**PROPOSED EXTENSION OF THE MUSOEDU WEBSITE TO PROVIDE FEE PAYMENT VIA MPESA AND RELATED FUNCTIONALITY**

By;

MURITHI ROY KATHURIMA

TED/142/14

A PROJECT PROPOSAL SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF BACHELOR OF EDUCATION IN TECHNOLOGY EDUCATION

MOI UNIVERSITY

MAY 2019

# DECLARATION

**Declaration by the Student**

I hereby declare that this project proposal work is my original work in substance and form, and has not been submitted elsewhere by any other person for the award of Degree in Bachelor of Education in Technology Education to the best of my knowledge.

Signature………………………………………………….

Name………………………………………………………

Date………………………………………………………...

**Declaration by the Supervisor**

I confirmed that the work reported in this proposal was carried out by the candidate under my supervision.

Signature…………………………………………………

Name…………………………………………………….

Date……………………………………………………..

# DEDICATION

I dedicate this project to my mother, Ruth Murithi, for her unwavering support, both financially and psychologically and her grand sacrifices. Thank you.

# ACKNOWLEDGEMENT

I’d like to appreciate my project’s supervisor Mr. Oloo for the unique perspective he accorded and elevating me to think of the bigger picture. Due gratitude also goes to the project’s coordinator Dr. Keter for successful guidance and many directives on how to get necessary materials required for the project. I’d also like to appreciate all the lecturers in the department for all their advice and guidance throughout my stay in the department; special thanks to Dr. Mutai and Mr. Kemei who have been pivotal in my understanding of the various daunting concepts in the Electrical department.

# ABSTRACT

In the last year or two, Moi University has made a technological leap by moving some of the services that were offered manually to the web. The services include; signing of the nominal roll, registration of courses, printing of the exam card, releasing of results and also checking the fee balance. The students however, still have to queue at the bank in order to pay their fees. In addition, upon printing the exam card, they have to queue yet again at the accountant’s office in order to get their exam cards stamped. Furthermore, in the event that the fee has not reflected, especially when doing last minute payment, the student has to queue at the accountant’s office to have their balance confirmed and then updated to enable them to print their exam card. The project aims at eradicating these inconveniences to the students while concurrently making the administration tasks exponentially easy. Moreover, the process uses the latest technology to ensure reliability and security. The project also aims at targeting those students who may not feel confident with making payments via a web interface and implements an intuitive USSD solution. In addition, the students can also check their fee balance via a telegram and whatsapp chat service depending on their preference which they can opt out as they wish. The projects further eliminates the need of having students exam cards stamped after downloading them. It does this by generating a unique string along with the timestamp of the time the card was generated. These two values are placed on the exam card while concurrently being sent to the student’s phone number. The students will then be prompted to produce the text and the exam card during the exam.

# TABLE OF CONTENTS

[DECLARATION ii](#_Toc17388076)

[DEDICATION iii](#_Toc17388077)

[ACKNOWLEDGEMENT iv](#_Toc17388078)

[ABSTRACT v](#_Toc17388079)

[TABLE OF CONTENTS vi](#_Toc17388080)

[CHAPTER 1 1](#_Toc17388081)

[1.0 Background information 1](#_Toc17388082)

[1.1 Statement of the problem 1](#_Toc17388083)

[1.2 Objectives of the study 1](#_Toc17388084)

[2.0 Research questions 1](#_Toc17388085)

[3.0 Justification and significance 1](#_Toc17388086)

[1.5 Merits and demerits 1](#_Toc17388087)

[1.6 Assumption and precautions 1](#_Toc17388088)

[CHAPTER 2 1](#_Toc17388089)

[4.0 Literature review 1](#_Toc17388090)

[CHAPTER 3 1](#_Toc17388091)

[3.1 Methodology 1](#_Toc17388092)

[3.3 Observations and result 1](#_Toc17388093)

[CHAPTER 4 1](#_Toc17388094)

[4.1 Data analysis 1](#_Toc17388095)

[CHAPTER 5 1](#_Toc17388096)

[5.1 Conclusions and recommendations 1](#_Toc17388097)

[5.2 References 2](#_Toc17388098)

# CHAPTER 1

## Background information

As a Moi University student, I am well versed with the logistics that go into fee payment especially if the parents have come up with the money in the last minute. The hustle includes making very long queues at the National bank in the student centre. The bank at the students’ centre is preferred due to its close proximity to the school in the event that something goes wrong, the complains can be resolved expeditiously. The student would then have to take the receipt to their respective school accountant, another long queue, so that the fee balance can be updated so that they can be able to download the exam card if the balance did not reflect automatically. Which is ultimately followed by yet another queue to have their exam cards stamped.

Also, I have differed once and I know the red tape one has to cross in order to get their fee reversed after deferring for an academic year. A lot of sketchy fixes had to be done over the years before I had the issue fixed 3-4years after I applied for readmission. The proposal also provides a solution that prevents the deferred students from being billed.

Definition of terms:

* HTTP: - According to webopedia, HyperText Transfer Protocol is the underlying protocol used by the World Wide Web. This protocol defines how messages are formatted and transmitted, and what actions web servers and browsers should take in response to various commands.
* API: - According to Google, an Application Programming Interface, in computer programming, is a set of functions and procedures allowing the creation of applications that access the features or data of an operating system, application, or other service.
* REST: - According to Wikipedia, Representational State Transfer is a software architectural style that defines a set of constraints to be used for creating web services. Thus, a RESTful API is an API that uses HTTP requests to GET, POST, PUT and DELETE data.
* PoC: - According to Wikipedia, proof of concept is the realization of a certain method or idea in order to demonstrate its feasibility, or a demonstration in principle with the aim of verifying that some concept or theory has practical potential.

## Statement of the problem

It is a challenge to get a fee statement on demand. The staff do not readily issue fee statements unless there is a good reason to it, otherwise, they just tell the student their balance. This is understandable because imagine if all students decided to ask for their statement. It would be chaos. This feature was also made available in the current website but the fee structure itself is usually not available.

Also, a lot of time and man power is expended due to long queues and inconvenience to the staff, coupled with the inconvenience of having to withdraw from MPESA before banking and the delay before the payment reflects in the system makes the event of paying school fees a nightmare.

There is also the problem of availability of information on demand. Fee statements and the reflection of fees upon payment usually takes some time and this can be really frustrating especially when the exams are around the corner. This is where one would have to queue again so as to have their fee balances updated.

## Objectives of the study

The project, in a nutshell hopes to ultimately accomplish the following objectives: -

* Eradicate queueing at the bank in students centre
* Eradicate the queueing at the accountant’s office for update of fee balances in the event of late payment.
* Provide real time update of the fee balance
* Provide an efficient alternative for stamping of exam cards. Provide a way that minimizes forgery of exam cards.
* Provide more efficient ways of getting fee balances i.e. getting fee balance via common messaging applications such as Whatsapp and Telegram and also being able to get fee balance and pay via a USSD application.
* Allow students to make complaints to the finance department in a timely fashion
* Provide the staff and the administrators with an easy method of administering the payment system. Also, the staff can also be able to manage various issues remotely. If for instance an administrator is not within the school and a staff member requires something done using his credentials, he can simply login to his account and get it done off site.
* Consolidate all the information about the transactions in one place.
* Provide the students with a convenient means of payment i.e. allow the students to make payments from anywhere even at the comfort of their homes.

Offer an online payment mechanism where the payments are done and updates provided to the students and staff in real time without the need for paperwork and additional third-party involvement unless in exception cases

## Research questions

Do students get fee statements on demand?

How often do students require the fee statement on demand?

How efficient and flexible is the current payment mechanism?

How efficient and flexible is the current exam card verification mechanism?

Would students wish for a more flexible payment method?

Would the staff wish that the last-minute queues be minimized?

## Justification and significance

This project is in no means meant to replace the old system but rather a complement to it. However, this does not mean modifications wouldn’t be made to the old system to provide for communication between the two entities. The program can either be assimilated into the old system or they can be used in compliment.

For instance, all the database records can be migrated into the website via a simple HTTP POST request. The alternative is that the database can be configured to be read by the website instead.

The website also allows transactions related to a given student to be updated from a third party. This is flexible and the issue of security is considered as the site can be configured to employ OAuth to make sure that the requesting site is not only authorized to make such transactions but also can only access the features that it has only been approved to use. Furthermore, every time a request has to be made, certain tokens have to be provided whose expiry can be set as desired by the site administrators.

