1 Introduction

1.1 Purpose of this document

This is a system for Pharaohs Experimental Language School that covers most of the operations related to the school, including: teachers, students, parents, employees, schedules, classes, subjects, buses and tuition fees, In order to reduce paper handling and the use of modern methods in recording data which will ensures that the work will be done in a shorter time, more efficient and saves data from being lost.

1.2   Scope of this document

This system Concentration is divided into three factors as following:

Teacher: The ability of the teacher to add homework and exams for the student, Also, the teacher will be able to add student's grades.

Parent: The ability of a parent to check the status of their student (grades, exams, courses), also, the parent now has the ability of paying his student fees online.

Employee: The employee has the ability of managing student's application request, creating the schedule and manage parents , teachers and students information.

Student: The ability of the student to see their schedules, submit his homework, exam and the ability to mark his attendance.

 As for the team we are four members in the team working on both front end and back end

- Our customer is**: Pharaohs Experimental Language School**

- Our project has a delivery date between: **2 - 3 months**

- Our budget to develop this whole system: **As we will work on Web by PHP, so we need domain, hosting service to host our website to the world.**
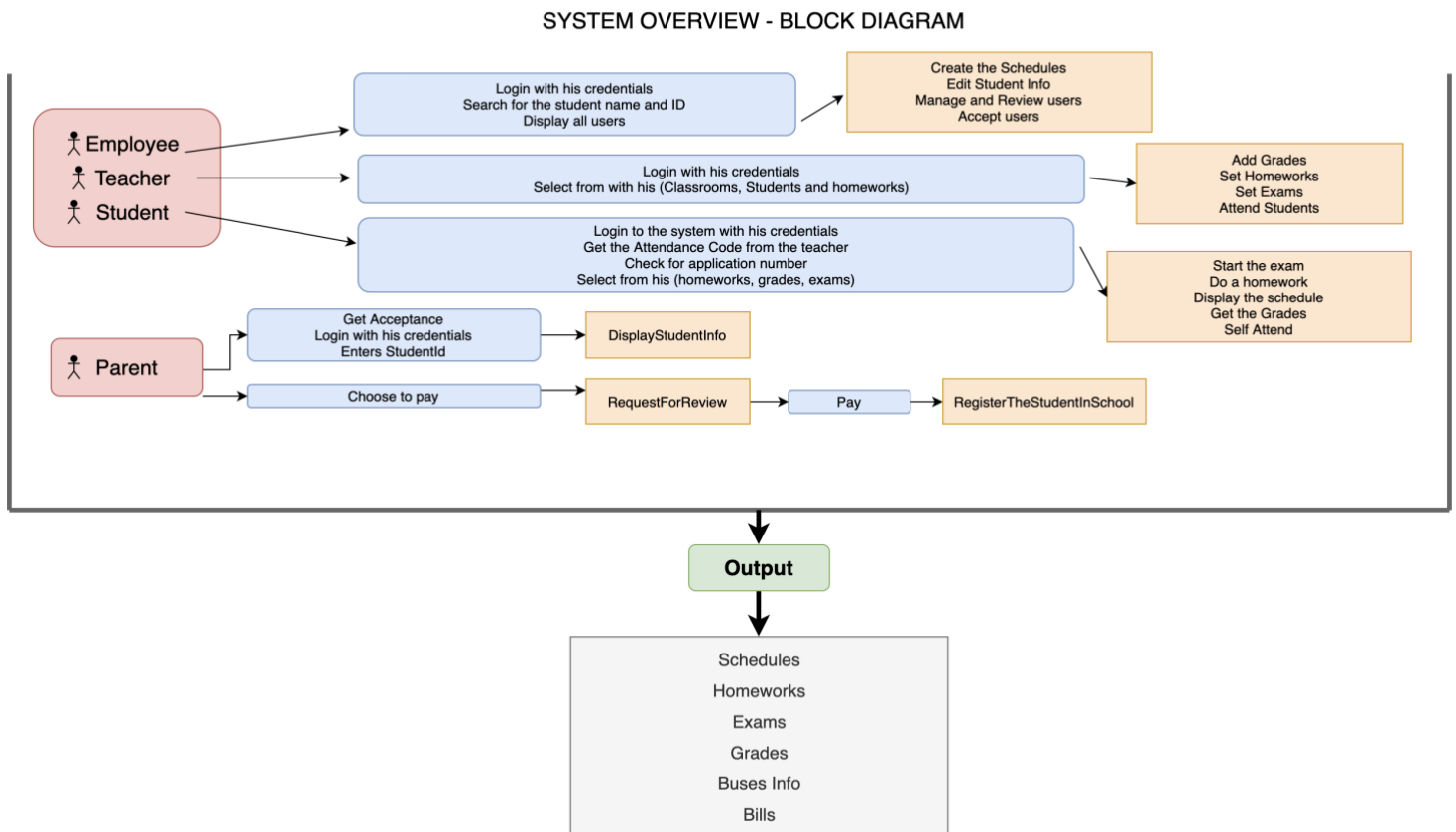
1- **pharaoh.school** domain in Namecheap.com is = **5.88 USD/yr**

2- **Steller Plus** Hosting service in Namecheap.com = **2.68 USD/month**

So, total = 5.88 USD/year, and 2.68 USD/month

1.3   Overview

As it is shown in the next **Figure (1)**

SYSTEM OVERVIEW - BLOCK DIAGRAM

1.4   Business Context

The main target was to provide to the users of the system less time consumption and more efficiency. To be clearer, Ease of accessing data or creating it anywhere with less effort. In order to achieve all of this, the idea was to create a system on which every information about the student is recorded, such as: his grades, his homework, his courses, his exams and his application's data. The system also allows the teacher to easily create an homework and follow up the level of the student by checking his grades, rates, and assignments, and the most important point that the system allow to the parents to check theirs students stats and paying his fees without the need of going to the school. Finally, the system allows the employee to create the student's schedule without the need of doing it manually, check his data and

application request online. All of that will assure less time consuming, less effort and more efficiency.

