LEVEL 5

COMP50001: Commercial Computing

Sprint Document 1

Batch code : IFK2321COM

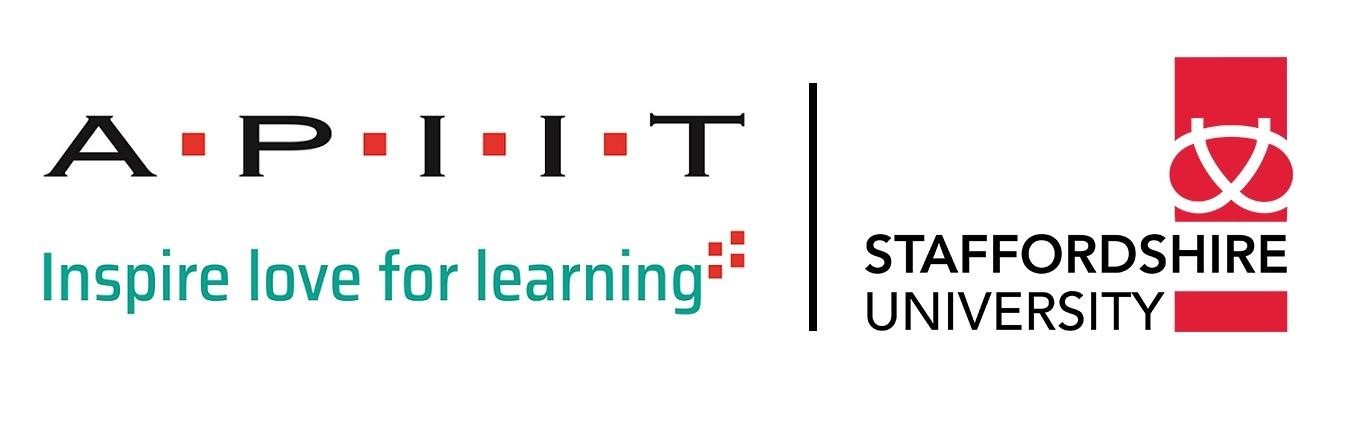
Group members :

Ayantha: CB010357/ 21035802

Bashini: CB010292/ 21035792

Kavinda: CB010406/ 21035795

Maryam: CB010658/ 21035976



# Sprint goals

1. Complete implementation of a role-based authentication process, including login and password reset to ensure data security and seamless user experience.
2. Design a user-friendly interface for the admin dashboard that is easy to use and navigate.
3. Complete implementation of functionality in the admin pages which allows the admin user to perform CRUD operations on users and courses.
4. Implement messaging functionality between admin and normal users.

# Sprint backlog

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | User story | Story points | Priority | Sprint | Status |
| U001 | As an admin, I want to be able to login to the system, so that I can manage the LMS. | 2 | High | 1 | Done |
| U002 | As an admin, I want to have access to a comprehensive admin dashboard that provides real-time information, including user analytics, and course overview. | 8 | High | 1 | Done |
| U003 | As an admin, I want to be able to create and manage user accounts, so that I can provide users with access to contents of the LMS depending on their role. | 13 | High | 1 | Done |
| U004 | As an admin, I want to be able to create and manage courses, so that users can access them. | 8 | Medium | 1 | Done |
| U006 | As an admin, I want to be able to view and reply to messages from users, so that I can help them with their inquiries. | 5 | Medium | 1 | Done |
| U007 | As an admin, I want to be able to change my password, so that I can renew the old password. | 5 | High | 1 | Done |
| U008 | As an admin, I want to be able to edit my profile, so that I can change my profile details. | 5 | Low | 1 | Done |
| U009 | As an admin, I want to be able to log out of the system, so that I can prevent other users from accessing the system without verifying their credentials. | 1 | High | 1 | Done |
| U010 | As a user, I want to login to the system, so that I can access the content of the LMS that’s specific to my role. | 1 | High | 1 | Done |
| U011 | As a user, I want to be able to change my password, so that I can renew the old password. | 5 | High | 1 | Done |
| U012 | As a user, I want to be able to log out of the system, so that I can prevent other users from accessing the system without verifying their credentials. | 1 | High | 1 | Done |
| U025 | As a user, I want to be able to send messages to the admin, so that I can communicate any questions, concerns, or requests about the content of the LMS. | 5 | High | 1 | Done |

# Sprint roles

|  |  |
| --- | --- |
| Name | Role |
| Bashini Weerasinghe | Project Manager |
| Maryam Mashkoora | Developer |
| Ayantha Seneviratne | UI/UX Engineer |
| Kavinda Rajapaksha | QA Engineer |

# Solution conceptualizing

## Use case diagram



The use case “Manage User” includes add, edit and delete use cases for the users.

