Tanglewood LMS

**Development Company: Dev Fusion**

**Client: Roxanne Decoito**

## Proposal Summary

The development project revolves around the creation of a Learner Management System (LMS) for use by teachers in a primary school setting.

**Pages:**

* Log-In Page
  + Allows user to log-in
  + Has a remember me option
* Announcements Page
  + List of announcements
  + Allows user to make a new announcements
  + Announcements are manually written
  + Announcements can be pinned by the admin
  + A search bar for announcements
* Add Students Page
  + Allows you to add a new student to the database
  + Starts with standard details, then leads into assigning them a Grade
* Disable Student Page
  + Allows you to remove a student from the available list of students
  + Likewise removes all links to the student, such as their grade, subjects and marks.
* Alter Student Page
  + Allows you to edit student details and alter links to students, such as their grade, subjects and marks.
* Markbook Page
  + Presents the student’s current marks for each subject, and a total of their marks per subject.
  + Should be filterable based on Year, Grade, Subject, Term
  + Should allow the user to add or modify the marks assigned to the student
  + For Term 4, the student’s markbook should show the average for the year (assuming even weighting across all subjects)
* Admin Page
  + Should allow an Administrative User to Create or Disable Teachers Accounts.
  + Should allow an Administrative User to Create, Modify or Disable Subjects
* Report Page
  + Should display a given student’s entire report card, and have a button used to download the report as a pdf.
* Account Management Page
  + Allows any teacher with an account to modify their password.

**UI Details**

* Pastel pink color scheme with a blend of light blue to break up the pinkness..
* Staff Orientated
* Font should be more professional
* Sizing should be done according to an approximate 100 students / 20 teachers

