

Requirements Specification

1. Overview

The Gym Class Reservation System is a browser-based, single-page web application that enables gym members to:

- Browse a catalog of gym classes.
- Reserve seats in eligible classes.
- Manage their personal reservations.

Additionally, the system provides an **admin role** for managing the class catalog:

- Creating, editing, reordering, and deleting classes.
- Overriding class status when necessary.

The system is intended to run entirely in the browser with in-memory data, without requiring a backend server.

2. Actors

2.1 Regular User

A gym member who:

- Identifies themselves via a User ID.
- Views and filters the class schedule.
- Creates reservations subject to business rules.
- Cancels their own reservations within allowed limits.

2.2 Administrator

A privileged actor who:

- Logs in using an admin credential.
- Manages the class catalog (create, edit, delete).
- Manually reorders classes.
- Overrides class status and capacity when needed.
- Uses the same reservation features as a regular user if desired.

2.3 System

The Gym Class Reservation System web application which:

- Maintains in-memory data about classes and reservations.
 - Enforces reservation and cancellation rules.
 - Enforces permission rules between regular users and admins.
 - Provides UI elements for viewing, sorting, filtering, and editing.
-

3. Functional Requirements

3.1 User Identification

- **FR-1:** The system shall provide a text field for entering a User ID.
- **FR-2:** The system shall allow setting the currently active user via an explicit action (e.g., button).
- **FR-3:** The system shall display the currently active User ID and role (User/Admin) in the UI.
- **FR-4:** All reservations shall be associated with the active User ID at the time of creation.
- **FR-5:** Reservation and cancellation actions shall be disabled when no User ID is set.

3.2 Class Browsing

- **FR-6:** The system shall display a list of all gym classes in the “Available Classes” panel.
- Each class entry shall include:
 - Title
 - Type (Strength, Cardio, Mind & Body, Mobility)
 - Instructor
 - Date/time
 - Capacity
 - Remaining slots
 - Status badge (Available / Full / Past)
- **FR-7:** The system shall compute and update the status badge for each class based on current system time and remaining slots.
- **FR-8:** The system shall update the displayed remaining slots and status badges when reservations or cancellations modify capacity usage.

3.3 Sorting

- **FR-9:** Users shall be able to sort classes by **date/time**.
- **FR-10:** Users shall be able to choose ascending or descending sort order.
- **FR-11:** Optional: the system may allow different sort fields (e.g., by title), but date/time sorting shall always be available.
- **FR-12:** Sorting shall apply to the currently filtered and searched result set, not just the full set of classes.

3.4 Filtering and Search

- **FR-13:** Users shall be able to filter classes by **Type** (Strength, Cardio, Mind & Body, Mobility, or All).
- **FR-14:** Users shall be able to filter classes by **Instructor** (specific instructor or All).
- **FR-15:** Users shall be able to search classes by **keyword** in the class title.

- **FR-16:** Filtering and search criteria shall be combinable (e.g., filter by type and instructor and search term simultaneously).
- **FR-17:** Filters and search shall maintain their state until explicitly changed by the user.
- **FR-18:** Sorting shall respect the currently active filters and search criteria and produce consistent, deterministic ordering.

3.5 Sort + Filter Linkage

- **FR-19:** Whenever sort criteria change, the system shall re-apply both sorting and filtering to produce a consistent class list.
- **FR-20:** Whenever filter or search criteria change, the system shall re-apply both filtering and sorting to produce a consistent class list.
- **FR-21:** The UI shall consistently reflect the current combination of sort, filter, and search parameters.

3.6 Reservations

- **FR-22:** Users shall be able to create reservations for classes via a dedicated action (e.g., “Reserve” button).
- **FR-23:** The system shall allow reservations only when:
 - The class is in the future.
 - The class is not full.
 - The user does not already hold a reservation for another class at the same date/time.
 - The user has fewer than 3 future reservations.
- **FR-24:** When a reservation is created:
 - The reservation shall be stored with the User ID and class ID.
 - The class remaining slots shall be decreased by 1.
 - The class status badge shall be updated if necessary (e.g., from Available to Full).
 - The “My Reservations” panel shall refresh to show the new reservation.
- **FR-25:** If a reservation request violates a business rule (e.g., full class, past class, double booking, max reservations), the system shall:
 - Reject the reservation.
 - Display a clear error message indicating the reason.

