

AC32006 / AC52001 - Database Systems

Coursework 1 - Cover Sheet

TEAM NUMBER **14**

TEAM MEMBERS **Kamal Kaskasoli**
Brendan Wood
Hasna El-Nakla
Johana Horvathova
Nazat Hossain

<input type="checkbox"/> Company description	Number of pages	1
<input type="checkbox"/> Specification report	Number of pages	3
<input type="checkbox"/> E-R diagram	Number of pages	1
<input type="checkbox"/> User interface designs	Number of pages	3
<input type="checkbox"/> SQL statements	Number of pages	3
<input type="checkbox"/> Peer assessment forms (if included)	Number of pages	1

Total pages **12**

Word count: 935

COMPANY DESCRIPTION

Introduction

Royal Language Academy (RLA) is an established nationwide school for studying foreign languages. It operates across four branches throughout Scotland. Students from all walks of life are welcomed to study any of the languages taught; with six levels of programmes taught corresponding to Common European Reference for Framework of Reference (CEFR) for Languages level grades.

RLA has decided to go online, allowing potential students to enroll, and current students to view their courses and grades. Going online would expand the business by reaching out to more students who are able to explore the course selection and enroll from anywhere.

Teaching overview

Royal Language Academy offers courses in the following languages in all branches; -

- German
- Spanish
- French
- English as a Foreign Language

With a possibility of adding more in the future once the business expands. Each course has many classes running at once, in order to provide classes at different times to suit each of our student's schedule. Students are recommended to start from the basic level and progress through each grade to fully experience and understand the language. To assist the students there are online lessons provided for each class. These lessons recap concepts covered during class. Students access these online resources through the website with their given login credentials.

Business Overview

Royal Language Academy has four Branches across Scotland in Dundee, Edinburgh, Glasgow and Aberdeen. Each branch is run by a Branch Administrator, who is responsible for hiring staff, approving teaching resources and managing the business. Each Branch has multiple teachers capable of teaching one or more languages. Teachers are responsible for organising the classes that they teach and planning the syllabus.

Company Key Values

1. Providing top class education
2. Ensuring customer security
3. Immersing students in foreign culture

SPECIFICATION REPORT

The purpose of this database is to house student and staff records in relation to which course they are attending. In order to facilitate this users must be able to register an account on the system, which they can then log into. Typical users include; staff, students, teachers, and guests.

Users will have varying degrees of permissions to view and edit the database. They will not be granted direct access to the database. Instead their queries conducted via forms and drop down menus will return user-friendly reports. To the user it will not be explicit that there are actually accessing a database.

Requirements for each user type

Staff - Administrative/management staff within each branch

- Create new classes and assign students/tutor to them
- (de)-activate student/teacher online accounts
- View all outstanding balances
- Enroll new students on courses

Students - enrolled in a course

- View online course material
- View class scheduling
- View and edit their own personal information
- Pay their own outstanding balance

Teachers - Teachers for the courses

- View all students enrolled in their class(es)
- Upload online course material
- Edit class scheduling information for their class(es)

Guests - Anyone visiting the website

- View course information
- Register an account to begin enrollment

Database Design

Refer to E-R Diagram on page 6.

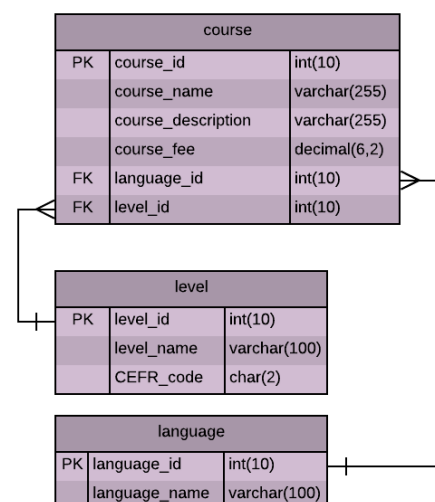
Course:

This area stores information about the courses available to enroll in, and the language and level that they are taught at.

The *course* table stores information about the courses available to students. It stores the name of the course, a description of what is taught and the price of enrolling in that course.