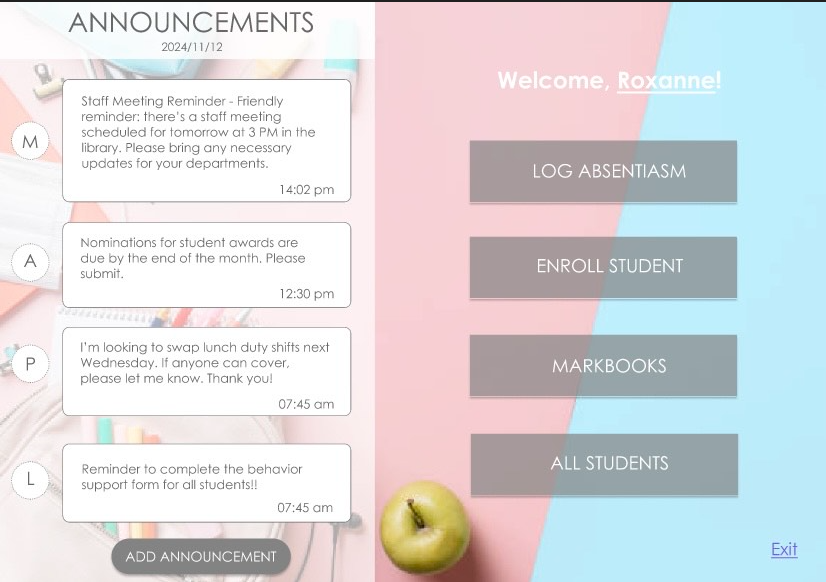
**Additional Details:**

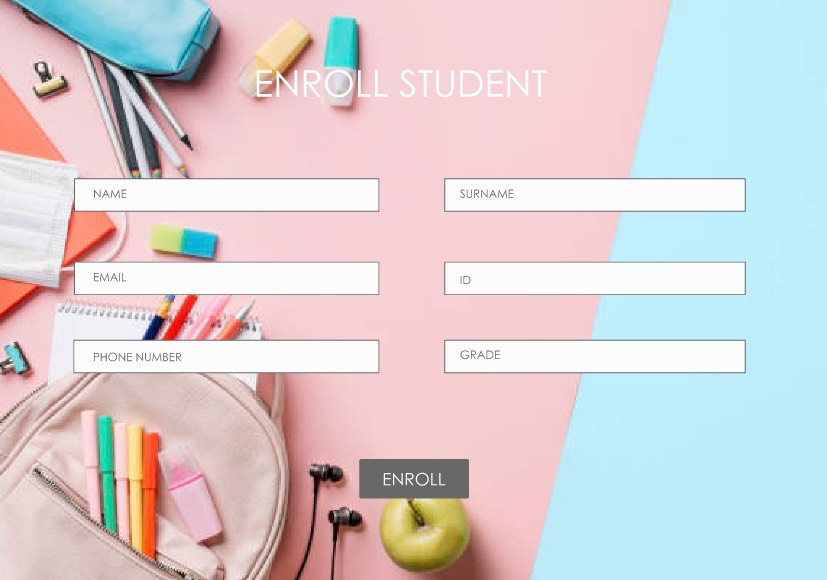
* Should accommodate grades 5-7, with CAPS Curriculum subject options.
* Student details include: name, surname, date of birth, gender, biological sex, primary address, ID, parent email, parent phone number, student ID number, learning difficulties, any important notes.