The site is powered by Django which is python framework that has been tried and tested and is constantly reviewed and updated providing its users with the latest technology. Python is a powerful scripting language that reduces complexity and focuses on the task at hand. It’s a language commonly used in scientific applications such as a machine learning and data science. Since it is a python framework, it inherits all the powerful features that come with Python.

Instead of having the students queue at the accountant’s office, during the generation of exam cards a unique key and timestamp are generated which appear on the exam card and a copy is sent to the student’s phone number via SMS. For authenticity, the student would have to produce both the text message with sender ID of say, ‘MUSOEDU’ and their exam card and the two values counterchecked.

MPESA is the most common form of money transfer in Kenya. Virtually, all students have an MPESA account and chances are that their fees is sent to them via MPESA with the exception of maybe bursaries and HELB for instance.

The site also identifies that some students might also not have access to a website and provides a PoC of how students could pay and access their account via a USSD code and for the majority who love messaging can also be able to check their fee balance in a chat.

## 1.5 Merits and demerits

Allows the admin to bill the students remotely

Students also do not have to be around school so as to pay fees. This is usually not the case when paying at the bank. Students prefer to pay at the National bank at the students centre. This is due to the fact that in case of claims, the proximity allows ease of airing them. Additionally, the bank has a seamless integration with the University.

## 1.6 Assumption and precautions

In the design of the website the major assumption is that in order to utilize the services provided by the proposed project herein must abide to a RESTful. This is also justified by the axiom that probably the University might run on relatively older technology and in order to comply an interface has to be created.

Since for instance we cannot get the current website to communicate with the proposed site, proofs of concepts via tools that speak HTTP such as postman or curl will be used to simulate the scenario.

Assumes that transaction data that is not natively generated; i.e. not paid to MPESA via the site.

Some of the subtle but important features related to the project and already present in the actual site such as quick request codes (QR) present in the exam have not been re-implemented unless where necessary. This is because the project is simply aimed at extending the already available site.

The API requests have not been secured so as to make demonstration less taxing. In a production environment

Care should also be taken when handling the authentication keys as this can lead to people being able to create or modify transactions arbitrarily.

Due to the escalated privileges of the admin, superuser users have to be selected carefully.

# CHAPTER 2

## Literature review

According to: https://djangostars.com/blog/top-14-pros-using-django-web-development/

### 1. Django is simple

Django’s documentation is exemplary. It was initially launched with high-quality docs, and they are still maintained at the same level, which makes it easy to use.

More than that, one of Django’s main purposes is to simplify the development process: it covers the basics, so you can focus on the more unique and/or complex features of your project.

### 2. Django works on Python

The framework is based on Python — a high-level, dynamic, and interpreted programming language, well-loved by developers.

Although it’s hard to find a language that can cover most programming tasks and problems, Python is a great choice for many of them. It’s one of the [most popular languages of 2018](https://www.economist.com/graphic-detail/2018/07/26/python-is-becoming-the-worlds-most-popular-coding-language), competing with C/++ and Java.



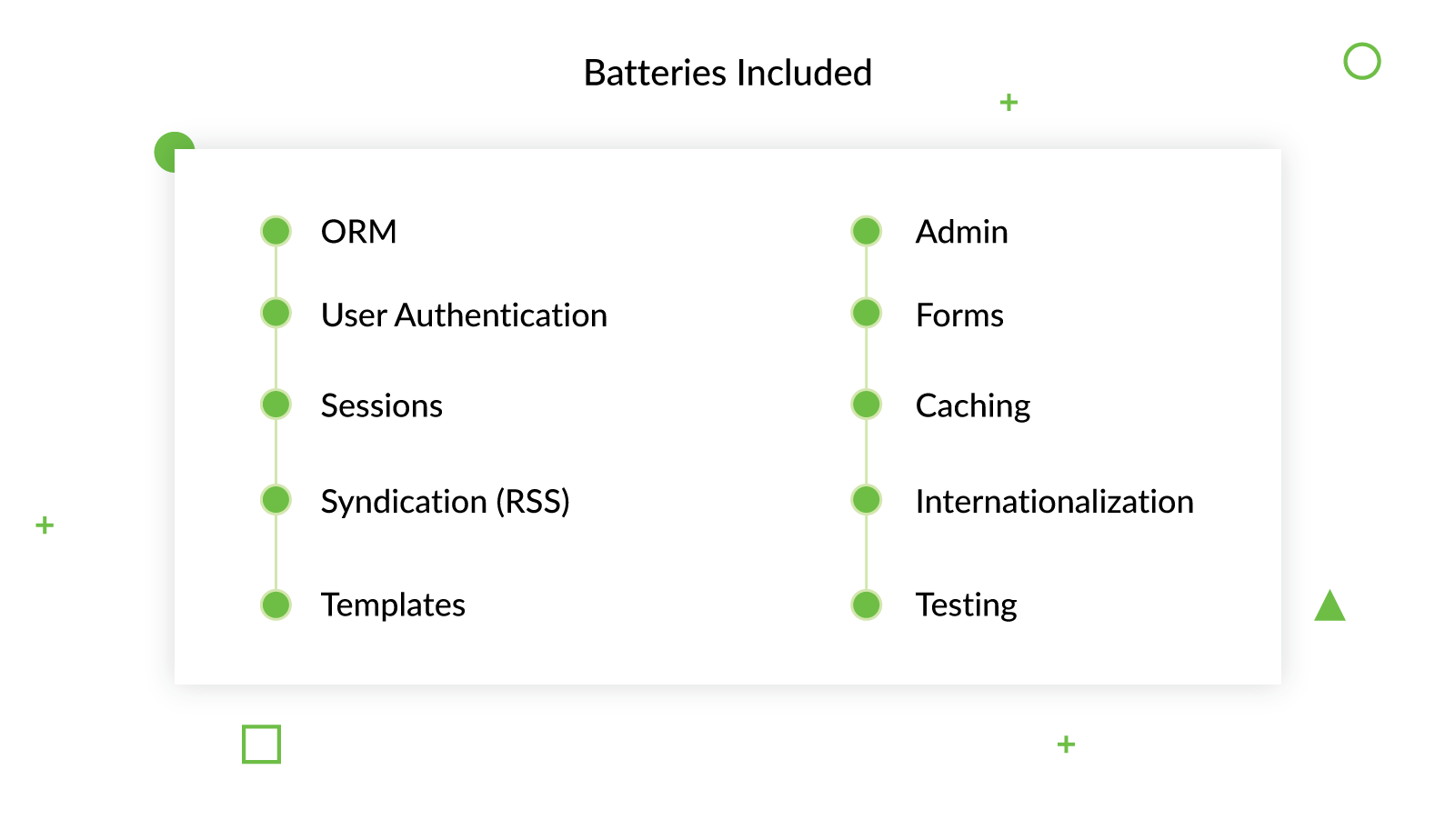
Python is:

* Portable. Your code can be ported to many platforms, from PC and Linux to PlayStation.
* Multi-paradigm. It supports object-oriented programing, which is a simple way to code, as well as imperative programming.
* More interactive than most other languages. It resembles a pseudo-code language and helps you focus on solving a task, rather than on syntax.

[Python web application development with Django](https://djangostars.com/services/python-django-development/) requires less code and less effort. Also, Python has extensive libraries, which make it easy to learn or switch to this language from another one. Customers like Python since it usually takes less time to write the code and, thus, less money to complete the technical part of a project.

### 3. Django has many useful features and extras to simplify development

Django has adopted Python’s “batteries included” approach — the framework has everything necessary to develop a fully fledged application out of the box.



You don’t need to spend hours customizing it to build a simple application or a prototype since all of the essentials are already available. But if you need additional features for a more complex app, there are well over 4,000 packages for Django to cover profiling, testing, and debugging.

The framework also has tool packages for working with cutting-edge technology such as data analysis, AI, and machine learning. They are easy to set up and use in your project. Plus, they’re great if you’re [using Django for FinTech](https://djangostars.com/industries/fintech/) or other math-heavy industries.

### 4. Django is time-effective

Is Django development good for  MVPs and prototypes? Yes, thanks to multiple features that make it time- and cost-effective.

