# Courses and attendance management for Cretan club of Anthousa-Gerakas-Pallini

**1. Introduction**

**Purpose of the Document**

This document outlines the requirements for the development of a web application for managing the dance school of the Cretan Association of Anthousa-Gerakas-Pallini. The application will allow administrators to manage courses, student attendance, and payments, as well as provide tools for easy student and class management.

**Needs Addressed**

* Management of the school year, courses, and classes.
* Management of student attendance and payments.
* Easy student data entry and management, including bulk entry via CSV or Excel files.
* Ensuring security and data protection via login with username and password.

**2. Product Description**

**Purpose of the Application**

The application will allow administrators to organize dance school courses and classes. It will assist in managing students, tracking their attendance, and handling their monthly payments for each school year.

**Key Features**

1. **School Year Management**: Allows the creation and management of a school year, including courses and classes.
2. **Course and Class Management**: Each course belongs to a school year and is divided into classes.
3. **Weekly Course Calendar**: Classes will be placed in a weekly calendar.
4. **Student Management**: Includes adding, editing, and deleting students. Students can be added in bulk using CSV or Excel files.
5. **Student Attendance**: Manage student attendance on a daily basis for each class.
6. **Payment Management**: Manage students' tuition payments on a monthly basis.
7. **Security and Access**: Administrators can log in using a username and password to access the application and perform necessary actions.

**3. Functional Requirements**

**3.1 School Year Management**

* Ability to create and edit school years.
* Each school year will include courses that are divided into classes.

**3.2 Course and Class Management**

* Ability to create courses within a school year.
* Ability to create classes for each course.
* Each class will have a weekly schedule (course calendar).

**3.3 Student Management**

* Add new students with the following fields:
  + First Name
  + Last Name
  + Phone Number
  + Email Address
  + Guardian’s Name (for minors)
* Bulk student import via CSV or Excel files.
* Assign students to courses and classes.

**3.4 Student Attendance**

* Manage student attendance on a monthly basis for each course.
* Administrators will check a box for each student’s attendance on the days the class takes place.
* A table will display:
  + Columns: Courses of the month.
  + Rows: Students of each class.
  + Last column: Checkbox for marking monthly payment.

**3.5 Payment Management**

* Administrators can record tuition payments for each student on a monthly basis.
* In the attendance management page, there will be a checkbox for marking the student's payment for the month.

**3.6 Security and Access**

* Administrators will log in with a username and password to access the application.
* Roles and permissions for monitoring and editing data will be implemented.

**4. User Interface (UI) Requirements**

**4.1 Visual Appearance**

* The UI design will be user-friendly and follow modern UI practices. The language has to be in Greek.
* Colors and styles should reflect the identity of the Cretan Association of Anthousa-Gerakas-Pallini.

**4.2 Attendance Table**

* The attendance page will feature a table with columns for the courses and rows for students.
* Each table cell will be a checkbox for marking attendance.
* The last column will be for the monthly fee payment checkbox.

**5. Performance Requirements**

* The application must load quickly, with a loading time of less than 2 seconds for any page.
* The application must support at least 10 simultaneous users without significant delays or errors.

**6. Security Requirements**

* All student data must be encrypted during storage (e.g., database encryption).
* Administrator login will require username and password.
* All connections to the application must be secured via HTTPS to protect data.

**7. Technology Requirements**

* **PHP 8.1** for backend development.
* **MySQL** for data storage.
* **Apache 2.4** for server hosting.
* Frontend will be developed using HTML, CSS, and JavaScript.

**8. Timeline and Deliverables**

* **Phase 1**: Requirement analysis and system design.
* **Phase 2**: Development and integration of course, class, and attendance management features.
* **Phase 3**: Development of student management and payment features.
* **Phase 4**: Testing, evaluation by the administrator, and corrections.
* **Phase 5**: Delivery and training of administrators on how to use the application.