Software requirement specification document for a School System

Mahmoud Hossam 222125, Mahmoud Ibrahim 213907,
Mohamed Magdy 213059, Omar AbdelFattah 212325,
Youssef Sharabas 212091

Supervised by: Dr. Ayman Ezzat

April 25, 2022

1 Introduction

1.1 Purpose of this document

This document describes the software specifications of a School System designed to fully digitise all processes that take place within a given school. The audience of this document is any relevant personnel within a school who require sufficient knowledge of the system.

1.2 Scope of this document

The system is designed and built by 5 Software Engineers for a School. The requirements of this system were specified by the school's management. In addition, the system is built using PHP. It is expected to be delivered within 3 months, while being completely free of cost.

1.3 Overview

This system will be used by Students, Parents, Teachers, The Principle, The Secretary, HR Employees, Job Candidates, The Transportation Manager, Drivers, The Librarian, and Nurses. In short; every employee in a school will be able to use the system which will help facilitate their jobs.

1.4 Business Context

This system is built for a Governmental School in Egypt. Being a Governmental School, the focus of the school is to provide the best learning environment for both students, and their parents. The goal of this system is to lift some of the monotonous load of the school's employees, resulting in saving the school's time and resources, while ensuring that they keep-up with the ever-changing world of technology.

2 General Description

2.1 Product Functions

As previously stated, the system will be used by several positions within the school. It will be used to store and access data, issue requests, track progress, and much more. The exact functionalities of the system will be described in more detail below.

2.2 Similar System Information

This system is designed to be a stand-alone system, with possible modifications and increments to be added down the line.

2.3 User Characteristics

The system requires fundamental and primary understanding of how the system works. A short training period would be sufficient for an efficient use of the system.

2.4 User Problem Statement

The most common problems that might face a user is the system being down; causing progress to be unsaved. Another possible problem is the system being hacked; and thus, proper security measures must be taken to ensure that the system is only accessed by authorised users.

2.5 User Objectives

The requirements of the system as defined by the client:
Student:
- A Student should be able to register for a course
- A Student should be able to access a course's learning material
- A Student should be able to access a course's results
- A Student should be able to rate a course
- A Student should be able to request a course drop
- A Student should be able to get the diploma

Teachers:
- A Teacher should be able to take the students' attendance
- A Teacher should be able to send out warnings for students who skip classes
- A Teacher should be able to provide the students with all learning materials
- A Teacher should be able to provide students with their results

- A Teacher should be able to test students on covered material
- A Teacher should be able to send a monthly review letter to the students' guardians

Parent:

- A Parent should be able to access the student's warnings
- A Parent should be able to access the student's results
- A Parent should be able to access the student's review letter

Principle:

- The Principle should be able to schedule meetings with a teacher
- The Principle should be able to check a teacher's performance
- The Principle should be able to send emails to teachers
- The Principle should be able to view a teacher's schedule
- The Principle should be able to schedule a meeting with a student's guardian
- The Principle should be able to check students' complaints
- The Principle should be able to check whether students paid the tuition fees
- The Principle should be able to check the school's budget
- The Principle should be able to check HR issues
- The Principle should be able to check whether salaries are paid
- The Principle should be able to approve a job posting
- The Principle should be able to interview a candidate
- The Principle should be able to approve the issuing of a diploma
- The Principle should be able to approve a request to purchase new medicine
- The Principle should be able to approve a request to purchase new books

HR:

- An HR employee should be able to post new vacancies
- An HR employee should be able to see everyone who applied for a certain job
- An HR employee should be able to access a candidate's data
- An HR employee should be schedule interviews with a potential candidate
- An HR employee should be able to specify an employee's salary
- An HR employee should be able to access a candidate's letter of recommendation
- An HR employee should be able to hire a driver
- An HR employee should be able to access a driver's information
- An HR employee should be able to update a driver's information
- An HR employee should be able to terminate a driver's contract
- An HR employee should be able to assign teachers to bus lines

Candidate:

- A candidate should be able to submit a letter of recommendation

Secretary:

- The Secretary should be able to add new students to the database
- The Secretary should be able to access a student's information
- The Secretary should be able to update a student's information
- The Secretary should be able to delete a student from the database
- The Secretary should be able to add a new course

- The Secretary should be able to access the information of current courses
- The Secretary should be able to update the information of current courses
- The Secretary should be able to delete a course
- The Secretary should be able to add a new course to a student
- The Secretary should be able to access a student's current courses
- The Secretary should be able to update a student's current courses
- The Secretary should be able to delete a course from a student

Transportation Manager:
- The Transportation Manager should be able to assign drivers to certain lines
- The Transportation Manager should be able to check a driver's performance
- The Transportation Manager should be able to add a student to the already-paid list

Bus Driver:
- A Bus Driver should be able to pick up students
- A Bus Driver should be able to request line transfer

School Nurse:
- The School Nurse should be able to report incidents
- The School Nurse should be able to check current medicine inventory
- The School Nurse should be able to request new medicine inventory

Librarian: - The Librarian should be able to check whether certain books are available
- The Librarian should be able to request a book purchase
- The Librarian should be able to buy a certain book
- The Librarian should be able to lend a book to a student
- The Librarian should be able to return a book

2.6 General Constraints

The design team are to ensure that the system is not down for more that 10 minutes, in order to ensure a smooth user-experience.

2.7 User Interfaces

The system is built for the web, and so it can be accessed by the users from several terminals such as mobile phones, laptops, etc. The system is not defined by a certain area i.e. school grounds, and so it can be accessed from different locations.

3 Functional Requirements

This section lists the functional requirements in ranked order. Functional requirements describes the possible effects of a software system, in other words, what the system must accomplish.

Table 1:

Function Name	Schedule meeting with teacher
Description	This function allows the principal to schedule a meeting between him and teacher
Critically	5/10
Technical issues	teacher could be not available at the meting time
Cost and schedule	N/A
Risks	Meetings file could be closed,so the meetings file should be opened
Dependencies with other requirements	Principal should be logged in
input	this function's input is the date and time of meeting and teacher object
output	this function's output is the scheduled meeting between principal and teacher
Pre-condition	There was no meeting scheduled between principal and teacher
Post-condition	A meeting is scheduled between the principal and a teacher

Table 2:

Function Name	Schedule meeting with a student guardian
Description	this function allows the principal to schedule a meeting between him and student's parents
Critically	5/10
Technical issues	There must be free capacity in hard disk to store meeting details
Cost and schedule	N/A
Risks	Student's parent contact information could be unrecorded,so every student must put his parent's contact information
Dependencies with other requirements	Principal should be logged in
input	this function's input is the date and time of meeting and parent object
output	this function's output is the scheduled meeting between principal and student's parents
Pre-condition	There was no meeting scheduled between principal and parent
Post-condition	A meeting is scheduled between the principal and parent

Table 3:

Function Name	Send emails to teacher
Description	This function allows the principal to send an email to teachers
Critically	5/10
Technical issues	Poor internet connection
Cost and schedule	N/A
Risks	Teacher's email could be not recorded, so all teachers should put their emails
Dependencies with other requirements	Principal should be logged in
input	this function's input is the teacher's email and the email itself
output	this function's output is the email sent to the teacher
Pre-condition	There was no email sent from principal to teacher
Post-condition	An email is sent from principal to teacher