Let’s sum them up:

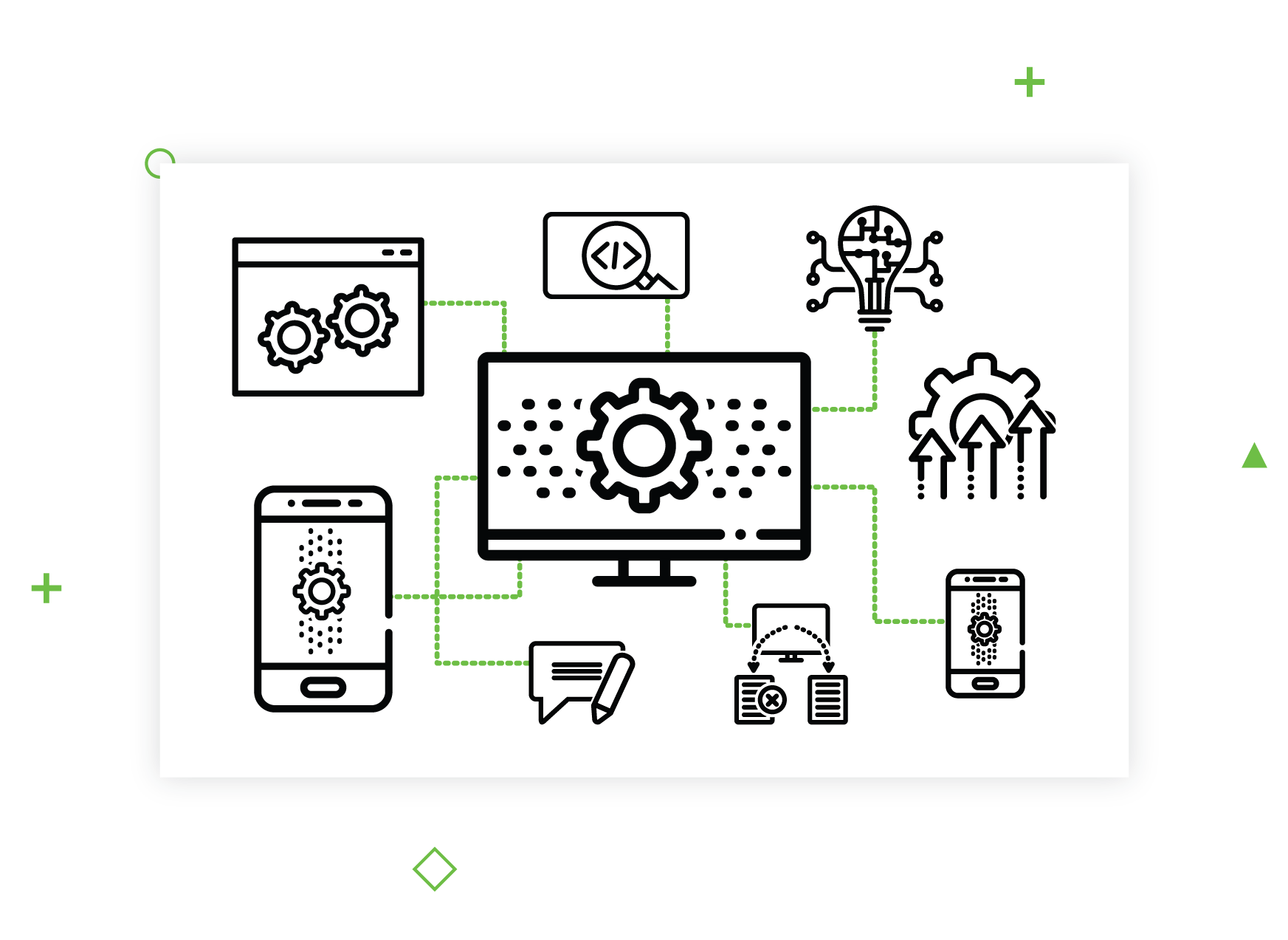
* There’s a flexible, well-structured admin panel, better than Laravel or Yii’s, for example.
* It allows you to reuse code from current or other projects (there is also a [library](https://djangopackages.org/) of reusable apps, tools, and features).
* It has great templates and forms; they were even copied by other projects.
* It has many out-of-the-box libraries and tools that allow you to assemble a good prototype in record time.

### 5. Django suits any kind of project

Django is not an enterprise solution like C# or Java, yet it suits most types of projects, no matter their size. For example, if you’re building a social media type web application, Django can handle the growth at any scale and capacity, be it heavy traffic or volumes of information. But if you want to make something simple, using Django for web development of a blog or a book database, for instance, is an excellent choice as well since it has everything you need to quickly assemble a working application.

In addition to that, Django is:

* Cross-platform. You can create applications that will run on Windows, as well as on Mac or Linux.
* Compatible with most major databases. You can use one or several different databases in one project thanks to Django’s ORM, and switch between the databases with only one line of code.



### 6. Django is DRY and KISS compliant

Django follows the DRY (Don’t Repeat Yourself) principle, which means you can replace frequently repeated software patterns with abstractions, or use data normalization. This way, you avoid redundancy and bugs. Plus, reusing the code simplifies development so you can focus on coding unique features.

KISS means “Keep It Short and Simple”, among its many variations. In Django, it means simple, easy to read, and understandable code. For example, methods shouldn’t be longer than 40-50 lines.

### 7. Django is secure and up-to-date

Django is always kept up to a high standard, following the latest trends in website security and development. That definitely answers the question “Is Django good for web development?” — as security is a priority in any project. Django is updated regularly with security patches, and even if you’re using an older version of the framework, its security is still maintained with new patches. It’s no wonder since Django has an LTS (Long-term Support) version.

### 8. Django is backward-compatible

You can use the interface of Django’s older versions, and most of its features and formats. In addition, it has an understandable roadmap and descriptions — the release notes contain all the information you need to know about changes and, more importantly, when new changes become incompatible with previous releases.

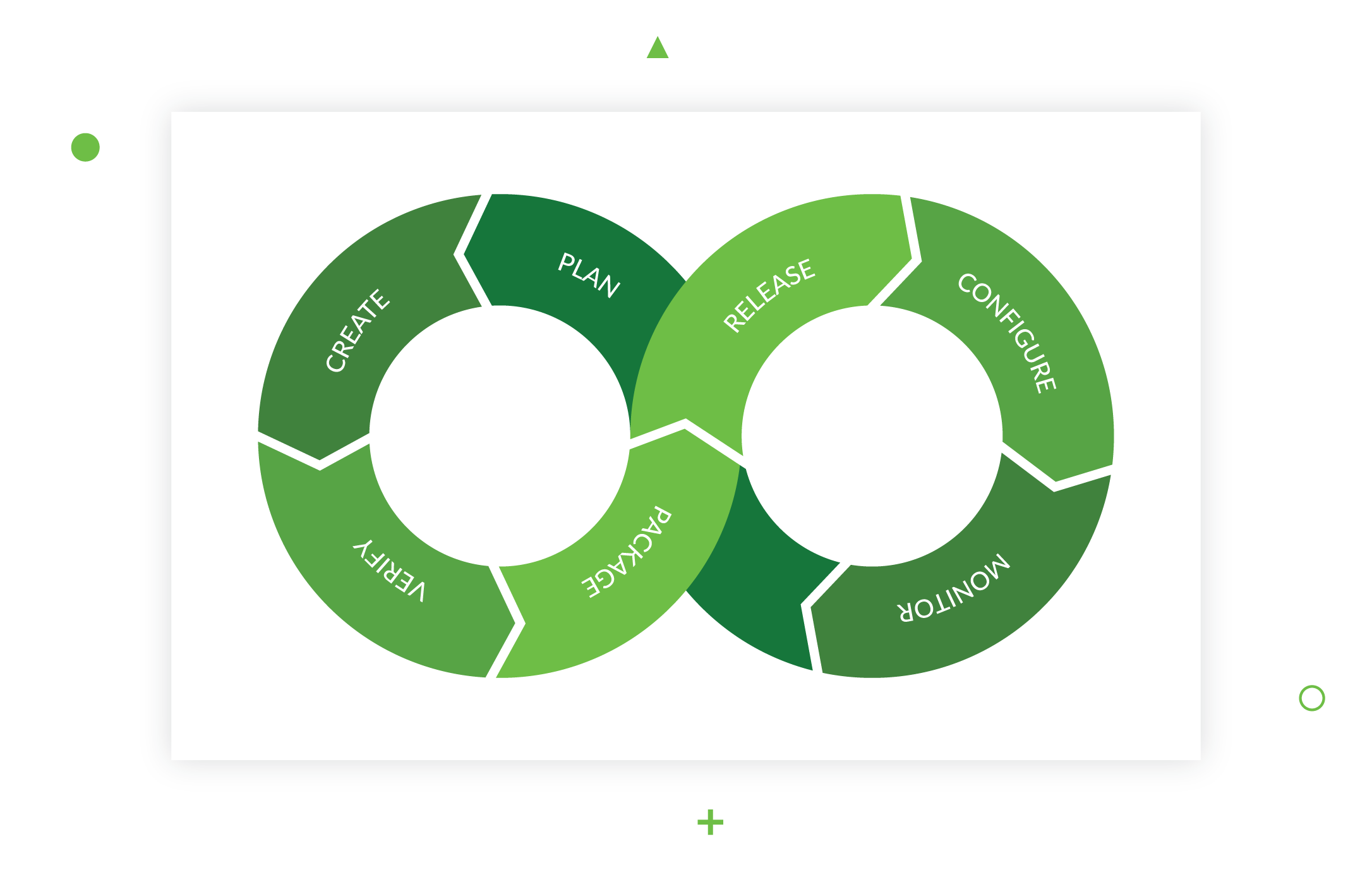
### 9. Django is compatible with DevOps

You can also enhance your project using the DevOps methodology, which aims to shorten lifecycles while maintaining business objectives. It’s especially good if you’re using Django for banking web applications since they are quite complex.