2 General Description

2.1 Product Functions

The system mainly can be classified into:

Teacher Main Functions:

1- <u>Add Exam or homeworks</u>: The teacher has the ability of creating exams or homework whenever without the need of going to school.

2- <u>Add Attendance</u>: The teacher can attend students who are presents in the class.

3- <u>Add students' grades</u>: The teacher now can mark the student paper easily online.

Student Main Functions:

1- <u>Display schedule</u>: The ability of the student to see their schedules anywhere, anytime.

2- <u>Submit exam or homework:</u> The student has now the ability of submitting his exam or homework online easily.

3- <u>Display grades:</u> The student now can display his homeworks and the overall grades anytime.

4- <u>Add Attendance:</u> The student also can attend himself but by the order of his teacher as he has the ability to share the attendance code so students can attend by this way.

Parents Main Functions:

1. <u>View student page:</u> The parent can check their students' info without the need to go to school.
2. <u>Pay student fees:</u> The parent can pay his student fees without the need of going to the school.

Employee Main Functions:

1. <u>Do schedule</u>: The employee has the ability of creating or editing the student schedule.
2. <u>Edit Students Info</u>: Another thing an employee can do is to edit students' info easily with his admin panel page.

2.2 Similar System Information

1- **esKooly**

This is a system that has the ability of managing the entire school management system for both the student and the teacher. The teacher has the ability of create online class, add students to the class, add quizzes and assignments. The student can attend the class, do the quiz and submit the assignment.
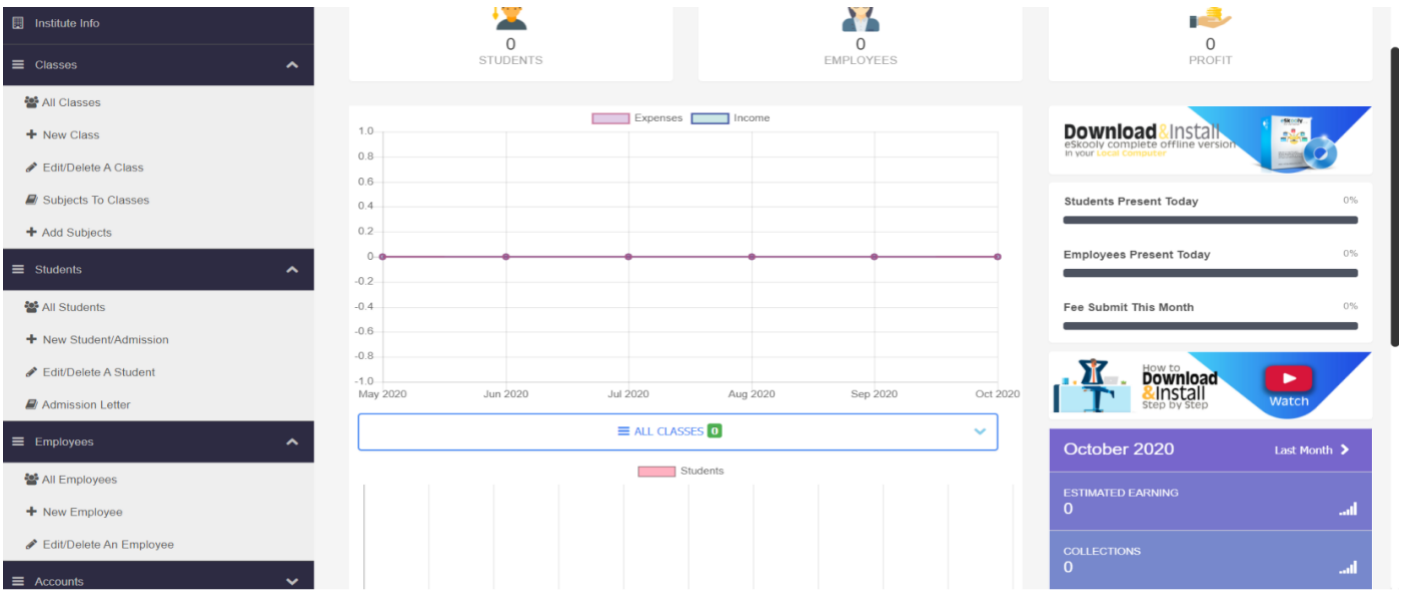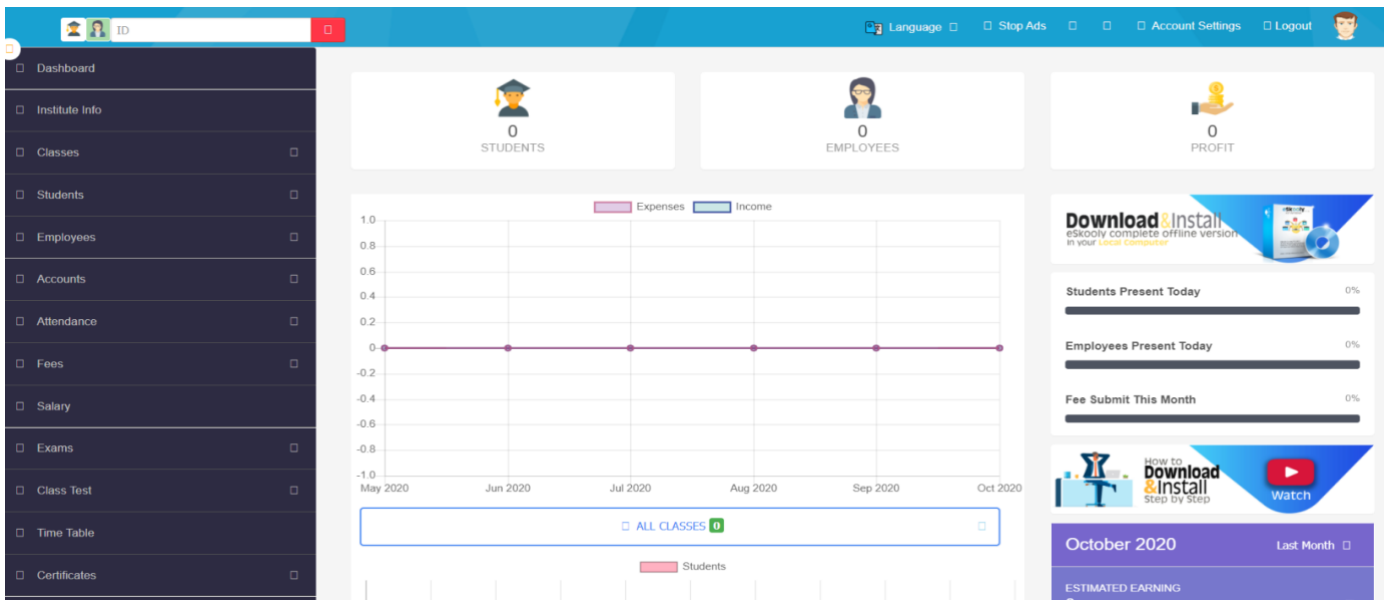
2- EZ School:

This is a school management app that manage the school system for the teacher in wish he can put lessons and the parent can download and view it.
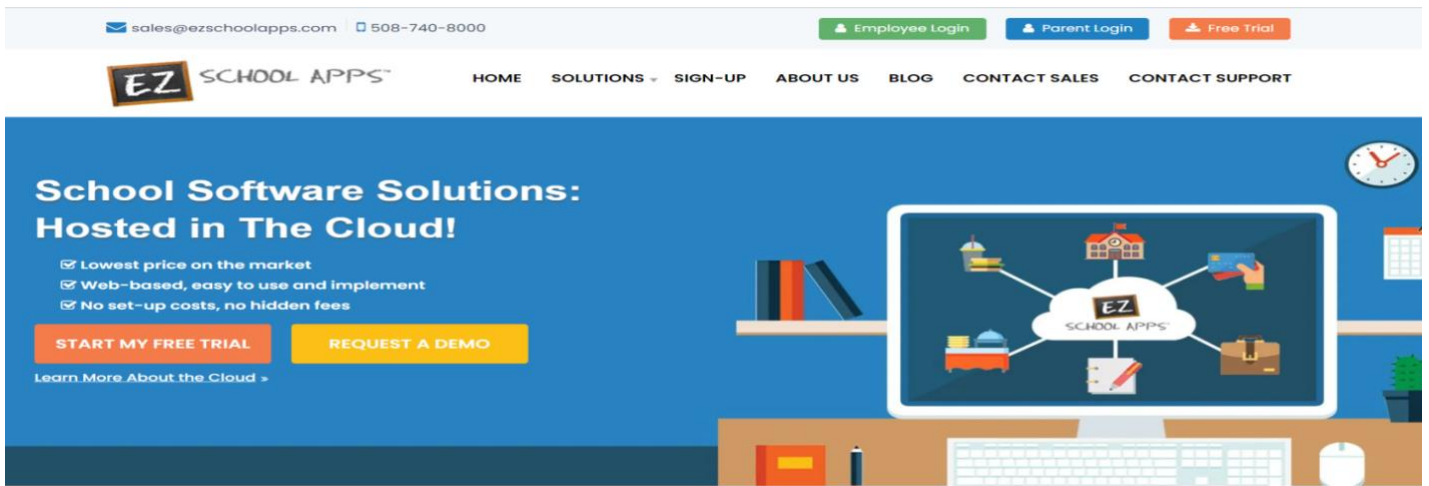
- Our system will be different of these systems because it provides to the parents to check their students stats and fees which is not provided in **esKooly.**

- Also it provide to the students to check their schedules and classes which is not provided in **EZ school.**

Systems	Student	Teacher	Parents
esKooly	Yes	Yes	No
EZ school app	No	Yes	Yes
Our system	Yes	Yes	Yes

2.3   User Characteristics
The user community can be classified mainly into **4** categories:

Teacher: This type of user will do most of the functions such as: creating exams, view student grades etc.... so it require a bit of experience in dealing with such systems and technology.

Student: This type of user will deal mainly with two functions, one of them is submit exams and assignments and the other is viewing his schedule, so, it doesn't require a specific experience to deal with the system.

Parent: This type of user will deal with simple functions in the system so mainly he will do only the following functions:

View student stats and pay or view student fees, it needs experience in dealing with the system and technology.

Employee: This type of user has the ability of creating and editing the student's schedule. This type of user must have an experience because he is the core of the system.

2.4   User Problem Statement

The problems will be divided according to the user classification in the following points:

Parents: The biggest problem was that if he must do anything related to his student he must go to the school.

Student: The student's problems were the disorganization and that he deal with a numerous amount of papers.

Teacher: the teacher problem was how to handle and search in a paper if the amount is large as much.

Employee: the employee's problem was that he was doing the student's schedule manually so it takes much time to do it for all the students.

