Introduction to Software Engineering

Requirements Analysis

The student team is required to complete the Software Requirements Specification (SRS) document for the assigned course project, following the attached template.

Software Engineering Department
Faculty of Information and Technology
University of Science

Table of Contents

Software Requirements Specification

Objectives

This document focusses on the following topics:

- ✓ Complete the Software Requirements Specification (SRS) document with the following contents:
  - ▪ Elaborate on the Problem Statement
  - ▪ Overview of Requirements (Functional and Non-Functional), Stakeholders
  - ▪ Use Case Model
  - ▪ Use Case Specifications
  - ▪ Create Prototype and Mock-up Diagrams of the System Interface
- ✓ Đọc hiểu tài liệu phân tích yêu cầu.

1 Member Contribution Assessment

ID	Name	Contribution (%)	Signature
23127069	Nguyen Minh Khoi	100	
23127359	Vo Tran Quoc Duy	100	
23127367	Ha Thu Hoang	100	
23127455	Truong Cong Thien Phu	100	
23127511	Vo Minh Tuan	100	

2 Problem Statement

English Tutoring Centre Management System

Business Problem

The tutoring centre currently faces several challenges in managing its day-to-day operations. The absence of an integrated digital solution necessitates significant manual effort, increases the likelihood of administrative oversights, and limits the centre's ability to leverage data for informed decision-making and proactive student support.

The core of the problem lies in several key areas. The process of creating, tracking, and updating student contracts, including details on fee payments and course durations, is handled manually, making it prone to errors. Academic tracking is fragmented, as monitoring student attendance, daily performance, and test results across various classes is cumbersome. This makes it difficult to gain a holistic view of student progress and to efficiently generate comprehensive report cards. Furthermore, the centre lacks a formalised system for staff coordination, making it difficult to track the working hours and specific tasks of staff and to manage teacher leave requests. There is also a lack of centralised class information, which complicates access to up-to-date data on both ongoing and completed classes, including enrolment and schedules. These issues create communication bottlenecks, particularly when conveying contract details to teachers or academic updates to administrative staff. Resource management, such as the tracking of classroom supplies, is also handled as an ad-hoc process.

Operating Environment

The proposed software solution is designed to function within a specific operating environment.

- **Client-Side:** The application will require a modern web browser that supports

HTML5 and contemporary web standards (e.g., Google Chrome, Firefox, etc.).

- **Server-Side:** The server-side of the application will be hosted on an Apache HTTP Server.
- **Database:** The system will connect to a MySQL database to store and manage all data.

Design and Implementation Constraints

The development of this solution will be governed by several key constraints.

- **Programming Language:** The front-end will utilise HTML, CSS, and JavaScript with the ReactJS framework, while the back-end will be developed using Python.
- **User Interface (UI):** The UI must be "simple", intuitive, and user-friendly to cater to users with varying levels of technical proficiency.
- **Security:** The system will feature robust role-based access control to ensure data confidentiality and integrity, and only authorised users will be able to access specific modules and data.
- **Scalability:** The system must be designed to accommodate future growth in the number of students, staff, and classes over time.
- **Data Integrity:** Mechanisms will be implemented to ensure the accuracy, consistency, and reliability of all data, particularly student academic records and contract information.
- **Documentation:** The project will produce standard software development lifecycle documentation, which will include user manuals.

3 Requirements Overview

3.1 Stakeholders

STT	Stakeholder	Description
1	Centre Manager	Provides the financial resources and overall strategic direction for the project. This individual is the primary decision-maker, responsible for approving the budget, defining high-level business goals (like viewing revenue and enrolment). The Manager is also a key user who will manage staff accounts and permissions.
2	Teachers	Constitute the principal user group, engaging with the system for daily academic administration and operational tasks. Their primary requirement is for a system that enhances operational efficiency by streamlining workflows for student evaluation (grade and comment input), attendance tracking, leave request management, and issue reporting.
3	Learning Advisors (Administrative Staff)	Core operational users responsible for executing the centre's key administrative functions. The system is expected to automate and streamline the administrative lifecycle, including class and contract management, report card generation and exportation, and technical system maintenance duties such as data backup.
4	Students	Represent the primary data subjects of the system, whose academic lifecycle and performance metrics are systematically managed. As indirect stakeholders, their interest is vested in the integrity and accuracy of their data, which directly impacts

		the performance evaluations they receive via system-generated report cards.
5	Parents/Guardians	Constitute the key external beneficiaries of the system's reporting capabilities. Their primary interest is in leveraging the system as an enhanced communication channel for receiving timely, transparent, and comprehensive reports on their child's academic performance.
6	Development Team	The executing body responsible for the end-to-end software development lifecycle, from requirements analysis through to deployment and maintenance planning. Their fundamental responsibility is the successful delivery of a high-quality software solution that adheres strictly to the technical specifications and functional requirements outlined in the project proposal, thereby satisfying the objectives of all other stakeholders.

3.2 Requirements

3.2.1. Functional Requirements Specification

- Manager

Requirement ID	Requirement Description
FR-MGR-01	The system must allow the Manager to view centre-wide statistics. This includes viewing the total number of student enrolments and centre revenue.
FR-MGR-02	The system must provide the Manager with the ability to manage user permissions.
FR-MGR-03	The system must allow the Manager to create new accounts for staff members.

- Teacher

Requirement ID	Requirement Description
FR-TEA-01	The system must allow teachers to check their own attendance online. This is part of the account check-in function.
FR-TEA-02	The system must allow teachers to check their students' attendance online.

FR-TEA-03	The system must allow teachers to input students' grades and provide written comments for quizzes and tests.
FR-TEA-04	The system must allow teachers to submit leave requests online. This includes selecting dates, reasons, and a substitute teacher.
FR-TEA-05	The system must provide teachers with the ability to report class issues, including student behavioural or technical issues.
FR-TEA-06	The system must allow teachers to view their colleagues' schedules to find available substitutes.

- Learning Advisor (Staff)

Requirement ID	Requirement Description
FR-STF-01	The system must allow staff to create classes and add students and teachers to those classes.
FR-STF-02	The system must allow staff to view class statistics, including the number of students, class progress, and course duration.
FR-STF-03	The system must provide a contract generator for staff to create contracts online, including details like course roadmaps and tuition fees.
FR-STF-04	The system must allow staff to generate report cards using the grades and comments entered by teachers.
FR-STF-05	The system must provide the option to export generated report cards to PDF and Excel formats.
FR-STF-06	The system must allow staff to view the issues reported by teachers.

- Admin

Requirement ID	Requirement Description
FR-ADM-01	The system must provide functions for the Admin to perform system maintenance.
FR-ADM-02	The system must allow the Admin to manage and monitor system performance.
FR-ADM-03	The system must have features for the Admin to handle and view system errors.

FR-ADM-04	The system must allow the Admin to perform data backups.
FR-ADM-05	The system must allow an Admin to add a student to a make-up class. This may be an automated function triggered when a student is marked absent.

- System

Requirement ID	Requirement Description
FR-SYS-01	The system must automatically send various notifications. This includes notifications for attendance checks, class times, deadlines, and staff check-ins.
FR-SYS-02	The system must provide a tutorial to guide beginner-friendly usage.

- Account (General)

Requirement ID	Requirement Description
FR-ACC-01	The system must allow any user with an account to sign in using their credentials to gain authenticated access.
FR-ACC-02	The system must allow staff members (Teachers and Staff) to perform a daily check-in to record their work attendance after signing in.

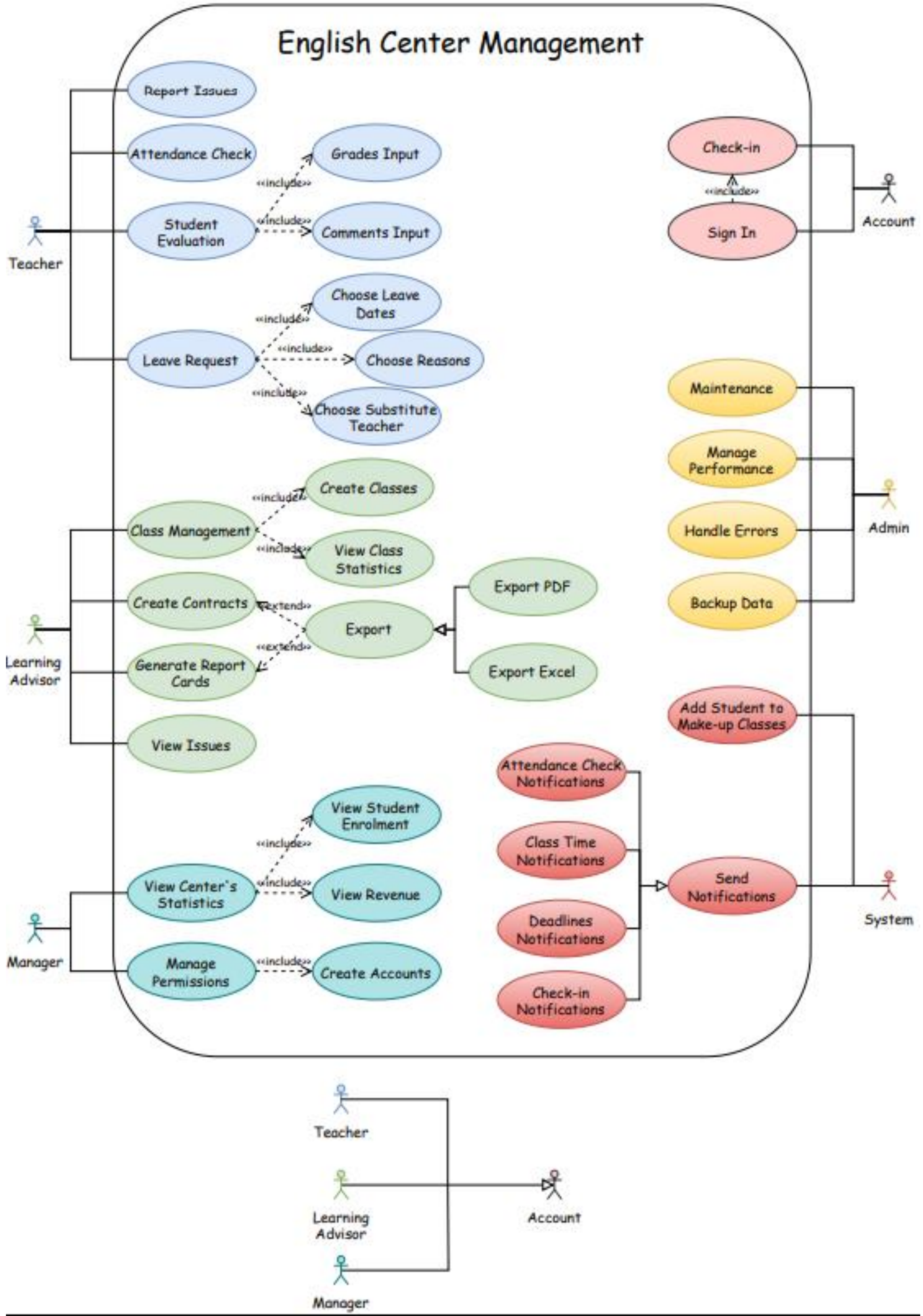
3.2.2. Non-Functional Requirements Specification

Requirement ID	Category	Requirement Description
NFR-USA-01	**Usability**	The User Interface (UI) must be "simple", intuitive, and user-friendly.
NFR-USA-02	**Usability**	The system must cater to users with varying levels of technical proficiency.
NFR-PRF-01	**Performance**	The system should be designed to accommodate growth in the number of students, staff, and classes over time.
NFR-SEC-01	**Security**	The system must implement robust role-based access control.

