to the online payment systems that people use but is specifically targeted for Santa Clara University Students. It solves certain issues regarding digital payments that are done by students at Santa Clara University for their on-campus purchases. A brief overview, at Santa Clara University, students are provided Flex points in their wallets as university currency and they use these points for dining and other on-campus services. These Flex points have a fixed expiry date and it is noticed that the flex points sometimes get expired and are wasted, thus loss of significant currency value. Also, there are cases where students use up all their Flex points and have to add more as per need in each quarter. Currently, the students are unable to transfer these Flex points to peers, in case someone has used up their points and need them on an urgent basis. This application would enable transfer of Flex points among Santa Clara University users, withdrawal and deposit of flex points, keeping in mind the security aspects of digital transfers. With the various cloud technologies learnt during the course, out of which

some of the services like AWS and GCP were known by us as we had utilized it during our previous commitments, we collectively concluded to tap into Oracle Cloud Infrastructure (OCI), and employed its services in our application. Oracle, being one of the latest generations of Cloud Computing technologies, has showcased remarkable performance in numerous real-world applications, including data security and reliability [1]. Here, the students of Santa Clara University provide a digital token as a form of currency which can be used on and around campus which comes with a troublesome nature called Flex Points. The current architecture provided by the University restricts the students in utilizing and loading the flex points to designated terminals located at a single point with cumbersome process. We aim to unbound such limitations and build a system that would most importantly, enable students to utilize flex points in a more constructive way by giving them a choice for peer-to-peer transfer within their friends and classmates, in order to relieve them from the constant fear of over using the flex points on expiration.

II. Background And Related Work

The survey on usage of online payment apps by customers [3] looks at customer behavior on using online payment apps, examines the problems that clients have with online transactions, identifies the problems and proposes potential improvements to online payment apps. According to the research outcome, more people are using methods of electronic modes and doing electronic banking than ever before, suggesting that these additional expenses will always be accepted. Yet, the installation and development of safe electronic ways of paying solutions within current as well as the foreseeable future face novel possibilities and difficulties as a consequence of the widespread use and operation of various technological innovations. The paper doesn't investigate the advantages and security of utilizing online payment applications, and also the problems users are having with online payment applications, concerning how users will use online transactions as a mode for bill payments.

The research paper about [4] electronic payments security gives us an overview of the two different categories of electronic payment involving e-wallet and digit payment methods which elaborates on the security issues of user's data. Although studies about e-wallets looked at how they have grown increasingly popular in contactless transactions, research on online payment methods placed a greater emphasis on digital payment privacy and protection. The evaluation of the chosen papers reveals a number of issues and topics to further study, particularly issues pertaining to employing money transfer mechanisms to enhance the safety as well as compatibility of e-wallets as well as e-commerce platforms. Data protection and scalability using digital payment methods have raised various concerns with respect to various banks using third-party networking applications.

The article on exploring how university students utilize the help of digital payments [5] gives us a brief overview on an application called as Polines-pay that can reduce the amount of currency used in payments and offer ease, speed, and economy in payment processes. Nevertheless, polines-pay does have a flaw in that it can't check its remaining amount before executing a payment with the retailer. Learners propose adding polines-pay functionalities in the near future. For instance, best efficiency by the ability to recharge cards, but also by best at the retailer, and the capacity to verify the remaining amount. Learners often mention the possibility of using polines-pay for transactions off campus.

The next research paper gives us a good idea about the construction and observations of payment systems that are virtually used in colleges and universities [6] that need to provide several ways for both educators and pupils to handle payments with adequate usage data; it can help schools better determine underprivileged students. Payment systems like prepaid debit cards, loss reporting, and termination can also be offered. The digital payment service includes both digital and physical money transfers at once, and it has a comprehensive accounts monitoring platform. Bank customers now have access to campus payment solutions via card readers, QR codes, as well as other means, thereby breaking the campus card's communication limitations.

The study on tuition fee payments in educational institutes [8] urges financial institutions to set up offices or departments on campuses and also to implement the theory-based money transfer supply chain, whereby financial transactions like cell phones might reduce anxiety before enrollment. Also, the choice of a payment mechanism must be carefully considered mostly by the administration of the college, the students council, and other sponsors to lessen the burden on students throughout administering and managing the financial transactions within the application.