The use case “Manage Course” includes add, edit and delete use cases for the courses.

## Use case scenarios

### Login

|  |  |
| --- | --- |
| Use Case Name | User Login |
| Description | Allows users to login to the system. |
| Primary Actor | All users |
| Pre-Conditions | System is up and running. |
| Basic Flow | 1. Use Case begins when the user opens the application. 2. The system loads the login page to the user. 3. The user can enter credentials and click on login. 4. The system validates the entered credentials.   Use Case ends after the user is successfully logged into the system. |
| Extensions | If the operation is not successful,  1. System display message “Invalid credentials. Please try again”  2. Use case ends |
| Post Conditions | The user can perform all his operations after a successful login. |
| Optional Information | - |

### Add User

|  |  |
| --- | --- |
| Use Case Name | Add User |
| Description | Allows the admin to add users(students, parents, teachers) to the system. |
| Primary Actor | Admin |
| Pre-Conditions | System is up and running, admin has to be logged into the system, and an Users(Students/Teachers) page opened. |
| Basic Flow | 1. Use case begins when Admin clicks on “Register User” button at the top of the Users list in the opened users page. 2. System prompts an “Add User” form to the admin. 3. Admin adds user details and clicks on “Save” button. 4. Use case ends after the new user is added to the system and a success message is displayed. |
| Extensions | If the entered user details are not valid  1. System display error message  2. Redirects user to retry. |
| Post Conditions | The admin can add users to the system and view the added users as a list in the respective page. |
| Optional Information | - |

### Edit User

|  |  |
| --- | --- |
| Use Case Name | Edit User Details |
| Description | Allows the admin to change the details of users (Students/ Teachers) in the system. |
| Primary Actor | Admin |
| Pre-Conditions | System is up and running, admin has to be logged into the system, and an Users(Students/Teachers) page opened. |
| Basic Flow | 1. Use case begins when Admin select a user and clicks on the edit button. 2. System prompts an “Edit User” form to the admin with editable details. 3. Admin alters the data and clicks on the save button. 4. Use case ends after the system save the altered information in the database and displays a success message. |
| Extensions | If the entered/edited details are not valid  1. System display error message  2. Redirects user to retry. |
| Post Conditions | The admin can view the alterations which he/she has done for user. |
| Optional Information | - |

### Delete User

|  |  |
| --- | --- |
| Use Case Name | Delete User |
| Description | Allows admin to remove users(Students/Teachers) from the system. |
| Primary Actor | Admin |
| Pre-Conditions | System is up and running, admin has to be logged into the system, and an Users(Students/Teachers) page opened. |
| Basic Flow | 1. Use case begins when Admin selects a user and click on the delete button. 2. System prompts a message to Admin to assure the user removal with Yes and No options. 3. Use case ends after Admin selects the Yes option, and the system removes the user from the system. |
| Extensions | 1. If admin selects No, the system stop the delete query execution. 2. Admin will not be able to see any changes in the users list. |
| Post Conditions | The deleted user is no longer available in the users list, and they can no longer access the system |
| Optional Information | - |

### Add Course

|  |  |
| --- | --- |
| Use Case Name | Add course |
| Description | Allows the admin to add courses to the system. |
| Primary Actor | Admin |
| Pre-Conditions | System is up and running, admin has to be logged into the system, and courses page opened. |
| Basic Flow | 1. Use case begins when Admin clicks on “Register Course” button at the top of the course list in the Courses page. 2. System prompts an “Add Course” form to the admin. 3. Admin adds course details and clicks on “Save” button. 4. Use case ends after the new course is added to the system and a success message is displayed. |
| Extensions | If the entered course details are not valid  1. System display error message  2. Redirects user to retry. |
| Post Conditions | The Admin can add courses to the system and view the added courses as a list in the Courses page. |
| Optional Information | - |