The *level* table stores the level codes of the course, following Common European Framework for Reference for Languages guidelines. E.g. a code will be A1, meaning that the course is for beginners.

The *language* table stores the list of languages taught at the school.

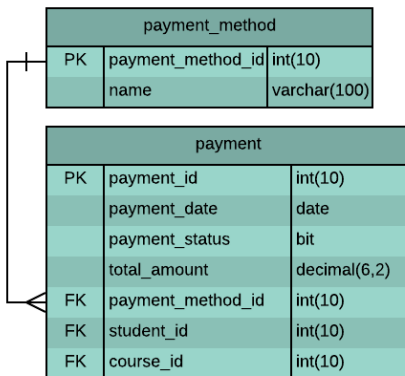


Payment:

This area stores information about the payments.

The *payment* table stores the date the payment was made, the status of whether or not it has been paid and the total amount paid.

The *payment_method* table stores the type of payment used. E.g. Visa Debit, paypal, etc.

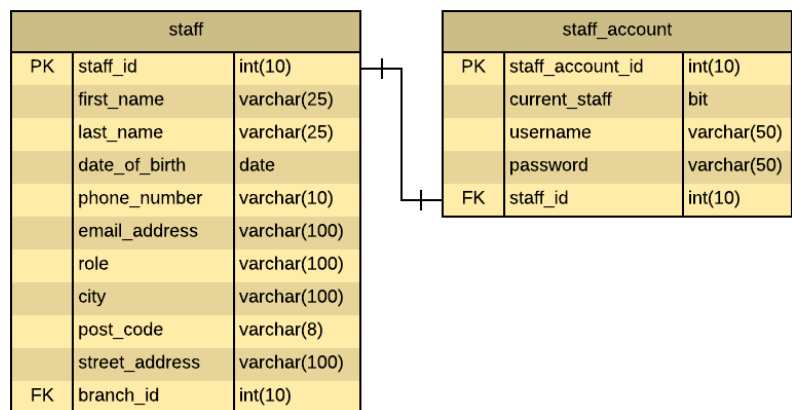


Staff:

This area of the diagram contains information about the employees of Royal Language Academy. E.g. Managers, Cleaning Staff, etc.

The *staff* table contains information about the staff members personal details, contact information and the role they work as in the School.

The *staff_account* table contains the login details of staff members and the *current_staff* attribute which denotes whether or not the person is currently a member of staff or not.

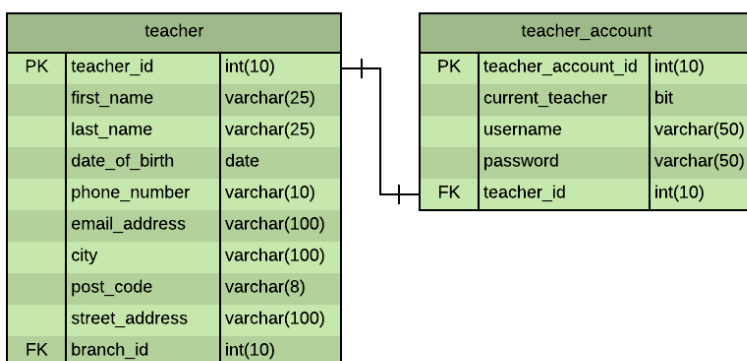


Teacher:

This area of the diagram contains information about the teachers of Royal Language Academy.

The *teacher* table stores the personal information of teachers and their contact information.

The *teacher_account* table contains the login details of teachers and the *current_teacher* attribute which denotes whether or not the person is currently a member of staff or not.