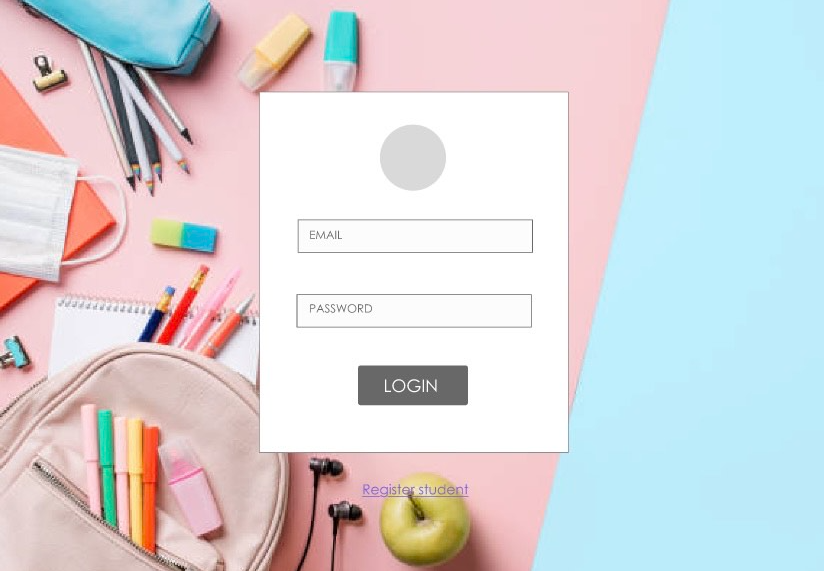
## Logo Design

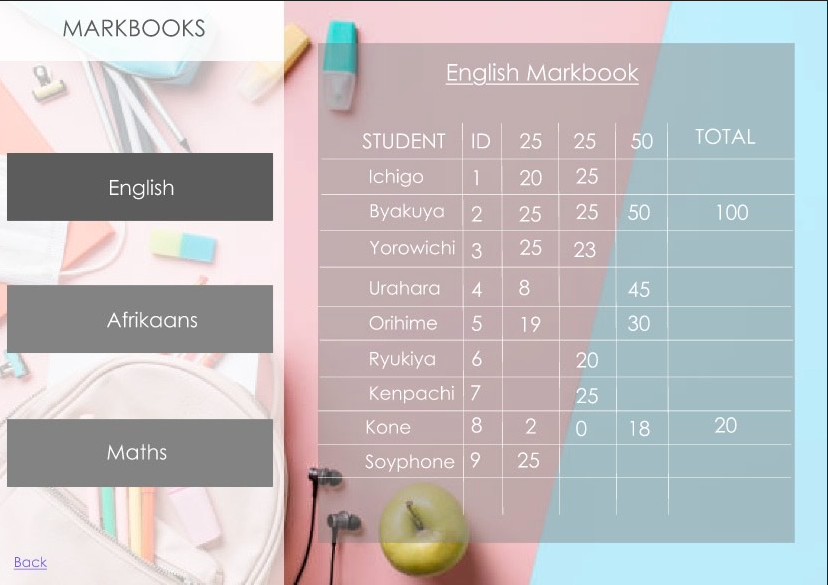


## Selected Website Mock Design

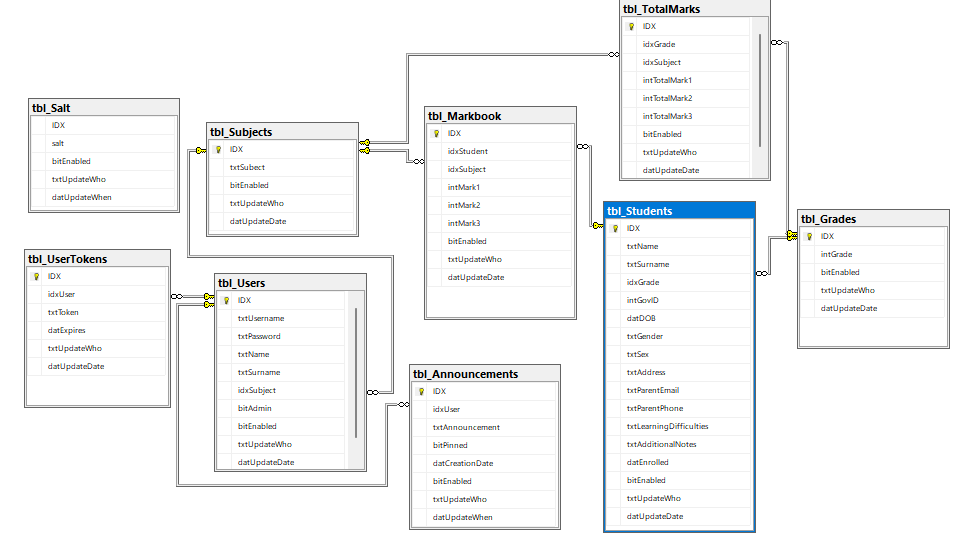








## Database Schema Rough Draft

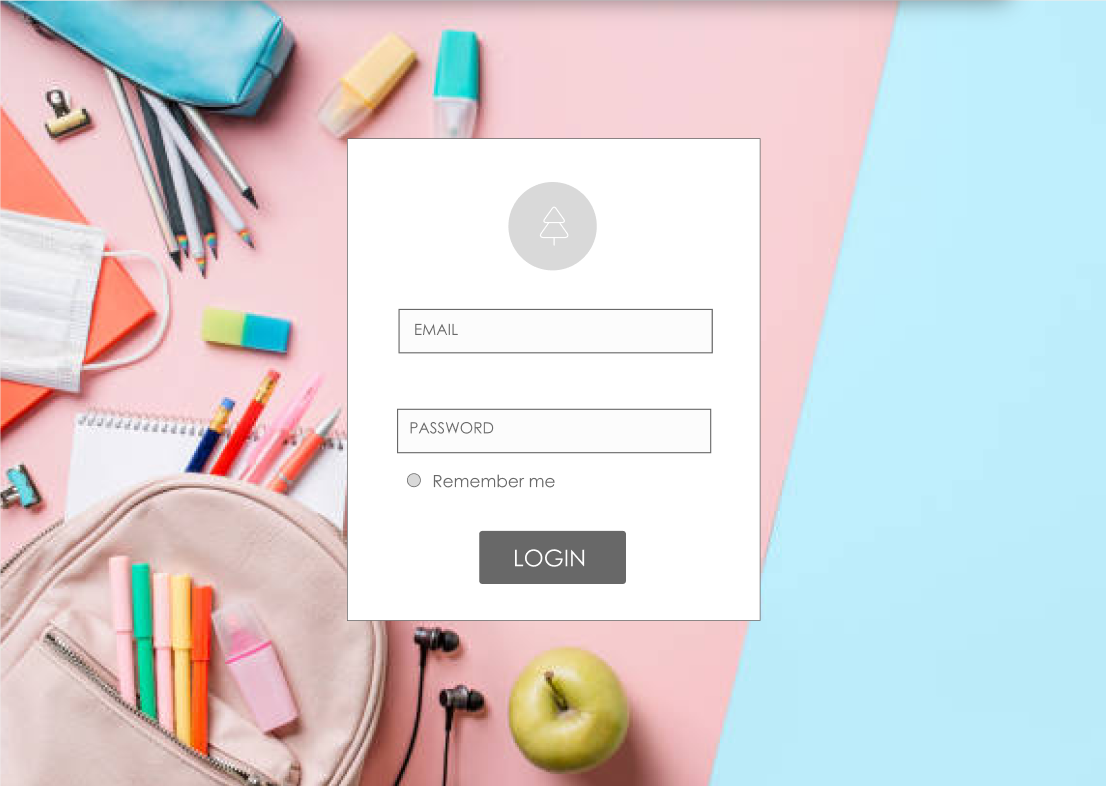