### Edit Course

|  |  |
| --- | --- |
| Use Case Name | Edit course information |
| Description | Allows the admin to change the details of courses in the system. |
| Primary Actor | Admin |
| Pre-Conditions | System is up and running, admin has to be logged into the system, and courses page opened. |
| Basic Flow | 1. Use case begins when Admin select a course and click on the edit button. 2. System prompts a Course form for admin with editable details. 3. Admin alters the data fields and click on the save button.  Use case ends after the system save the altered information in the database and displays a success message. |
| Extensions | If the entered/edited details are not valid  1. System display error message  2. Redirects user to retry. |
| Post Conditions | The admin can view the alterations which he/she has done for courses. |
| Optional Information | - |

### Delete Course

|  |  |
| --- | --- |
| Use Case Name | Delete course |
| Description | Allows admin to remove Courses from the system. |
| Primary Actor | Admin |
| Pre-Conditions | System is up and running, admin has to be logged into the system, and courses page opened. |
| Basic Flow | 1. Use case begins when Admin selects a course and click on the delete button. 2. System prompts a message to Admin to assure the course removal with Yes and No options. 3. Use case ends after Admin selects the Yes option, the system removes the course from the database. |
| Extensions | 1. If admin selects No, the system stop the delete query execution. 2. Admin will not be able to see any changes in the course list. |
| Post Conditions | The deleted course is no longer accessible/available to users. |
| Optional Information | - |

### Edit Profile

|  |  |
| --- | --- |
| Use Case Name | Edit profile. |
| Description | Allows admin to alter their profile details. |
| Primary Actor | Admin |
| Pre-Conditions | System is up and running, admin has to be logged into the system. |
| Basic Flow | 1. Admin click on the settings link in the side menu. 2. System displays an Admin profile edit form with Save and Cancel buttons . 3. Admin fill up the details he/she wishes to update and click on save button. 4. System displays a success message “Details are saved”. |
| Extensions | If Admin click on the Cancel button   1. System will redirect the admin to the admin dashboard. 2. Use case ends   If entered details are invalid   1. An error message is displayed, and user redirected to retry. |
| Post Conditions | Admin can edit their profile details, and view the changes made. |
| Optional Information | - |

### Change Password

|  |  |
| --- | --- |
| Use Case Name | Change Password |
| Description | Allows users(admin and other users) to change their login password |
| Primary Actor | All users |
| Pre-Conditions | System is up and running, user has to be logged into the system |
| Basic Flow | 1. Use Case begins when the user clicks on the “Change Password” option in their profile. 2. The system displays “Change Password” form to admin 3. Admin fills out the old password and new password 4. System validates the inputs 5. Use Case ends after the password is renewed successfully |
| Extensions | If the operation is not successful,  1. System display message “Could not change password. Please try again”  2. Use case ends |
| Post Conditions | The user can login with their password. |
| Optional Information | - |

### Message

|  |  |
| --- | --- |
| Use Case Name | User Message |
| Description | Allows user to send messages to the admin. |
| Primary Actor | User |
| Pre-Conditions | System is up and running, User has to be logged into the system |
| Basic Flow | 1. Use Case begins when the user clicks on the “Send message” option in the dashboard.  2. The system displays, the message window with the description space and a send option.  3. User can type in a message and click on send.  4. The message is recorded in the Admins inbox. |
| Extensions | If the operation is not successful,  1. The message is not sent.  2. Use case ends |
| Post Conditions | The user can communicate with admin through this message portal. |
| Optional Information | - |

### Logout

|  |  |
| --- | --- |
| Use Case Name | User Logout |
| Description | Allows users to logout from the system. |
| Primary Actor | All users |
| Pre-Conditions | System is up and running, user has to be logged into the system |
| Basic Flow | 1. Use Case begins when the user clicks on the “Logout” option. 2. The system displays “Are you sure you want to logout” message with yes and no options. 3. If user clicks on yes, the user is successfully logged out from the system. |
| Extensions | If the operation is not successful,  1. User redirected to their previous page  2. Use case ends |
| Post Conditions | - |
| Optional Information | - |

