Integrated CRM, LMS, and Accounting System for Educational Consultancy Service Provider

Document Version: 1.3

Date: December 17, 2025

Author: Senior Business Technical Analyst (built by xAI)

Purpose: This updated document specifies the technology stack, incorporating PHP for the backend and MySQL for the relational database as required. Additional recommendations include Laravel as the PHP framework for structured development, React.js for the frontend to ensure a responsive UI, and MongoDB for handling unstructured data like documents and materials. The architecture remains microservices-based for modularity, with updates reflected in relevant sections. This builds on previous updates for multi-role assignments, visa tracking, document storage, and study abroad features.

---

1. Introduction

1.1 Purpose

The purpose of this document is to define the technical requirements for a unified, embedded system that streamlines operations for an educational consultancy. The system integrates CRM for student acquisition and management, LMS for course delivery and learning, and Accounting for financial tracking. This integration aims to enhance efficiency, reduce data silos, and improve user experience for counselors, teachers, students, and administrators.

1.2 Scope

- **In Scope:** Development of core modules for inquiry handling, student onboarding, course management, class assignments, material sharing, task management, payment processing, reporting, visa tracking, document storage, multi-role assignments, role-based menus, and newly added features like university databases and application management.

- **Out of Scope:** Hardware procurement, third-party integrations beyond specified APIs (e.g., payment gateways), and ongoing maintenance post-deployment.

- **Assumptions:** The system will be web-based with potential mobile extensions, using cloud hosting for scalability. Users have basic digital literacy.

1.3 System Overview

The system is a web application with role-based access, built on a microservices architecture for modularity. Key components:

- **Frontend:** Responsive UI using React.js for dynamic, interactive interfaces.

- **Backend:** API-driven services using PHP with the Laravel framework for robust, secure, and scalable development.

- **Database:** Relational database using MySQL for structured data (e.g., student profiles, payments); NoSQL using MongoDB for unstructured data (e.g., documents, materials).

- **Integration:** Internal APIs between CRM, LMS, and Accounting; external for email notifications and payment gateways.

- **Additional Recommendations:**

  - **Server Environment:** Apache or Nginx for hosting PHP applications.

  - **Caching and Queueing:** Redis for caching and Laravel's built-in queue system for background jobs (e.g., notifications).

  - **Security Tools:** Laravel's Sanctum or Passport for API authentication.

  - **DevOps:** Docker for containerization, Composer for PHP dependency management, and npm for frontend builds.

  - **Testing Frameworks:** PHPUnit for backend unit/integration tests, Jest for frontend tests.

High-Level Architecture Diagram (Text Representation):

```

[User Interfaces (Web/Mobile)]

        |

[API Gateway]

        |

[Microservices: CRM | LMS | Accounting] (PHP/Laravel)

        |

[Database Layer: MySQL (Relational) + MongoDB (NoSQL) + File Storage]

        |

[External Services: Payment Gateway, Email/SMS, University APIs]
```

```

---

## 2. User Roles and Permissions

The system supports multiple user roles with granular permissions. Users (except Students) can be assigned one or multiple roles dynamically via the Admin interface. Access to modules and menus is cumulative based on assigned roles (e.g., a user with Counselor and Accountant roles sees combined menus and features). Admin role grants full access overriding all restrictions.

- **Multi-Role Assignment:**
  - Admin can assign roles to backend users (e.g., User1: Counselor + Accountant; User2: Counselor + Accountant + Teacher).
  - System checks roles at login to generate dynamic permissions and menus.
  - Students are a separate role without multi-assignment, as they are end-users.

| Role | Description | Key Permissions |
|---|---|---|
| Admin | Oversees the entire system with full access. | Full access: User management (including role assignments), system configurations, all modules, reports. |
| Counselor | Handles inquiries, onboarding, consultations, visa/applications. | Create/edit student profiles, assign courses/classes, manage visas/applications, view payments. |
| Teacher | Manages classes, uploads materials, assigns tasks. | Access class-specific data, grade assignments, communicate with students. |
| Student | Accesses courses, materials, tasks, applications, and personal dashboard. | View/submit assignments, track payments/visas/applications, update profile, upload documents. (No multi-role). |
| Accountant | Manages financial transactions. | Process payments, generate invoices/reports, view accounting ledgers. |

### 2.1 Role-Based Menu Constants

Menus are defined as constants in the frontend code (e.g., in a configuration file or database-driven) and rendered dynamically based on user roles. Below are the menu structures for each role. For multi-
```