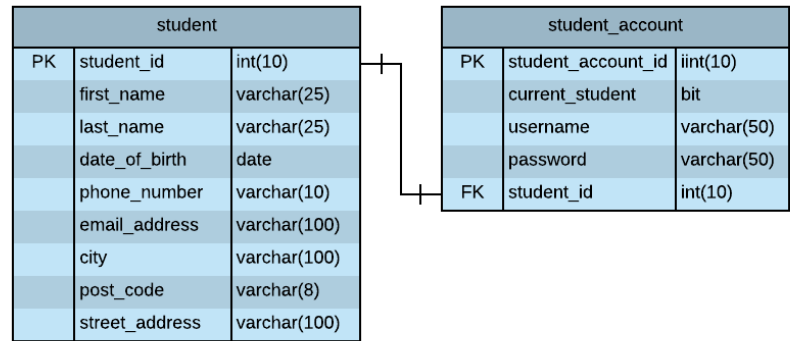


Student:

This area of the diagram contains information about the students of Royal Language Academy.

The *student* table contains personal information about the students.

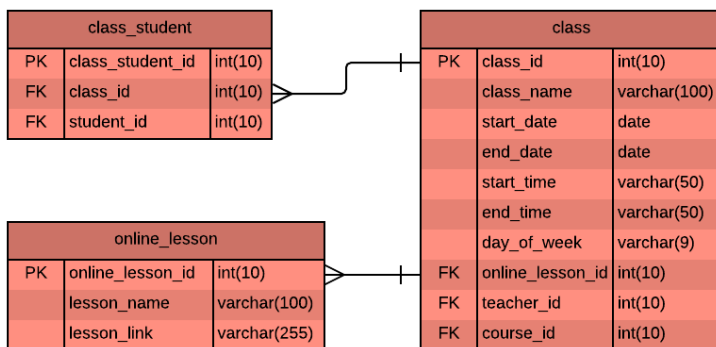
The *student_account* table contains login details for students.



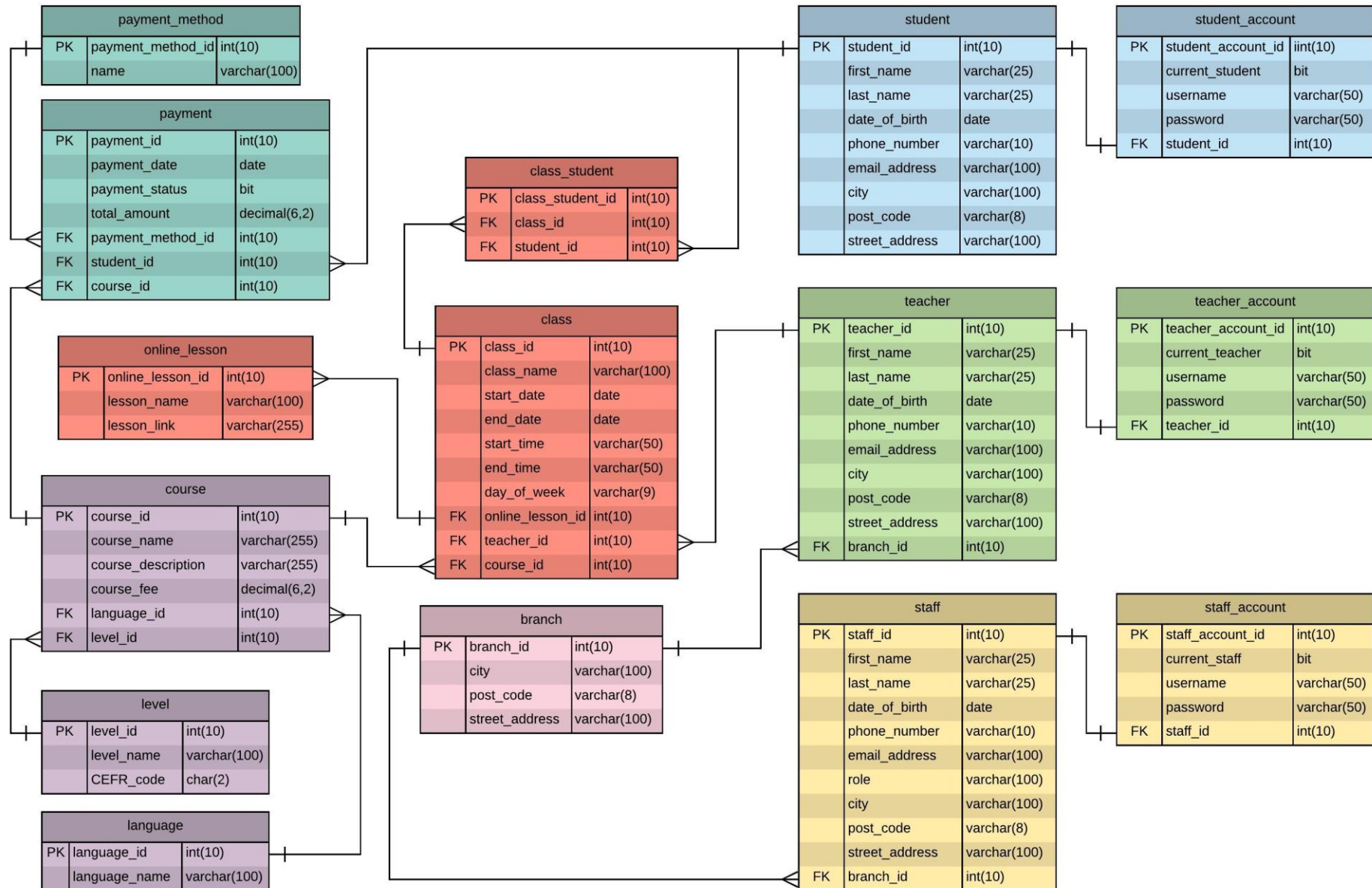
Class:

This area of the diagram contains information about the different classes in a course. A course is made up of one or more classes. Each class has weekly lessons over a period of time. There is a *class_student* table that associates classes and the students in them. Online lessons are supplemented to recap material covered in class. There are three tables, *class*, *online_lesson* and *class_student*.

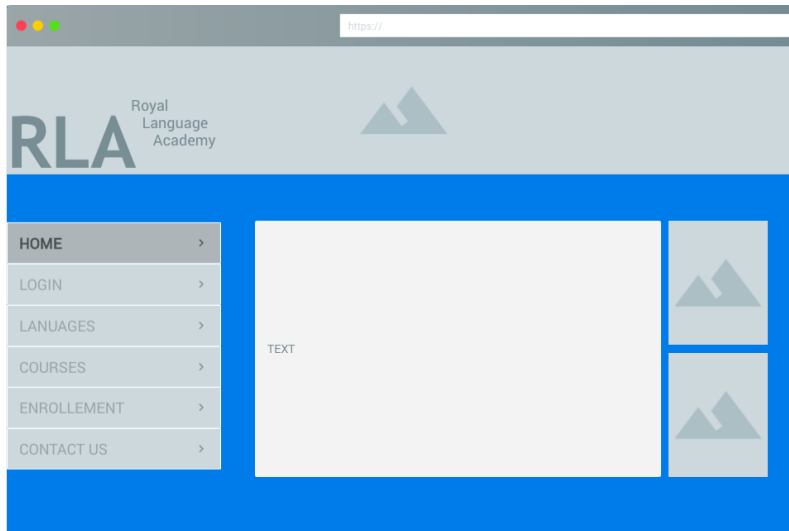
The *class* table is essential to the area. It stores the id and name of the class, the dates it runs between and the time the class is held. The *online_lesson* table stores the name of the lessons covered and a link to a file or resource. The *class_student* table is a joining table associating the classes to the students.