## Activity diagrams

### Login

### 

### Add User

### 

### Edit User



### Delete User



### Add Course



### Edit Course



### Delete Course



### Edit Profile



### Change Password



### Message



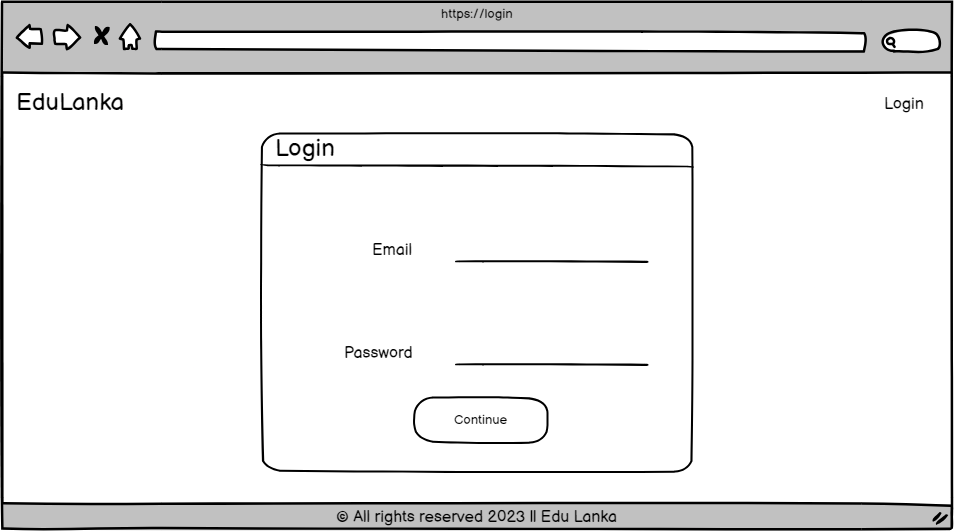
### Logout



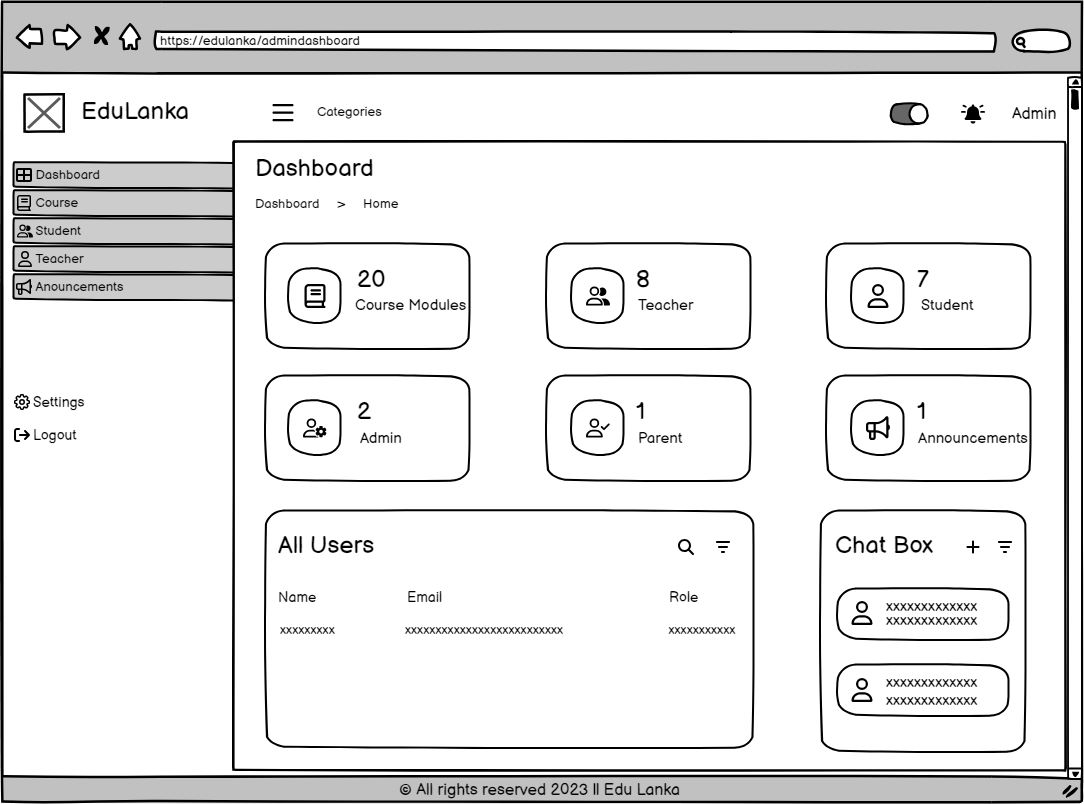
# Solution Design

## Wireframes

### Login



### Admin Dashboard



### Students

A screenshot of a computer

Description automatically generated

### Teachers

A screenshot of a computer

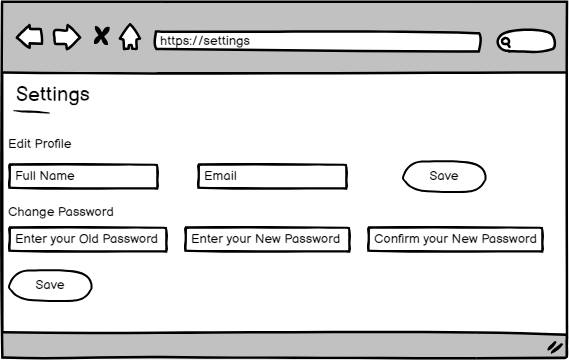
Description automatically generated

### Courses

A computer screen shot of a computer

Description automatically generated

### Settings



### Add Student

A computer screen with a login box

Description automatically generated

### Add Teacher

A computer screen shot of a computer

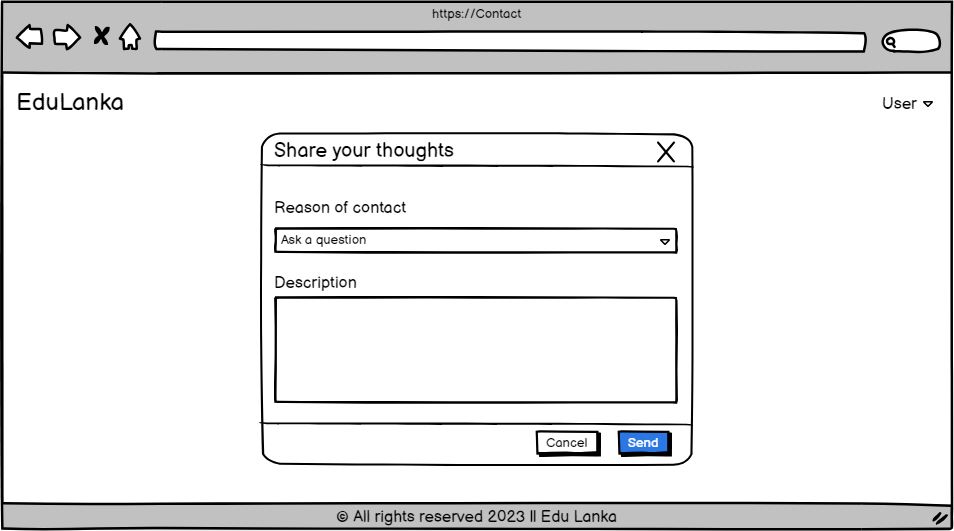
Description automatically generated

### Add Course

A screenshot of a computer

Description automatically generated

### Contact Admin



### Change Password

A screenshot of a computer

Description automatically generated

## Interfaces

### Login

A login page with books and a magnifying glass

Description automatically generated

### Admin Dashboard

A screenshot of a computer

Description automatically generated

### Students

A screenshot of a computer

Description automatically generated

### Teachers

A screenshot of a computer

Description automatically generated

### Courses

A screenshot of a computer

Description automatically generated

### Settings

A screenshot of a computer

Description automatically generated

### Add Student

A screenshot of a computer

Description automatically generated

### Add Teacher

A screenshot of a computer

Description automatically generated

### Add Course

A screenshot of a computer

Description automatically generated

### Contact Admin

A screenshot of a computer

Description automatically generated

### Change Password

### A screenshot of a computer Description automatically generated

# Implementation

## Code Refactoring

During the quality assurance procedure done by the sprint’s QA engineer, some errors and areas of improvement were discovered.

The below table is a summary of such code refactoring carried out.