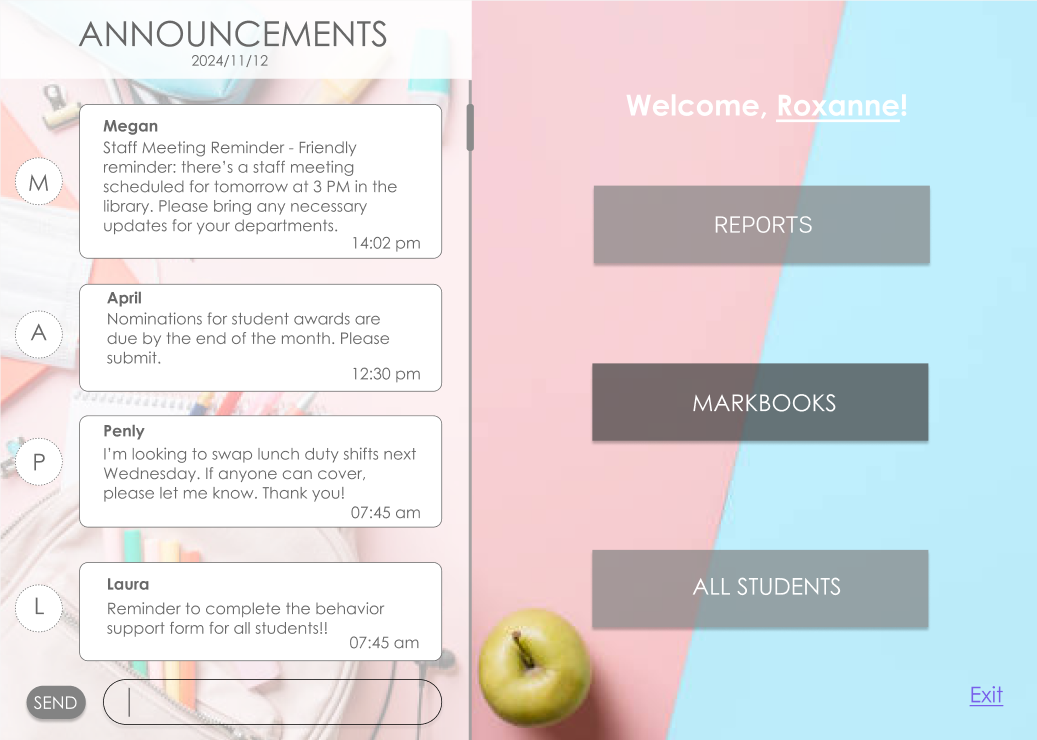
## 

## Followup Requests & Alterations

* Implement a textbox for the announcement creation.
* Filter markbooks and reports more precisely.
* Accommodate future subjects being added.
* Implement a “remember me” feature.
* Make announcements scrollable.
* Make a dedicated report page and make the report downloadable.
* Manage Staff in addition to students for administrator accounts.