NFR-SEC-02	Security	Access control is critical to ensure data confidentiality and integrity.
NFR-SEC-03	Security	Only authorised users should be able to access specific modules and data.
NFR-REL-01	Reliability & Data Integrity	The system must include mechanisms to ensure the accuracy, consistency, and reliability of all data.
NFR-REL-02	Reliability & Data Integrity	Data integrity mechanisms must particularly apply to student academic records and contract information.
NFR-CMP-01	Compatibility (Operating Environment)	The client-side of the application must be accessible on a modern web browser with support for HTML5.
NFR-CMP-02	Compatibility (Operating Environment)	The server-side will be hosted on an Apache HTTP Server.
NFR-CMP-03	Compatibility (Operating Environment)	The system will use a MySQL database.
NFR-MNT-01	Maintainability (Documentation)	Standard software development lifecycle documentation will be required.
NFR-MNT-02	Maintainability (Documentation)	Required documentation will include user manuals.

4 Requirements Analysis

4.1 Use Case model


English Center Management
Report Issues
Attendance Check
Student Evaluation
Grades Input
Comments Input
«include»
«include»
Teacher
Choose Leave Dates
Leave Request
Choose Reasons
Choose Substitute Teacher
«include»
«include»
«include»
Create Classes
Class Management
View Class Statistics
«include»
«include»
Create Contracts
Export
Generate Report Cards
«extend»
«extend»
Export PDF
Export Excel
View Issues
Learning Advisor
View Student Enrolment
View Center's Statistics
View Revenue
«include»
«include»
Manage Permissions
Create Accounts
«include»
Manager
Check-in
Sign In
«include»
Account
Maintenance
Manage Performance
Handle Errors
Backup Data
Admin
Add Student to Make-up Classes
Attendance Check Notifications
Class Time Notifications
Deadlines Notifications
Check-in Notifications
Send Notifications
System
Teacher
Learning Advisor
Manager
Account

4.2 Use Case Specification

4.2.1. Use Case 1

Use case ID	U001
Use Case	Report Issues
Brief Description	Allows a teacher to report classroom issues, such as student behavioural or technical problems, to the appropriate administrative staff.
Actor	Teacher
Pre-Condition	The Teacher must be authenticated and logged into the system.
Result	An issue report is successfully submitted and logged in the system for a Learning Advisor to review.
Main Scenario	1. The Teacher selects the "Report Issues" option in the system. 2. The system presents a form to create a new report. 3. The Teacher selects the issue type (e.g., "Behavioural", "Technical") and provides a detailed description. 4. The Teacher submits the form. 5. The system saves the report and displays a confirmation message.
Alternative Scenarios	**User Cancels:** The Teacher can choose to cancel the report at any point before submission. The system discards the data and returns to the previous screen.
Non-Functional Constraints	**Usability:** The form for reporting issues must be simple and quick to complete. **Security:** Only authenticated users with a "Teacher" role can submit issue reports.

4.2.2. Use Case 2

Use case ID	U002
Use Case	Attendance Check
Brief Description	Enables a teacher to take and record student attendance for a specific class session online.
Actor	Teacher
Pre-Condition	The Teacher must be logged into the system. A class session

	must be scheduled and have students enrolled.
Result	The attendance status (e.g., Present, Absent, Late) for each student in the selected class session is saved in the system database.
Main Scenario	1. The Teacher navigates to the "Attendance" section and selects a class and session date. 2. The system displays a list of enrolled students for that session. 3. The Teacher marks the attendance status for each student. 4. The Teacher clicks "Save Attendance”. 5. The system persists the attendance records and displays a success message.
Alternative Scenarios	**Save Incomplete List:** If the Teacher saves before marking all students, the system saves the entered data and marks the remaining students as a default status (e.g., "NaN"), possibly with a warning message.
Non-Functional Constraints	**Performance:** The class list should load within 3 seconds to ensure efficiency. **Data Integrity:** The system must ensure attendance records are accurately linked to the correct student, class, and date.

4.2.3. Use Case 3

Use case ID	U003
Use Case	Student Evaluation
Brief Description	Allows a Teacher to input grades and provide written comments on a student's performance on quizzes, tests, or other assignments.
Actor	Teacher
Pre-Condition	The Teacher must be logged into the system. The student being evaluated must be enrolled in one of the Teacher's classes.
Result	The student's academic record is updated with a new grade and/or comment for a specific assessment.
Main Scenario	1. The Teacher selects the "Student Evaluation" or "Gradebook" feature. 2. They choose a class, a student, and a specific assessment item. 3. The system displays fields for the evaluation. 4. <<include>> The Teacher inputs the student's grade ("Grades Input").

	5. <<include>> The Teacher writes relevant feedback or observations ("Comments Input"). 6. The Teacher saves the evaluation. 7. The system confirms that the record has been successfully saved.
Alternative Scenarios	**Invalid Grade Input:** If the Teacher enters a grade in an invalid format (e.g., text in a numeric field) or outside the valid range, the system displays an error message and prevents saving until the data is corrected.
Non-Functional Constraints	**Data Integrity:** The system must ensure that grades and comments are accurately stored and associated with the correct student and assessment item. **Scalability:** The system must handle a growing number of student evaluation records over time.

4.2.4. Use Case 4

Use case ID	U004
Use Case	Leave Request
Brief Description	This use case allows a Teacher to submit an online leave request for a day off.
Actor	Teacher
Pre-Condition	The Teacher must be authenticated and logged into the system.
Result	A leave request is successfully submitted to the system with a "Pending Approval" status, and a notification is sent to the Centre Manager.
Main Scenario	1. The Teacher selects the "Leave Request" option. 2. They initiate a new request. 3. The system presents a form. 4. <<include>> The Teacher **chooses the leave dates** (start and end). 5. <<include>> The Teacher **chooses the reasons** for the absence from a predefined list or enters a custom reason. 6. The Teacher views the schedules of available colleagues. 7. <<include>> The Teacher **chooses a substitute teacher** from the list of available staff. 8. The Teacher submits the form. 9. The system validates the request, saves it, and displays a confirmation message.

Alternative Scenarios	**No Substitute Available:** If no substitute teachers are available for the selected dates, the system displays a warning but may still allow the Teacher to submit the request, flagging it for manager review. Scheduling Conflict: If the chosen substitute already has a conflicting schedule, the system displays an error and prompts the Teacher to select another substitute.
Non-Functional Constraints	**Usability:** The process of requesting leave, from selecting dates to choosing a substitute, must be a seamless and user-friendly experience.

4.2.5. Use Case 5

Use case ID	U005
Use Case	Class Management
Brief Description	Allows a Learning Advisor to create new classes, assign teachers and students, and view statistics for existing classes
Actor	Learning Advisor
Pre-Condition	The Learning Advisor must be authenticated and logged into the system. Teacher and student profiles must exist in the system to be assigned.
Result	A new class is successfully created and stored in the system, or statistics for a selected class are displayed.
Main Scenario	1. The Learning Advisor navigates to the "Class Management" section. 2. The system displays options to create a new class or view existing classes. 3. <<include>> To create a class, the user selects "Create Classes". They enter class details (name, subject, schedule), assign a teacher, add students from a list, and save the new class. 4. <<include>> To view statistics, the user selects an existing class, triggering "View Class Statistics". The system displays key metrics like the number of enrolled students and class progress
Alternative Scenarios	**Scheduling Conflict:** If the Learning Advisor tries to assign a teacher or student to a new class at a time that conflicts with their existing schedule, the system will display a warning and prevent the assignment until resolved.
Non-Functional Constraints	**Usability:** The interface for creating classes and adding students must be intuitive and minimise clicks.

	Data Integrity: The system must ensure class data, schedules, and enrolment lists are accurate

4.2.6. Use Case 6

Use case ID	U006
Use Case	Create Contracts
Brief Description	Enables a Learning Advisor to create and save new student contracts online, detailing course information and tuition fees.
Actor	Learning Advisor
Pre-Condition	The Learning Advisor must be authenticated and logged into the system. The student's profile must exist in the system.
Result	A new student contract is created with a unique ID and saved in the system database.
Main Scenario	1. The Learning Advisor selects the "Create Contracts" option. 2. The system displays a new contract form. 3. The Learning Advisor selects a student and enters all required contract details, including course roadmaps, tuition fees, and contract duration. 4. The Learning Advisor saves the contract. 5. The system validates the data, creates the contract record, and displays a success confirmation message.
Alternative Scenarios	**Missing Information:** If required fields (e.g., student name, fees) are not filled in, the system displays an error message highlighting the missing fields and prevents saving. **Duplicate Contract:** If an active contract for the same student and course already exists, the system will show a warning and ask for confirmation before proceeding.
Non-Functional Constraints	**Security:** Access to contract creation and financial details must be restricted to authorised administrative staff (Learning Advisors, Managers). **Data Integrity:** Financial figures and course details must be stored with high accuracy

4.2.7. Use Case 7

Use case ID	U007

Use Case	Generate Report Cards
Brief Description	Allows the Learning Advisor to generate student report cards based on academic data. The user can optionally export the generated report
Actor	Learning Advisor
Pre-Condition	The Learning Advisor is logged into the system. Student evaluation data (grades, comments) has been entered by teachers.
Result	A report card for a selected student or class is generated and displayed on screen, and optionally downloaded as a PDF or Excel file.
Main Scenario	1. The Learning Advisor navigates to the "Report Card Generation" section. 2. The system provides options to generate reports for a specific class or an individual student. 3. The Learning Advisor makes a selection and clicks "Generate". 4. The system retrieves all relevant student data and displays the formatted report card.
Alternative Scenarios	**Export Report (<<extend>>):** 4a. After the report is displayed, the Learning Advisor can choose to trigger the "Export" use case. 4b. The system presents two options: "Export PDF" or "Export Excel". 4c. The Learning Advisor selects a format, and the system generates the file and initiates a download. **No Data Available:** If a student has no evaluation data, the system will display a message: "No data available to generate a report." and the process ends.
Non-Functional Constraints	**Performance:** Generating a report for a full class should take no longer than 15 seconds. **Scalability:** The system must be able to handle generating reports for a growing number of students

4.2.8. Use Case 8

Use case ID	U008
Use Case	View Issues
Brief Description	Provides a centralised interface for the Learning Advisor to view and manage classroom issues reported by teachers

Actor	Learning Advisor
Pre-Condition	The Learning Advisor must be authenticated and logged into the system.
Result	The Learning Advisor can view a comprehensive list of all reported issues and their details.
Main Scenario	1. The Learning Advisor navigates to the "Issues Dashboard". 2. The system displays a list or table of all reported issues, showing key information like issue date, teacher, type, and status (e.g., Open, Resolved). 3. The Learning Advisor can filter or sort the list. 4. The Learning Advisor clicks on a specific issue to view its full description and any attached notes.
Alternative Scenarios	**No Issues:** If no issues have been reported, the system displays a message, such as "There are no issues to display."
Non-Functional Constraints	**Usability:** The issue list must be easy to read, sort, and filter to allow for efficient management. **Security:** This view is restricted to administrative staff to maintain the confidentiality of reported issues.