|  |  |  |
| --- | --- | --- |
| Section(s) | Error / Missing Component | Alteration(s) |
| setting.blade.php | <title>Document</title> | <title>Admin Settings</title> |
| setting.blade.php | placeholder=”Enetr your first name” | placeholder=”Enetr your full name” |
| setting.blade.php | <label>Email</label> | <label>Email Address</label> |
| setting.blade.php | placeholder=”Enetr email” | placeholder=”Enetr your email address” |
| banner.blade.php | <title>Admin Dashborad</title> | <title>Admin Dashboard</title> |
| banner.blade.php | Edu Lanka All Right Reseved | Edu Lanka All Right Reserved |
| student1.blade.php | Placeholder missing in input forms.  <input type=”email” class=”form-control” id=”editEmail” name=”email”> | <input type=”email” class=”form-control” id=”editEmail” name=”email” placeholder=”Enter your email address”> |
| teacher1.blade.php | Typo in the copyright section. | Edu Lanka All Right Reserved |
| teacher1.blade.php | Typo in the alert message.  “Are you sure you want to delete this courser?” | “Are you sure you want to delete this teacher?” |
| login.blade.php | Placeholder missing in input forms. | <input *id*="email" *type*="email" *class*="form-control @error('email') is-invalid @enderror" *name*="email" *value*="{{ old('email') }}" *placeholder*="Enter your email address" *required* *autofocus*> |
| admin.blade.php | Edu Lanka All Right Reseved | Edu Lanka All Right Reserved |
| developer.blade.php | Incorrect label definition and input placeholder field unavailability. |  |
| parent.blade.php | Incorrect label definition and input placeholder field unavailability. |  |
| student.blade.php | Incorrect label definition and input placeholder field unavailability. |  |
| teacher.blade.php | Incorrect label definition and input placeholder field unavailability. |  |
| All html Files. | Did not include the PWA icon pack. |  |