Table 4:

Function Name	Check teacher's performance
Description	This function allows the principal to check teacher's performance
Critically	5/10
Technical issues	Poor internet connection
Cost and schedule	N/A
Risks	Teacher's performance might not be recorded, so all teacher's performances should be recorded
Dependencies with other requirements	Principal should be logged in
input	this function's input is a teacher object
output	this function's output is the teacher's performance details
Pre-condition	The principle didn't know how teacher is performing
Post-condition	The principal knows how the teacher is performing

Table 5:

Function Name	View teacher's schedule
Description	This function allows the principal to check teacher's schedule
Critically	4/10
Technical issues	Poor internet connection
Cost and schedule	N/A
Risks	Teacher could forget updating his schedule, so all teachers must update their schedule every term
Dependencies with other requirements	Principal should be logged in
input	this function's input is a teacher object
output	this function's output is the teacher's schedule
Pre-condition	The principle didn't know the teacher's schedule
Post-condition	The principal knows the teacher's schedule

Table 6:

Function Name	Check invoices that need signing
Description	This function allows the principal to check unsigned invoices
Critically	6/10
Technical issues	Poor internet connection
Cost and schedule	N/A
Risks	The invoice information could be missing, all invoices detail should be recorded
Dependencies with other requirements	Principal should be logged in
input	this function's input is an invoice object
output	this function's output is displaying the invoice details on screen
Pre-condition	The principle didn't know about unsigned invoices
Post-condition	The principal knows about unsigned invoices

Table 7:

Function Name	Check school budget
Description	This function allows the principal to check the school budget
Critically	5/10
Technical issues	Poor internet connection
Cost and schedule	N/A
Risks	The budget file could be not updated, the budget file should be updated daily
Dependencies with other requirements	Principal should be logged in
input	this function's input is the school budget
output	this function's output is displaying the budget's details on screen
Pre-condition	The principal didn't know the budget
Post-condition	The principal knows the budget

Table 8:

Function Name	Approve employees request
Description	This function allows the principal to check on and allow employees requests
Critically	5/10
Technical issues	Server breaking down
Cost and schedule	N/A
Risks	The request could be not recorded yet, so the principal should check the requests daily
Dependencies with other requirements	Employee should make a request
input	this function's input is the employee's request
output	this function's output is displaying the request details on screen and whether the request is accepted or rejected
Pre-condition	The principle didn't know about employee's requests
Post-condition	The principle knows about employee's requests

Table 9:

Function Name	Check student complain
Description	This function allows the principal to check on student's complains
Critically	5/10
Technical issues	Server breaking down
Cost and schedule	N/A
Risks	Complain could not be saved in complains file, every complain should be saved in the complains file
Dependencies with other requirements	Student should file a complain
input	this function's input is the student and complain object
output	this function's output is displaying the complain details on screen
Pre-condition	The principle didn't know about student's complains
Post-condition	The principle knows about student complains

Table 10:

Function Name	Check HR issues
Description	This function allows the principal to check HR issues
Critically	4/10
Technical issues	Server breaking down
Cost and schedule	N/A
Risks	HR could forget to record issues, HR should record issues and send the to the principal immediately
Dependencies with other requirements	HR should record issues
input	this function's input is the HR object
output	this function's output is that it displays HR's issues on screen
Pre-condition	The principle didn't know about HR's issues
Post-condition	The principle knows about HR's issues

Table 11:

Function Name	Check whether salaries are paid
Description	This function allows the principal to check if salaries are paid
Critically	6/10
Technical issues	Server breaking down
Cost and schedule	N/A
Risks	Salaries file could be empty, all salaries should be recorded in the salaries file
Dependencies with other requirements	Principal should be logged in
input	this function's input is the salary object
output	this function's output is displaying the salaries status
Pre-condition	The principle didn't know if salaries are paid
Post-condition	The principle knows if salaries are paid

Table 12:

Function Name	Check whether student paid fees
Description	This function allows the principal to check if student paid fees
Critically	6/10
Technical issues	Poor internet connection
Cost and schedule	N/A
Risks	It could be unrecorded whether the student paid fees or not, if student paid fees it must be recorded immediately
Dependencies with other requirements	principal should be logged in
input	this function's input is student object
output	this function's output is displaying the student's fees status
Pre-condition	The principle didn't if student paid fees or not
Post-condition	The principle knows if student paid fees or not

Table 13:

Function Name	Check book
Description	This function allows the librarian to check where a book is
Critically	5/10
Technical issues	Book could be not recorded
Cost and schedule	N/A
Risks	Librarian could forget book's id, all books' id must be recorded
Dependencies with other requirements	Book's info should be recorded
input	this function's input is book object
output	this function's output is displaying the book's location in library
Pre-condition	The librarian didn't know where the book is
Post-condition	The librarian knows where the book is

Table 14:

Function Name	Lend book
Description	This function allows the librarian to lend a book to student
Critically	5/10
Technical issues	Server breaking down
Cost and schedule	N/A
Risks	Librarian could forget book's id, all books' id must be recorded
Dependencies with other requirements	Librarian should be logged in
input	this function's input is student and book objects
output	this function's output is that book's status changed and recorded who it is lend to
Pre-condition	The librarian couldn't lend book to student
Post-condition	The librarian could lend book to student

Table 15:

Function Name	request new book
Description	This function allows the librarian to request new book. this request could be seen by principal
Critically	5/10
Technical issues	Server breaking down
Cost and schedule	N/A
Risks	Librarian could forget what books are needed, librarian should record all books that need to be bought
Dependencies with other requirements	Librarian should be logged in
input	this function's input is the book's name
output	this function's output is that a request is made to buy new book
Pre-condition	There were books that needed to be bought
Post-condition	New books were bought for the library

Table 16:

Function Name	record students who pay buses
Description	This function allows transportation manager to record students who pay buses
critically	7/10
Technical issues	server breaking down
cost and schedule	N/A
Risks and prevention	Transportation manager could miss record a student;double checking the entries
Dependencies with other requirements	Transportation manager should be logged in
input	this function's input is student object
output	this function's output is that student is added to the list of students who paid bus fees
Pre-Condition	There were no records of students who paid for their buses
Post-Condition	Student's bus payments were recorded

Table 17:

Function Name	check drivers' reports
Description	This function allows transportation manager to check drivers' reports
critically	5/10
Technical issues	Poor internet connection
cost and schedule	N/A
Risks and prevention	Driver's reports may not be recorded; request driver's report
Dependencies with other requirements	Transportation manager should be logged in
input	this function's input is report object
output	this function's output is that it displays report on screen
Pre-Condition	The transportation manager didn't know the driver's reports
Post-Condition	The transportation manager knows the driver's reports

Table 18:

Function Name	check drivers' requests
Description	This function allows transportation manager to check drivers' requests
critically	4/10
Technical issues	Server breaking down
cost and schedule	N/A
Risks and prevention	Driver's requests could not be saved in requests file; have more file storage
Dependencies with other requirements	Transportation manager should be logged in
input	this function's input is request object
output	this function's output is that it displays the request on screen
Pre-Condition	The transportation manager didn't know the driver's requests
Post-Condition	The transportation manager knows the driver's requests

Table 19:

Function Name	check drivers' performance
Description	This function allows transportation manager to check drivers' performance
critically	6/10
Technical issues	Poor internet connection
cost and schedule	N/A
Risks and prevention	Driver's performance could not be recorded in requests file; assign a survey
Dependencies with other requirements	Transportation manager should be logged in
input	this function's input is driver object
output	this function's output is that it displays the driver's performance details
Pre-Condition	The transportation manager didn't know the driver's performance
Post-Condition	The transportation manager knows the driver's performance

Table 20:

Function Name	assign drivers
Description	This function allows transportation manager to assign drivers
critically	4/10
Technical issues	Poor internet connection
cost and schedule	N/A
Risks and prevention	Driver trip could not be recorded and request driver's trip
Dependencies with other requirements	Transportation manager should be logged in
input	this function's input are line object and driver object
output	this function's output is driver assigned to a line
Pre-Condition	The transportation manager didn't assign the driver to a line
Post-Condition	The transportation manager assign the driver to a line

Table 21:

Function Name	pick up
Description	This function allows driver to pick up students who needs to be picked up
critically	7/10
Technical issues	Poor internet connection
cost and schedule	N/A
Risks and prevention	Driver trip could not be recorded and request driver's trip
Dependencies with other requirements	student address should be recorded
input	this function's input is student object
output	this function's output is that it displays the students who need to be picked up
Pre-Condition	The driver didn't know the the students who needs to be picked up
Post-Condition	The driver knows the the students who needs to be picked up

Table 22:

Function Name	report any accidents
Description	This function allows driver to report any accidents
critically	5/10
Technical issues	Poor internet connection
cost and schedule	N/A
Risks and prevention	Driver forgets to report accidents; record accidents when it happens
Dependencies with other requirements	student address should be recorded
input	this function's input is accident object
output	this function's output is that a new accident is recorded
Pre-Condition	The driver didn't report any accidents
Post-Condition	The driver did report all his accidents

Table 23:

Function Name	report any mechanical issues
Description	This function allows driver to report any mechanical issues
critically	6/10
Technical issues	Poor internet connection
cost and schedule	N/A
Risks and prevention	Driver forgets to report mechanical issues; record mechanical issues as soon as it occurs
Dependencies with other requirements	driver should be logged in
input	this function's input is mechanical issues object
output	this function's output is that a new mechanical issue is recorded
Pre-Condition	The driver didn't record any mechanical issues
Post-Condition	The driver recorded his mechanical issues

Table 24:

Function Name	check inventory
Description	This function allows school nurse to check the medications inventory
critically	7/10
Technical issues	server breaking down
cost and schedule	N/A
Risks and prevention	medications could be expired; check its date
Dependencies with other requirements	school nurse should be logged in
input	this function's input is medicine object
output	this function's output is that it displays the inventory of this medicine
Pre-Condition	The school nurse didn't know the quantity of medicine left
Post-Condition	The school nurse does know the quantity of medicine left

Table 25:

Function Name	request inventory
Description	This function allows school nurse to request new medicine inventory
critically	7/10
Technical issues	server breaking down
cost and schedule	N/A
Risks and prevention	request could not be saved in request file; need more storage
Dependencies with other requirements	school nurse should be logged in
input	this function's input is medicine object
output	this function's output is that a request for medicine is made
Pre-Condition	The school nurse didn't have enough medicine in inventory
Post-Condition	The school nurse has enough medicine in inventory

Table 26:

Function Name	report incidents
Description	This function allows school nurse to report any incidents to the principal
critically	6/10
Technical issues	Poor internet connection
cost and schedule	N/A
Risks and prevention	request could not be saved in request file; need more storage
Dependencies with other requirements	school nurse should be logged in
input	this function's input is principle object and parent object
output	this function's output is that a report is made
Pre-Condition	The principle doesn't know any incidents
Post-Condition	The principle knows all the incidents

Table 27:

Function Name	access medical records
Description	This function allows school nurse access students medical records
critically	6/10
Technical issues	Poor internet connection
cost and schedule	N/A
Risks and prevention	outdated students medical records; student need to update his medical record
Dependencies with other requirements	should have students medical records
input	this function's input is medical record object
output	this function's output is that a report is made
Pre-Condition	The school nurse couldn't access students medical records
Post-Condition	The school nurse could access students medical records

Table 28:

Function Name	Post New vacancies
Description	This function Puts new Jobs Vacancies for people to apply for
Critically	5/10
Technical issues	Some jobs won't be meet the qualifications of some people
Cost and schedule	N/A
Risks	Vacancies file were not open
Risk prevention	Vacancies file should be open
Dependencies with other requirements	HR should be logged in
input	this function's input is job object
output	this function's output is that it displays job vacancies
Pre-condition	Doesn't have the job requirements
Post-condition	Test for job requirements

Table 29:

Function Name	See who applied for certain jobs
Description	HR would be able to see who applied for a job
Critically	5/10
Technical issues	There must be free capacity in hard disk to store Candidates data
Cost and schedule	N/A
Risks	Candidate's contact information weren't available
Risk prevention	Candidate's contact information should be available
Dependencies with other requirements	HR should be logged in
input	this function's input is candidate object
output	this function's output is that it displays who applied for a job
Pre-condition	No data stored about the Candidate
Post-condition	data stored about the Candidates

Table 30:

Function Name	Access candidate's data
Description	This function access the candidates data by the HR
Critically	5/10
Technical issues	There must be free capacity in hard disk to store candidate's data
Cost and schedule	N/A
Risks	Candidate's contact information weren't available
Risk prevention	Candidate's contact information should be available
Dependencies with other requirements	HR should be logged in
input	this function's input is candidate object
output	this function's output is that it displays candidate information
Pre-condition	There was no stored about candidate
Post-condition	Candidate data is stored

Table 31:

Function Name	Schedule interviews
Description	This function allows the HR to schedule an interview
Critically	5/10
Technical issues	candidate not available for the interview
Cost and schedule	N/A
Risks	Candidate might be busy
Dependencies with other requirements	HR should be logged in
input	this function's input is interview object
output	this function's output is that it records interview information
Pre-condition	There were no interview scheduled for the candidate
Post-condition	Interview scheduled for the candidate

Table 32:

Function Name	Specify an employee salary
Description	This function allows the HR to specify the employee's salary
Critically	5/10
Technical issues	salary folder isn't available
Cost and schedule	N/A
Risks	HR changing salary to a wrong one
Risk prevention	HR be more accurate and change the salary to the correct worker
Dependencies with other requirements	HR should be logged in
input	this function's input is salary object
output	this function's output is that it update or insert employee's salary
Pre-condition	The HR didn't know the employee's salary
Post-condition	The HR knows the employee's salary

Table 33:

Function Name	Access complains provided by the working staff
Description	This function allows the HR to check complains made by the working staff
Critically	5/10
Technical issues	Poor internet connection
Cost and schedule	N/A
Risks	The complain contains swear words or empty
Risk prevention	filter the complain form from swear words
Dependencies with other requirements	HR should be logged in
input	this function's input is complain object
output	this function's output is that it access the complain form
Pre-condition	The HR didn't know about the complain
Post-condition	The HR knows about the complain

Table 34:

Function Name	Access learning material
Description	This function allows the students to access there learning material
Critically	7/10
Technical issues	Poor internet connection
Cost and schedule	N/A
Risks	student isn't enrolled in the course on the system
Risk prevention	enroll the student in the course on the system
Dependencies with other requirements	student should be logged in
input	this function's input is student and course object
output	this function's output is that it access the learning material
Pre-condition	The student didn't know about the learning materials
Post-condition	The student knows about the learning materials

Table 35:

Function Name	Access results
Description	This function allows the students to see there results
Critically	8/10
Technical issues	Poor internet connection
Cost and schedule	N/A
Risks	results are given to the wrong student
Risk prevention	give results to the right student
Dependencies with other requirements	student should be logged in
input	this function's input is student object and course object
output	this function's output is that it access the results
Pre-condition	The student didn't know about the results
Post-condition	The student knows about the results

Table 36:

Function Name	Register course
Description	This function allows the students to register there courses
Critically	7/10
Technical issues	Poor internet connection
Cost and schedule	N/A
Risks	courses aren't available for students who are allowed to take it
Risk prevention	make courses available for students who are allowed to take it
Dependencies with other requirements	student should be logged in
input	this function's input is courses object and student object
output	this function's output is that it register the course
Pre-condition	The student didn't which courses are available for them
Post-condition	The student knows which courses are available for them

Table 37:

Function Name	Get diploma
Description	This function allows the students to get the diploma
Critically	8/10
Technical issues	Poor internet connection
Cost and schedule	N/A
Risks	student didn't finish the required courses for the diploma
Risk prevention	make sure the student finish the required courses
Dependencies with other requirements	student should be logged in
input	this function's input are student object and diploma object
output	this function's output is that it gets the diploma information
Pre-condition	The student didn't they achieved the diploma
Post-condition	The student knows they achieved the diploma

Table 38:

Function Name	take attendance
Description	This function allows the teacher to record the attendance
Critically	5/10
Technical issues	Poor internet connection
Cost and schedule	N/A
Risks	students share the attendance code
Risk prevention	use different method to prevent the share of the code
Dependencies with other requirements	teacher should be logged in
input	this function's input is attendance object
output	this function's output is that it record the attendance
Pre-condition	teacher didn't know who attended
Post-condition	The teacher knows who attended

Table 39:

Function Name	send warnings
Description	This function allows the teacher to send warnings to parents of the students
Critically	5/10
Technical issues	Poor internet connection
Cost and schedule	N/A
Risks	parents didn't reply
Risk prevention	resend the warning
Dependencies with other requirements	teacher should be logged in
input	this function's input is warnings object
output	this function's output is that it sends warnings to parents
Pre-condition	The teacher didn't send the warning
Post-condition	The teacher sent the warning

Table 40:

Function Name	provide learning materials
Description	This function allows the teacher to upload the learning materials
Critically	6/10
Technical issues	Poor internet connection
Cost and schedule	N/A
Risks	wrong material were uploaded
Risk prevention	allow to edit the uploaded material
Dependencies with other requirements	teacher should be logged in
input	this function's input is learning material object
output	this function's output is that it provide the learning material
Pre-condition	The teacher didn't upload the material
Post-condition	The teacher uploaded the material

Table 41:

Function Name	provide results
Description	This function allows the teacher to give the results of the students
Critically	7/10
Technical issues	Poor internet connection
Cost and schedule	N/A
Risks	wrong results given to the student
Risk prevention	provide the correct results to the student
Dependencies with other requirements	teacher should be logged in
input	this function's input is results object
output	this function's output is that it provide results information
Pre-condition	The teacher didn't know the result
Post-condition	The teacher knows the result

Table 42:

Function Name	tests
Description	This function allows the teacher to make tests for the students
Critically	7/10
Technical issues	Poor internet connection
Cost and schedule	N/A
Risks	wrong test were provided
Risk prevention	make a new test
Dependencies with other requirements	teacher should be logged in
input	this function's input is tests object
output	this function's output is that it provide tests to students
Pre-condition	The teacher didn't know who will take the test
Post-condition	The teacher knows who take the test

Table 43:

Function Name	send parents review letter
Description	This function allows the teacher to send review letter to the parents
Critically	5/10
Technical issues	Poor internet connection
Cost and schedule	N/A
Risks	parents didn't see the letter
Risk prevention	resend the letter
Dependencies with other requirements	teacher should be logged in
input	this function's input is parents review letter object
output	this function's output is that it send parents review letter
Pre-condition	The teacher didn't know the letter
Post-condition	The teacher knows the letter

Table 44:

Function Name	access warning
Description	This function allows parent to access warnings sent by teacher
Critically	4/10
Technical issues	Poor internet connection
Cost and schedule	N/A
Risks	Parent didn't check his/her mail, so teacher should call parent
Dependencies with other requirements	Parent should be logged in
input	this function's input is parent object and student object
output	this function's output is that it access the warning information
Pre-condition	Teacher couldn't warn student's parent
Post-condition	It's possible for parent to access warnings sent by teacher

Table 45:

Function Name	access results
Description	This function allows parent to access results sent by teacher
Critically	5/10
Technical issues	Poor internet connection/ Server breaking down
Cost and schedule	N/A
Risks	Parent didn't check his/her mail, so teacher should call parent
Dependencies with other requirements	Parent should be logged in
input	this function's input is parent object and student object
output	this function's output is that it access the results
Pre-condition	Parent couldn't get his/her student's results without going to school
Post-condition	It's possible for parent to get his/her student's results by checking his/her mail

Table 46:

Function Name	access review letter
Description	This function allows parent to access review letters sent by teacher
Critically	4/10
Technical issues	Poor internet connection
Cost and schedule	N/A
Risks	Parent didn't check his/her mail
Dependencies with other requirements	Parent should be logged in
input	this function's input is parent object and student object
output	this function's output is that it access the review letters information
Pre-condition	Parent couldn't know how his/her student is performing without going to school
Post-condition	It's possible for parent to know his/her student's performance by checking his/her mail

Table 47:

Function Name	record new visitors
Description	This function allows receptionist to record visitors' data and their reason for visiting
Critically	6/10
Technical issues	Server breaking down
Cost and schedule	N/A
Risks	Visitors' data could not be saved, so receptionist should check if data were saved
Dependencies with other requirements	Visitor should sign in before entering
input	this function's input is visitors object
output	this function's output is that it record new visitors information
Pre-condition	There is no plan for security risk management
Post-condition	It's simpler to identify and prioritise actions school can take to deter vandals and thieves

Table 48:

Function Name	identify parent
Description	This function allows receptionist to identify parent by access his/her data
Critically	6/10
Technical issues	Server breaking down
Cost and schedule	N/A
Risks	Data could be mistakenly deleted, so data should be stored on two or more servers
Dependencies with other requirements	parents' data should be recorded
input	this function's input is parent object
output	this function's output is that it identify the parent of the student
Pre-condition	There is no plan for security risk management
Post-condition	It's simpler to identify and prioritise actions school can take to deter vandals and thieves

Table 49:

Function Name	check new inventory
Description	This function allows cafeteria employee to identify every inventory item and its associated information
Critically	5/10
Technical issues	Server breaking down
Cost and schedule	N/A
Risks	Data for new inventory aren't available, so cafeteria employee should check if data for new inventory are provided or not
Dependencies with other requirements	A database for all inventory should be provided
input	this function's input is inventory object
output	this function's output is that it checks new inventory
Pre-condition	There is no plan for inventory management
Post-condition	It's simpler to spot any problems right away