4.2.9. Use Case 9

Use case ID	U009
Use Case	View Centre’s Statistics
Brief Description	This use case allows the Manager to view a high-level dashboard of the centre's key performance indicators
Actor	Manager
Pre-Condition	The Manager must be authenticated and logged into the system.
Result	An up-to-date dashboard of the centre's key statistics is displayed successfully to the Manager
Main Scenario	1. The Manager logs into the system and navigates to the "Dashboard" or "Statistics" section. 2. The system retrieves and presents a dashboard with various statistical summaries. 3. <<include>> The system displays the total student enrolment count ("View Student Enrolment"). 4. <<include>> The system displays key centre revenue figures for specified time periods ("View Revenue"). 5. The Manager can filter the statistics by date range or other

	parameters.
Alternative Scenarios	**Data Load Error:** In the event the system cannot retrieve the data from the database, a user-friendly error message is displayed, such as "Could not load statistics at this time. Please try again later."
Non-Functional Constraints	**Security:** Access to this use case and its financial data must be strictly restricted to users with the "Manager" role. **Performance:** The dashboard and all statistical data should load in under 5 seconds. **Data Integrity:** All financial and enrolment figures must be accurate and up-to-date.

4.2.10. Use Case 10

Use case ID	U010
Use Case	Manage Permissions
Brief Description	Enables the Manager to manage user roles and system permissions, which includes the creation of new staff accounts
Actor	Manager
Pre-Condition	The Manager must be authenticated and logged into the system with top-level administrative privileges.
Result	A user's role is successfully updated, or a new user account is created with an assigned role.
Main Scenario	1. The Manager navigates to the "User Management" or "Permissions" section. 2. To edit an existing user, they select the user from a list. 3. The Manager modifies the user's role (e.g., "Teacher", "Learning Advisor") or specific permissions and saves the changes. 4. <<include>> To create a new account, the Manager selects the "Create Accounts" option. They enter the new user's details, assign a role, and save. 5. The system confirms the action was successful.
Alternative Scenarios	**Username Exists:** When creating a new account, if the chosen username already exists, the system displays an error and prompts for a different username. **Self-Permission Lockout Prevention:** The system should prevent the Manager from removing their own administrative permissions in a way that would lock them out of this function.

Non-Functional Constraints	**Security:** This is a highly sensitive function. All permission changes must be logged for auditing purposes. Access is strictly limited to the Manager role. **Usability:** The interface for managing roles and permissions must be clear to prevent accidental misconfiguration.

4.2.11. Use Case 11

Use case ID	U011
Use Case	Sign In
Brief Description	This use case allows an existing user (Account) to provide their credentials to gain authenticated access to the system.
Actor	Account (Teacher, Learning Advisor, Manager)
Pre-Condition	The user must have an active account that was previously created by a Manager. The user is currently not authenticated.
Result	The user is successfully authenticated, a session is established, and they are redirected to their role-specific dashboard.
Main Scenario	1. The user navigates to the application's login page. 2. The system displays fields for a username and password. 3. The user enters their credentials and clicks the "Sign In" button. 4. The system validates the credentials against the user database. 5. Upon successful validation, the system grants access and directs the user to their main dashboard.
Alternative Scenarios	**Invalid Credentials:** If the username or password does not match a record in the database, the system displays an "Invalid username or password" error message and denies access. **Account Locked:** If a user enters incorrect credentials multiple times (e.g., 5 attempts), the system may temporarily lock the account for a short period and display a corresponding message for security purposes.
Non-Functional Constraints	**Security:** User passwords must be stored in a securely hashed format. The system must use secure methods for session management. **Performance:** The authentication process should complete within 2 seconds.

4.2.12. Use Case 12

Use case ID	U012
Use Case	Check-in
Brief Description	Allows a staff member (Teacher or Staff) to check in for their workday, creating a digital record of their attendance.
Actor	Account (Teacher, Learning Advisor, Manager)
Pre-Condition	The user must not have already checked in for the current workday.
Result	A timestamped check-in record is successfully created for the user for the current day. The system may send a notification confirming the check-in.
Main Scenario	1. <<include>> The user first performs the "Sign In" use case to gain access to the system. 2. Once logged in, the user navigates to their dashboard or a dedicated staff attendance section. 3. The user clicks the "Check-in" button. 4. The system records the current date and time as the user's official check-in time. 5. The system displays a confirmation message on the screen, such as "You have successfully checked in at [time]."
Alternative Scenarios	**Already Checked In:** If the user has already checked in for the day, the "Check-in" button is disabled or replaced with a message indicating the time they checked in (e.g., "Checked-in at 08:30 AM"). **System Time Error:** If the system is unable to retrieve a reliable timestamp from the server, it displays an error and prevents the check-in from completing.
Non-Functional Constraints	**Data Integrity:** Timestamps for check-ins must be accurate, generated by the server, and stored securely to prevent tampering. **Availability:** The check-in function must be highly available, especially during the start of normal business hours.

4.2.13. Use Case 13

Use case ID	U013
Use Case	Maintenance
Brief Description	This use case allows the Admin to perform routine system maintenance tasks, such as applying software patches, updating dependencies, or clearing temporary files to ensure system

	health.
Actor	Admin
Pre-Condition	The Admin must be authenticated with the highest-level system privileges.
Result	A specified maintenance task is completed successfully, and the system's operational status is verified.
Main Scenario	1. The Admin logs into the system's administrative console. 2. The Admin navigates to the "System Maintenance" section. 3. They select and initiate a maintenance task (e.g., "Apply Security Patches"). 4. The system executes the task, providing real-time log output. 5. Upon completion, the system displays a success message with a summary of the actions taken.
Alternative Scenarios	**Maintenance Task Fails:** If a task fails, the system immediately halts the process, attempts to roll back any changes to maintain stability, logs the critical error, and sends an alert to the Admin.
Non-Functional Constraints	**Availability:** Major maintenance tasks that may cause service interruption should be scheduled during off-peak hours to minimise impact on users. **Security:** This function must be protected by strong authentication and is accessible only to the Admin role.

4.2.14. Use Case 14

Use case ID	U014
Use Case	Manage Performance
Brief Description	Enables the Admin to monitor key system performance metrics and make necessary adjustments to optimise responsiveness and stability.
Actor	Admin
Pre-Condition	The Admin must be authenticated and logged into the system.
Result	The Admin has a clear, real-time view of system performance and has taken action to resolve any identified bottlenecks.
Main Scenario	1. The Admin logs into the admin panel and accesses the "Performance Monitoring" dashboard. 2. The system displays real-time metrics, such as server CPU

	and memory usage, database query times, and API response times. 3. The Admin identifies a performance issue (e.g., a slow database query). 4. The Admin uses system tools to analyse and optimise the identified issue.
Alternative Scenarios	**Monitoring Service Unavailable:** If the performance monitoring service fails to load, the system displays an error message, and a notification is sent to the Admin's configured alert channel.
Non-Functional Constraints	**Performance:** The monitoring dashboard itself must be lightweight and have a minimal impact on the overall system's performance. **Usability:** Performance data should be presented in clear, understandable graphs and tables.

4.2.15. Use Case 15

Use case ID	U015
Use Case	Handle Errors
Brief Description	Allows the Admin to view, diagnose, and manage system-level errors that have been automatically logged by the application.
Actor	Admin
Pre-Condition	The Admin must be authenticated. The system has previously logged one or more errors.
Result	System errors are reviewed, their root causes are investigated, and their status is updated (e.g., Acknowledged, In-Progress, Resolved).
Main Scenario	1. The Admin navigates to the "Error Log" section in the admin panel. 2. The system displays a filterable list of logged errors, sorted by date. 3. The Admin selects an error to view its detailed information (e.g., stack trace, user context, request data). 4. After investigating, the Admin updates the error's status in the log and, if necessary, creates a task for the development team.
Alternative Scenarios	**No Errors:** If the error log is empty, the system displays a "No new errors reported" message.

Non-Functional Constraints	**Usability:** Error logs must be easily searchable and filterable by severity, date, and error type to allow for efficient diagnostics.

4.2.16. Use Case 16

Use case ID	U016
Use Case	Backup Data
Brief Description	This use case allows the Admin to manually initiate a complete backup of the application database to prevent data loss.
Actor	Admin
Pre-Condition	The Admin must be authenticated with backup privileges. There must be sufficient storage space in the target backup location.
Result	A complete, consistent, and secure backup of the database is created and stored in a predefined location.
Main Scenario	1. The Admin navigates to the "Backup & Restore" section. 2. They select the "Create New Backup" option. 3. The system temporarily puts the database in a read-only state (if necessary) to ensure consistency. 4. The system performs the data dump, compresses the file, and stores it in the designated secure location. 5. The system confirms the successful completion of the backup with details like file size and location.
Alternative Scenarios	**Backup Fails:** If the backup process fails (e.g., due to insufficient disk space or database errors), the system immediately logs the failure, cleans up any partial backup files, and sends a critical alert to the Admin.
Non-Functional Constraints	**Data Integrity:** Backups must be complete and validated to ensure they are restorable. **Security:** Backup files should be encrypted and stored in a secure, geographically separate location if possible.

4.2.17. Use Case 17

Use case ID	U017
Use Case	Send Notifications

Brief Description	This is a generalized use case where the system automatically sends various types of event-driven or scheduled notifications to relevant users.
Actor	System
Pre-Condition	The system is operational. Target users have contact information registered in the system. A trigger event has occurred.
Result	A notification is successfully generated and delivered to the target user's specified channel. The action is logged to ensure a reliable information flow.
Main Scenario	1. A trigger event occurs (e.g., the time is 15 minutes before a scheduled class). 2. The system identifies the type of notification (Attendance Check, Class Time, Deadlines, Check-in) to send and the target user(s). 3. The system formats the notification message with relevant information. 4. The system sends the notification. 5. The system logs the sent notification for auditing purposes.
Alternative Scenarios	**Notification Failure:** If the notification fails to send (e.g., due to a service outage), the system logs the failure and may retry a set number of times before flagging the issue for an Admin.
Non-Functional Constraints	**Reliability:** Notifications must be sent reliably to ensure a smooth information flow. **Timeliness:** Time-sensitive notifications must be sent promptly at the scheduled time.

4.2.18. Use Case 18

Use case ID	**U018**
Use Case	Automatically Add Student to Make-up Classes
Brief Description	This use case automatically finds and assigns a student to a suitable make-up class immediately after a teacher marks them as absent for a session.
Actor	System
Pre-Condition	A Teacher has submitted an attendance record with a student marked "Absent". A suitable make-up class (matching subject and with available capacity) exists in the schedule.
Result	The student is automatically enrolled in the make-up class. A notification is sent to relevant parties, and the student's

	attendance record is updated to "Make-up Scheduled".
Main Scenario	1. A Teacher marks a student as "Absent" and saves the attendance record. 2. This action triggers the system to search for an appropriate make-up class for the student. 3. The system identifies a suitable make-up class from the schedule based on subject, level, and availability. 4. The system automatically enrols the student in the identified make-up class. 5. The system updates the student's schedule and the original attendance record. 6. The system sends notifications to the student/parent and the relevant teachers.
Alternative Scenarios	**No Suitable Classes:** If the system cannot find any suitable make-up classes, it will flag the student's absence for manual review by a Learning Advisor or Admin and will not perform an automatic assignment. **Multiple Suitable Classes:** If multiple suitable classes exist, the system will use a predefined rule (e.g., assign to the earliest available class) to make a selection.
Non-Functional Constraints	**Data Integrity:** The system must ensure that student enrolment records are accurately updated in both the original and the make-up class to prevent data conflicts. **Reliability:** The automated assignment process must be highly reliable to ensure no students are missed. **Security:** The process must be an internal, secure system function that cannot be triggered externally except by an authorized attendance submission.