It’s great because you can:

* Solve problems faster with improved operational support.
* Use the continuous delivery approach (an app is produced in short cycles to ensure it is reliable enough to be released at any time).
* Increase the productivity of your team through collaborative working.

**Read More:** [**What is DevOps and Why You Should Have It**](https://djangostars.com/blog/what-is-devops/)



### 10. Django has its own infrastructure

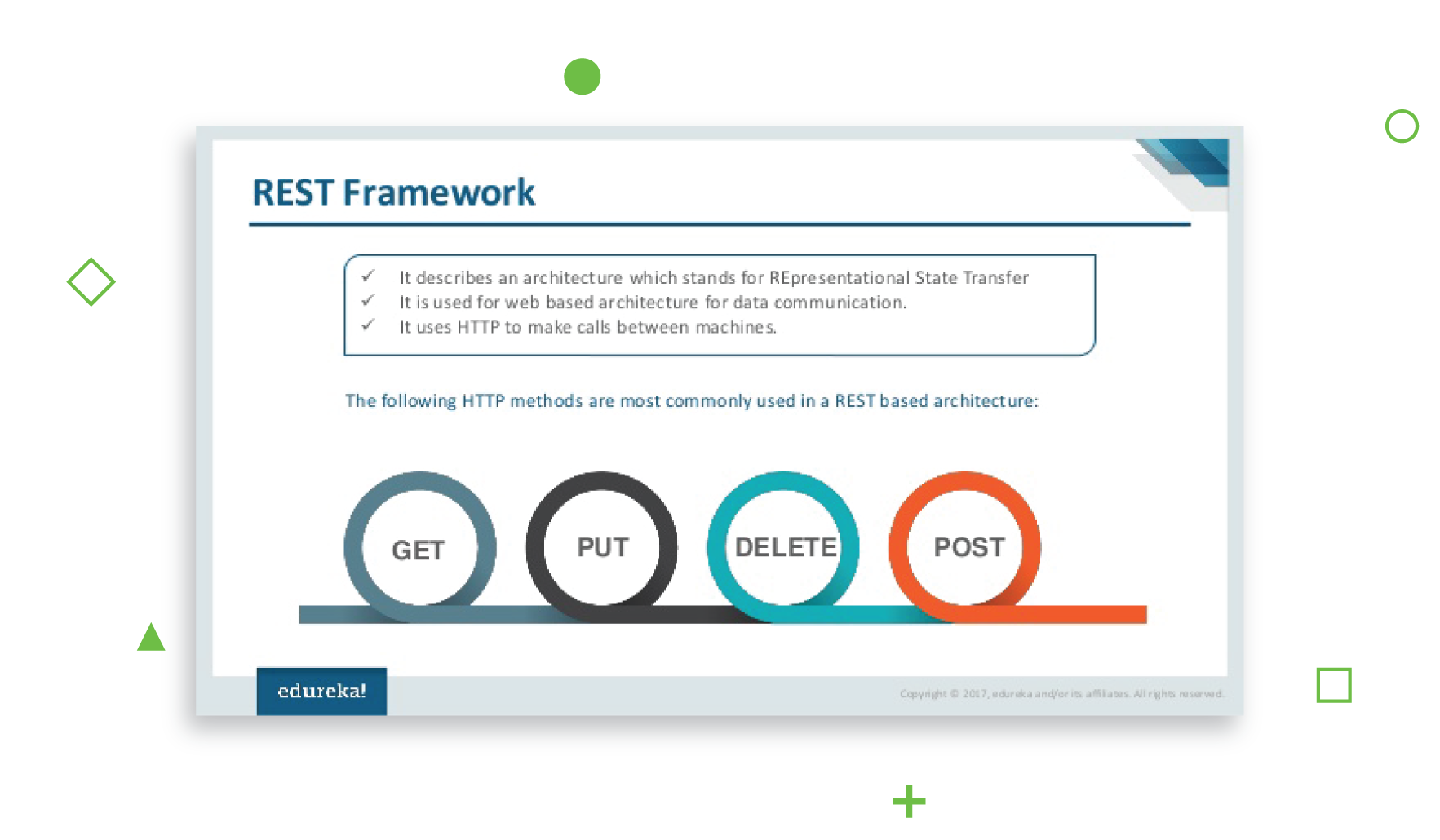
Django doesn’t depend on any outside solutions. It has pretty much everything, from a web server and a templating engine to an Object Relational Mapper (ORM), which allows the framework to use different databases and switch between them within one project.

Plus, Django has libraries and tools for building forms to receive input from users. That’s important for any website that’s supposed to do more than just publish content.

### 11. Django has a REST framework for building APIs

The benefits of using Django for web development also include its Representational State Transfer (REST) framework — a popular toolkit for building web APIs. Django’s REST is powerful enough to build a ready-to-use API in just three lines of code.

One of its key advantages is that it’s extremely flexible: data is not tied to any methods or resources, so REST can return different data formats and handle multiple types of calls. As a result, it can meet the requirements of different customers.



### 12. Django is time-tested

The Django framework has been around for more than a decade, and during that time, it has become the choice of many companies for creating their web applications.

A few of the famous examples are:

* Instagram;
* Spotify;
* NASA;
* Disqus.

10 Popular Websites Built With Django

What is the framework Regardless of the sphere you work in, one of your most important tasks is to create a fast, good-looking website. Today, almost every business needs a website, which acts as a sort of business card for a company or online service. It helps you engage with customers, promote your business, increase sales and so on. In every case, the website should be fast, scalable and dynamic....



### 13. Django has a big, supportive, and professional community

Advantages of Django also include its big, professional community. It’s quite easy to find good developers who know Django inside out and have experience coding with it.

That’s a good testament to the framework’s popularity – but it also means that:

* You can find help or, at least, the right direction in solving harder programming cases;
* The Django community is quick to respond to bugs and fix them;
* As an open source framework, Django is constantly improving – by means of new libraries, for example.

### 14. It’s easy to find Django developers to hire

A huge advantage of the large Django community is that it’s easy to find good developers for your team. Moreover, you can extend an existing team, since all [Django developers](https://djangostars.com) use the same documentation, code pretty much the same way, and can easily read each other’s code.

Top famous projects built with Django

Django allows you to build projects that have different goals and functionality. There are many applications that exist out there. We’ve listed the top popular Django apps below.

### NASA

The National Aeronautics and Space Administration’s official website is a go-to app if you are interested in space. The site features news, pictures, and videos. Django allows the company handle the big number of views and traffic.

### Pinterest

Pinterest allows users to share pictures that showcase their interests. Also, users can follow and share pictures with others as well as content. Pinterest is powered by Django and lets users interact with the platform easily.

### Instagram

You all heard of Instagram. It hosts more than 400 million people and allows them to share photos and videos. Django lets Instagram users browse the app, find and post photos with ease.

### YouTube

We all go to YouTube for fun and serious videos to watch. Initially, YouTube was a PHP-based application. However, the constant need to improve and add new functionality made the company choose Python and Django in particular. The Django framework lets YouTube developers add new features and make upgrades effortlessly.

### DropBox

Dropbox is powered by Python and lets users manage their files effectively. Familiar to all kinds of devices, Dropbox uses the Django framework to enable file storage, synchronization, and sharing.

### Reddit

Reddit is known as the place to search for information and entertainment based on different categories. All posts and links generated by users are promoted to the top via voting. Due to the Django framework, Reddit’s users can enjoy its capabilities and functionality.

# CHAPTER 3

## 3.1 Methodology

The project has been implemented to tackle problems in both the students’ and the administrators’ domain

The Student’s perspective

The student is able to create an account with the site which defines their identity. The students will then be required to login in each and every time to access their account. The students can then navigate to view their fee statement, their details and finally they can be able to pay their school fees via MPESA.