3.7 Viewing “My Reservations”

- **FR-26:** The system shall display all reservations associated with the current User ID in the “My Reservations” panel.
- **FR-27:** For each reservation, the system shall display:
 - Class title
 - Type
 - Instructor
 - Date/time

- Status label based on current time (e.g., Past / Starting Soon / Upcoming).
- **FR-28:** The system shall update the “My Reservations” panel whenever:
 - The active User ID changes.
 - A reservation is created.
 - A reservation is cancelled.
 - Class details (title, time, instructor, etc.) are edited by an admin.

3.8 Cancellations

- **FR-29:** Users shall be able to cancel their own reservations via a dedicated action in the “My Reservations” panel.
- **FR-30:** The system shall prevent users from cancelling:
 - Reservations for past classes.
 - Reservations within 1 hour before the class start time.
- **FR-31:** When a cancellation is accepted:
 - The reservation shall be removed.
 - The corresponding class remaining slots shall be increased by 1.
 - The class status badge shall be recalculated and updated.
 - The “My Reservations” panel and “Available Classes” panel shall refresh to reflect the change.
- **FR-32:** If a cancellation request is not allowed due to business rules, the system shall:
 - Reject the cancellation.
 - Display an appropriate error message.

3.9 Admin Class Management

- **FR-33:** The system shall provide an **Admin Panel** that is visible only when an admin is logged in.
- **FR-34:** Admins shall be able to **create** new classes by entering title, type, instructor, date/time, capacity, and initial remaining slots.
- **FR-35:** Admins shall be able to **edit** existing classes:
 - Title
 - Type
 - Instructor
 - Date/time
 - Capacity
 - Remaining slots or status override
- **FR-36:** Admins shall be able to **delete** existing classes.
- **FR-37:** Admins shall be able to **manually reorder** classes (e.g., up/down controls), defining an admin-specific ordering.
- **FR-38:** The “Available Classes” panel shall reflect any admin-defined order when an admin-specific view is active, or fall back to the configured sort order.
- **FR-39:** Editing or deleting a class shall propagate updates to:

- The “Available Classes” panel.
 - The “My Reservations” panel (e.g., updated titles and times).
 - **FR-40:** Admins shall be able to override a class status (e.g., mark as Full or Cancelled) if required by configuration.
-

4. Business Rules

4.1 Reservation Rules

- **BR-1:** A user shall not be allowed to reserve a class whose start time is in the past (based on current system time).
- **BR-2:** A user shall not be allowed to reserve a class when remaining slots are 0 (full class).
- **BR-3:** A user shall not be allowed to hold two reservations for classes that start at the same date/time.
- **BR-4:** A user shall not be allowed to hold more than **three future reservations** at any time.
- **BR-5:** A reservation shall be considered “future” if the class start time is strictly later than now.

4.2 Cancellation Rules

- **BR-6:** A user shall not be allowed to cancel reservations for classes whose start time is in the past.
- **BR-7:** A user shall not be allowed to cancel reservations within **1 hour** before the class start time.
- **BR-8:** Users may only cancel their own reservations (matching their User ID).

4.3 Status Computation

- **BR-9:** Class status shall be:
 - **Past** if the start time is earlier than current time.
 - **Full** if remaining slots are 0 and the class is in the future.
 - **Available** if remaining slots > 0 and the class is in the future.
- **BR-10:** Reservation status for display in “My Reservations” shall be:
 - **Past** for classes whose start time is in the past.
 - **Starting soon** for classes starting within the next hour.
 - **Upcoming** for classes starting more than an hour in the future.

4.4 Sorting & Filtering

- **BR-11:** Sorting by date/time shall always use a consistent comparison based on the class start time.
- **BR-12:** Filters and search criteria shall not change the underlying data; they only affect the view.

- **BR-13:** Sorting and filtering shall always produce a deterministic order for the same parameters.