5 Prototype/Mockup

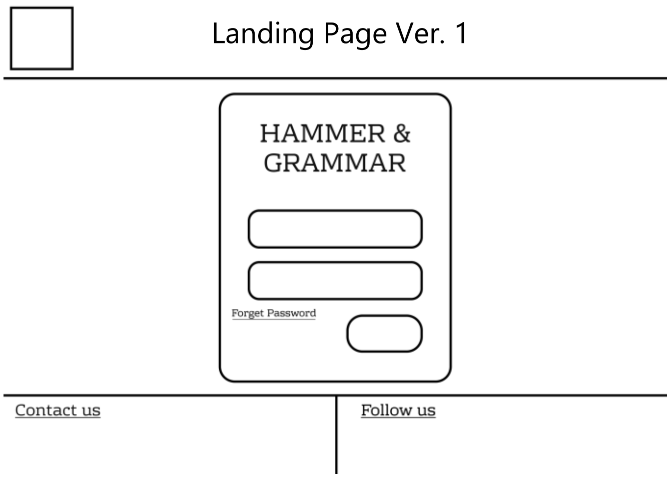
This section presents the visual design plan for the English Tutoring Centre Management System. It includes low-fidelity wireframe diagrams that outline the application's structure and layout, as well as higher-fidelity prototypes that demonstrate key user workflows. The goal of this phase is to translate the functional requirements into a tangible user interface design, which can be reviewed and validated by stakeholders before development commences.

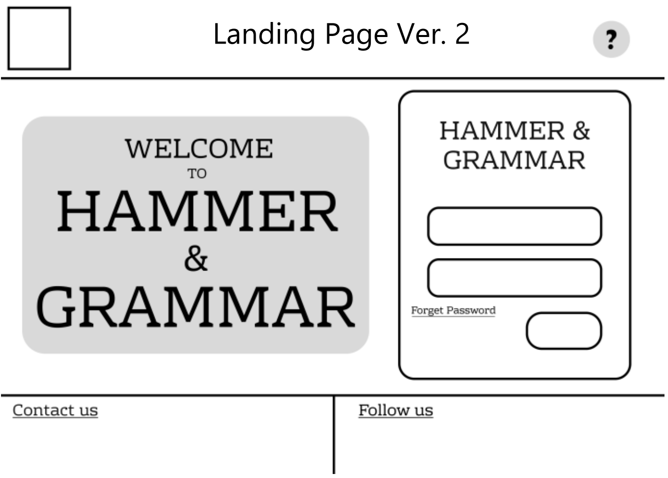
Tool Selection

For this project, our team will be using **Figma** to create all wireframes, mock-ups, and interactive prototypes. Figma has been chosen for its robust collaborative features, which allow our team members to work on designs simultaneously, and its powerful tools for creating component-based design systems and interactive user flow simulations.

Wireframe

Landing & Login Page: These wireframes illustrate two proposed designs for the application's primary landing page. The main purpose of this page is to serve as the main entry point for the "HAMMER & GRAMMAR" tutoring centre's management system, providing a secure way for authorised users to access their accounts. Both designs directly support the "Sign In" functional requirement (FR-ACC-01).





Version 1

This version features a minimalist and function-focused design.

- **Layout:** The page is structured with a simple header, a central content area, and a two-column footer.
- **Header:** Contains a placeholder box in the top-left corner, intended for the Centre’s logo.
- **Login Form:** The central focus of the page is a clearly defined login form.
  - It is titled "HAMMER & GRAMMAR".
  - It includes two rounded input fields, designated for the user's username and password.
  - A "Forget Password" link is situated directly below the input fields.
  - A rounded rectangular button is provided for submitting the credentials to log in.
- **Footer:** The footer is split into two sections, with a "Contact us" link on the left and a "Follow us" link on the right.

This design prioritises immediate access to the login functionality for returning staff members.

Version 2

This version introduces more visual elements to create a more welcoming experience.

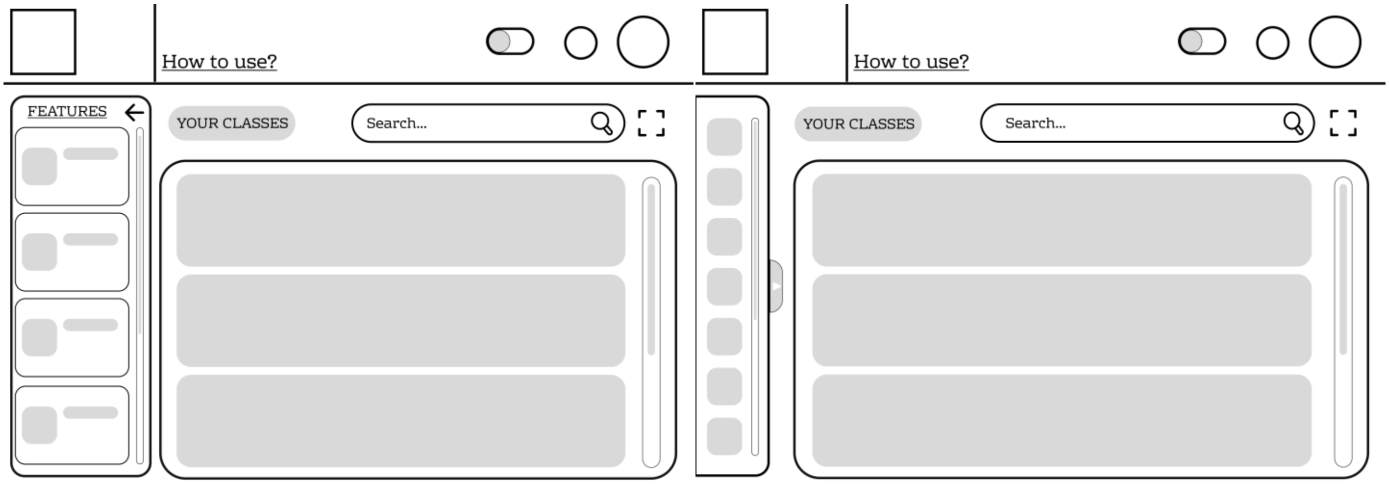
- **Layout:** This design modifies the central content area into a two-column layout.
- **Header:** The header maintains the logo placeholder in the top-left and adds a help icon, represented by a question mark inside a circle, to the top-right corner.
- **Left Column:** This area is dedicated to a large, prominent welcome graphic with

the text "WELCOME TO HAMMER & GRAMMAR". This serves as a strong branding element.

- **Right Column:** This column contains the same functional login form as described in Version 1, with two input fields, a "Forget Password" link, and a submission button.
- **Footer:** The footer structure remains consistent with Version 1, providing "Contact us" and "Follow us" links.

This alternative design aims to be more engaging for all visitors, clearly separating the welcome message from the login function.

Home screen / Dashboard Wireframes: These wireframes illustrate the primary user dashboard or "Home screen" that users see after logging in. The design is clean and modern, providing access to core functionalities via a collapsible sidebar and a focused main content area. This screen serves as the central hub for users like Teachers and Learning Advisors to access their classes and other features.



Both versions of the wireframe share several common elements:

- **Header:** The application features a consistent header at the top of the page.
  - It includes a square placeholder in the top-left for the Centre’s logo.
  - A "How to use?" link is present to provide users with guidance and support.
  - On the right side, there is a set of UI controls, including a toggle switch (likely for a light/dark mode), and several circular icons representing user settings and the user's profile avatar.
- **Main Content Area:** This area is dedicated to displaying the user's primary

information.

- It is clearly titled "YOUR CLASSES".
- A prominent search bar with a magnifying glass icon allows users to quickly find specific classes.
- A full-screen toggle icon is available next to the search bar.
- The main view consists of a scrollable list of large, rounded panels, each representing a single class.

The two wireframes demonstrate the two states of the collapsible navigation sidebar.

Home screen with Fully Extended Sidebar

This view shows the sidebar in its default, expanded state, providing clear and descriptive navigation.

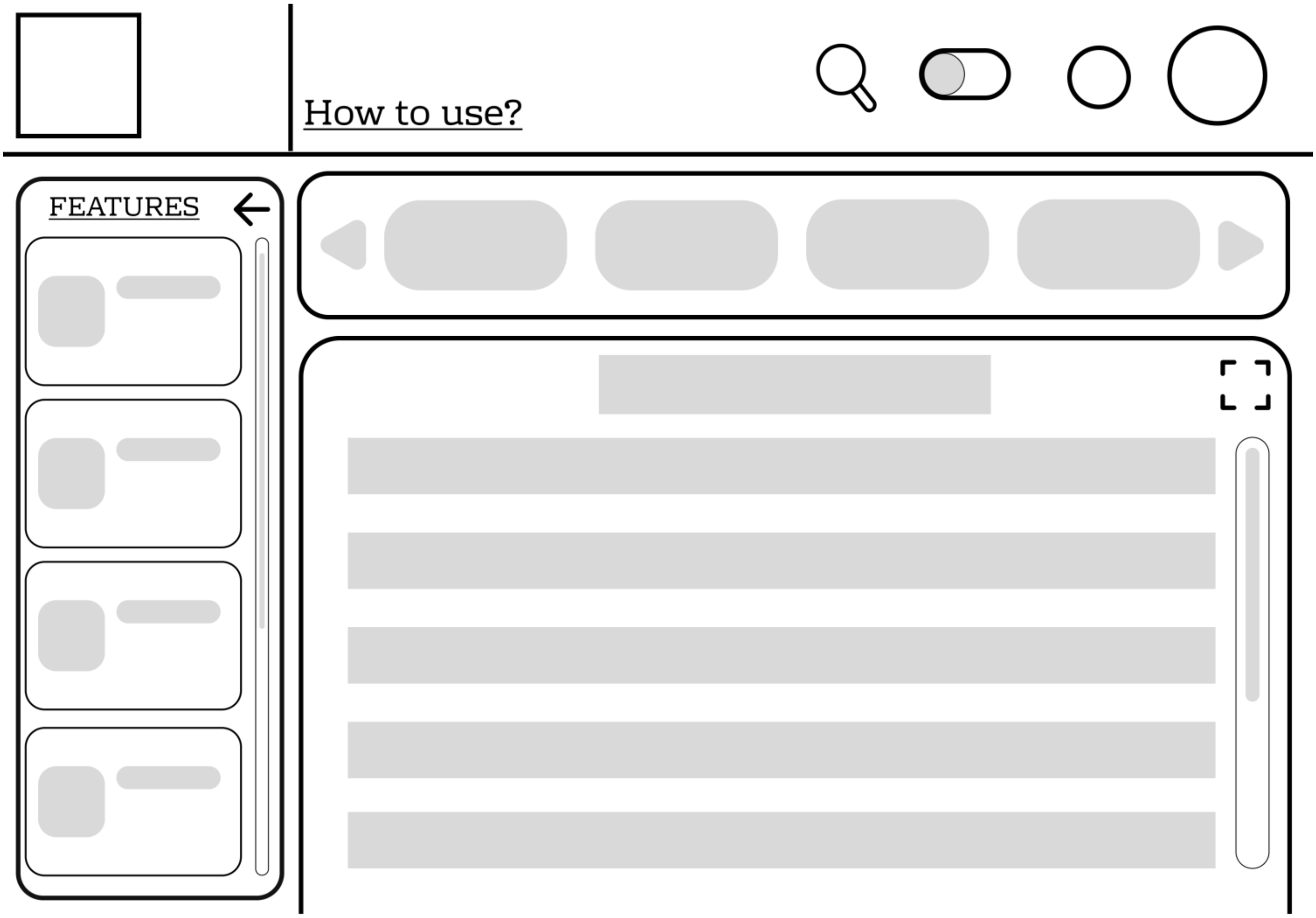
- **Sidebar Header:** The sidebar is titled "FEATURES" and includes a back-arrow icon (←) that allows the user to collapse it into the mini version.
- **Navigation Items:** It displays a vertical list of navigation links. Each link is composed of an icon placeholder on the left and a text label on the right, ensuring clarity for the user.