# Test cases

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Description | Pre-condition | Test Procedure | Test inputs | Expected output | Actual output | Result |
| TC001 | Admin login into the system using correct credentials (Email address and password). | The system should be up and running. | Admin should enter login credentials and click on the **“Continue”** button. | Email address = [admin@gmail.com](mailto:admin@gmail.com)  Password = 1234567890  Button click **“Continue”** | Directs admin to the dashboard. | Directed admin to the dashboard. | **Pass** |
| TC002 | Admin login into the system using null credentials (without inserting any values). | The system should be up and running. | Admin should leave the input fields null and click on the **“Continue”** button. | Email address =  Password =  Button click **“Continue”** | Prompts **“Please fill out this field”** message to the admin. | Prompted **“Please fill out this field”** message to the admin. | **Pass** |
| TC003 | Admin login into the system using an invalid email address. | The system should be up and running. | Admin should enter a wrong email address and the correct password. Next, he should click on the **“Continue”** button. | Email address = [admingmail.com](mailto:admin@gmail.com)  Password = 1234567890  Button click **“Continue”** | Prompts **“Please include an “@” in the email address”** message to the admin. | Prompted **“Please include an “@” in the email address”** message to the admin. | **Pass** |
| TC004 | Admin login into the system using an invalid password. | The system should be up and running. | Admin should enter the correct email address and a wrong password. Next, he should click on the **“Continue”** button. | Email address = [admin@gmail.com](mailto:admin@gmail.com)  Password = admin  Button click **“Continue”** | Prompts **“Invalid credentials. Please try again.”** message to the admin. | Prompts **“Invalid credentials. Please try again.”** message to the admin. | **Pass** |
| TC005 | Admin accessing the course materials. | Admin should be logged into the system. | Admin should log into the system and click on the **“Course”** section in the left side menu bar. | Click on **“Course”** hyperlink in the left menu bar. | Prompts admin with currently registered courses along with **Edit** and **Delete** icons and also displays a **“Add Course”** button as well. | Prompted admin with currently registered courses along with **edit** and **delete** icons and also displayed a “**add course”** button as well. | **Pass** |
| TC006 | Admin adding a course to the system. | Admin should be logged into the system, and he should be in the course details frame. | Admin should click on the **“Add Course”** button. After that he should fill out all form details and click on **“Save”** button. | Button click **“Add Course”**  Level = Grade 5  Subject = ICT  Button click **“Save”** | Should display the course adding form to the admin. After clicking on the **“save”** button, the admin should be redirected back to the course section. | Displayed the course adding form to the admin. After clicked on the “**save”** button, admin is redirected back to the course section. | **Pass** |
| TC007 | Admin editing details about a course. | Admin should be logged into the system, and he should be in the course details frame. | Admin should click on the **course edit icon.** After that he should fill out form details and click on **“Save Changes”** button. | Icon clicks **“Edit Course.”**  Level = Grade 5  Subject = English  Button click **“Save”** | Should display the course editing form to the admin. After clicking on the **“save changes”** button, the admin should be redirected back to the course section. | Displayed the course editing form to the admin. After clicked on the “**save changes”** button, the admin is redirected back to the course section. | **Pass** |
| TC008 | Admin removing a course from the system. | Admin should be logged into the system, and he should be in the course details frame. | Admin should click on the **course delete icon.** After that he should click on the **OK** button in the alert form. | Icon clicks **“Delete Course”**  Button click “**OK”** option in browser alert window. | Should prompt **“Are you sure you want to delete this course?”** message and when the admin clicks on the **OK** button, the system redirects the admin to the course details section. | Prompted **“Are you sure you want to delete this course?”** message and when the admin clicked on the **OK** button, the system redirected the Admin to the course details section. | **Pass** |
| TC009 | Admin registering a student into the system. | Admin should be logged into the system, and he should be in the student details frame. | Admin should click on the **Register Student** button. After that he should fill out form details and click on **“Save”** button. | Button click **“Register Student”**  First Name = Henry  Last Name = Hues  Email Address = henry@gmail.com  Birthday = 1996/12/08  Level = Grade 5  School = Gateway  Guardian Name = Jhon  Guardian Telephone Number = 0772562164  Guardian Occupation = Lawyer  Guardian Email = jhon@gmail.com  Button click **“Save”** | Should display the student registration form to the admin. After clicking on the **“save”** button, the admin should be prompted with **“Student added Successfully”** message. Next, he should be redirected to student frame. | Displayed the student registration form to the admin. After clicked on the **“save”** button, the admin is prompted with **“Student added Successfully”** message. Next, he is redirected to student frame. | **Pass** |