For most of our background research on our application, we have taken into consideration the three main areas of research and analysis, which are the payments methods, user's overall experience on usage of such methods, and security issues of such payment methods. Our applications overcome these limitations by using Oracle Cloud Infrastructure (OCI) which makes it possible to develop and operate a number of different applications in an exposure to elevated settings with high availability and scalability. In order to gain access to a service like an application, a netbanking, or a Network, the user must submit two or more verification criteria, which is known as multi-factor authentication (MFA) where we use Microsoft MFA for authentication.

III. Approach

The entire system is designed in such a way that it can be accessed easily through the mobile application. It is primarily designed for Santa Clara University (SCU) students where the flex points are used for internal transaction purposes. This system involves technologies, each playing a crucial role in the development and working of the system. After careful analysis through various cloud technologies, the cloud services utilized for our project is Oracle Cloud Infrastructure (OCI) because of its extensive range of services provided in a consistently high speed environment. The services employed from OCI have aided us in building an effective and responsive application that is deployed over the cloud. The computing capabilities provided by OCI are unparalleled, boasting physical hardware instances and extensive storage capacity. These resources are housed within a secure and flexible overlay virtual network that can be accessed from your on-premises network, providing a reliable and efficient computing environment. [1]. With its all-inclusive platform, Oracle Cloud Infrastructure (OCI) delivers a range of services, such as IaaS, PaaS, SaaS, and Data as a Service (DaaS), to provide a holistic solution to its users. We have utilized Compute Instance, which is a configurable computing service available through OCI, which empowers users to create and execute virtual machines while overseeing their resources, such as CPUs, memory, and network resources. Oracle Cloud Infrastructure's IaaS presents these Compute Instances for the server [2]. With the Oracle Cloud Infrastructure Load Balancing service, users can avail a load balancer that can be tailored to their needs and can be assigned either a public or private IP address, in addition to provisioned bandwidth and availability features that span two Availability Domains. This service enhances resource utilization, simplifies scaling, and ensures high availability [7]. A Virtual Cloud Network (VCN) with two subnets that are each in a separate access domain, a network gateway, a pair of instances running, one in every subnet, and an HTTP endpoint server built using Node.js that is operating on each instance are required for the installation of the Oracle load balancer. For additional load balancing for maintaining the HTTP traffic flow, Nginx has been available, where it supports various deployment situations with HTTP flow [9].

The system comprises users' data that will be collected from the internet via encrypted services and stored in MongoDB, which is a hybrid NoSQL document-based database service [10]. MongoDB Atlas is a noSQL database service that operates across multiple clouds, created by the same team that developed MongoDB. Atlas streamlines the process of deploying and managing your databases, while also providing the versatility necessary to construct resilient and high-performing global applications on the cloud providers of your preference. We have created database deployments in the form of clusters to manage our data. The data will be stored in its Atlas Cluster. The Node.js server is hosted over the Oracle Cloud Compute Instance where Oracle load balancer and Nginx runs in the background in order to maintain the performance of the application. The mobile application is user friendly, built using Flutter and primarily for the given educational institution. The users' data collected is secured and private.

The objective of this project is to facilitate the easy use of flex points for Santa Clara University students. We have designed our application to be versatile, having scalability and downtime free performance as and when required. The front-end mobile application was developed using Flutter SDK with Dart programming language, which can be deployed as an iOS, Android as well as a web application from a single codebase which makes it a hybrid, cross-platform system. The server side of the application was developed using JavaScript as a base programming language to write the server-side code with Node.js - a JavaScript runtime environment and library that is open-source and cross-platform, designed to run web applications outside of the client's browser and Express, which is a dynamic online application framework for Node.js that offers a strong range of features for both mobile and website applications and MongoDB stack.