role users, menus merge without duplicates (e.g., Counselor + Accountant shows both CRM and Accounting sections).

- **Admin Menu (Full Access):**
  - Dashboard (Overview of all metrics)
  - User Management (Create/Edit Users, Assign Roles)
  - CRM (Inquiries, Onboarding, Visa Tracking, Applications)
  - LMS (Courses, Classes, Materials, Tasks)
  - Accounting (Fees, Payments, Ledgers, Reports)
  - Reports & Analytics (Custom Reports, Financial Dashboards)
  - System Settings (Configurations, Integrations)
  - Document Repository (All Documents)
  - Partner Management (Universities/Agents)

- **Counselor Menu:**
  - Dashboard (Pending Inquiries, Applications, Visa Statuses)
  - CRM (Inquiries, Student Onboarding, Visa Tracking, University Recommendations, Applications)
  - Student Management (Profiles, Enrollments, Communications)
  - Reports (Enrollment by Country, Lead Conversion)

- **Teacher Menu:**
  - Dashboard (Upcoming Classes, Student Progress)
  - LMS (My Classes, Upload Materials, Assign Tasks, Grade Assignments)
  - Student Interaction (Forums, Attendance, Feedback)
  - Reports (Class Performance, Completion Rates)

- **Student Menu:**
  - Dashboard (Upcoming Classes, Payments, Visa/Application Status)
  - My Courses (Materials, Tasks, Submissions)

  - Profile (Update Details, Upload Documents)

  - Applications (University Apps, Visa Tracking)

  - Payments (View Invoices, Make Payments)

- **Accountant Menu:**

  - Dashboard (Outstanding Dues, Revenue Overview)

  - Accounting (Fee Management, Payments, Invoices, Refunds)

  - Reports (Financial Ledgers, Tax Reports, Analytics)

  - Student Finances (View/Payment History per Student)

- **Dynamic Menu Generation:**

  - On login, fetch user's roles from database.

  - Merge menu items from assigned roles (e.g., User2 with Counselor + Accountant + Teacher sees CRM + Accounting + LMS sections).

  - Use RBAC to hide/show sub-items based on permissions.

  - Implementation: JSON-based menu config in backend, rendered via frontend navigation component.

---

3. Functional Requirements

3.1 CRM Module

Handles student acquisition from inquiry to onboarding.

- **Inquiry Management:**

  - Students submit inquiries via a web form capturing: Name, Email, Phone, Intended Countries (multi-select: e.g., USA, UK, Canada), Education Level (e.g., Undergraduate, Postgraduate, Diploma).

  - System auto-assigns inquiries to counselors based on workload or region.

  - Notification: Email/SMS to counselor upon new inquiry.

- **Student Onboarding:**

  - Counselor creates student profile with additional details: Passport info, Academic history (e.g., transcripts, GPA), Test scores (e.g., IELTS band), Travel history/documents.

  - Option to enroll in preparatory courses (PTE, IELTS, SAT, Languages).

  - Assign classes: Link student to class ID, teacher, and schedule.

- **Additional Features (Suggested):**

  - Lead Scoring: AI-based scoring of inquiries based on completeness and match to services (e.g., high score for students with clear country preferences).

  - Communication Log: Track all interactions (calls, emails) with timestamps and notes.

  - Follow-up Reminders: Automated calendar events for counselors to follow up on inquiries.