Home screen with Mini Sidebar

This view shows the sidebar in its collapsed, icon-only state, which maximises the screen real estate for the main content.

- **Compact Design:** The sidebar is reduced to a narrow vertical strip displaying only the icons for each feature. The text labels are hidden.
- **Scroll Functionality:** A scroll handle is visible, indicating that the user can scroll through more feature icons than are currently displayed.
- **Functionality:** This "mini sidebar" allows for quick navigation while taking up minimal space. It is expected that hovering over an icon would display a tooltip with the feature's name, and clicking the expand icon (the arrow from the extended view) would return it to its full state.

Class Screen Wireframe: This wireframe illustrates the design for the main "Class Screen", which is the dedicated interface for viewing and managing all information and activities related to a single, specific class. This screen maintains the application's consistent layout, featuring the standard header and the collapsible features sidebar.



The primary components of the Class Screen are:

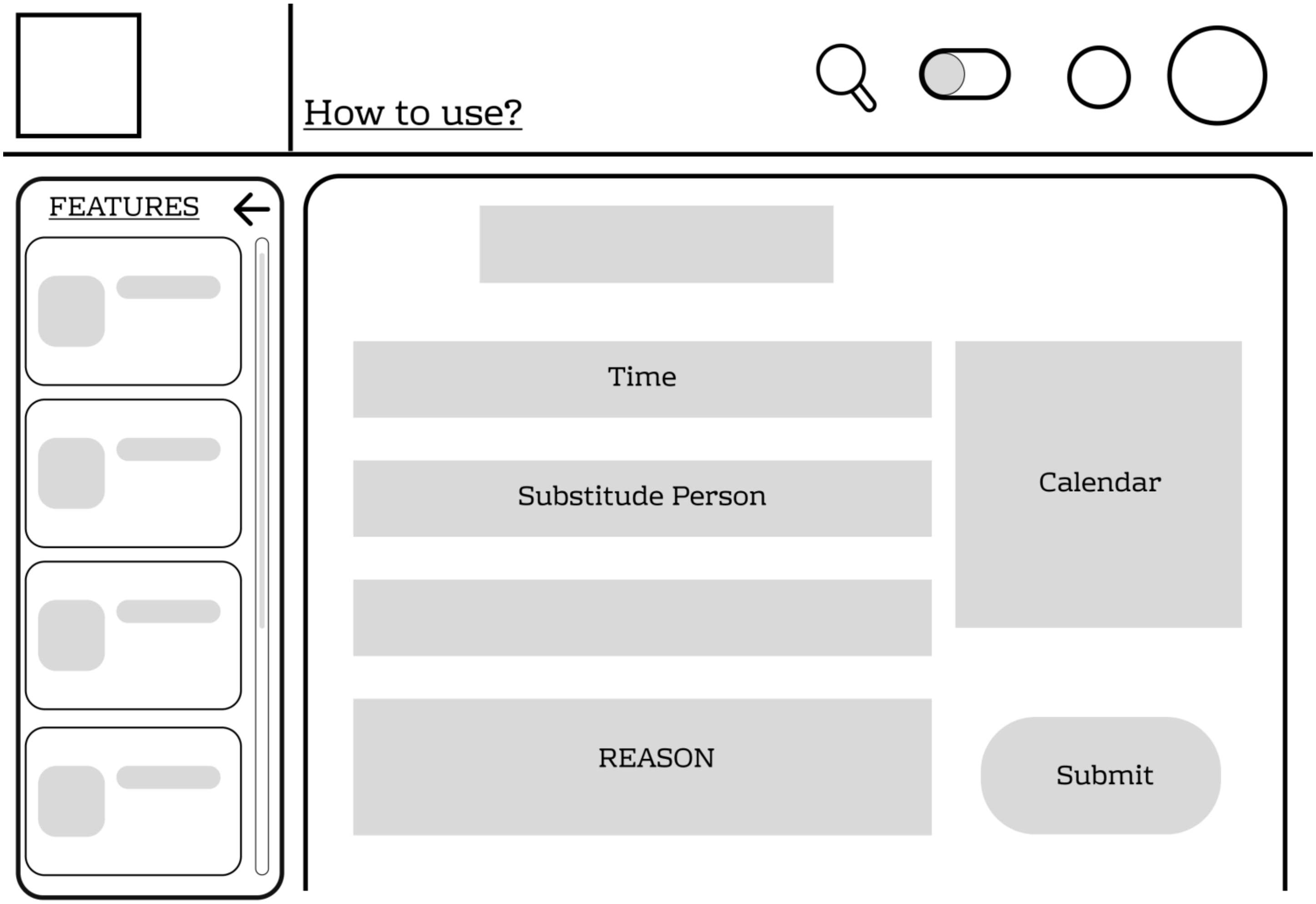
- **Header and Sidebar:** The screen includes the standard application header with a logo placeholder, a "How to use?" link, and user interface controls (search, mode toggle, profile icons). The "FEATURES" sidebar is shown in its expanded state, providing navigation to other parts of the system.
- **Main Content Area:** This section is specifically designed to present class-related content.
    - **Horizontal Navigation Bar:** At the top of the content area, there is a horizontally scrollable navigation bar. This bar contains several rounded tabs or "pills" that act as filters or links to different subsections of the class. Left and right arrows indicate that more options are available. This navigation allows users to switch between views such as "Announcements", "Assignments", "Student List", or "Attendance Records".
    - **Content Feed:** Below the navigation bar is the main content feed. This area

includes a title placeholder and a vertical list of items, represented by multiple horizontal bars. A scrollbar on the right indicates that the list is scrollable, allowing for a large amount of content. This feed will dynamically display information based on the subsection selected in the navigation bar above it (e.g., showing a list of announcements or a list of students).

- **Full-Screen Toggle:** An icon to expand the content feed to a full-screen view is located in the top-right corner of this section.

This design provides a structured and organised way for users, particularly Teachers, to access all the different types of information associated with a single class in one place.

Leave Request Screen Wireframe: This wireframe illustrates the design for the "Application for temporary absence" screen, which serves as the primary interface for Teachers and administrative staff to submit and manage leave requests. The design maintains the application's standard layout with the persistent header and features sidebar.



The key components of this screen are within the main content area:

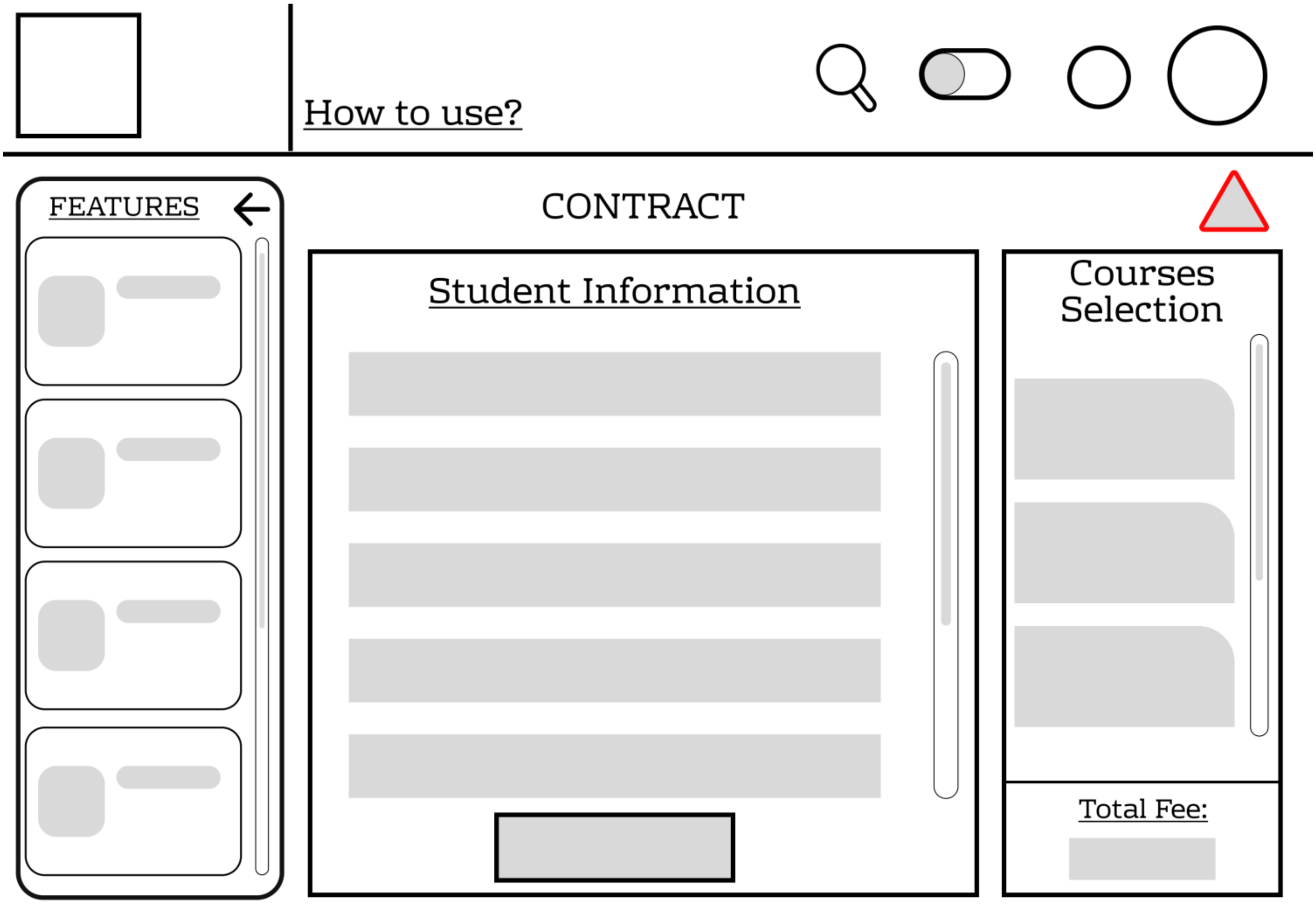
- **Header and Sidebar:** The screen includes the standard application header with a logo placeholder, a "How to use?" link, and user interface controls. The "FEATURES" sidebar is shown in its expanded state, providing navigation to other parts of the system.
- **Leave Request Form:** The central part of the screen is dedicated to the form for submitting a leave request.
  - A **title placeholder** is located at the top of the form.
  - The form consists of several clearly labelled input fields, including:
    - **Time:** A field for selecting the date or date range for the requested absence.
    - **Substitute Person:** A field to select or designate a colleague to cover the absence.
    - **REASON:** A large text area for the user to provide a detailed reason

for their leave.

