EduTraker Product Requirements Document

EduTraker is a Software-as-a-Service (SaaS) platform designed to support educational institutions, particularly those in crisis-affected regions[1]. The platform aims to ensure educational continuity, enhance administrative efficiency, and facilitate data-driven decision-making[2].

1. Purpose and Scope

- **Primary Purpose:** To mitigate the disruptive impact of crises on education by providing a centralized platform for managing student information and communication[3].
- **System Scope:** Covers functional and non-functional requirements, including the system architecture and technology stack[4].
- **Development Focus:** The initial focus is on the backend API, with a roadmap for future frontend development[5].

2. User Roles and Personas

The system utilizes a strict Role-Based Access Control (RBAC) model to define permissions across various users[6666].

Role	Description
Admin	Superuser with full system access; manages

	Workstream Managers[7].
Workstream Manager	Manages a group of schools and assigns School Managers[8].
School Manager	Manages a single school, overseeing teachers, secretaries, and operations[9].
Teacher	Manages courses, lesson plans, attendance, and grading[10].
Secretary	Provides administrative support, such as registering students and guardians[11].
Student	Accesses academic information, grades, assignments, and attendance[12].
Guardian	Monitors child's academic progress and communicates with teachers[13].
Guest	Default role for unauthenticated users with limited access[14].

3. Key Product Features

3.1. Academic & Content Management

- **Progress Tracking**: Teachers can manage grades, assignments, and record student attendance[15].
- **Personalized Dashboards**: Students and guardians can view specific academic records and upcoming tasks[16].
- **Lesson Planning**: Teachers have tools to create lesson plans and upload learning materials[17].

3.2. Communication & Analytics

- **Messaging & Notifications**: Includes an internal messaging system and notifications for deadlines or announcements[18].
- **Reporting**: School and Workstream Managers can generate performance reports[19].
- **System Analytics**: High-level analytics regarding user activity and system health are available to Administrators[20].

---

4. Technical Specifications

4.1. Technology Stack

- **Backend**: Django and Django REST Framework[21].
- **Database**: MySQL[22].
- **Authentication**: JSON Web Tokens (JWT)[23].
- **Documentation**: OpenAPI/Swagger via drf-spectacular[24].

4.2. Data Model

The system uses a multi-tenant architecture with a clear hierarchy[25]:

- **WorkStream**: Represents a collection of schools[26].
- **School**: Represents an individual institution[27].
- **CustomUser**: The central model for all users, extended by specific profile models (Student, Teacher, etc.)[28].
- **Academic Structure**: Managed via models for Course, ClassRoom, and AcademicYear[29].

---

5. Non-Functional Requirements

- **Performance**: Dashboard pages are designed to load within 2 seconds for 95% of requests[30].
- **Security**: Uses HTTPS encryption and securely hashes passwords using **bcrypt**[31].
- **Reliability**: Target uptime of 99% with a Progressive Web App (PWA) supporting offline functionality[32].
- **Usability**: The interface is designed to be intuitive across desktop and mobile devices[33].

---

6. Future Roadmap

Future development phases will include a **React-based frontend** [34], a dedicated **mobile application** [35], and **advanced analytics** for deeper performance insights[36].

Would you like me to provide more detail on the specific fields required for the **Data Model** (such as Assignments or Attendance) or explain the **future scope** for the React frontend?