Table 50:

Function Name	check current inventory
Description	This function allows cafeteria employee to identify every inventory item and its associated information
Critically	5/10
Technical issues	Server breaking down
Cost and schedule	N/A
Risks	Data for current inventory aren't available, so cafeteria employee should ensure that data for current inventory are provided
Dependencies with other requirements	A database for all inventory should be provided
input	this function's input is inventory object
output	this function's output is that it checks the current inventory
Pre-condition	There is no plan for inventory management
Post-condition	It's simpler to spot any problems right away

Table 51:

Function Name	update inventory
Description	This function allows cafeteria employee to store inventory records
Critically	5/10
Technical issues	Server breaking down
Cost and schedule	N/A
Risks	Data for new inventory aren't available, so cafeteria employee should ensure that data for new inventory are provided
Dependencies with other requirements	A database for all inventory should be provided
input	this function's input is inventory object
output	this function's output is that it updates the inventory
Pre-condition	There is no plan for inventory management
Post-condition	It's simpler to spot any problems right away

Table 52:

Function Name	check product pricing
Description	This function allows cafeteria employee to check every product price
Critically	5/10
Technical issues	Server breaking down
Cost and schedule	N/A
Risks	Data for inventory products aren't available, so cafeteria employee should ensure that data for every product are provided
Dependencies with other requirements	A database for all inventory should be provided
input	this function's input is product object
output	this function's output is that it checks the product pricing
Pre-condition	Relying on memory or a visit to the warehouse to check product price
Post-condition	It's easier to check price of every product

Table 53:

Function Name	request inventory
Description	This function allows cafeteria employee to request additional inventory of a particular item
Critically	5/10
Technical issues	Poor internet connection
Cost and schedule	N/A
Risks	Employee's request could be not recorded, so cafeteria employee should contact requester as soon as possible for confirmation
Dependencies with other requirements	A database for all inventory should be provided
input	this function's input is inventory object
output	this function's output is that it requests the inventory
Pre-condition	Losing sales during stock out
Post-condition	Employee uses inventory data to fulfill the requested inventory

Table 54:

Function Name	report profit
Description	This function allows cafeteria employee to know the performance of business over a specified period of time
Critically	4/10
Technical issues	Poor internet connection
Cost and schedule	N/A
Risks	Sales could be not be recorded, so cafeteria employee should update sales daily
Dependencies with other requirements	Employee should update sales daily
input	this function's input is profit object
output	this function's output is that it report the profit made
Pre-condition	Employee couldn't know the performance of his/her business
Post-condition	Employee know the performance of his/her business

Table 55:

Function Name	add course to the student
Description	This function allows the principal to add a new course to the student
Critically	6/10
Technical issues	Poor internet connection
Cost and schedule	N/A
Risks	students file and course file could be closed, so the students file should be opened when adding a new course to the student
Dependencies with other requirements	principal should be logged in
input	this function's input is student and course object
output	this function's output is that it adds new course to the students
Pre-condition	student was not enrolled to the course
Post-condition	The student is enrolled to the course

Table 56:

Function Name	add new student
Description	This function allows the principal to add a new student
Critically	7/10
Technical issues	Poor internet connection
Cost and schedule	N/A
Risks	students file could be closed, so the students file should be opened when adding a new student
Dependencies with other requirements	principal should be logged in
input	this function's input is student object
output	this function's output is that it adds new student
Pre-condition	student was not added to the system
Post-condition	The student is added to the system

Table 57:

Function Name	access students information
Description	This function allows the principal to see student's information
Critically	5/10
Technical issues	Poor internet connection
Cost and schedule	N/A
Risks	student information could be unrecorded, so every student's information should be stored in the students file
Dependencies with other requirements	principal should be logged in
input	this function's input is student object
output	this function's output is that it access the information of the student
Pre-condition	The principal didn't know the student's information
Post-condition	The principal knows the student's information

Table 58:

Function Name	update student information
Description	This function allows the principal to update student's information
Critically	6/10
Technical issues	Poor internet connection
Cost and schedule	N/A
Risks	Student file could be closed, therefore the student file should be opened when updating student's information
Dependencies with other requirements	principal should be logged in
input	this function's input is student object
output	this function's output is that it updates the students information
Pre-condition	Student information was not updated in the system
Post-condition	Student information was updated in the system

Table 59:

Function Name	delete student
Description	This function allows the principal to delete a student
Critically	5/10
Technical issues	Poor internet connection
Cost and schedule	N/A
Risks	Student file could not be located, therefore the student file should be located to be eradicated
Dependencies with other requirements	principal should be logged in
input	this function's input is student object
output	this function's output is that it deletes the student
Pre-condition	Student information was not deleted from the system
Post-condition	Student information was deleted from the system

Table 60:

Function Name	Access student's courses
Description	This function allows the principal to Access student's courses
Critically	6/10
Technical issues	Poor internet connection
Cost and schedule	N/A
Risks	students file could be closed, so the students file should be opened when accessing student's current courses
Dependencies with other requirements	principal should be logged in
input	this function's input is student object
output	this function's output is that it access the student's current courses
Pre-condition	principal was not accessing student's current courses
Post-condition	The principal is accessing student's current courses

Table 61:

Function Name	Update student's courses
Description	This function allows the secretary to update student's courses
Critically	6/10
Technical issues	Poor internet connection
Cost and schedule	N/A
Risks	Students file could be closed, so the students file should be opened when updating student's courses
Dependencies with other requirements	secretary should be logged in
input	this function's input is student and courses object
output	this function's output is that it updates the student's courses
Pre-condition	secretary was not accessing student's current courses
Post-condition	The secretary is updating student's courses

Table 62:

Function Name	Delete course from student
Description	This function allows the secretary to delete a course from student's courses
Critically	6/10
Technical issues	Poor internet connection
Cost and schedule	N/A
Risks	Students file could be closed, so the students file should be opened when deleting a course from student's courses
Dependencies with other requirements	secretary should be logged in
input	this function's input is student and courses object
output	this function's output is that it deletes the student's course and delete the student from the course
Pre-condition	secretary was not able to delete a course from student's courses
Post-condition	The secretary is able to delete a course from student's courses

Function Name	login
Description	This function allows the user to login into the system
Critically	9/10
Technical issues	Poor internet connection
Cost and schedule	N/A
Risks	user type file could be closed, so the user file should be opened when logging into the system
Dependencies with other requirements	admin should be logged in
input	this function's input is student and user object
output	this function's output is that it will either allow the user to access the system or deny if it was invalid user
Pre-condition	user was not able to login into the system
Post-condition	The user is able to login into the system

Function Name	rate course
Description	This function allows the student to rate a course from student's courses
Critically	4/10
Technical issues	Poor internet connection
Cost and schedule	N/A
Risks	Students file could be closed, so the students file should be opened when deleting a course from student's courses
Dependencies with other requirements	Principal should be logged in
input	this function's input is student and courses object
output	this function's output is that it rate the course
Pre-condition	student was not able to rate a course from student's courses
Post-condition	The student is able to rate a course from student's courses