- **Calendar Component:** To the right of the input fields, there is a large square component labelled "Calendar". This provides a visual and interactive tool for users to select their leave dates, directly supporting the "Choose Leave Dates" part of the use case.
- **Submit Button:** A prominent, rounded "Submit" button is located at the bottom right of the form, allowing the user to finalize and send their request for approval.

This interface is designed to streamline the leave request process (FR-TEA-04) by providing all necessary fields and tools in a single, organised view.

Contract Creation Screen Wireframe: This wireframe illustrates the interface for an Administrator or Learning Advisor to create a new student contract. This screen is a crucial part of the administrative workflow, designed to digitise and streamline the student enrolment process. The design uses the application's standard layout with a persistent header and sidebar.



The primary components of the Contract Creation screen are:

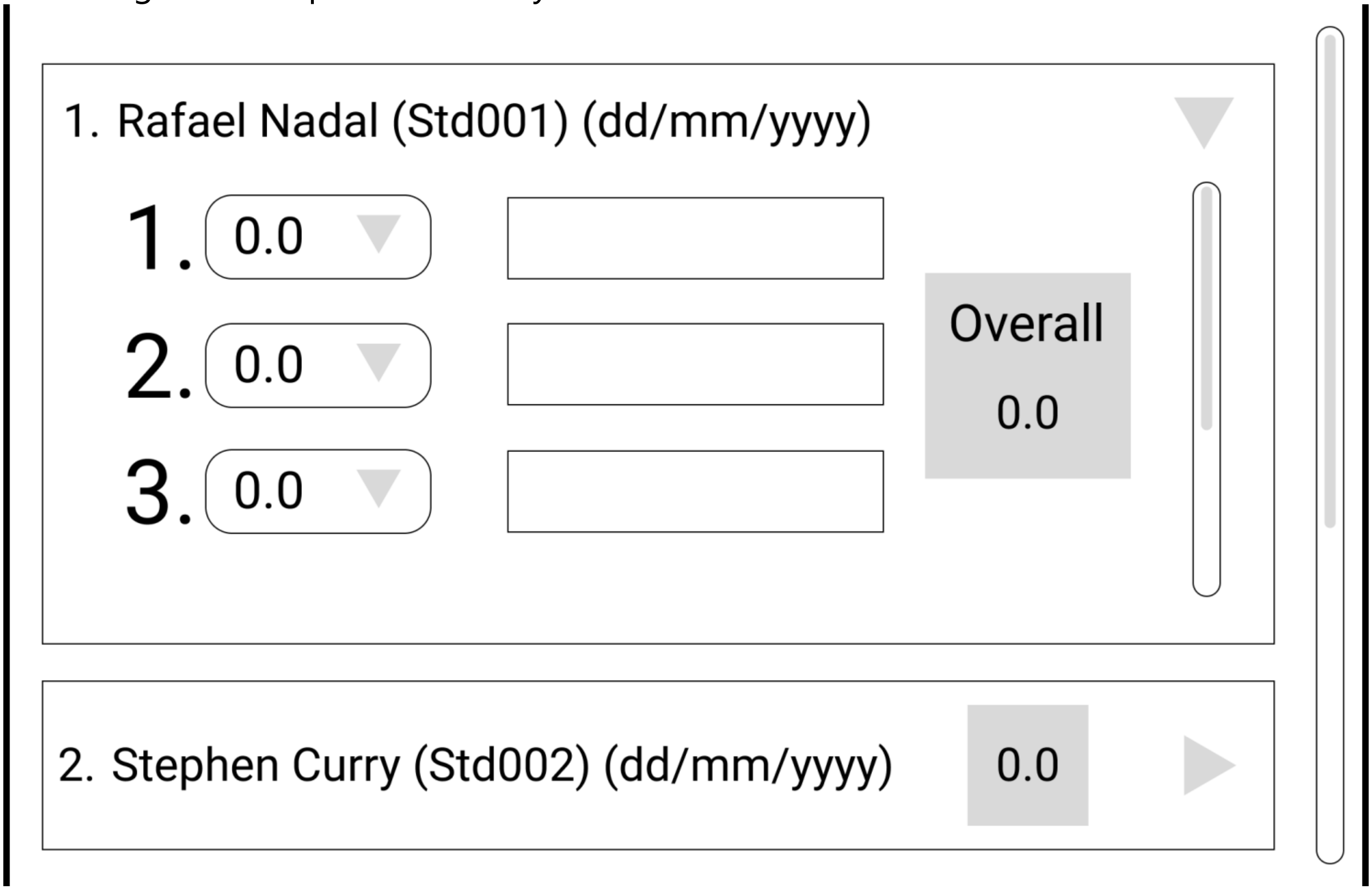
- **Header and Sidebar:** The screen includes the standard application header with its logo placeholder, "How to use?" link, and user controls. The "FEATURES" sidebar is shown expanded, providing access to other system functions.
- **Main Content Area:** The central workspace is titled "CONTRACT" and is organised into a two-panel layout to separate different types of information.
  - **Student Information Panel:** The larger panel on the left is titled "Student Information". It contains a scrollable series of rectangular input fields designed to capture all necessary student details (e.g., name, contact information, etc.). A button is located at the bottom of this panel, likely to save the entered information.
  - **Courses Selection Panel:** The panel on the right is titled "Courses Selection". It features a scrollable list of available courses that can be added to the contract. Below the course list, there is a "Total Fee:" display, which

would calculate the total cost based on the selected courses.

- **Cancel Icon:** A red, triangular icon is positioned in the top-right corner of the content area. This icon functions as a button to **cancel the contract creation process** and return the user to the previous screen.

This interface directly supports the functional requirement for a contract generator (FR-STF-03) and addresses the business problem of manual contract management.

Teacher Grading Component Wireframe: This wireframe provides a detailed look at the core component of the "Teacher Scoring Screen". It is designed to allow teachers to efficiently enter detailed scores and comments for each student in a class list. The component uses an expandable and collapsible accordion-style layout to manage screen space effectively.


1. Rafael Nadal (Std001) (dd/mm/yyyy)
1.
0.0
2.
0.0
3.
0.0
Overall
0.0
2. Stephen Curry (Std002) (dd/mm/yyyy)
0.0

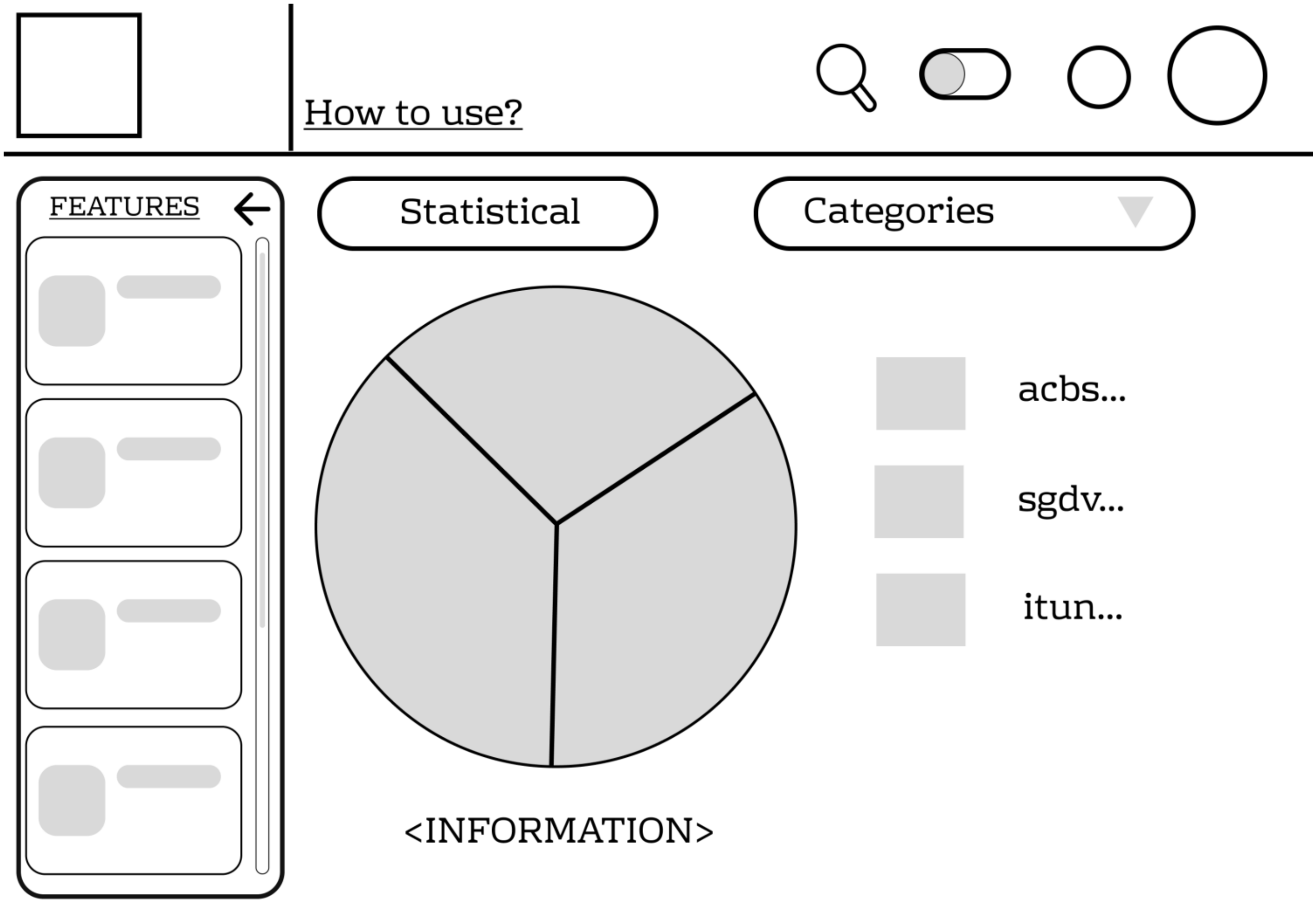
The wireframe illustrates two states for a student entry:

- **Expanded View:**
  - The entry for the first student, "1. Rafael Nadal (Std001)", is shown in its expanded state.

  - **Component Scores:** It features a numbered list of input fields, allowing the teacher to enter marks for different components of an assessment. Each score input has a dropdown arrow, suggesting predefined score options or steppers.
  - **Comment Fields:** Adjacent to each component score input, there is a rectangular text field for the teacher to provide specific written feedback on that part of the assessment.
  - **Overall Score:** A dedicated box on the right displays the "Overall" score, which would be automatically calculated based on the component scores entered.
  - **Scrollbar:** A scrollbar within the expanded section indicates that a student's assessment may have more score components than can be displayed at once.
- **Collapsed View:**
  - The entry for the second student, "2. Stephen Curry (Std002)", is shown in a collapsed, summary state.
  - This view displays only the student's name and ID, their final overall score, and a right-pointing arrow, which the teacher would click to expand the view for detailed score entry.

This design directly supports the functional requirement for teachers to input detailed grades and comments (FR-TEA-03) by providing a structured and organised interface that is both comprehensive when needed and compact for general overview.