Royal Language Academy



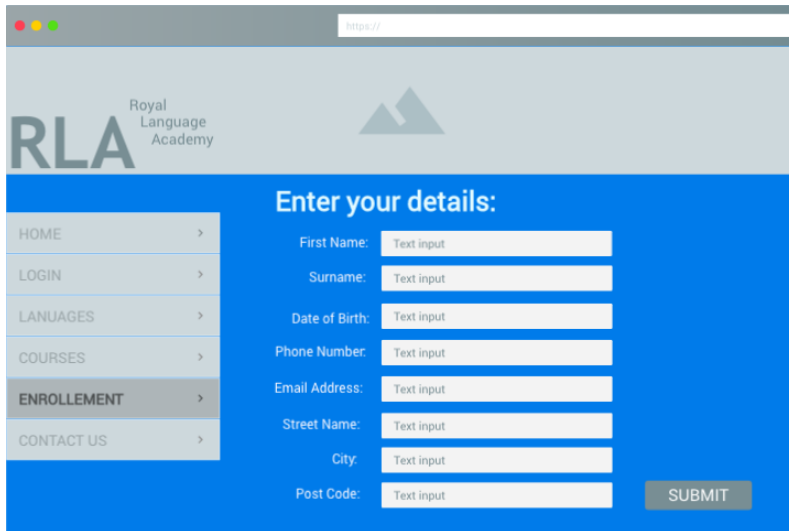
USER INTERFACE DESIGNS



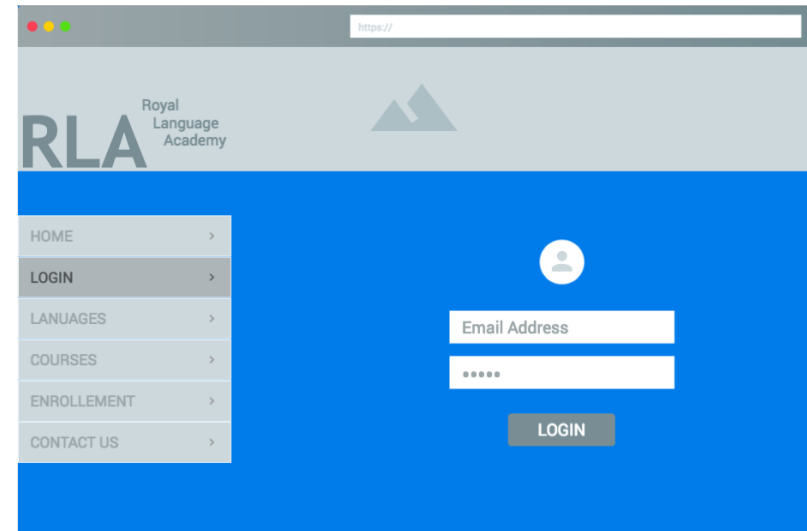
1. Home Page



2. Guest (Courses Overview) Page

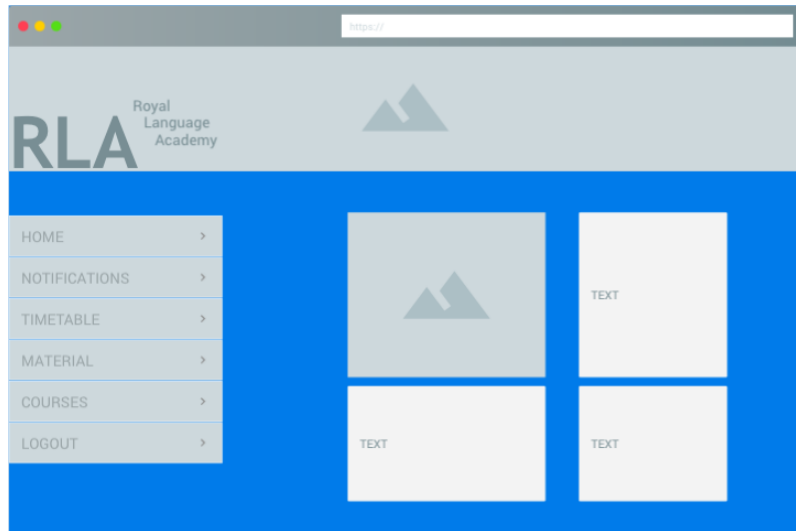


3. Enrollment Form Page

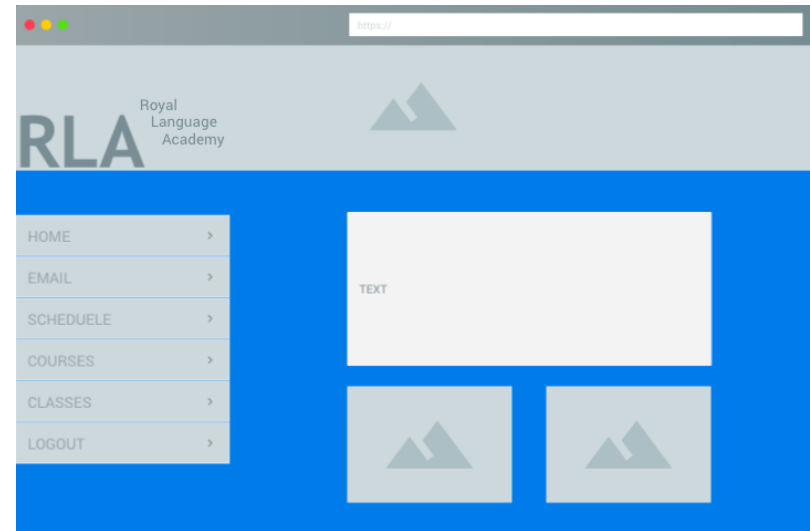


4. Login Page

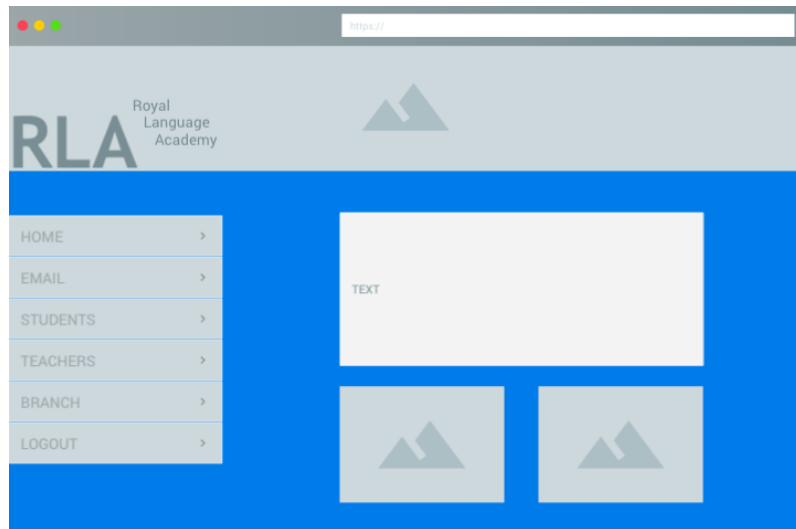
USER INTERFACE DESIGNS



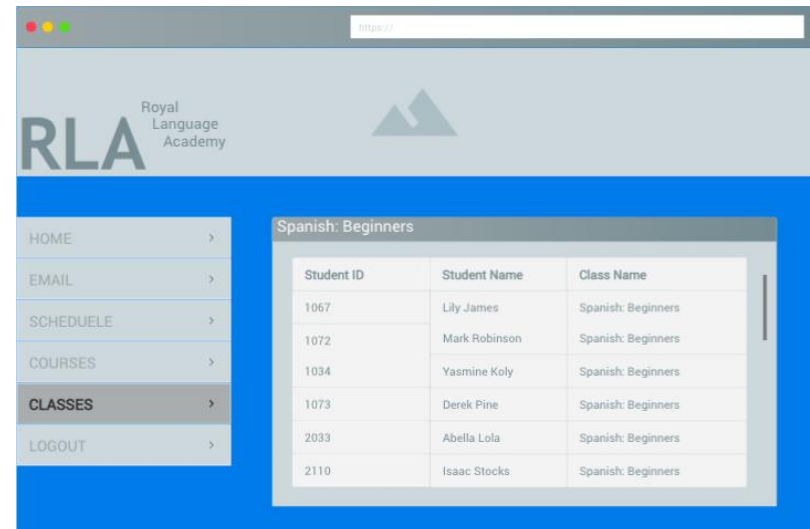
5. Student Account Page



6. Teacher Account Page



7. Staff Account Page



8. Class Register Page (from Teacher Account)

USER INTERFACE DESIGNS - explained

1. Home Page

The home page is the most significant interface screen as it serves as a landing page for all the different types of users. It consists of the company name to make users of the page aware about who we are and it has a permanent menu to help direct users on where to go next. All the users would essentially use the page in a different way. Existing Staff members, Teachers and Students can access their login page directly from the home page. Whereas guests who are just visiting the site can view information such as courses offered or company contact details and guests who are potential students can have a look at enrollment.