| TC10 | Admin editing details about a student. | Admin should be logged into the system, and he should be in the student details frame. | Admin should click on the **student edit icon.** After that he should fill out form details and click on **“Save Changes”** button. | Icon click **“Edit Student”**  First Name = Henry  Last Name = Hues  Email Address = henry@gmail.com  Birthday = 1996/12/08  Level = Grade 6  School = Gateway  Guardian Name = Jhon  Button click **“Save Changes”** | Should display the student editing form to the admin. After clicking on the **“save changes”** button, the admin should be redirected back to the student section. | Displayed the student editing form to the admin. After clicked on the **save changes** button, the admin is redirected back to the student section. | **Pass** |
| TC11 | Admin removing a student from the system. | Admin should be logged into the system, and he should be in the student details frame. | Admin should click on the **student delete icon**. After that he should click on the **OK** button in the alert form. | Icon click **“Delete Student”**  Button click “**OK”** option in browser alert window. | Should prompt **“Are you sure you want to delete this student?”** message and when the admin click on the OK button, the system redirects the Admin to the student details section. | Prompted **“Are you sure you want to delete this student?”** message and when the admin clicked on the **OK** button, the system redirected the admin to the student details section. | **Pass** |
| TC12 | Admin registering a teacher into the system. | Admin should be logged into the system, and he should be in the teacher details frame. | Admin should click on the **Register Teacher** button. After that he should fill out form details and click on **“Save”** button. | Button click **“Register Teacher”**  First Name = Adam  Last Name = Sandler  Email Address = adam@gmail.com  Level = Grade 12  School = CIS  Button click **“Save”** | Should display the teacher registration form to the admin. After clicking on the **“save”** button, the admin should be prompted with **“Teacher added Successfully”** message. Next, he should be directed to teacher frame. | Displayed the teacher registration form to the admin. After clicked on the **“save”** button, the admin is prompted with “**Teacher added Successfully”** message. Next, he is redirected to teacher frame. | **Pass** |
| TC13 | Admin editing details about a teacher. | Admin should be logged into the system and he should be in the teacher details frame. | Admin should click on the **teacher edit icon**. After that he should fill out form details and click on **“Update”** button. | Icon click **“Edit Teacher”**  First Name = Adam  Last Name = Sandler  Email Address = adam@gmail.com  Level = Grade 12  School = Cambridge  Button click **“Update”** | Should display the teacher editing form to the admin. After clicking on the **“save changes”** button, the admin should be redirected back to the teacher section. | Displayed the teacher editing form to the admin. After clicked on the **update** button, the admin is redirected back to the teacher section. | **Pass** |
| TC14 | Admin removing a teacher from the system. | Admin should be logged into the system and he should be in the teacher details frame. | Admin should click on the **teacher delete icon**. After that he should click on the **OK** button in the alert form. | Icon click **“Delete Teacher”**  Button click “**OK”** option in browser alert window. | Should prompt **“Are you sure you want to delete this teacher?”** message and when the admin click on the **OK** button, the system redirects the Admin to the teacher details section. | Prompted **“Are you sure you want to delete this teacher?”** message and when the admin clicked on the **OK** button, the system redirected Admin to the teacher details section. | **Pass** |
| TC15 | Admin editing his profile details. | Admin should be logged into the system. | Admin should log into the system and click on the **Settings** section in the left side menu bar. After that he should fill out form details and click on **“Save”** button. | Click on **“Settings”** hyperlink in the left menu bar.  Full Name = Jean Jacques  Email Address = jean@gmail.com  Button click **“Save”** | Should display the profile editing form to the admin. After clicking on the **“save”** button, the admin should be redirected back to the admin dashboard. | Displayed the profile editing form to the admin. After clicked on the **“save”** button, the admin is redirected back to the admin dashboard. | **Pass** |
| TC16 | Admin changing his password. | Admin should be logged into the system. | Admin should log into the system and click on the **Settings** section in the left side menu bar. After that he should fill out form details and click on **“Save”** button. | Click on **“Settings”** hyperlink in the left menu bar.  Old Password = 1234567890  New Password = admin@123  Confirm New Password = admin@123  Button click **“Save”** | Should display the profile editing form to the admin. After clicking on the **“save”** button, the admin should be redirected back to the admin dashboard. | Displayed the profile editing form to the admin. After clicked on the “save” button, the admin is redirected back to the admin dashboard. | **Pass** |
| TC17 | Student sending a message to admin. | Student should be logged into the system. | Student should log into the system and click on the **Contact Admin** button. Next he should fill out the input fields in the message form and click on the **“Send”** button. | Button click **“Contact Admin”**  Reason of Contact = Report a bug  Description = Cannot send messages  Button click **“Send”** | After clicking on the **Contact Admin** button. The system should prompt the message form. When the student clicks on **“send”** button, it should redirect the student to the dashboard. | Clicked on the **contact admin** button. Prompted message form. When the send button is clicked, the system redirected the student to the dashboard. | **Pass** |