Statistical Screen Wireframe: This wireframe illustrates the design for the "Statistical" screen, which serves as a data visualisation dashboard primarily for the Manager role. The purpose of this screen is to provide a clear, at-a-glance overview of key performance indicators for the tutoring centre. The design utilises the application's standard layout.



The primary components of the Statistical Screen are:

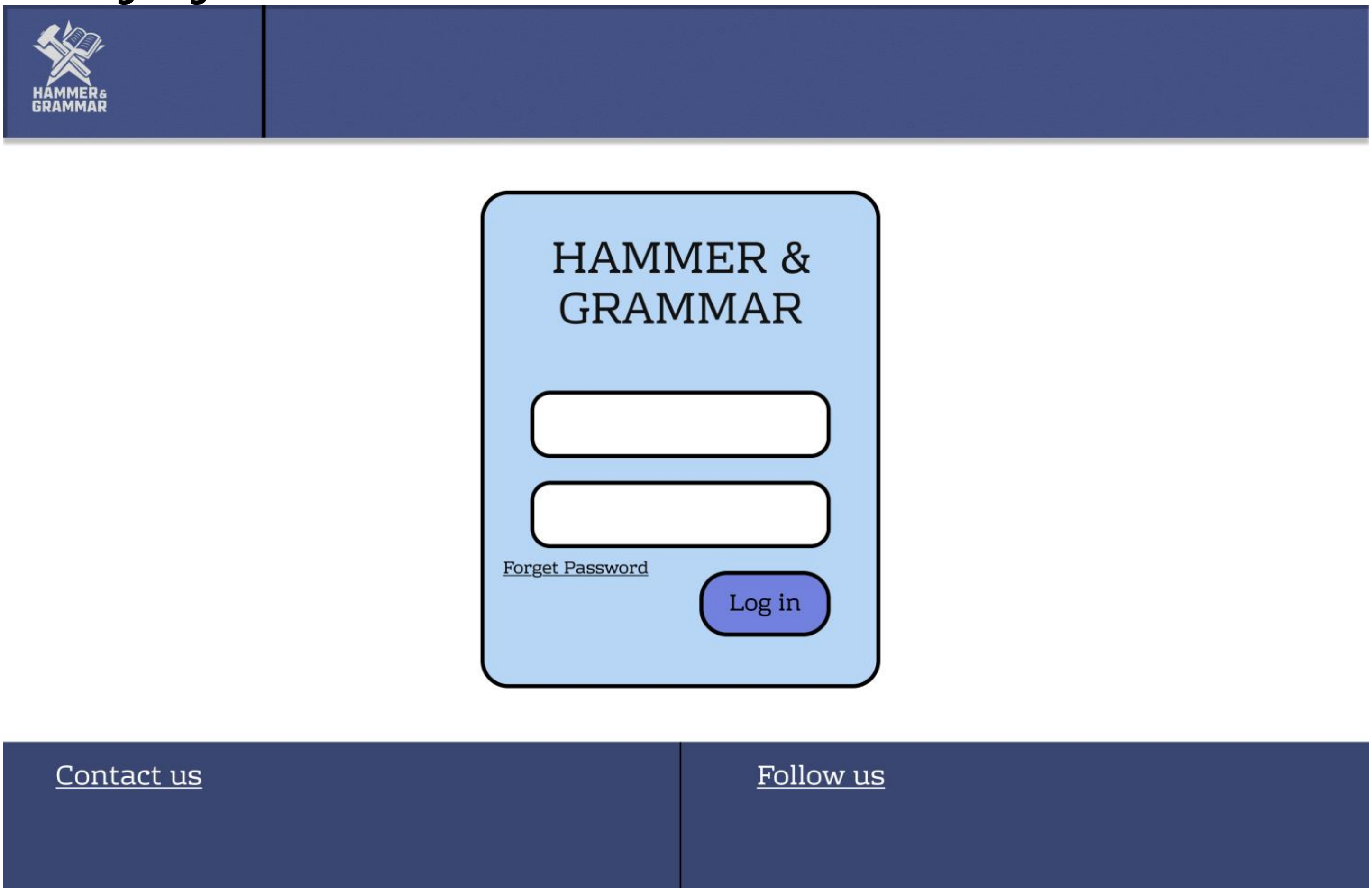
- **Header and Sidebar:** The screen includes the standard application header with a logo placeholder, a "How to use?" link, and user interface controls. The "FEATURES" sidebar is shown in its expanded state, providing navigation to other system functions.
- **Main Content Area:** This section is dedicated to displaying statistical data.
  - **Filter Controls:** At the top of the content area, there is a filter bar containing a selected "Statistical" button and a "Categories" dropdown menu. This dropdown allows the Manager to select the specific type of data to be displayed, such as revenue, student enrolment, or class performance.
  - **Data Visualisation:** The central element is a large pie chart, shown here divided into three segments. This chart visually represents the breakdown of the data from the selected category.

- **Legend:** To the right of the pie chart, a legend is provided. It consists of colour-coded squares and corresponding labels (e.g., "acbs...", "sgdv...") to explain what each segment of the chart represents.
- **Information Display:** Below the pie chart, there is a text placeholder labelled <INFORMATION>. This area is intended to display a summary, raw figures, or a brief analysis of the data shown in the chart.

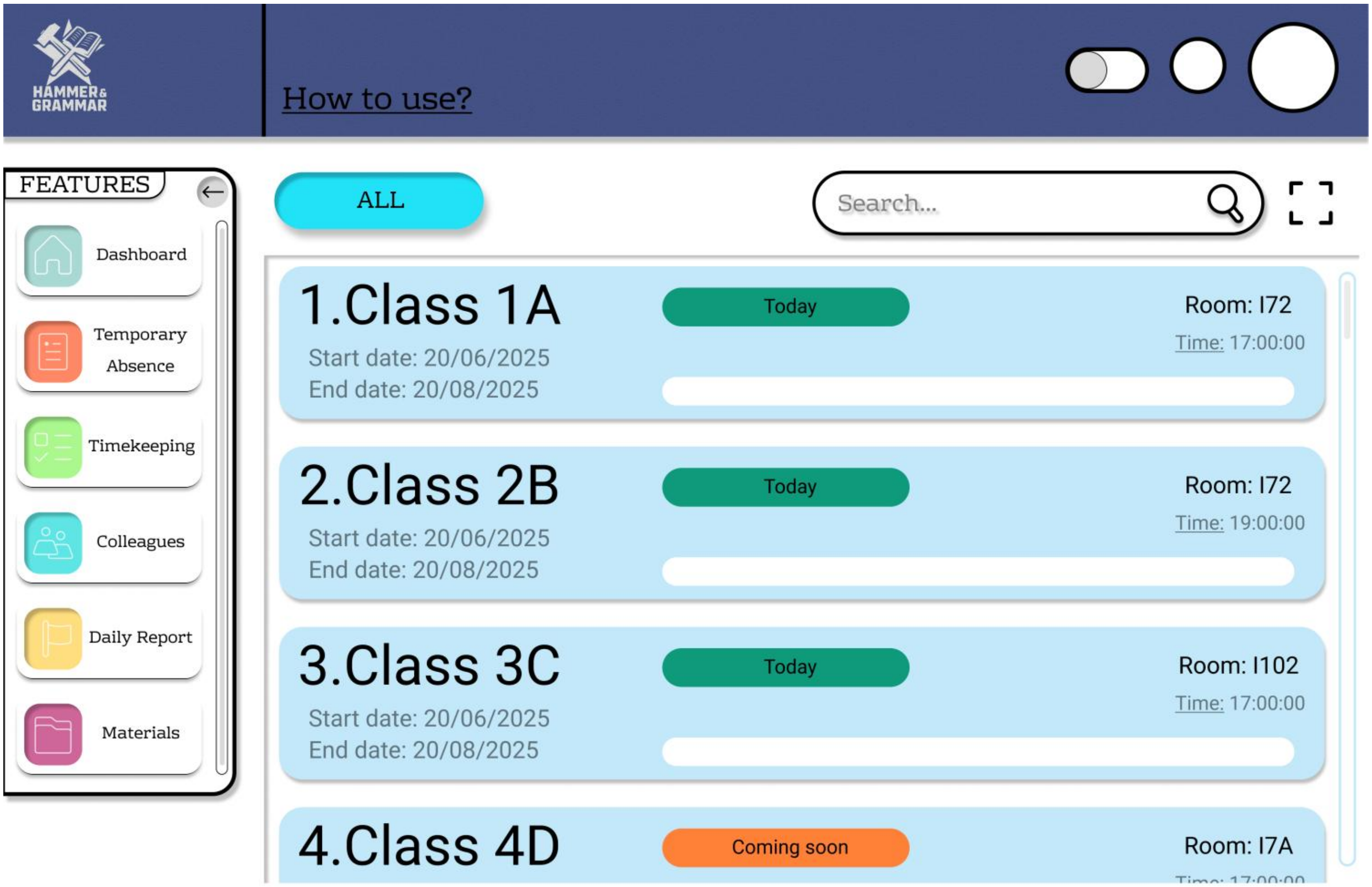
This screen directly supports the functional requirement for the Manager to view centre-wide statistics (FR-MGR-01) by presenting complex data in an easily digestible visual format.

PROTOTYPE

Landing Page



Dashboard



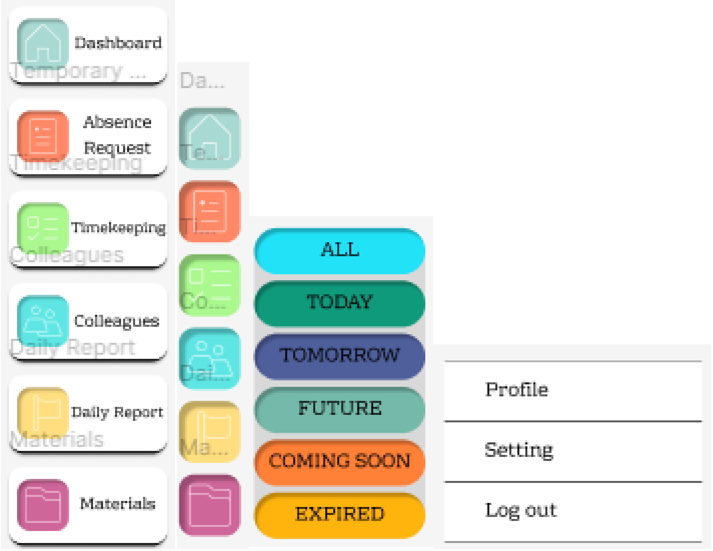
1.Class 1A
Today
Room: I72
Time: 17:00:00
Start date: 20/06/2025
End date: 20/08/2025

2.Class 2B
Today
Room: I72
Time: 19:00:00
Start date: 20/06/2025
End date: 20/08/2025