2. Guest (Courses Overview) Page

This interface screen design shows an example of a guest user using the website. The guest has loaded the Courses page, which displays all things about the courses offered at the company, from the Home page. Once the user has clicked on a course such as German for beginners, a new page will load with more information on that course such as payment amount and start/end date.

3. Enrollment Form Page

This interface screen design shows another example of a guest user, who might be a potential student, using the website. The enrollment page displays a form that must be filled out and submitted in order for the data to be stored in the database.

4. Login Page

The Login page is used by the Staff, Teacher and Student users. They can access their personal accounts from this page by entering their email address used to create the account and password. Although the Login page is the same for every user, it differentiates on their account page as shown in the other interface screen designs.

5. Student Account Page

The Student account page is the page loaded when a student user has logged into their account. They are immediately taken to their account home page where they can access notifications; announcements, emails sent from teachers, timetable; information on when, where and what classes the student is taking, material; online lessons, courses; looking at offered courses with permission to add more courses to your timetable and finally logout; which signs the student out of their account and back to the home page (interface screen 1).

6. Teacher Account Page

This page is for when a Teacher logs into their account. From this page they have access to their email, schedule, courses, classes and just like the Student account page, they can logout.

7. Staff Account Page

This is the page loaded when a Staff user logs into their account. They can access their email and logout from this page. They can also access records and information on students, teachers and the other branches if needed.

8. Class Register Page

This page is loaded when a teacher clicks on the Classes menu option from their account home page. The page presents a form for the classes they currently teach that consists of the student id, name and class they are taking. All of this information is stored in the database as shown in the ER diagram.