2.5   User Objectives

The user objectives can be classified into four groups:

Parents: Every Parent will be able to check the stats of his student and pay his fees without the need of going to school.

Teacher: The Teacher now is not forced to deal with these huge numbers of papers, now he can add or mark a quiz online whenever he wants.

Student: The Student will be able to check easily his schedule and submit his quiz or assignment anywhere at any time.

Employee: The Employee will have the ability to display all users in the system and review everyone's data and then decide whether accept him or no. Also he can edit any of users' data and he is responsible to create the schedules within its subjects, classrooms, students and teachers.

2.6   General Constraints

The system does not require many requirements to run it, but there are technical problems that the user may face, we will talk about it from three perspectives:

Parents: The parent may face some problems while viewing his student's stats or paying his student fees as a result of pressure on the system or problems in the network.

Students: The student may face some difficulties during viewing his schedule or submitting the assignment due to a pressure on the system or weakness in the network.

Teachers: The teacher may face some difficulties during viewing the student stats or adding a quiz or an assignment due to a pressure on the system or weakness in the network.

Employees: the employee may face some conflicts while creating the schedule for the school's students.

3 Functional Requirements

Function name (3.1)	CreateAccount($firstName, $secondName, $thirdName, $email...)
Description	Take the data of the user that is registering and insert it in the database with respect to every use type by checking the $_Request url parameter.
Input	- Users' information's (first name, second name, third name, password, address, phone, gender, email, images) and more as a parameters
output	- had submitted an application so that he can be accepted and then have access to the system
critically	This function is the most important one, without it the user can't access to the system.
Technical issues	- Error during insertion - Repeated username, phone number, email
Risks	- Disconnection from the system during registrations
Cost and schedule	- The create account function doesn't require any cost, it would take only minutes.
Dependencies with other requirements	It does not depend on other function
Pre-Condition	-The database is empty -The user can't use the system
Post-Condition	The worker user will be able to manage and use the functions of the system.

Function name (3.2)	insertNewBus($busRoute,$meetAt,$busCode,$driverName, $supervisorName,$supervisorPhoneNumber,$busSeats,$timeMove, $timeArrive)
Description	Take the values as a $_POST method and insert $supervisorPhoneNumber to its own database called **bus_SV_numbers**, also, insert $timeMove and $timeArrive to their own database called **bus_timing** and get rid of duplicate timing. Then, insert the rest values to the database called **Bus**
Input	busRoute, meetAt, busCode, driverName, supervisorName, supervisorPhoneNumber, busSeats, timeMove, timeArrive
output	- true/false
critically	This function will let the students able to find the buses they like and register.
Technical issues	- Database connection problem
Risks	Improper Use of Data and inserting
Cost and schedule	This function doesn't need a cost and doesn't need much time to proceed, depending on the server quality.
Dependencies with other requirements	No dependencies, just the employee need to have the access to do this function.
Pre-Condition	There is no bus available in the bus schedule so that students can't find bus to register into.
Post-Condition	A new bus is added to the **Bus** database and hence displayed in the bus schedule that displayed by the students contains table full of rows and columns showing the details/values for each column with a button to either **REGISTER** or **CANCEL**, but if the seats are full, then display the button with **FULL.**

Function name (3.3)	displayStudentPage($studentId)
Description	- get access to the users and students data table and then loop over the rows there and display the columns in which the student id matches the id with the parameter
Input	-Student Id
output	-Student grades and info
critically	This function allows for the parent to check the student stats and keep updated with his grades and HomeWorks.
Technical issues	- StudentId is not valid
Risks	Improper Use of Data and inserting
Cost and schedule	This function doesn't need a cost and doesn't need much time to proceed.
Dependencies with other requirements	This function can't happen without that the parent has an account (Function 3.1)
Pre-Condition	Student info were unknown by the parent
Post-Condition	Student info is now displayed to the parent as a table showing every subject with its grade and shows the attendance of the student.

Function name (3.4)	addHomework($class_id, $title, $degree, $details, $image, $deadline)
Description	- Take the inputs as a $_POST from text fields and then insert it to the database table named homework.
Input	Homeworks' attributes (class_id, title, degree, details, image and deadline)
output	The homework will be saved and inserted in database and ready to be displayed to the students.
critically	This function will help the teacher to create an online homework.
Technical issues	Conflict between classes
Risks	Database failure
Cost and schedule	This function doesn't require a specific cost; it might take a few minutes to create the exam in database.
Dependencies with other requirements	The teacher must have an account in the system with a user type as a teacher so it depends on(function 3.1)
Pre-Condition	The teacher hasn't created the homework.
Post-Condition	The homework is added to the database and ready to be shown to the students.

Function name (3.5)	addGrade($studentId, $subjectId, $grade, $gradeId, $teacher_Id)
Description	1- Take student Id, Subject Id, the grade, the grade Id and the teacher who is adding this grade as a parameters and then insert these values to the database called gradingMethodValues.
Input	- The studentId, subjectId, grade, gradeMethodID and TeacherId
output	- Grades has been added and the student can check it.
critically	This function allows the teacher to add the student's grades online
Technical issues	-Grade exceeded the grading method limit -Grade is less than zero
Risks	-Data corruption
Cost and schedule	No cost, it might take a couple of minutes to add student's grades
Dependencies with other requirements	The teacher must have an account on the system, the teacher must have added an homework and the student must have submitted his homework in the database (function 3.1, function 3.4, function 3.7)
Pre-Condition	The student's grades were unknown
Post-Condition	The Student can display his grades in his dashboard, in a table full of every subject followed by each grading method and the grade for the student, and the overall grade is also displayed(Decorative dp), also after this function done, employee can generate the certificate and call Function(3.9)