The figure 1 provides the system architecture for our project FlexPay. We have utilized Oracle Cloud's Virtual

Private Cloud (VPC) for deployment of our server. A virtual cloud network is a computer-generated copy of a customary network that involves subnets, route tables, and gateways, and acts as a base for your instances. This network is confined to a specific region but encompasses all the availability domains within that region. The Virtual Private Cloud (VPC) was divided into two subnets, i.e. Public Subnet and Private Subnet. The Public subnet is accessible by the resources on the internet via Internet Gateway whereas the Private Subnet can only be accessed by internal resources deployed within the VPC. In the public subnet, we deployed two compute instances and a load balancer for increasing the throughput of the server.. The two Compute Instances used were added to the network security exception list so that they can be directly accessed individually for troubleshooting and maintenance in a controlled environment. In addition to this, we also deployed a Virtual Private Network (VPN) Tunnel for connecting MongoDB and Oracle's Virtual Private Cloud (VPC). For additional security, we deployed the Autonomous Database (ADB), which delivers a suite of data management services that automates patching, upgrades, and tuning, as well as all the routine database maintenance tasks, even while the system is in use in a private subnet of the VPC which is entirely cut-off from the internet.

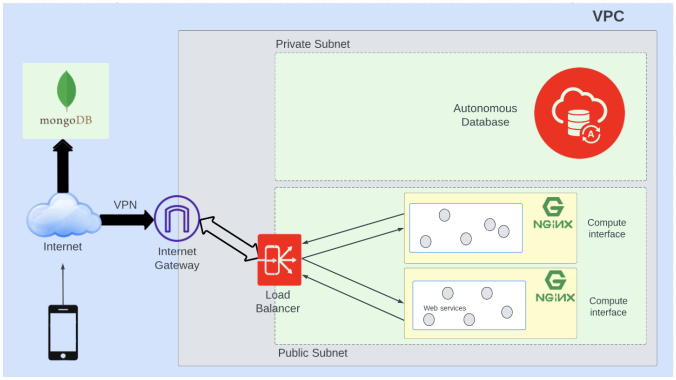

Fig. 1 System Architecture for FlexPay

Figure 2 below shows the console for Oracle Cloud Infrastructure (OCI) and the various services within OCI utilized for our project. The services pinned on the dashboard include Autonomous Database, Compute Instances for deployment of the virtual server and Virtual Cloud Networks (VCNs) for networking purposes. We have also employed Oracle Load Balancers for controlling the traffic loads for smooth usage and improved performance of the application.

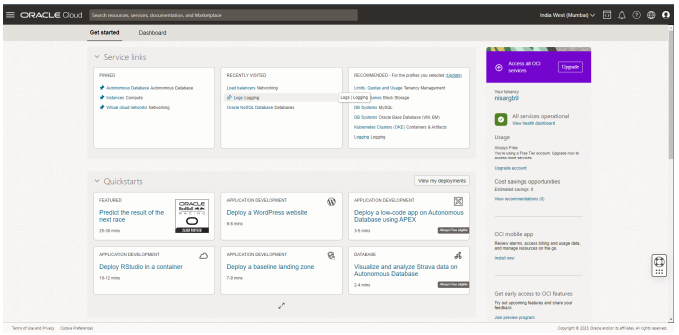

Fig. 2 Oracle Cloud Infrastructure (OCI) UI

The Nginx web server board in Figure 3 below illustrates the traffic flow in the log chart which is distributed over the virtual server, along with a graphical analysis provided in the log chart for monitoring the traffic load and usage of the virtual server over the Oracle Cloud Dashboard.

