**Comprehensive Use Case Specification for Admin in FYP Management System**

**1. System Overview**

The **Final Year Project (FYP) Management System** allows university administrators to manage student projects, supervisors, evaluations, and system configurations. The **Admin** is the primary actor responsible for maintaining data, generating reports, and ensuring smooth system operations.

**2. Actors**

| **Actor** | **Description** |
| --- | --- |
| **Admin** | Manages students, supervisors, projects, and system settings. |
| *Student* | (Secondary) Submits project proposals and reports. |
| *Supervisor* | (Secondary) Guides students and evaluates projects. |

**3. Detailed Use Cases**

**Use Case 1: Login to System**

* **Actor:** Admin
* **Description:** Admin logs into the FYP Management System.
* **Preconditions:** Admin has valid credentials.
* **Basic Flow:**
  1. Admin enters email and password.
  2. System verifies credentials.
  3. On success, redirects to Admin Dashboard.
* **Alternative Flow:**
  1. If credentials are invalid, system displays an error.
* **Post conditions:** Admin gains access to the system.

**Use Case 2: Manage Students**

* **Actor:** Admin
* **Description:** Admin adds, edits, or removes student records.
* **Preconditions:** Admin is logged in.
* **Basic Flow:**
  1. Admin clicks **"Manage Students"**.
  2. System displays a list of students.
  3. Admin selects:
     + **Add Student**: Enters (Name, ID, Email, and Program).
     + **Edit Student**: Modifies existing details.
     + **Delete Student**: Removes record after confirmation.
  4. System validates and updates the database.
* **Alternative Flows:**
  1. If student ID already exists, system rejects duplication.
* **Post conditions:** Student records are updated.

**Use Case 3: Manage Supervisors**

* **Actor:** Admin
* **Description:** Admin adds or assigns faculty supervisors.
* **Preconditions:** Admin is logged in.
* **Basic Flow:**
  1. Admin clicks **"Manage Supervisors"**.
  2. System shows the list of supervisors.
  3. Admin:
     + **Adds Supervisor** (Name, Department, Max Projects).
     + **Assigns Supervisor** to a project.
  4. System checks availability and updates records.
* **Alternative Flows:**
  1. If supervisor’s project limit is exceeded, system shows an error.
* **Post conditions:** Supervisor assignments are updated.

**Use Case 4: Manage Projects**

* **Actor:** Admin
* **Description:** Admin creates, assigns, and tracks FYP projects.
* **Preconditions:** Students and supervisors are registered.
* **Basic Flow:**
  1. Admin clicks **"Manage Projects"**.
  2. Chooses:
     + **Create Project** (Title, Description, and Deadline).
     + **Assign Student & Supervisor** (from dropdown lists).
  3. System validates and links them in the database.
* **Alternative Flows:**
  1. If a student is already assigned, system prevents duplication.
* **Post conditions:** Project is added and visible in tracking.

**Use Case 5: Generate Reports**

* **Actor:** Admin
* **Description:** Admin exports project progress or evaluation reports.
* **Preconditions:** Projects exist with submission data.
* **Basic Flow:**
  1. Admin clicks **"Generate Reports"**.
  2. Selects report type:
     + **Progress Report** (Student milestones).
     + **Evaluation Report** (Supervisor feedback).
  3. Filters by Department/Date Range.
  4. System compiles data into **PDF/Excel**.
  5. Admin downloads or prints.
* **Alternative Flows:**
  1. If no data exists, system displays "No records found."
* **Post conditions:** Report is generated for review.

**Use Case 6: System Configuration**

* **Actor:** Admin
* **Description:** Admin sets deadlines, grading rules, and permissions.
* **Basic Flow:**
  1. Admin clicks **"System Configuration"**.
  2. Updates:
     + **Submission Deadlines** (e.g., Proposal: 30-May-2024).
     + **Grading Criteria** (e.g., 70% for final report).
  3. System saves changes and notifies affected users.
* **Alternative Flows:**
  1. If a deadline is in the past, system rejects it.
* **Postconditions:** System settings are updated.