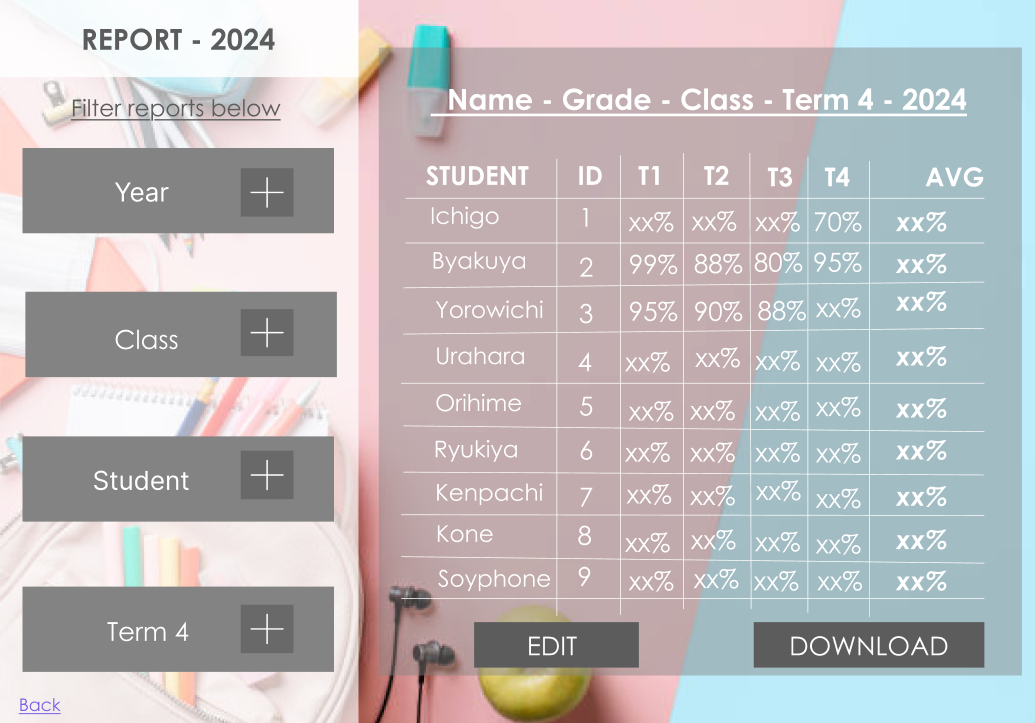
## Modified Website Updated Design



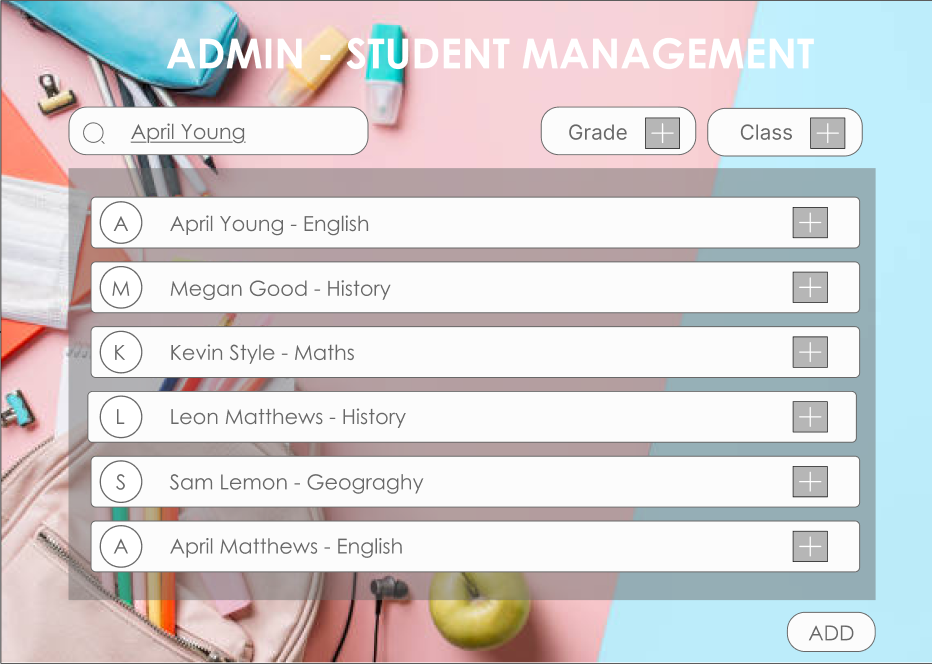


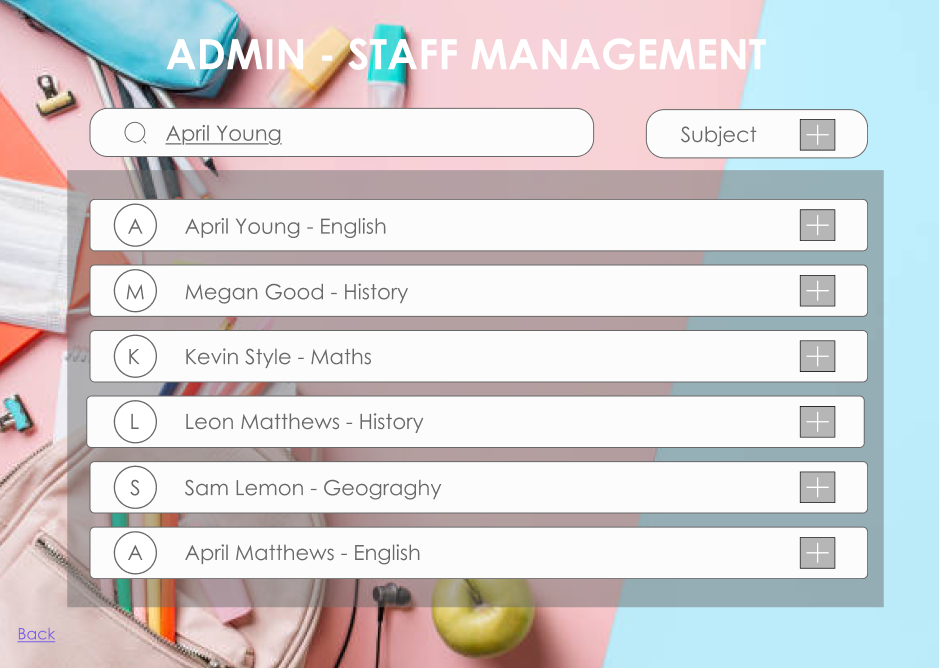


* The inclusion of a download button is an oversight and should not be considered