Also, in order to eliminate having to go to the accountant’s office in order to get their cards stamped, when the students generate an exam card, a unique string is generated along with a timestamp. These details not only appear on the exam card but are also sent along as text messages to the number forwarded by the user.

The students also have an alternative to paying via a USSD code which takes them through some simple and intuitive prompts.

A telegram interface also exists in the form of a chat between a bot and the student where the student can query it for their fee balance.

The Admin’s perspective

From this domain, the administrators are divided into either staff and superuser. The superuser is the arch admin with privileges akin to the root superuser in a Linux distribution. The staff user has privileges that allow them to do their work effectively but still limiting their capacity, i.e. things done by a staff user can be undone while some things executed by the superuser are potentially irreversible.

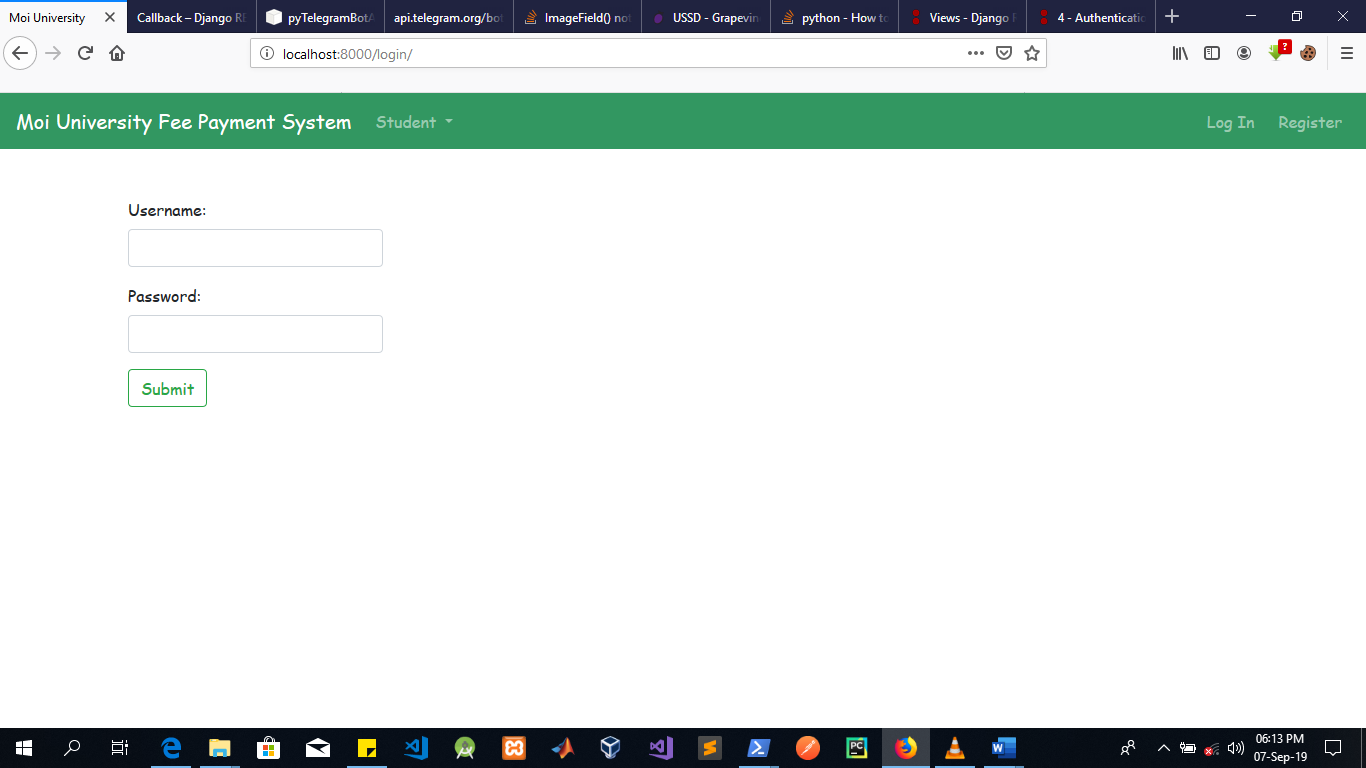
In the project thus far, the staff controls the student billing process. They can set which groups of students are in session and can also invoice them the fee amount that was set up in the admin.

## 3.3 Observations and result

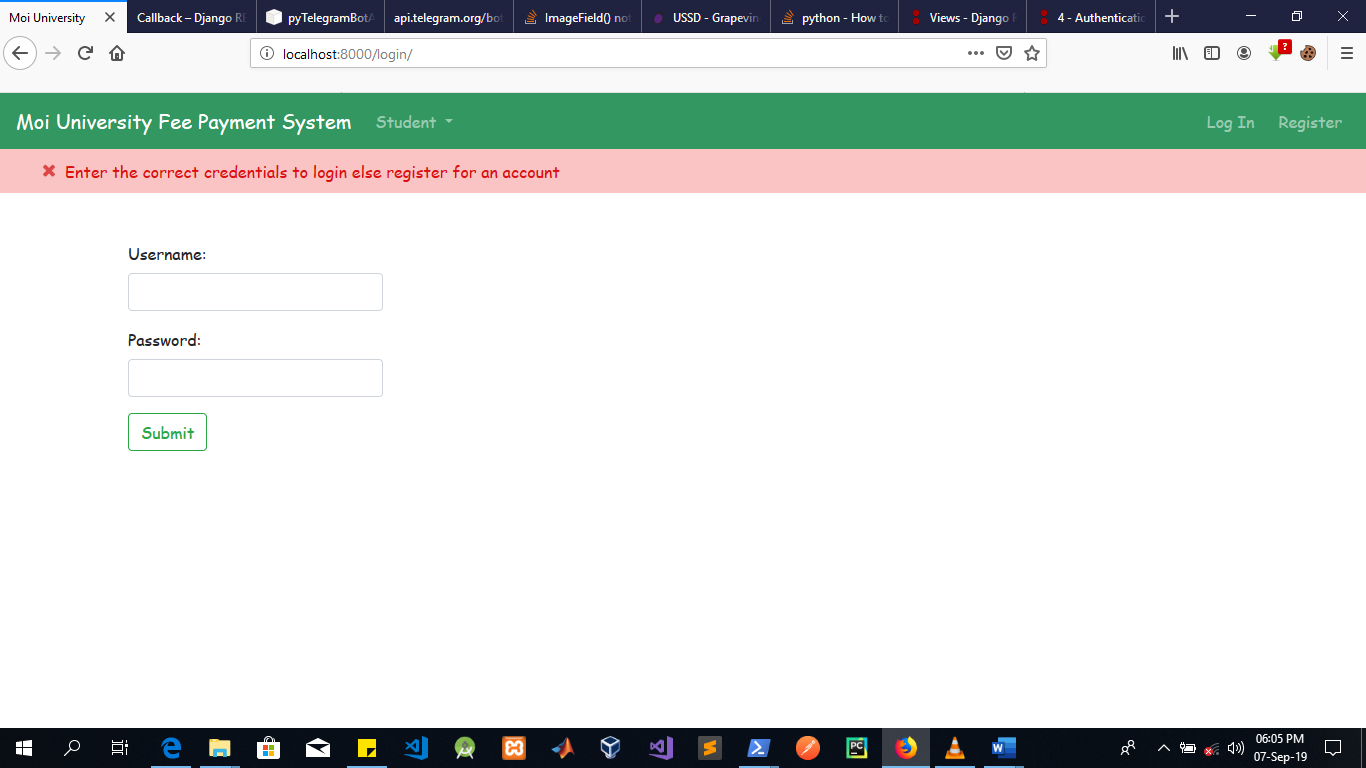
The website is responsive and keeps the users engaged with helpful messages that inform them of the state. When in loading status, the website also displays spinners that are intuitive to the user to inform them of the current action the system is performing.

The payment to MPESA takes a few seconds to process, validate and displays a response indicating whether the transaction was successful or otherwise.