Function Name	request course drop
Description	This function allows the student to course a course from student's courses
Critically	7/10
Technical issues	Poor internet connection
Cost and schedule	N/A
Risks	Students file could be closed, so the students file should be opened when deleting a course from student's courses
Dependencies with other requirements	Principal should be logged in
input	this function's input is student and courses object
output	this function's output is that it drops the student's course and delete the student from the course
Pre-condition	student was not able to drop a course from student's courses
Post-condition	The student is able to drop a course from student's courses

Function Name	access course
Description	This function allows the student to course a course from student's courses
Critically	7/10
Technical issues	Poor internet connection
Cost and schedule	N/A
Risks	Students file could be closed, so the students file should be opened when accessing a course from student's courses
Dependencies with other requirements	admin should be logged in
input	this function's input is student's courses object
output	this function's output is that it access the student's course
Pre-condition	student was not able to access a course from student's courses
Post-condition	The student is able to access a course from student's courses

Function Name	submit letter
Description	This function allows the candidate to submit a letter
Critically	6/10
Technical issues	Poor internet connection
Cost and schedule	N/A
Risks	candidate file could be closed, so the candidate file should be opened when submitting a letter
Dependencies with other requirements	HR should be logged in
input	this function's input is letter object
output	this function's output is that it submit the candidate's letter
Pre-condition	candidate was not able to submit a letter
Post-condition	The candidate is able to submit a letter

Function Name	request transfer
Description	This function allows the driver to transfer a line
Critically	6/10
Technical issues	Poor internet connection
Cost and schedule	N/A
Risks	driver file could be closed, so the driver file should be opened when transferring a course from student's courses
Dependencies with other requirements	transportation manager should be logged in
input	this function's input is line object
output	this function's output is that it transfers the drivers line
Pre-condition	driver was not able to request a transfer to the line
Post-condition	The driver is able to request a transfer to the line

4 Other non-functional attributes

4.1 Security

As mentioned above, the system must be secure in order to protect the client's data to ensure that no compromises face the client.

4.2 Reliability

As mentioned above, the system should also be reliable i.e. it should not be down for more than 10 minutes

4.3 Maintainability

The system should be designed in such a way that the system is maintainable, making future increments accessible and easy to implement.

5 Preliminary Object-Oriented Domain Analysis

The following Class Diagram maps the system out:

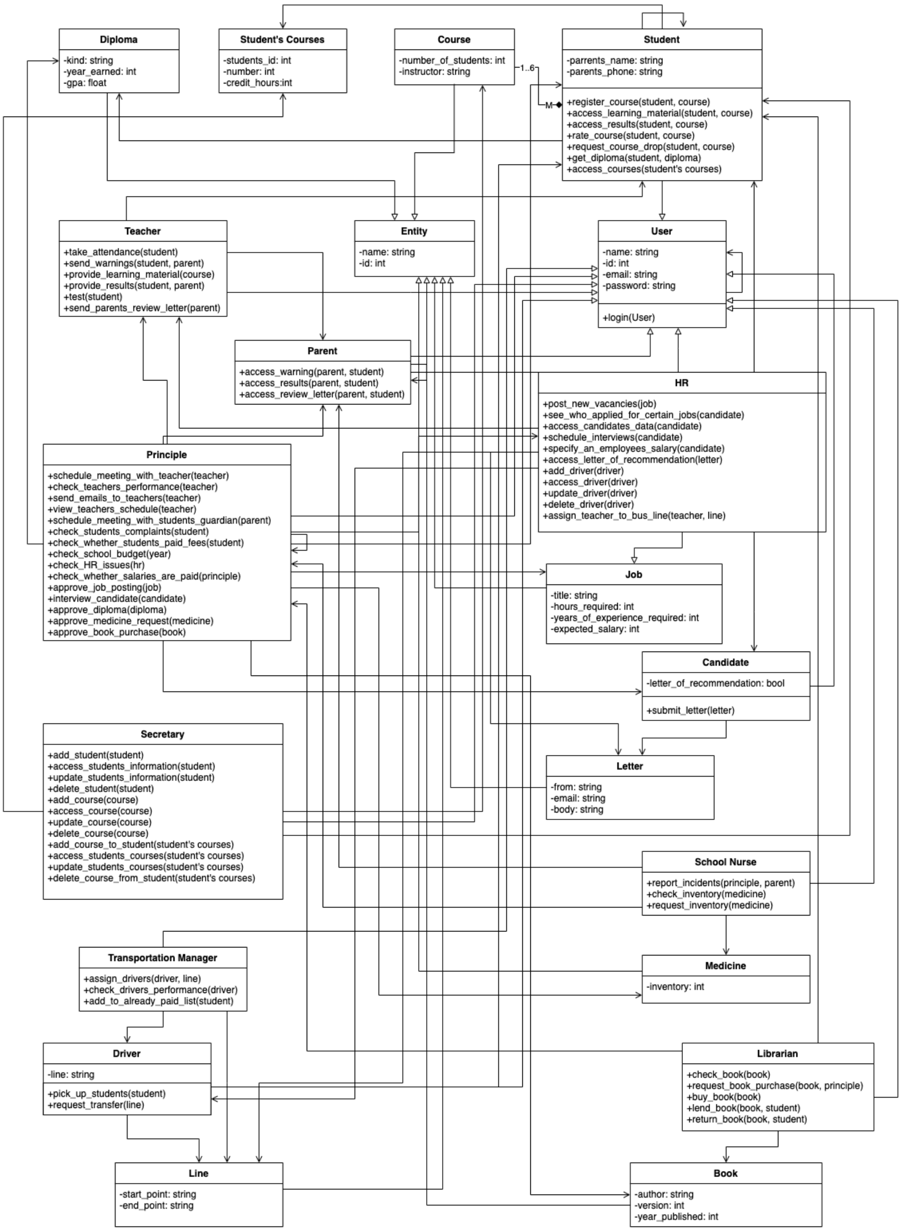

Diploma
-kind: string
-year_earned: int
-gpa: float
Student's Courses
-students_id: int
-number: int
-credit_hours:int
Course
-number_of_students: int
-instructor: string
1..6
M
Student
-parrents_name: string
-parents_phone: string
+register_course(student, course)
+access_learning_material(student, course)
+access_results(student, course)
+rate_course(student, course)
+request_course_drop(student, course)
+get_diploma(student, diploma)
+access_courses(student's courses)
Teacher
+take_attendance(student)
+send_warnings(student, parent)
+provide_learning_material(course)
+provide_results(student, parent)
+test(student)
+send_parents_review_letter(parent)
Entity
-name: string
-id: int
User
-name: string
-id: int
-email: string
-password: string
+login(User)
Parent
+access_warning(parent, student)
+access_results(parent, student)
+access_review_letter(parent, student)
HR
+post_new_vacancies(job)
+see_who_applied_for_certain_jobs(candidate)
+access_candidates_data(candidate)
+schedule_interviews(candidate)
+specify_an_employees_salary(candidate)
+access_letter_of_recommendation(letter)
+add_driver(driver)
+access_driver(driver)
+update_driver(driver)
+delete_driver(driver)
+assign_teacher_to_bus_line(teacher, line)
Principle
+schedule_meeting_with_teacher(teacher)
+check_teachers_performance(teacher)
+send_emails_to_teachers(teacher)
+view_teachers_schedule(teacher)
+schedule_meeting_with_students_guardian(parent)
+check_students_complaints(student)
+check_whether_students_paid_fees(student)
+check_school_budget(year)
+check_HR_issues(hr)
+check_whether_salaries_are_paid(principle)
+approve_job_posting(job)
+interview_candidate(candidate)
+approve_diploma(diploma)
+approve_medicine_request(medicine)
+approve_book_purchase(book)
Job
-title: string
-hours_required: int
-years_of_experience_required: int
-expected_salary: int
Candidate
-letter_of_recommendation: bool
+submit_letter(letter)
Secretary
+add_student(student)
+access_students_information(student)
+update_students_information(student)
+delete_student(student)
+add_course(course)
+access_course(course)
+update_course(course)
+delete_course(course)
+add_course_to_student(student's courses)
+access_students_courses(student's courses)
+update_students_courses(student's courses)
+delete_course_from_student(student's courses)
Letter
-from: string
-email: string
-body: string
School Nurse
+report_incidents(principle, parent)
+check_inventory(medicine)
+request_inventory(medicine)
Transportation Manager
+assign_drivers(driver, line)
+check_drivers_performance(driver)
+add_to_already_paid_list(student)
Medicine
-inventory: int
Driver
-line: string
+pick_up_students(student)
+request_transfer(line)
Librarian
+check_book(book)
+request_book_purchase(book, principle)
+buy_book(book)
+lend_book(book, student)
+return_book(book, student)
Line
-start_point: string
-end_point: string
Book
-author: string
-version: int
-year_published: int

Figure 1: System's Class Diagram

5.1 Class descriptions

This section presents a more detailed description of each class identified during the Object-Oriented Domain Analysis.

5.1.1 User

Abstract Class

5.1.2 List of Superclasses:

N/A

5.1.3 List of Subclasses:

Student, Parent, Teacher, Principle, Secretary, HR, Candidate, Transportation Manager, Driver, School Nurse, and Librarian

5.1.4 Purpose:

This is a Super Class/ Parent Class from which several other classes inherit attributes and methods to reach DRY (Don't Repeat Yourself) code

5.1.5 Collaborations:

This class associates itself in order to successfully login

5.1.6 Attributes:

This class has the following attributes: name (John Doe), id(202200), email(johndoe@doe.com), and password (John-Doe-1)

5.1.7 Operations:

This class has the following operations:
- login(User); it takes in the User class as a parameter

5.1.8 Constraints:

N/A

5.2 Entity

Abstract Class

5.2.1 List of Superclasses:

N/A

5.2.2 List of Subclasses:

Course, Diploma, Job, Letter, Medicine, Line, Book

5.2.3 Purpose:

This is a Super Class/ Parent Class from which several other classes inherit attributes and methods to reach DRY (Don't Repeat Yourself) code

5.2.4 Collaborations:

N/A

5.2.5 Attributes:

This class has the following attributes: name (John Doe), id(202200)

5.2.6 Operations:

N/A

5.2.7 Constraints:

N/A

5.3 Student

Concrete Class

5.3.1 List of Superclasses:

User

5.3.2 List of Subclasses:

N/A

5.3.3 Purpose:

This is a concrete class mapping one of the most crucial system users

5.3.4 Collaborations:

- This class aggregates the Course class
- It associates the Diploma class
- It associates the Student's Courses classs
- It associates itself

5.3.5 Attributes:

This class inherits its attributes from the User superclass and does not have additional attributes

5.3.6 Operations:

This class has the following operations:
- register_course(student, course); it takes in the Student and Course classes
- access_learning_material(student, course); it takes in the Student and Course classes
- access_results(student, course); it takes in the Student and Course classes
- rate_course(student, course); it takes in the Student and Course classes
- request_course_drop(student, course); it takes in the Student and Course classes
- get_diploma(student, diploma); it takes in the Student and Diploma classes
- access_courses(student's courses); it takes in the Student's Courses class

5.3.7 Constraints:

N/A

5.4 Teacher

Concrete Class

5.4.1 List of Superclasses:

User

5.4.2 List of Subclasses:

N/A

5.4.3 Purpose:

Maps the Teacher class

5.4.4 Collaborations:

This class associates the Student, Parent, and Course classes

5.4.5 Attributes:

This class inherits its attributes from the User superclass and does not have additional attributes

5.4.6 Operations:

- take_attendance(student); it takes in the Student class
- send_warnings(student, parent); it takes in the Student and Parent classes
- provide_learning_material(course); it takes in the Course class
- provide_results(student, parent); it takes in the Student and Parent classes
- test(student); it takes in the Student class
- send_parents_review_letter(parent); it takes in the Parent class

5.4.7 Constraints:

N/A

5.5 Parent

Concrete

5.5.1 List of Superclasses:

User

5.5.2 List of Subclasses:

N/A

5.5.3 Purpose:

Maps the Parent class

5.5.4 Collaborations:

This class associates the Parent and Student classes

5.5.5 Attributes:

This class inherits its attributes from the User superclass and does not have additional attributes

5.5.6 Operations:

This class has the following operations:
- access_warning(parent, student
- access_results(parent, student)
- access_review_letter(parent, student)

5.5.7 Constraints:

N/A

5.6 Principle

Concrete

5.6.1 List of Superclasses:

User

5.6.2 List of Subclasses:

N/A

5.6.3 Purpose:

Maps the Principle class

5.6.4 Collaborations:

This class associates the Teacher, Parent, Student, HR, Principle, Job, Candidate, Diploma, Medicine, and Book classes

5.6.5 Attributes:

This class inherits its attributes from the User superclass and does not have additional attributes

5.6.6 Operations:

This class has the following operations:
- schedule_meeting_with_teacher(teacher); it takes in the Teacher class
- check_teachers_performance(teacher); it takes in the Teacher class
- send_emails_to_teachers(teacher); it takes in the Teacher class
- view_teachers_schedule(teacher); it takes in the Teacher class
- schedule_meeting_with_students_guardian(parent); it takes in the Parent class
- check_students_complaints(student); it takes in the Student class
- check_whether_students_paid_fees(student); it takes in the Student class
- check_school_budget(year); it takes in an integer named year
- check_hr_issues(hr); it takes in the HR class
- check_whether_salaries_are_paid(principle); it takes in the Principle class
- approve_job_posting(job); it takes in the job class
- interview_candidate(candidate); it takes in the candidate class
- approve_diploma(diploma); it takes in the diploma class
- approve_medicine_request(medicine); it takes in the medicine class
- approve_book_purchase(book); it takes in the book class

5.6.7 Constraints:

N/A

5.7 HR

Concrete

5.7.1 List of Superclasses:

User

5.7.2 List of Subclasses:

N/A

5.7.3 Purpose:

Maps the HR class

5.7.4 Collaborations:

This class associates the Job, Candidate, Letter, Driver, Teacher, and Line classes

5.7.5 Attributes:

This class inherits its attributes from the User superclass and does not have additional attributes

5.7.6 Operations:

This class has the following operations:
- post_new_vacancies(job)
- see_who_applied_for_certain_jobs(candidate)
- access_candidates_data(candidate)
- schedule_interviews(candidate)
- specify_an_employees_salary(candidate)
- access_letter_of_recommendation(letter)
- add_driver(driver)
- access_driver(driver)
- update_driver(driver)
-delete_driver(driver)
-assign_teacher_to_bus_line(teacher, line)