Function name (3.6)	submitHomework($studentid, $homeworkid, $arrayOfAnswers)
Description	Take the answers as an array and insert it to the answers database with student id and homework id as a reference.
Input	- student id, homework id and the answers.
output	- Students' answers have stored in the database and then the teachers can display it.
critically	- This function allows the student to submit his homework online.
Technical issues	- Error while submitting the homework.
Risks	- System Failures: The result of pressure on the system due to the large number of submissions on the system
Cost and schedule	No cost, it might take a couple of minutes while submitting the exam
Dependencies with other requirements	In order to submit an homework the student must have an account , the homework must be created by the teacher in database and the student must paid his fees so it depends on(function 3.1, function 3.5)
Pre-Condition	The homework wasn't submitted in database
Post-Condition	The homework has been submitted and added to the homework database and ready to be graded

Function name (3.7)	checkApplicationStatus($applicationNumber)
Description	Take the application number as a parameter from the input and then loop over the users table and catch all the columns that related to the row with the application number same as the one as the parameter, then by an if condition ask if the accepted column == 0 then echo 'you still in the review' else echo 'you are accepted';
Input	- Application Number
output	- Acceptance status, either yes or no.
critically	- This function allows the users to check if they have reviewed by the school and accepted or not.
Technical issues	- Application number is not valid.
Risks	- System Failures: The result of pressure on the system due to the large number of submissions on the system
Cost and schedule	No cost, it might take a couple of nanoseconds while searching either accepted or not.
Dependencies with other requirements	In order to check the application status user have to CreateAccount and then receive the number that will be displayed to him function(3.1)
Pre-Condition	The users' don't know either they are accepted or not.
Post-Condition	The users' now know either they are accepted or not and will receive their id and can LOGIN to the system.

Function name (3.8)	generateBill($id)
Description	Take the student id as a parameter and then do SQL query that selects all the bills related to this id, push the array of items to a temp array and then push this temp array object to another temp2 array, by another meaning let it a 2D array
Input	- Student Id
output	- Array of bills
critically	- This function allows the employee to display the bill(s) for specific student.
Technical issues	- Student Id is not valid
Risks	- System Failures: The result of pressure on the system due to the large number of bills on the system
Cost and schedule	No cost, it might take a couple of seconds depending on number of rows
Dependencies with other requirements	In order to generate the Bill, employee should have an account and also the student, so it depends on Function(3.1)
Pre-Condition	The Student's bill is unknown and not displayed
Post-Condition	The Employee will display and see the bills as a table related to the student with the help properties of the bill array that is the output of this function

Function name (3.9)	getCertificate($studentId)
Description	Take the student id as a parameter and do SQL query that return all the properties related to that student id of the certificate which exist in Certificate class
Input	- Student Id
output	- Array certificate object
critically	- This function allows the employee to create the certificate for a student
Technical issues	- Student Id is not valid
Risks	- System Failures: The result of pressure on the system due to the large number of rows on the system
Cost and schedule	No cost, it might take a couple of seconds depending on number of rows
Dependencies with other requirements	In order to create the certificate, employee should have an account and also the student, so it depends on Function(3.1) and Teacher must addGrade Function(3.5)
Pre-Condition	The Student's bill is unknown and not displayed
Post-Condition	The Certificate of the student is displayed as a table showing every subject with its grade and the overall grade, and either passed or failed, also, showing the total percentage followed by a button to transfer him (update his semester) to next semester if and only if he passed.

4 Interface Requirements

4.1 User Interfaces

4.1.1 GUI

Teacher

Add New Homework

Homework title

Homework degree

Homework details

No image has uploaded. Upload Image

Deadline 10/31/2020

Select from your classes

Next

Add New Exam - Step 1

Exam title

Exam Degree

Exam info

Exam date | 10/31/2020 | 🗓

Start time

Exam duration

Select from your academic years | ▼

Next

Add New Exam/HW - Step 2

Question 1:

Question Type: "MCQ" -- It can be MCQ, Image or Text ▾

Question Title

Question Degree

Answer -- Ideal

Choice1   Choice2   Choice3   Choice4

Add New Question

Add the Exam/HW

Add Homework Grades

Select from your classes ▾

Select from your homeworks ▾

Select from your students ▾

See student answers

Add Note

Rate this student: ○ Very bad ○ Bad ⊙ Ok ○ Good ○ Very Good ○ Excellent

Submit Grade

Student Answers

Student Name: **Mahmoud Ahmed Ibrahim**

Student Academic Year: **Prep. 3**

Student Class: **3A**

Subject: **Math**

Question 1 - Title - Degree

Image if exist

Student's Answer/Choice

Student Degree

Add Attendance

Select from your classes --For example: 1B has selected	▼

Provide an Attendance Password	Generate New Password

Generate QR Code

Mahmoud Ahmed Ibrahim   ○ Present   ○ Absent

Display Your Childs' Activities

Select from your childs -- Mahmoud has selected   ▼

Select from Subjects -- Math has selected   ▼

◉ Homeworks   ○ Exams

Next

Display Your Childs' Activities

Student Name: **Mahmoud Ahmed Ibrahim**

Student Academic Year: **Prep. 3**

Student Class: **3A**

Subject: **Math**

Homework 1 - Title

Grade: **80%**

Note: **He should focus on geometry**

Rating: **Ok**

Homework 2 - Title

Has not submitted

Back

26

Create a new Bill

Student Name

Please upload your national ID Photo. [Upload Image]

Starting Semester -- 2021 Semester 1 has selected	▼

Select Type -- Tuition Fees has selected	▼

[Request for a Review]   Already a Student?