# Workload matrix

|  |  |
| --- | --- |
| Name | Contribution |
| Ayantha | Wireframes(100%)  UI/ UX(50%)  Activity diagrams(33%)  Use case scenarios(33%) |
| Bashini | Documentation(100%)  Activity diagrams(33%)  Use case scenarios(33%)  Development(5%) |
| Kavinda | Quality Assurance (100%)  Activity diagrams(33%)  Use case scenarios(33%) |
| Maryam | Development(95%)  UI/ UX(50%) |

# Sprint Retrospective

The main goal of this sprint was to complete implementation of the admin dashboard and make the CRUD operations functional for the various components of the LMS.

## What Went Well

* Good collaboration between the team members helped foster a positive and productive environment.
* To ensure steady progress, every team member dedicated at least 5 hours per day to the sprint.
* Open and respectful communication helped problem-solving and decision making.
* Managed to effectively complete the targeted 13 user stories in the sprint backlog ahead of the planned schedule, showcasing the team’s efficiency.
* Each team member had a thorough understanding of their roles and responsibilities to which they committed and was also willing to lend a hand when necessary.
* Continuous engagement with the product owner helped reduce ambiguity and receive insightful feedback.
* Stand up meetings helped keep everyone updated about the progress of the sprint and address concerns
* Comprehensive testing done by QA helped in finding and resolving bugs and inconsistencies in the system.

## What Went Wrong

* Starting development before finalizing activity diagrams led to inconsistencies, requiring extra time for rectification.
* Some delays and adjustments were brought on by conflicts of interest among team members about UI designs.
* Generalization of users and foreign key implementation made the database design more complex than anticipated.
* Although there was some scope creep throughout the sprint, we were able to minimize its negative impact on the schedule.

## Action Plan

* Prior to initiation of development, thoroughly analyze and implement use case and activity diagrams to enhance clarity and reduce ambiguity.
* Improve communication between developer and UI/UX designer to ensure consistency between designs and development
* Thoroughly evaluate and estimate all user stories in sprint backlog before starting the sprint, to identify potential challenges and allocate resources effectively.
* Continue holding daily stand-up meetings to keep the team on the same page.
* Project manager should make sure that the team has a deeper understanding of the project requirements and sprint goals.