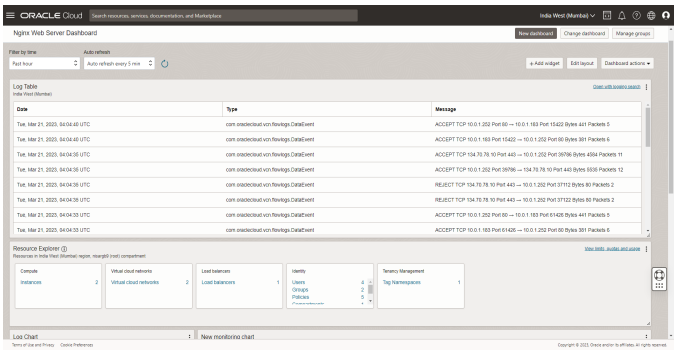

Fig. 3a Nginx Web Server Dashboard

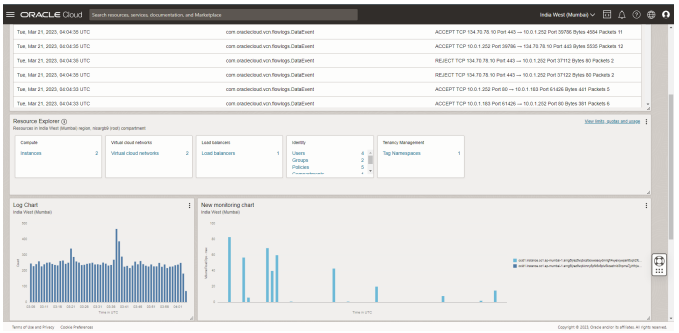

Fig. 3b Nginx Web Server Dashboard

Any requests sent by the client (front-end application) to the server (backend-side) would be routed through the internet gateway (IG), then to the load balancer (LB) and further to the Compute Instance (CI) where Nginx and PM2 (Production Process Manager) will bring the new threads up/down, in order to process the valid incoming requests in the server. Each step has their own security filters and firewalls to prevent malicious attacks from external third parties.

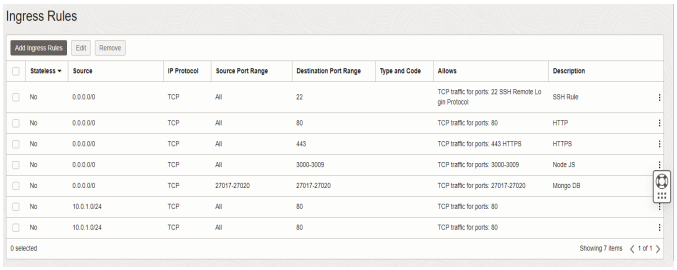

Fig. 4 Nginx Web Server Networking rules

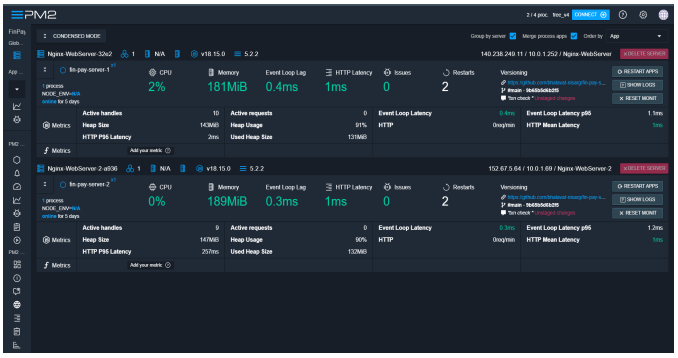

Fig. 5 Nginx Web Server PM2 Dashboard

We deployed our Node.js backend application on a Compute Instance provided by OCI which was managed by Nginx and monitored using the PM2 tool. Here, Nginx worked as a reverse proxy as well as a load balancer for scalability of the web application. Figure 5 illustrates the console for Nginx web server in PM2 where it monitors the active handles and the HTTP latency with error logs and an option to restart the server without any codebase.

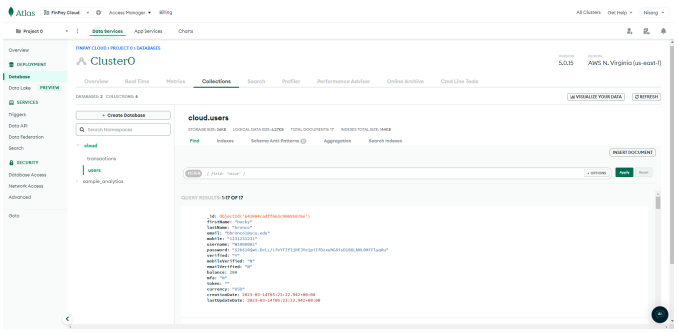

Fig. 6a MongoDB Database (Users)

Figure 6a illustrates the dashboard the registered user data stored in "cloud.users" database within the ClusterO, which can be retrieved with a unique object ID from the MongoDB database. It can be queried and fetched using the object ID allocated to each registered user in JSON format. A different collection to save the transactions has been created which will store the transaction success and details which includes the timestamp and amount. Figure 6b shows the transaction collections for the registered users.