Create a new Bill - Payment Process

Tuition Fees for: Student (**Mahmoud Ahmed Ibrahim**)

Starting Semester: **2021 Semester 1**

Net Amount: **2,500 EGP**

Payment Type: **VISA**

Pay with *VISA*

| Card Holder Full Name |
| Credit Card Number |
| Credit Card Type ▼ |

Expiry Date: mm/yyyy | 02 ▼ | 2022 ▼ |

| Verification Code |

[Confirm]

[Back]

Attendance Page

Please enter the attendance code provided by your teacher

Attend

Please enter the attendance code provided by your teacher

Student Dashboard

Hello, Mahmoud!

Homeworks

You have 2 new homework(s).

1 homework(s) still not submitted.

You have 3 homeworks have been graded.

[Homeworks Page]

Exams

You have 2 nearby exam(s).

[Exams Page]

[Weekly Schedule]
[Bus Info]

News

- Tomorrow 2/11/2020 is off
- 3/11/2020 is a regular school day

[Account Info]

30

Edit Student Info

For Example: 191583 Student has choosen

Fees Amount

Academic Year

Major

Net Grade

Class Name

Number of Attended Days

Control Student's Subjects

◉ Paid   ○ Not Paid

No image has uploaded.   Upload Image

Update Account

4.1.2 CLI

4.1.3 API

4.1.4 Diagnostics or ROM

We will use try and catch method, so, we will return the exception code, and then we will know what to diagnose and debug it.

4.2 Hardware Interfaces

4.3 Communications Interfaces

We will use local host in the beginning, using local host requires to install XAMPP or WAMPP applications. Once we finish the system, it will be hosted online via namecheap.com

4.4 Software Interfaces

The Web Application requires a browser to open.

It's highly recommended to use **Google Chrome** with its latest version, also, **Safari, Firefox, Opera and Microsoft Edge** will be able to open it, but maybe they will face some issues.

5 Performance Requirements

In PHP we write:

$appMemoryUsage = memory_get_usage();

$appMemoryPeak = memory_get_peak();

The 2 lines of code written above shows how much usage/peak the web application is using from the memory.

The system works in any CPU (64bit and 32bit) as it will be uploaded to the internet, which any CPU can access it.

6 Design Constraints

6.1 Standards Compliance

6.2 Hardware Limitations

Processor: Any

Memory: 2 GB minimum, 4 GB recommended

Internet Connection: Required

6.3 others as appropriate

7 Other non-functional attributes

7.1 Security

We will use a cryptography extension called OpenSSL to encrypt and decrypt what needs to.

Usage of this function in PHP requires these **parameters** as it follows:

- Get our **encryption string** and then, store the **cipher method**, choose the **openssl encryption method** we want to use, set the **options**, set the **encryption initialization vector** and finally our main key which is the **encryption key,** so, by now, we are ready to call the functions:

$encrypt = Openssl_encrypt($**theString, $cipherMethod, $enc_key, $options, $enc_iv**);

$decrypt = Openssl_decrypt($**encrypt, $cipherMethod, $enc_key, $options, $enc_iv**);

7.2   Binary Compatibility

We will use any CPU (64bit and 32bit) as we will use web application.

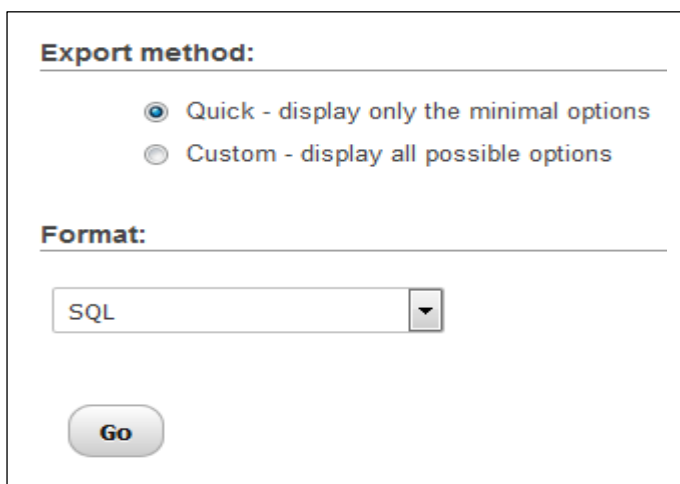## 7.3   Reliability

Not Applicable.

7.4   Maintainability

Not Applicable.

7.5   Portability

The whole system should be able to work with any operating system as it only needs an internet connection, and it will be coded by HTML, CSS, JS and PHP connected to a host.

7.6 Extensibility

In this part we will manage the database to avoid damage of the system if the storage is going to full, in this section, we are going to back up our data that is implemented in phpMyAdmin. The database is exposed to be damaged or deleted by any future reason, so we need to export the the database usually by the export method in phpMyAdmin which allows to save these database offline in a hard disk. There are two methods for exporting, which are quick and Custom, Quick are the default and Custom recovers all the possible options that will be available as shown in the figure.

7.7 Re-usability

The System is divided into components, which will be re-used to obtain information to benefit data from it and re-prioritize it to the system again. **Such as**: If the student got 95% overall grade or more, this student will be upgraded to a classroom for "**A**" students (For Excellent students), we can collect the grades (that added by the teachers) for the student to get the students that will be upgraded to that **A** classroom.