4.5 Permission Rules

- **BR-14:** Only authenticated admins may access the Admin Panel.
 - **BR-15:** Only admins may create, edit, delete, or reorder classes.
 - **BR-16:** Regular users shall never be able to modify class definitions, capacity, or schedule.
 - **BR-17:** Admin rights shall be granted only when both an admin User ID and a valid admin password are provided.
-

5. Non-Functional Requirements

5.1 Usability

- **NFR-1:** The UI shall be simple, with clear grouping of user functions (Available Classes, My Reservations, Admin Panel).
- **NFR-2:** Primary user actions (set user, reserve, cancel, sort, filter) shall be discoverable without training.
- **NFR-3:** Admin functions shall be grouped within a clearly labeled Admin Panel.

5.2 Performance

- **NFR-4:** UI updates (reserve, cancel, sort, filter, admin edits) shall complete within 1 second on a typical modern desktop browser.
- **NFR-5:** The system shall handle at least 200 classes and 1,000 reservations in-memory during a session without noticeable lag.

5.3 Reliability

- **NFR-6:** Data structures for classes and reservations shall remain internally consistent during a session.
- **NFR-7:** Class capacity and remaining slots shall never become negative under correct operation.

5.4 Portability

- **NFR-8:** The application shall run without plugins on modern browsers (Chrome, Firefox, Edge, Safari).
- **NFR-9:** The system shall be packaged as a single HTML file with embedded CSS and JavaScript.

5.5 Security

- **NFR-10:** Admin credentials shall not be exposed in the UI or stored persistently; they are checked only in memory.
 - **NFR-11:** The system shall not store or transmit personal data other than the free-text User ID.
-

6. UI Requirements

- **UI-1:** The top of the page shall contain:
 - User ID input
 - Admin password input
 - Set User / Role button
 - Current user and role label
 - **UI-2:** The main area shall be divided into:
 - “Available Classes” panel
 - “My Reservations” panel
 - **UI-3:** A dedicated message area shall display success, error, and informational messages.
 - **UI-4:** Sorting, filtering, and search controls shall be grouped together and clearly labeled.
 - **UI-5:** Each class card shall display all core attributes and a “Reserve” button (when user is set).
 - **UI-6:** Each reservation card shall display class details and a “Cancel” button (when cancellation is allowed).
 - **UI-7:** The Admin Panel shall be visually distinct and labeled, containing fields for editing and controls for create/update/delete/reorder.
-

7. Permission Model

- **PM-1:** The system shall support at least two roles: Regular User and Admin.
- **PM-2:** Role selection shall be determined by credentials (User ID + admin password).
- **PM-3:** Regular Users:
 - May view, sort, filter, search classes.
 - May create and cancel reservations according to business rules.
 - Shall not see or access admin edit controls.
- **PM-4:** Admins:
 - Have all Regular User capabilities.
 - May access the Admin Panel.
 - May create, edit, delete, reorder classes.
- **PM-5:** All admin operations shall verify that the current actor is an authenticated admin before applying changes.

8. Constraints

- **C-1:** No backend service shall be required; all logic runs client-side in JavaScript.
 - **C-2:** Data shall be in-memory only; refreshing the page resets all data.
 - **C-3:** System time shall rely on the client's local clock for "past/future" and "1-hour" computations.
 - **C-4:** Implementation shall use standard HTML5, CSS, and JavaScript (no frameworks required).
-

9. Glossary

- **User ID:** A free-text identifier used to associate reservations with a user.
- **Admin:** A privileged user who can manage class definitions and schedule.
- **Class Capacity:** Maximum number of participants allowed for a class.
- **Remaining Slots:** Available seats left for reservation in a class.
- **Reservation:** A link between a User ID and a specific class.
- **Past Class:** A class whose start time is earlier than the current time.
- **Future Class:** A class whose start time is later than the current time.
- **Starting Soon:** A class whose start time is within one hour of the current time.
- **Sort:** Ordering of classes according to specific criteria (e.g., date/time).
- **Filter:** Restricting the visible set of classes according to criteria (type, instructor, keyword).
- **Admin Panel:** UI section that exposes class management capabilities to admins.