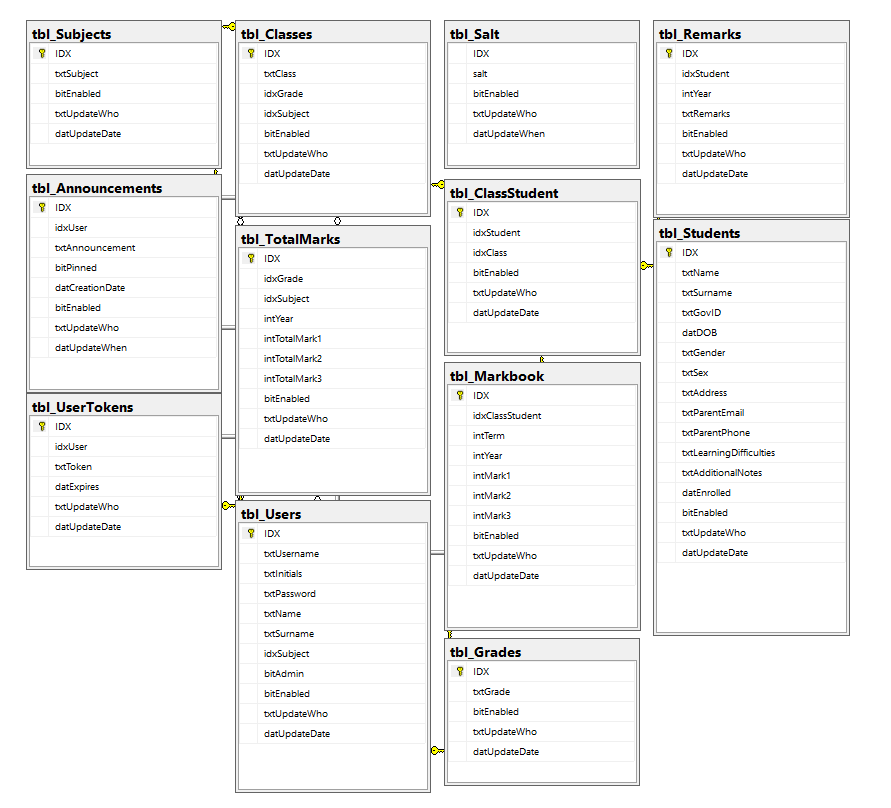


* The inclusion of an edit button is an oversight and should not be considered





## Database Schema Rough Draft



## 

## Final Client Requests & Alterations

* Add comment section to reports
* Automate report final calculation
* Make the display scalable.

## Backend Stored Procedures

* addClass [ To create a class-student relationship and populate the markbook with relevant records for each term ]
* AddToken [ Assigns a newly generated “remember me” token to the logged in user ]
* CheckAdmin [ Check if a given user record is marked as an Admin ]
* CheckDuplicate [ Checks if the username of a user being saved has already been taken ]
* CheckToken [ Checks if the current “remember me” token maps onto a User and if it has expired yet ]
* ClassDropdown [ Populates the Class DropDownList based on the selected Grade and Subject ]
* CreateAnnouncement [ Creates an Announcement record ]
* CreateStudent [ Creates a blank student record ]
* CreateUser [ Creates a blank user record if no other blank username record exists ]
* DisableStudent [ Switches the Student’s bitEnabled field to ‘false’ ]
* DisableUser [ Switches the User’s bitEnabled field to ‘false’ ]
* DisplayAnnouncements [ Retrieves announcements and their poster ]
* DisplayReport [ Pulls a collection of markbooks based on criteria for the code behind to work with ]
* DisplayStudent [ Pulls a student’s data ]
* EmptyCheck [ Checks if a user has no username ]
* EmptyStudents [ Displays only students with no classes assigned ]
* FetchMarkbook [ Pulls the markbook of a student based on class, and term ]
* FetchMaxMarks [ Pulls the max marks of a given subjects assessments ]
* fetchRemark [ Pulls the remark of a given class-student relationship, which is used to give reports remarks ]
* FetchStaff [ Pulls basic info about all staff to populate a list ]
* FetchStaffInfo [ Pulls advanced info about a staff member except username and password ]
* FetchStudent [ Pulls basic info about all students to populate a list ]
* FetchStudentClasses [ Pulls all classes that the student currently has assigned ]
* FetchStudentDetails [ Pulls advanced info about a student ]
* GradeDropdown [ Populates the Grade DropDownList ]
* LoginValidation [ Validates the user’s input username and the hash of their salted password against the stored hash ]
* removeClass [ Disables a given student-class relationship using bitEnabled ]
* SaltPassword [ Adds a salt to the end of a given password ]
* StudentDropdownReport [ Populates a student dropdown for the report page ]
* SubjectDropdown [ Populates a subject DropDownList ]
* UpdateMarkBook [ Updates all records for a given classes markbook ]
* UpdateRemark [ Updates the stored remark for a given class-student relationship ]
* UpdateStudent [ Updates a student records details ]
* UpdateUser [ Updates the users details as well as the password/username if those options are selected ]