7.8   Application Affinity/Compatibility
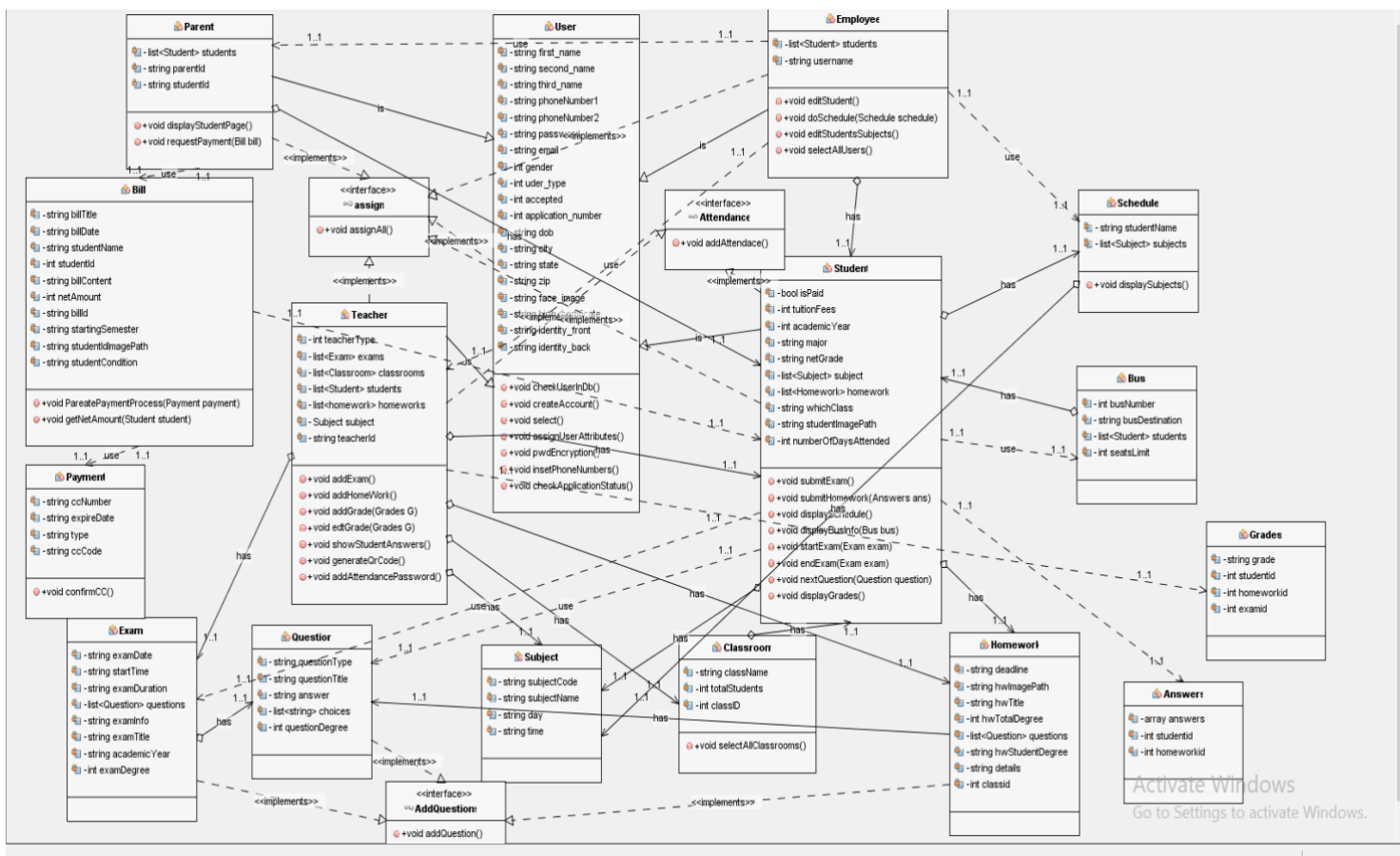
Not Applicable.

7.9   Resource Utilization

Not Applicable.

7.10   Serviceability

Not Applicable.

7.11   others as appropriate
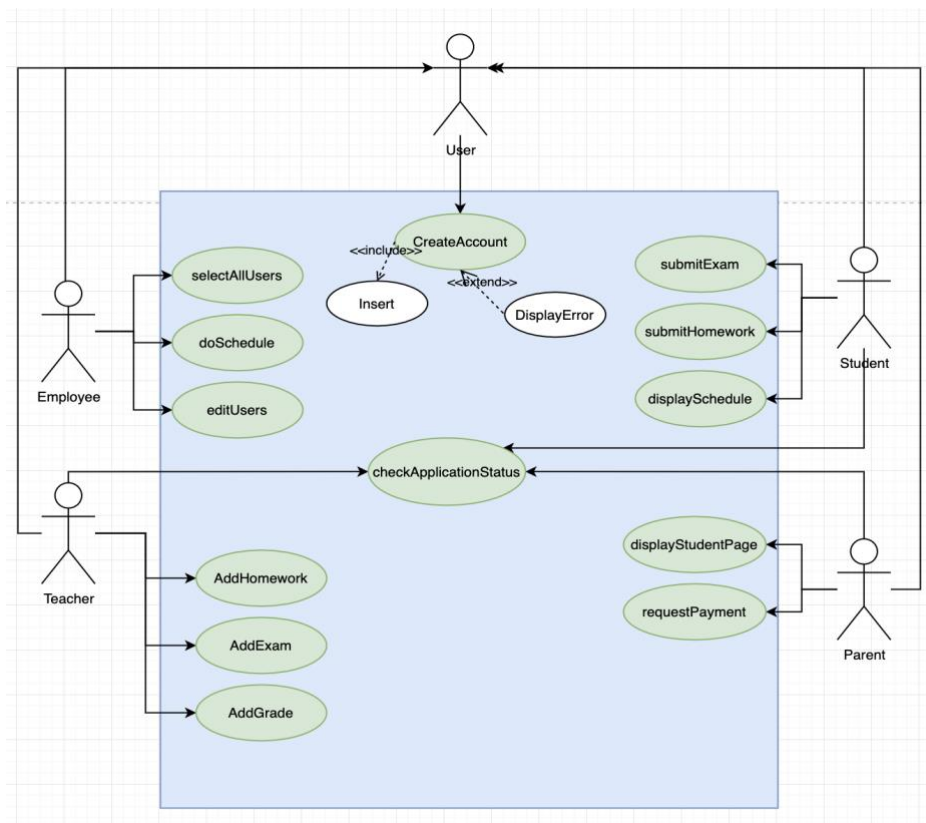
8   Preliminary Object-Oriented Domain Analysis