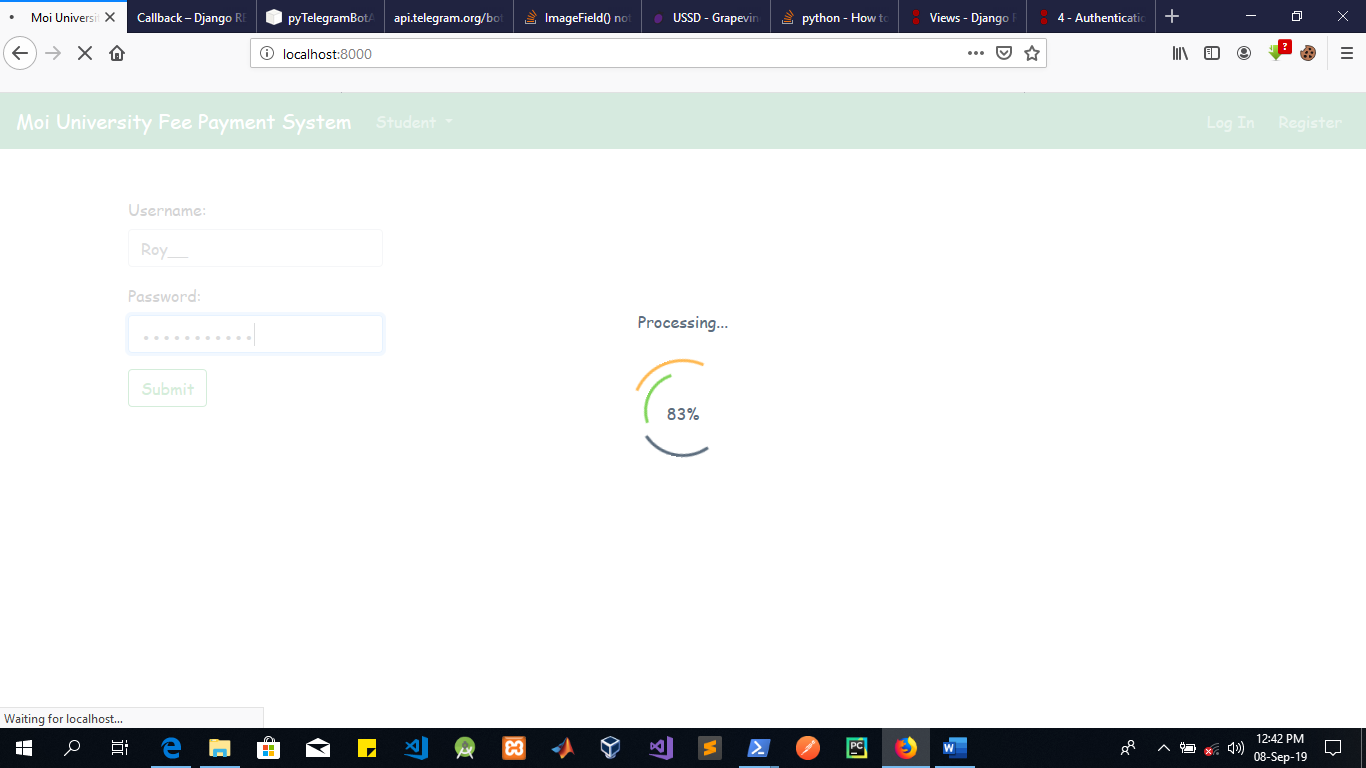
The UI is user friendly and is built to adjust to all screen sizes



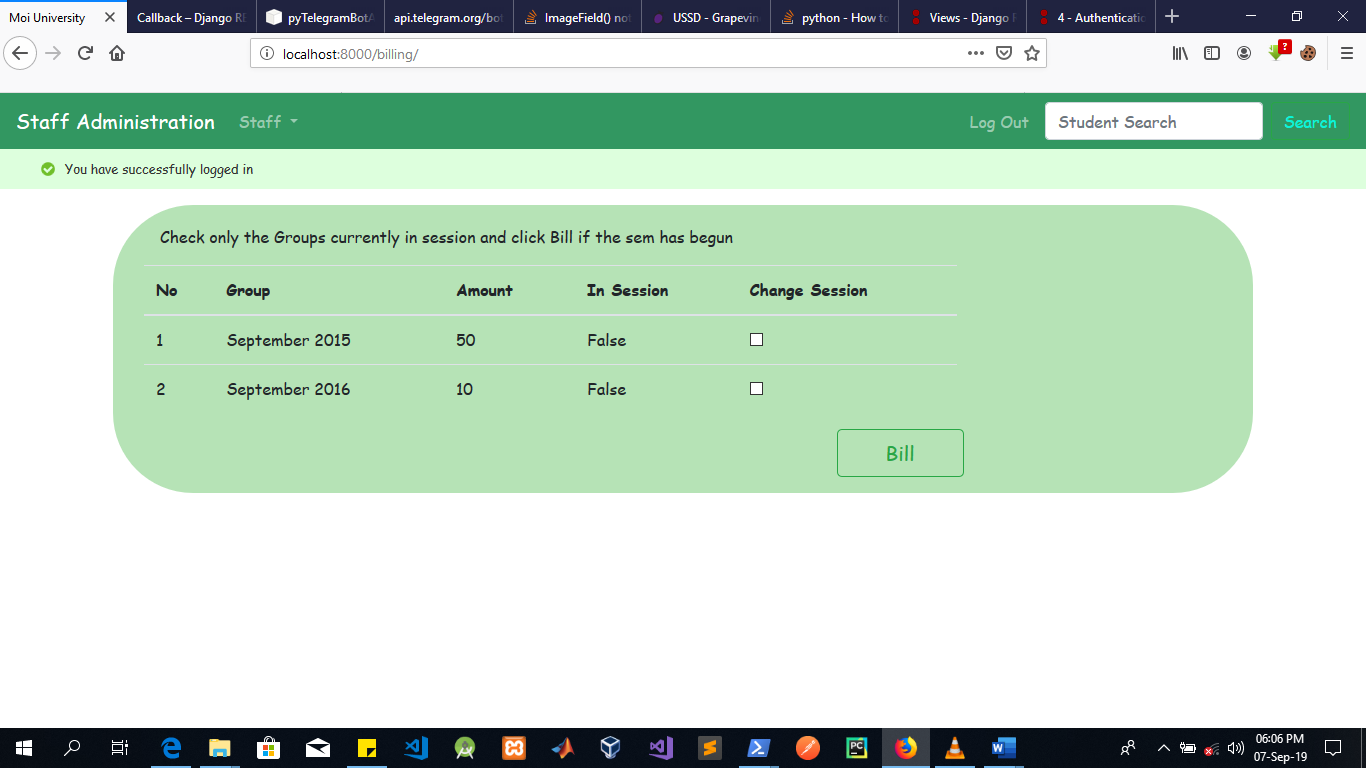
The image above shows the default login page presented to all users. Students and admins alike. The login basically dictates what content will be displayed based on the user that will be authenticated.



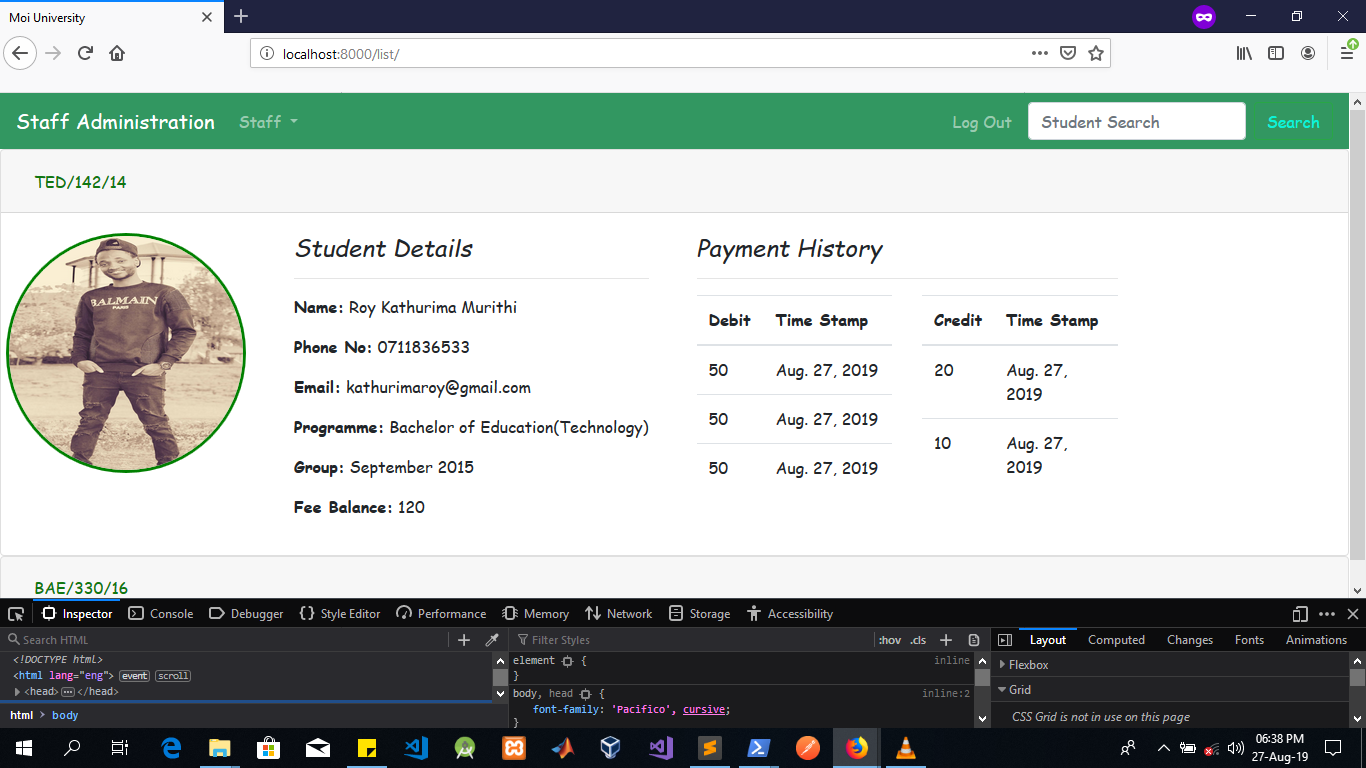
In the event that the user enters the wrong credentials, the pink ribbon with the red text is displayed and the login form re-rendered to give them an opportunity to login again.