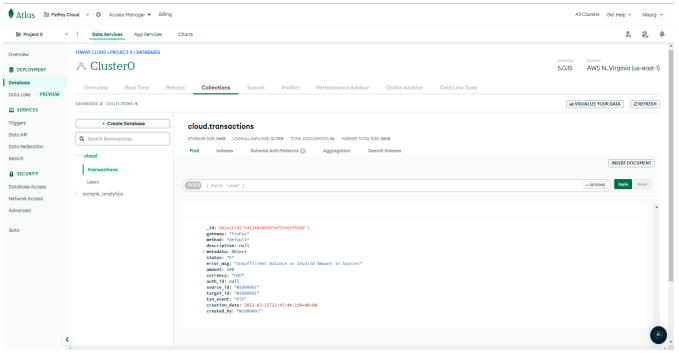

Fig. 6b MongoDB Database (Transactions)

The application is exclusively for the user, specifically SCU students. The sole requirement of the application is that the user should have a valid SCU student ID (which starts with "W") and an email account (ending with @scu.edu). The user first registers details which include full name, student ID, university email address, password and mobile number. Once the user is validated and registered in the database, they can log into the application using their SCU student ID where further authentication is employed in the Node.js server. The data is sent to MongoDB database via the Node.js backend layer where the data endpoints are devised through the HTTP GET/POST requests, integrated with the Flutter SDK frontend via Node.js and in-built Flutter pub packages to perform the endpoint functions when called. The user after logging into their account is routed to the dashboard page of the application where the list of users' contacts registered with the application is displayed in order to ease the flex point transactions between peer-to-peer. The database layer where the registered data is stored in JSON format is in consistent functioning with the application and backend layer.

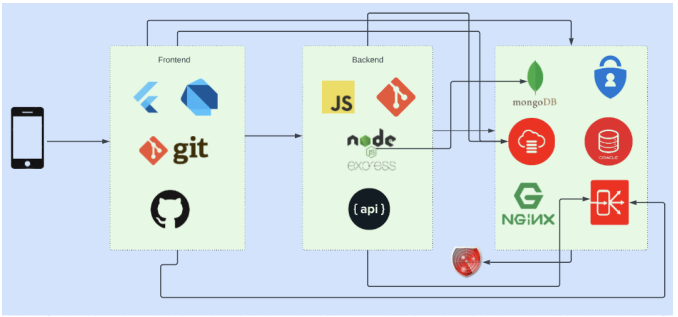

Fig. 7 Software Architecture for FlexPay

The API endpoints consist of functions for the application which includes withdrawal, deposit and

peer-to-peer transfer which is designed on the Payment tab of the application where necessary calculations are done from the backend on initiating any of the above payment methods. The multi-factor authentication (MFA) can be enabled and verified for registered email users only. The authentication is initiated with Microsoft authenticator where the registered user will be notified with a timed one-time password (OTP) which is 6 digits. The authentication is verified on entering the OTP. MFA can be enabled, verified and deleted as per the user in the Profile page which also provides logged in users' details and an option to edit their details and logout of the application.

IV. Outcome

We have successfully built a responsive application that broaden the usage for Santa Clara University (SCU) students where it handles multiple transactions of flex points which includes deposit, withdrawal, and peer-to-peer transfer, maintaining security and data protection using Oracle Cloud Infrastructure (OCI) where our server is hosted over the cloud using Oracle Compute Instance and handles data control using HTTP requests from the endpoints in the Node.js server. The Computer Instance allowed us to launch the server on which we can manage and control the applications. Furthermore, we built a server that is connected to MongoDB Atlas Clusters with load balancing implemented in order to maintain stability and performance without any downtime. For additional security and encryption for SCU only, Multi-Factor Authentication (MFA) was enabled which provides offline timed one-time password (TOTP) service.