8.1   Inheritance Relationships

8.2 Class descriptions

List of Super Classes	List of Subclasses
User	Teacher
	Student
	Parent
	Employee

8.2.(1,2,3,4,5,6) CRC Cards

8.2.7 Operations

8.2.8 Constraints:

9 Operational Scenarios

Initial assumption: Parent of a student tries to log into the system and pay the tuition fees online .

Normal: The parent log into the system , then the parent selects payment section to pay the tuition fees for his child ,after he enters card data and student data he presses confirm then a message pop up that the transaction process is done successfully.

What can go wrong: The transaction was made successfully but the student payment status was not recorded on the system . the student will not have accesses to his subjects and can not attend the school because he is payment status is not confirmed.

Other Activities: Transaction maybe made successfully but not recorded on the database or the transaction is made successfully and recorded but the payment status of the student is not updated.

System state on completion: Parent logged in . The transaction is made successfully but the status is not updated on the system . a record of transaction is add to the log showing when was it done .

10   Preliminary Schedule Adjusted

The project is plan is to create a well developed school management system the majors task that are going to be made are : Tuition fees online payment , parent checking student information , Teacher making exams and correcting homework, Employee assigning student schedule , student submitting his homework and exams . these tasks are planned to be finished by the next 2 months .

11   Preliminary Budget Adjusted

The initial budget for this project is 5.88 USD/year, and 2.68 USD/month and it remains the same, also, the implementation part is totally free which will be coded as a web application mostly by PHP.

12   Appendices

12.1 Definitions, Acronyms, Abbreviations

(IV): initialization vector

(**API**): Application Programming Interface

(**GUI**): Graphical User Interface

12.2 Collected material

إدارة أوسيم التعليمية
مدرسة يشتيل الثانوية المشتركة   كشف بدرجات طلاب الصف الأول الثانوي فى

الصف الأول الثانوى
العام الدراسى : ٢٠١٨ / ٢٠١٩

نشاط رياضى	تربية وطنية	تربية فنية	اسبوع وزارة مدرسى	المجموع الثانى	صحة	جغرافى	تاريخ	الفيزياء	كيمياء	أحياء	رياضيات	اللغة الانجليزية	اللغة العربية	اسم الطالب	رقم الجلوس	مسلسل
١٠	١٠	٩٠	٢٠	٢٠	٢٨٠	٢٠	٢٠	٢٠	٢٠	٢٠	٢٠	٤٠	٤٠	٤٠		
															شهد اشرف عبد الحليم عبد الصمد	١٠٠٠	٢٩٢
															شهد اسامة شعبان على	١٠١٢	٢٩٣
															شهد حماد عبد الرؤف حسانين	١٠٠٠	٢٩٤
															شهد خليله على خليفه	١٠١٠	٢٩٥
															شهد ضياء الدين يحيى سعد		٢٩٦
															شهد طارق عبد الرحمن ثابت		٢٩٧
															شهد عمرو محى الدين عبد العزيز		٢٩٨
															شهد ماهر حسان محمد		٢٩٩
															شهد مبروك حسن احمد		٣٠٠
															شهد محمد احمد حسن		٣٠١
															شهد محمد قرنى محمود عبد الواحد		٣٠٢
															شهد هشام محمد اسماعيل		٣٠٣
															شيماء حجاجى سيد رضوان		٣٠٤
															شيماء سمير عبد الفتاح منصور		٣٠٥
															شيماء صلاح ابراهيم عبد الباسط		٣٠٦
															شيماء ممدوح حسن محمد		٣٠٧
															صابرين احمد محمد احمد		٣٠٨
															صفاء على رمضان حسن		٣٠٩
															ضحى احمد عبد الواحد ابو عوف		٣١٠
															ضحى عمرو محمد عبد الحميد		٣١١
															علياء عبد شعبان عبد العال		٣١٢
															فاطمة صابر عامر صالح		٣١٣

لجنة النظم والمراقبة
أ/ عبد الجواد عبد السميع

(١٥)

مدير المدرسة
أ/ مرفت ابراهيم

Payment confirmation

Order reference : G0023201_5848581443299605
Total charge : 36.00 EUR
Beneficiary : Universiteit Gent

Pay with : **VISA**

Card holder's name* :

Card number* :

Expiry date (mm/yyyy)* : ▼ / ▼

Card verification code : _What is this ?_

* Mandatory fields.

Yes, I confirm my order

Payment processed by
ogone
About Ogone | Privacy policy | Security

Elavon
Merchant Services

VeriSign Secured
VERIFY ▸

Cancel

13 References

References

https://alexwebdevelop.com/monitor-script-memory-usage/

https://www.geeksforgeeks.org/how-to-encrypt-and-decrypt-a-php-string/

https://help.dreamhost.com/hc/en-us/articles/214395738-phpMyAdmin-How-to-backup-or-export-a-database-or-table

https://www.tecmint.com/secure-phpmyadmin-centos-ubuntu/