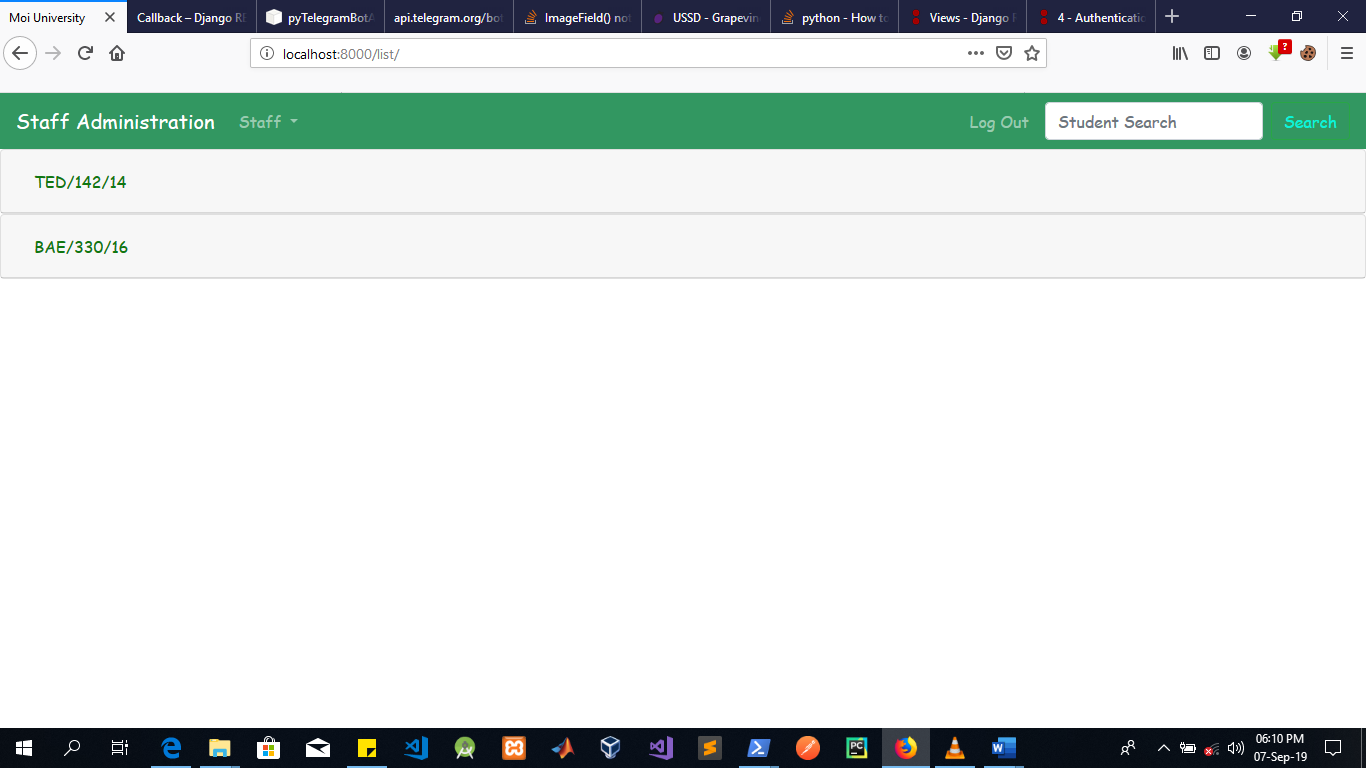
During processing, a loader is displayed that shows the user the progress of the current activity.



The image above shows the billing page. Here the staff member has the capacity to set the cohorts currently in session and be able to bill them. The billing process is applied to all students with the exception of those that have deferred or otherwise completed their studies. It is also visible beneath the navbar a message pops up when the user logs in for their first time.

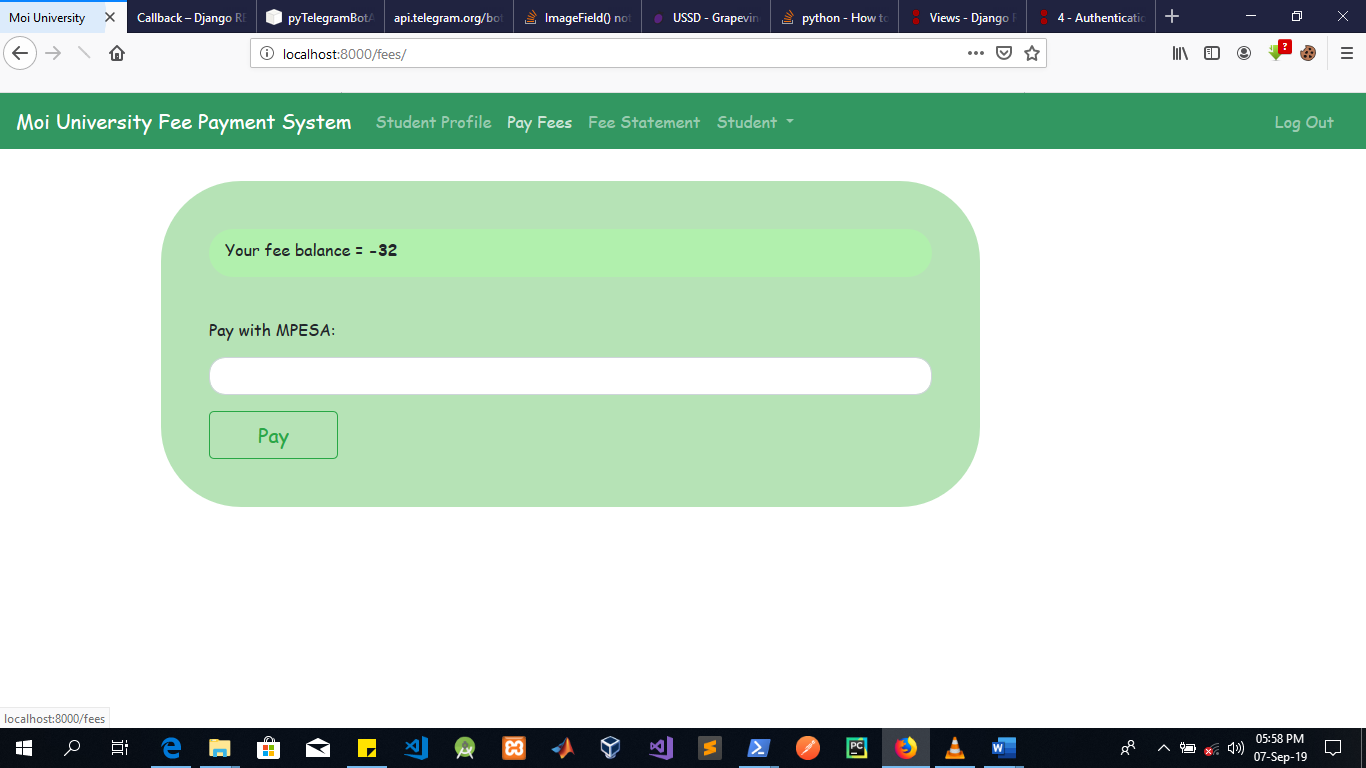


The image above shows the list view from the staff’s perspective. The list is collapsible allowing for the staff user to be able to view the details of the student along with their payment history and fee balance. The navbar which persists in all pages contains a navbar which is only visible to the admin that allows them to search for the given student. The transactions are categorized as either credit or debit, representing cash paid or owed by the student respectively. Below is the same list but in the un-collapsed state.