SQL STATEMENTS

```
CREATE DATABASE RLA;
```

```
CREATE TABLE student (  
  student_id INT(10) NOT NULL,  
  first_name VARCHAR(25) NOT NULL,  
  last_name VARCHAR(25) NOT NULL,  
  date_of_birth DATE NOT NULL,  
  phone_number VARCHAR(10) NOT NULL,  
  email_address VARCHAR(100) NOT NULL,  
  city VARCHAR(100) NOT NULL,  
  post_code VARCHAR(8) NOT NULL,  
  street_address VARCHAR(100) NOT NULL,  
  PRIMARY KEY (student_id)  
);
```

```
CREATE TABLE student_account (  
  student_account_id INT(10) NOT NULL,  
  current_student BIT NOT NULL,  
  username VARCHAR(50) NOT NULL,  
  password VARCHAR(50) NOT NULL,  
  student_id INT(10) NOT NULL,  
  PRIMARY KEY (student_account_id),  
  FOREIGN KEY (student_id) REFERENCES student(student_id)  
);
```

```
CREATE TABLE branch (  
  branch_id INT(10) NOT NULL,  
  city VARCHAR(100) NOT NULL,  
  post_code VARCHAR(8) NOT NULL,  
  street_address VARCHAR(100) NOT NULL,  
  PRIMARY KEY (branch_id)  
);
```

```
CREATE TABLE staff (  
  staff_id INT(10) NOT NULL,  
  first_name VARCHAR(25) NOT NULL,  
  last_name VARCHAR(25) NOT NULL,  
  date_of_birth DATE NOT NULL,  
  phone_number VARCHAR(10) NOT NULL,  
  email_address VARCHAR(100) NOT NULL,  
  role VARCHAR(100) NOT NULL,  
  city VARCHAR(100) NOT NULL,  
  post_code VARCHAR(8) NOT NULL,  
  street_address VARCHAR(100) NOT NULL,  
  branch_id INT(10) NOT NULL,  
  PRIMARY KEY (staff_id),  
  FOREIGN KEY (branch_id) REFERENCES branch(branch_id)  
)
```

```
CREATE TABLE staff_account (
  staff_account_id INT(10) NOT NULL,
  current_staff BIT NOT NULL,
  username VARCHAR(50) NOT NULL,
  password VARCHAR(50) NOT NULL,
  staff_id INT(10) NOT NULL,
  PRIMARY KEY (staff_account_id),
  FOREIGN KEY (staff_id) REFERENCES staff(staff_id)
);
```

```
CREATE TABLE level (
  level_id INT(10) NOT NULL,
  level_name VARCHAR(100) NOT NULL,
  CEFR_code CHAR(2) NOT NULL,
  PRIMARY KEY (level_id)
);
```

```
CREATE TABLE language (
  language_id INT(10) NOT NULL,
  language_name VARCHAR(100) NOT NULL,
  PRIMARY KEY (language_id)
);
```

```
CREATE TABLE course (
  course_id INT(10) NOT NULL,
  course_name VARCHAR(255) NOT NULL,
  course_description VARCHAR(255) NOT NULL,
  course_fee DECIMAL(6,2) NOT NULL,
  language_id INT(10) NOT NULL,
  level_id INT(10) NOT NULL,
  PRIMARY KEY (course_id),
  FOREIGN KEY (language_id) REFERENCES language(language_id),
  FOREIGN KEY (level_id) REFERENCES level(level_id)
);
```

```
CREATE TABLE payment_method (
  payment_method_id INT(10) NOT NULL,
  name VARCHAR(100) NOT NULL,
  PRIMARY KEY (payment_method_id)
);
```

```
CREATE TABLE payment (
  payment_id INT(10) NOT NULL,
  payment_date DATE NOT NULL,
  payment_status BIT NOT NULL,
  total_amount DECIMAL(6,2),
  payment_method_id INT(10) NOT NULL,
  student_id INT(10) NOT NULL,
  course_id INT(10) NOT NULL,
  PRIMARY KEY (payment_id),
  FOREIGN KEY (student_id) REFERENCES student(student_id),
  FOREIGN KEY (course_id) REFERENCES course(course_id)
);
```

```

CREATE TABLE online_lesson (
online_lesson_id INT(10),
lesson_name VARCHAR(100),
lesson_link VARCHAR(255),
);

CREATE TABLE class (
class_id INT(10) NOT NULL,
class_name VARCHAR(100) NOT NULL,
start_date DATE NOT NULL,
end_date DATE NOT NULL,
start_time DATE NOT NULL,
end_time DATE NOT NULL,
course_id INT(10),
teacher_id INT(10),
PRIMARY KEY (class_id),
FOREIGN KEY (course_id) REFERENCES course(course_id),
FOREIGN KEY (teacher_id) REFERENCES teacher(teacher_id)
);

CREATE TABLE class_student (
class_student_id INT(10) NOT NULL,
class_id INT(10) NOT NULL,
student_id INT(10) NOT NULL,
PRIMARY KEY (class_student_id)
FOREIGN KEY (class_id) REFERENCES class(class_id),
FOREIGN KEY (student_id) REFERENCES student(student_id)
);

CREATE TABLE teacher (
teacher_id INT(10) NOT NULL,
first_name VARCHAR(25) NOT NULL,
last_name VARCHAR(25) NOT NULL,
date_of_birth DATE NOT NULL,
phone_number VARCHAR(12) NOT NULL,
email_address VARCHAR(100) NOT NULL,
city VARCHAR(100) NOT NULL,
post_code VARCHAR(8) NOT NULL,
street_address VARCHAR(100) NOT NULL,
branch_id INT(10) NOT NULL,
PRIMARY KEY (teacher_id),
FOREIGN KEY (branch_id) REFERENCES branch(branch_id)
);

CREATE TABLE teacher_account (
teacher_account_id INT(10) NOT NULL,
current_teacher BIT NOT NULL,
username VARCHAR(50) NOT NULL,
password VARCHAR(50) NOT NULL,
teacher_id INT(10) NOT NULL,
PRIMARY KEY (teacher_account_id),
FOREIGN KEY (teacher_id) REFERENCES teacher(teacher_id)
);

```