5.7.7 Constraints:

N/A

5.8 Candidate

Concrete

5.8.1 List of Superclasses:

User

5.8.2 List of Subclasses:

N/A

5.8.3 Purpose:

Maps the Candidate class

5.8.4 Collaborations:

This class associates the Letter class

5.8.5 Attributes:

This class inherits its attributes from the User superclass and does not have additional attributes

5.8.6 Operations:

This class has the following operations:
- submit_letter(letter)

5.8.7 Constraints:

N/A

5.9 Secretary

Concrete

5.9.1 List of Superclasses:

User

5.9.2 List of Subclasses:

N/A

5.9.3 Purpose:

Maps the Secretary class

5.9.4 Collaborations:

This class associates the Student, Course, and Student's Courses classes

5.9.5 Attributes:

This class inherits its attributes from the User superclass and does not have additional attributes

5.9.6 Operations:

This class has the following operations:
- add_student(student)
- access_students_information(student)
- update_students_information(student)
- delete_student(student)
- add_course(course)

- access_course(course)
- update_course(course)
- delete_course(course)
- add_course_to_student(student's courses)
- access_student's_courses(student's courses)
- update_student's_courses(student's courses)
- delete_course_from_student(student's courses)

5.9.7 Constraints:

N/A

5.10 School Nurse

Concrete

5.10.1 List of Superclasses:

User

5.10.2 List of Subclasses:

N/A

5.10.3 Purpose:

Maps the School Nurse class

5.10.4 Collaborations:

This class associates the Principle, Parent, and Medicine classes

5.10.5 Attributes:

This class inherits its attributes from the User superclass and does not have additional attributes

5.10.6 Operations:

This class has the following operations:
- report_incidents(principle, parent)
- check_inventory(medicine)
- request_inventory(medicine)

5.10.7 Constraints:

N/A

5.11 Transportation Manager

Concrete

5.11.1 List of Superclasses:

User

5.11.2 List of Subclasses:

N/A

5.11.3 Purpose:

Maps the Transportation Manager class

5.11.4 Collaborations:

This class associates the Driver, Line, and Student classes

5.11.5 Attributes:

This class inherits its attributes from the User superclass and does not have additional attributes

5.11.6 Operations:

This class has the following operations:
- assign_drivers(driver, line)
- check_drivers_performance(driver)
- add_to_already_paid_list(student)

5.11.7 Constraints:

N/A

5.12 Driver

Concrete

5.12.1 List of Superclasses:

User

5.12.2 List of Subclasses:

N/A

5.12.3 Purpose:

Maps the Driver class

5.12.4 Collaborations:

This class associates the Line and Student classes

5.12.5 Attributes:

This class inherits its attributes from the User superclass, and it has an additional attribute called line which is a string

5.12.6 Operations:

This class has the following operations:
- pick_up_students(student)
- request_transfer(line)

5.12.7 Constraints:

N/A

5.13 Librarian

Concrete

5.13.1 List of Superclasses:

User

5.13.2 List of Subclasses:

N/A

5.13.3 Purpose:

Maps the Librarian class

5.13.4 Collaborations:

This class associates the Book, Principle, and Student classes

5.13.5 Attributes:

This class inherits its attributes from the User superclass and does not have additional attributes

5.13.6 Operations:

This class has the following operations:
- check_book(book)
- request_book_transfer(book, principle)
- buy_book(book)

- lend_book(student)
- return_book(book, student)

5.13.7 Constraints:

N/A

5.14 Job

Concrete

5.14.1 List of Superclasses:

Entity

5.14.2 List of Subclasses:

N/A

5.14.3 Purpose:

Maps the Job class

5.14.4 Collaborations:

N/A

5.14.5 Attributes:

This class inherits its attributes from the Entity superclass and it has these additional attributes:
- title; which is a string
- hours_required; which is an integer
- years_of_experience_required; which is an integer
- expected_salary; which is an integer

5.14.6 Operations:

N/A

5.14.7 Constraints:

N/A

5.15 Letter

Concrete

5.15.1 List of Superclasses:

Entity

5.15.2 List of Subclasses:

N/A

5.15.3 Purpose:

Maps the Letter class

5.15.4 Collaborations:

N/A

5.15.5 Attributes:

This class inherits its attributes from the Entity superclass and it has these additional attributes:
- from; which is a string
- email; which is a string
- body; which is a string

5.15.6 Operations:

N/A

5.15.7 Constraints:

N/A

5.16 Medicine

Concrete

5.16.1 List of Superclasses:

Entity

5.16.2 List of Subclasses:

N/A

5.16.3 Purpose:

Maps the Medicine class

5.16.4 Collaborations:

N/A

5.16.5 Attributes:

This class inherits its attributes from the Entity superclass and it has these additional attributes:
- inventory; which is an int

5.16.6 Operations:

N/A

5.16.7 Constraints:

N/A

5.17 Book

Concrete

5.17.1 List of Superclasses:

Entity

5.17.2 List of Subclasses:

N/A

5.17.3 Purpose:

Maps the Book class

5.17.4 Collaborations:

N/A

5.17.5 Attributes:

This class inherits its attributes from the Entity superclass and it has these additional attributes:
- author; which is a string
- version; which is an integer
- year_published; which is an integer

5.17.6 Operations:

N/A

5.17.7 Constraints:

N/A

5.18 Line

Concrete

5.18.1 List of Superclasses:

Entity

5.18.2 List of Subclasses:

N/A

5.18.3 Purpose:

Maps the Line class

5.18.4 Collaborations:

N/A

5.18.5 Attributes:

This class inherits its attributes from the Entity superclass and it has these additional attributes:
- start_point; which is a string
- end_point; which is a string

5.18.6 Operations:

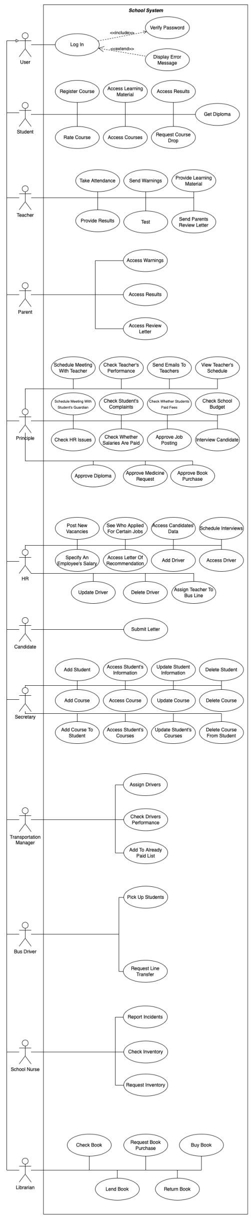
N/A

5.18.7 Constraints:

N/A

6 Operational Scenarios

This section should describes a set of scenarios that illustrate, from the user's perspective, what will be experienced when utilizing the system under various situations.



System's Use Case Diagram

7 Preliminary Budget Adjusted

As already mentioned, this project is Pro Bono.

8 Appendices

The following information is useful for a complete understanding of the document

8.1 Definitions, Acronyms, Abbreviations

- SRS: Software Requirements Specification
- OO: Object-Oriented