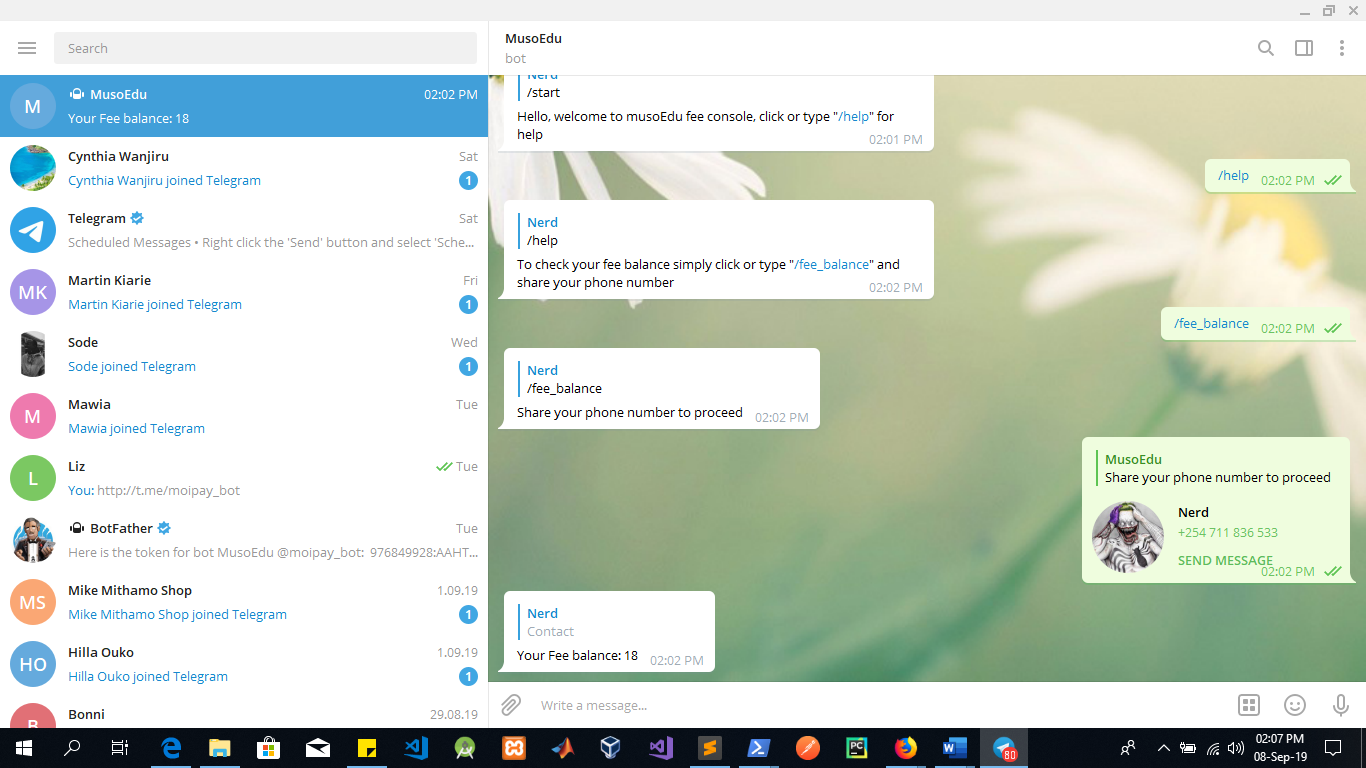




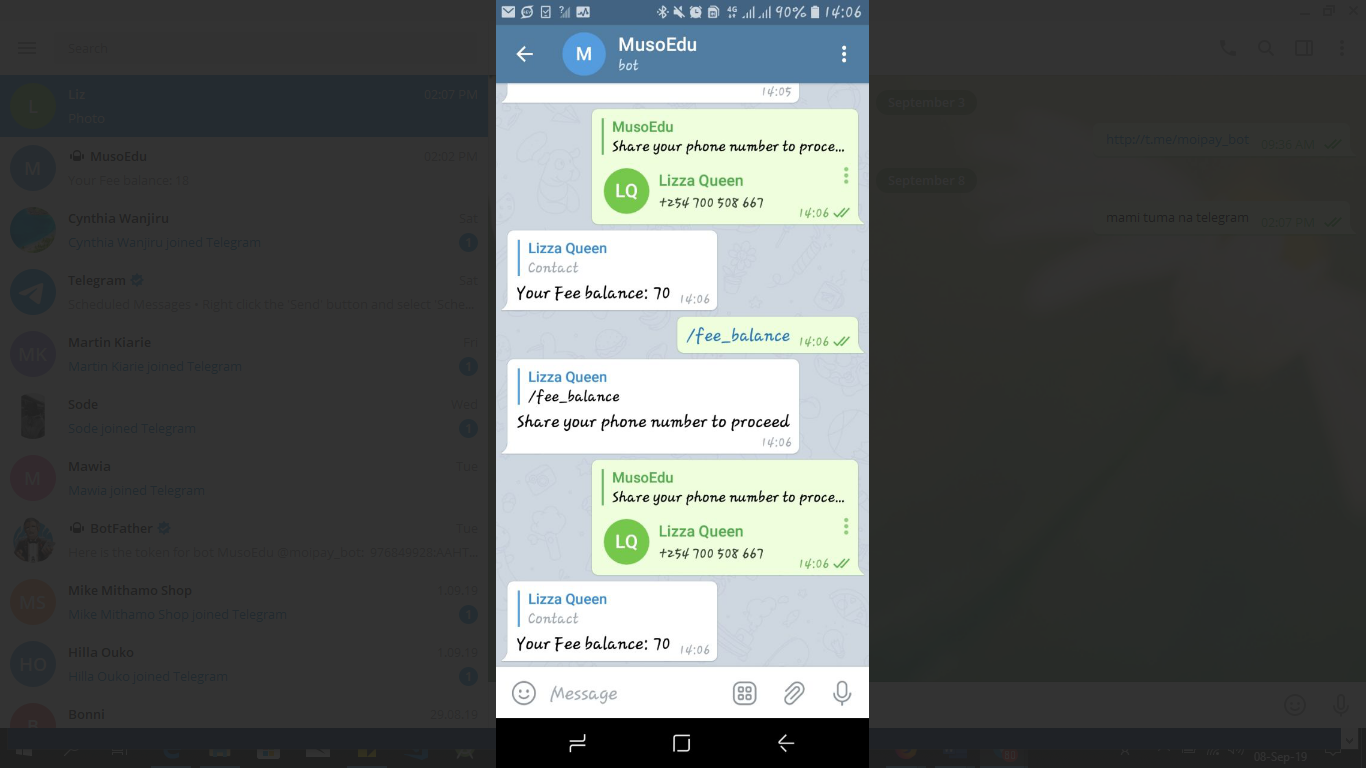
The image above shows the view from the perspective of the student just logged in. In the event of logging in, the student is first redirected to this page, where they can see their details.



The image above shows the payment interface where the student enters the amount they want to pay and immediately press pay. After a second or two, the student will be prompted, on their phone, to enter their MPESA pin number and if they indeed pay, the amount is deducted from their MPSEA account and the fee balance is successfully updated. In the event of transaction failure, the student the fee balance is not updated.



The image above was taken from a telegram client in windows and it shows how one can basically query their fee balance from within the telegram client. The image below, on the other hand shows the telegram client running on a mobile platform.



# CHAPTER 4

## 4.1 Data analysis

The request to mpesa takes a few seconds to process and upon a successful transaction, the Student’s fee balance is updated promptly. The MPESA transaction uses an STK push which prefills most of the information required for the transaction and presents the user with a prompt to solely confirm the amount and enter their MPESA pin.

# CHAPTER 5

## 5.1 Conclusions and recommendations

The old system should be updated so as either to use the site’s API or alternatively assimilate the site’s functionality so as to provide both the admins and the students with convenience

The current website should be made to conform to RESTful practices in order to be able to communicate with other applications.

The proposed project delivers on its claims and would be a better alternative

## 5.2 References

Holovaty, A., & Kaplan-Moss, J. (2009). *The definitive guide to Django: Web development done right*. Apress.

Sanner, M. F. (1999). Python: a programming language for software integration and development. *J Mol Graph Model*, *17*(1), 57-61.

(2019, September 1). Top 14 Pros of Using Django Framework for Web Development. Retrieved from https://djangostars.com/blog/top-14-pros-using-django-web-development/