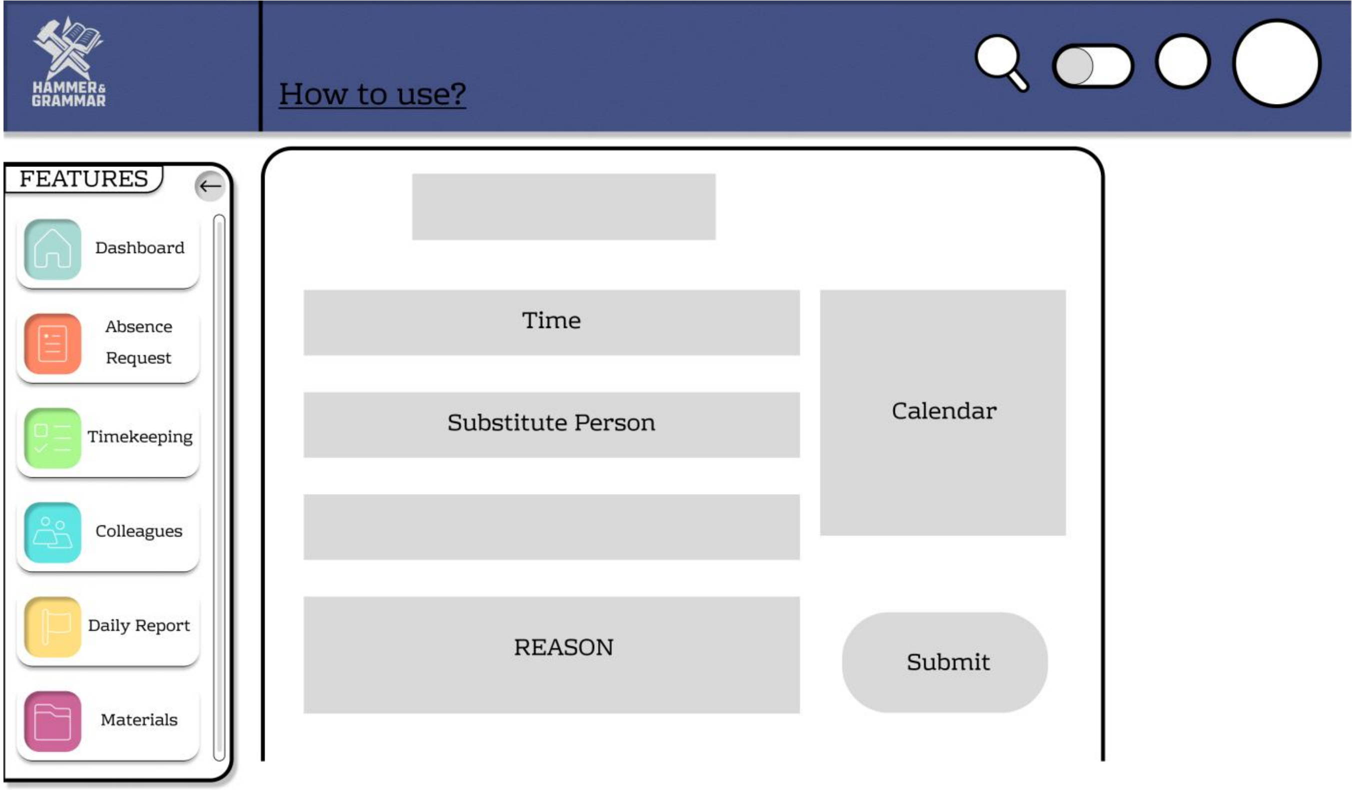
3.Class 3C
Today
Room: I102
Time: 17:00:00
Start date: 20/06/2025
End date: 20/08/2025

4.Class 4D
Coming soon
Room: I7A

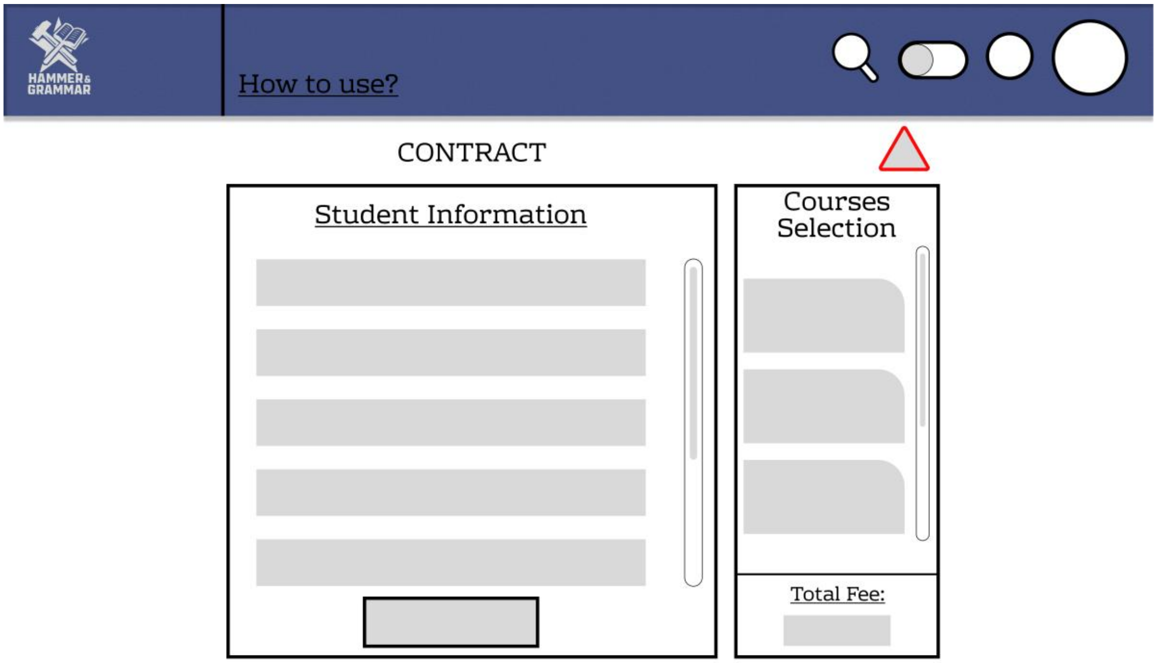
Side Bar, Mini Side Bar, Class Filter and User Setting



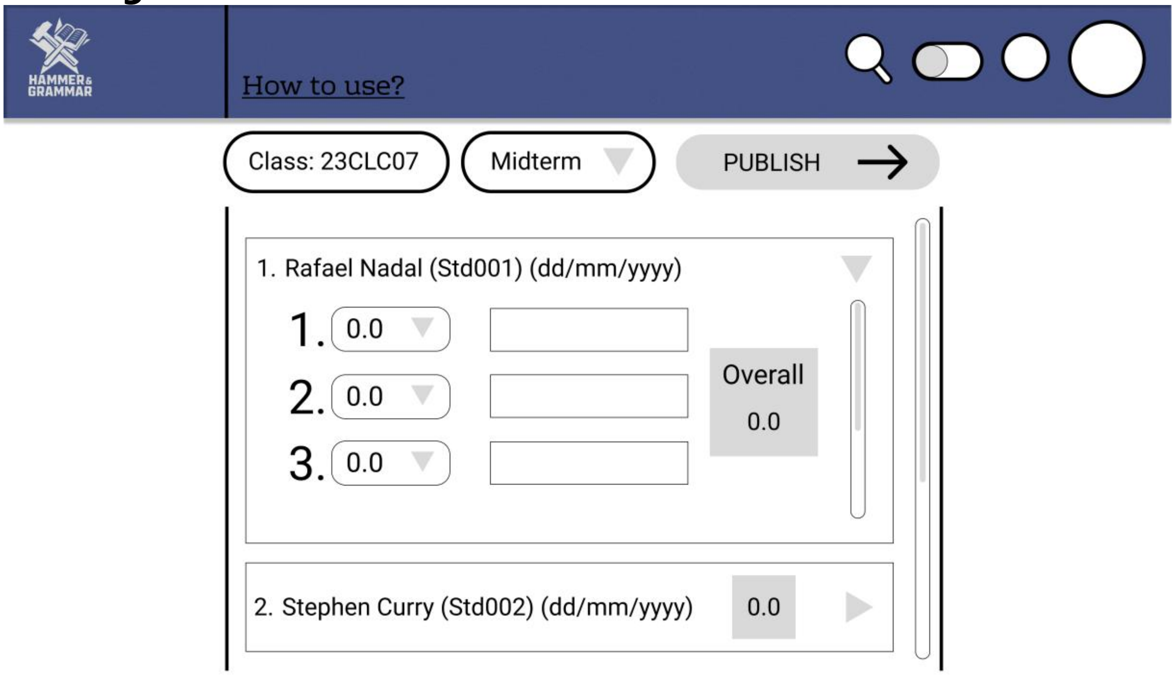
Absence Request



Contract Creation



Grading



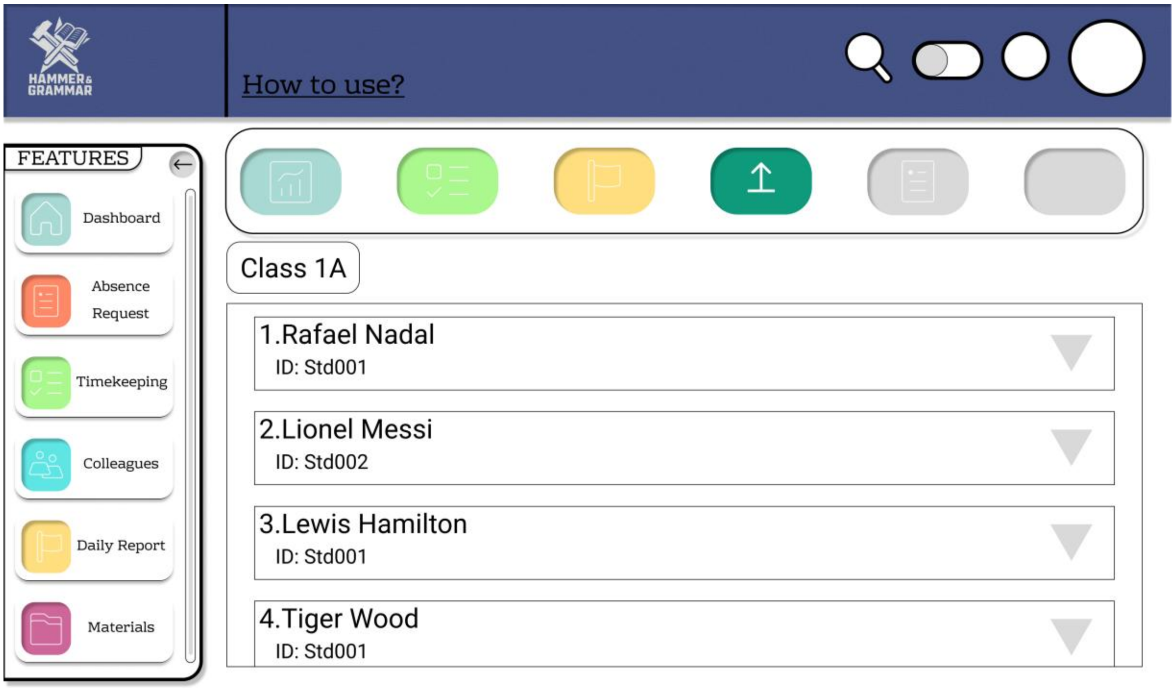
1. Rafael Nadal (Std001) (dd/mm/yyyy)

1. 0.0
2. 0.0
3. 0.0

Overall 0.0

2. Stephen Curry (Std002) (dd/mm/yyyy) 0.0

Class Screen



Class 1A

1.Rafael Nadal
ID: Std001

2.Lionel Messi
ID: Std002

3.Lewis Hamilton
ID: Std001

4.Tiger Wood
ID: Std001