V. Analysis And Future Work

This application currently supports flex points based transactions for Santa Clara University students. We can create a prime system which would provide other educational institutions to join the system and add their wallets with equivalent point based currency, similar to flex points, which would further aid in extensive use of the application and ease for other university students as well, in situations where the points have an expiry during the end of each quarter/semester system and the rush to overuse them even without any need. A feature would be enabled to allow peer-to-peer transfers between their contacts within and outside the university where specific point exchange value, similar to foreign exchange, can be agreed upon by the universities. The application would have an option to auto generate statements in order to track the expenses on a monthly basis. Push notifications will be implemented to notify the user on each transaction. Currently, our system uses Microsoft authenticator for TOTP, lasts for about 30 seconds. We can increase the expiration timestamp by using the subscribed version of Microsoft authenticator in order to increase the entry time. Moreover, authenticators like Google Authenticator can also be employed as an alternative for the current OTP system.

VI. Conclusion

We have demonstrated a system known as FlexPay application using Oracle Cloud Infrastructure (OCI) which we learnt and implemented over the course of the project with a user-friendly frontend and backend integration which interacted with each other using HTTP requests. The idea behind this project is to broaden the existing features in the payment application used by students of Santa Clara University (SCU) wherein we have successfully implemented flex point transactions which include peer-to-peer transfer and Multi-factor Authentication (MFA) for data protection and security.

Some of the limitations of our project include:

1. Timed One time password (TOTP) which expires within 30 seconds for MFA.
2. Deposit and withdrawal in your student account used by the university.
3. Official SCU login authentication which includes DUO mobile.
4. Currently, only student ID is used for login while we can authenticate using email as well.
5. Official permission access from SCU to validate university students.

Acknowledgment

We would like to thank our professor and our project mentor Prof. Sean Choi, Department of Computer Science and Engineering, School of Engineering at Santa Clara University for his guidance and constant support throughout the project and the course.

References

[1] "OCI Documentation" - Oracle Official Website.

[2] "Oracle Cloud Infrastructure Platform Overview" - Oracle Official Website

[3] V. Sanjai, Dr. Tr. Kalai Lakshmi, Keelkatalai, Nanganallur, "A Study on Usage of Online Payment Apps by Customers" - International Journal of Creative Research Thoughts (IJCRT) 2021, Vol. 9, Issue May 5, 2021, ISSN: 2320-2882

[4] Md Arif Hassan, Zarina Shukur, Mohammad Kamrul Hasan, Ahmed Salih Al-Khaleefa, “A Review on Electronic Payments Security” 2020 by the authors. Licensee MDPI, Basel, Switzerland, 12 August 2020

[5] Dina Yeni Martia, Afriyanti Hasanah, Ira Siti Sarah, Rola Nurul Fajria, Kenneth Pinandhito, “Exploring How University Students Utilize the Help of Digital Payment: a Qualitative Study” International Seminar of Science and Applied Technology (ISSAT 2020), Advances in Engineering Research, Vol. 198, 2020

[6] Sun Yang, Lixia Wen, “Design and Research of Virtual Payment System in Colleges and Universities” Open Journal of Social Sciences, Vol. 8 No. 6, June 2020, DOI: 10.4236/jss.2020.86035

[7] Michael Osei, Regina Anastasia Ahuren, “Financial Methods of Fees Payment in Academic Institutions: A Case of Takoradi Technical University” IOSR Journal of Economics and Finance (IOSR-JEF) e-ISSN: 2321-5933, p-ISSN: 2321-5925. Volume 12, Issue 4 Ser. VII (Jul. –Aug. 2021), PP 37-43

[8] “Creating a Load Balancer Using OCI Load Balancing” - Oracle Official Website

[9] “HTTP Load Balancing” - Nginx Official Website

[10] “MongoDB Atlas Documentation” - MongoDB Official Website

[11] GitHub Repository Link - https://github.com/benitarego/FinPay-App ; Commit ID: 03f1a7a44b3d3553ea67fb55e0abee3804bab849