  - Chatbot Integration: AI-powered chatbot for initial inquiry handling, qualifying leads, and scheduling consultations.

3.2 LMS Module

Facilitates online/offline course delivery and management.

- **Course Management:**

  - Admin/Teachers create courses: e.g., PTE Preparation, IELTS Speaking Module.

  - Define curriculum: Modules, durations, prerequisites.

- **Class Assignment:**

  - Counselor/Teacher assigns students to classes via Class ID.

  - Class dashboard: List of enrolled students, schedule, attendance tracking.

- **Material and Task Management:**

  - Teachers upload files (PDFs, videos, readings) tagged to Class ID.

  - Assign tasks/homework: Due dates, submission portal for students.

  - Students view/download materials, submit assignments, receive feedback/grades.

- **Additional Features (Suggested):**
  - Progress Tracking: Student dashboard showing completion percentages, quiz scores.
  - Live Sessions: Integration with Zoom/Teams for virtual classes, with recording storage.
  - Forum/Discussion: Class-specific chat for Q&A between students and teachers.
  - Certification: Auto-generate certificates upon course completion.
  - Pre-Departure Modules: Interactive courses on cultural adaptation, travel tips, and destination-specific orientation.

3.3 Accounting Module

Manages financial aspects integrated with CRM and LMS.

- **Fee Management:**
  - Define fee types: Class Fee, Exam Fee, Admission Fee, Consultation Fee, Processing Fee.
  - Group fees by student category (e.g., Abroad Study Package vs. Course-Only).
  - Generate invoices: Customizable templates with due dates, partial payments.

- **Payment Processing:**
  - Students pay via integrated gateway (e.g., Stripe, PayPal).
  - Track payments: Status (Paid, Pending, Overdue), receipts generation.
  - Refunds: Handle partial/full refunds with approval workflow.

- **Accounting Ledger:**
  - Real-time ledger for income/expenses.
  - Integration: Auto-update student status in CRM upon payment (e.g., activate class access).

- **Additional Features (Suggested):**
  - Installment Plans: Allow split payments with reminders.
  - Tax Compliance: Calculate taxes based on location, generate reports.

  - Financial Analytics: Dashboards showing revenue by course/country, outstanding dues.

  - Commission/Payout Management: Track payouts to partners/universities/agents based on successful enrollments.

3.4 Integrated Features

- **Dashboard:** Role-specific overviews (e.g., Student: Upcoming classes, payments, visa status; Counselor: Pending inquiries, applications). Dynamic based on multi-roles.

- **Notifications:** Email/SMS/Push for events like task due, payment reminders, class starts, visa updates.

- **Reporting:** Custom reports (e.g., Student enrollment by country, Revenue by fee type) exportable to PDF/Excel.

- **Document Management:** Secure upload/storage for student docs (e.g., academic transcripts, passports, visas, recommendation letters). Version control and access logs for compliance.

3.5 Study Abroad Specific Features

- **Visa Tracking Workflow:**

  - Dedicated workflow for visa applications: Students upload required documents (academic records like transcripts/degrees, travel documents like passports/visas, financial proofs).

  - Storage: Secure, encrypted repository with categorization (e.g., by document type) and expiration alerts (e.g., passport renewal reminders).

  - Status Tracking: Multi-stage workflow (e.g., Document Collection → Submission → Interview Scheduling → Approval/Rejection). Counselors update statuses; students view progress via dashboard.

  - Checklists: Dynamic checklists based on country (e.g., USA requires DS-160, UK needs TB test).

  - Reminders: Automated notifications for deadlines, interviews, or missing documents.

  - Integration: Link with external APIs for real-time visa status checks where available (e.g., embassy portals).

- **University Recommendation Engine:** Based on student profile (GPA, test scores, preferences), suggest universities/programs using an integrated database or API (e.g., from QS Rankings or custom abroad colleges database).

- **Test Score Integration:** Upload/track scores from official sources, with alerts for retakes.

- **Partner Management:** CRM extension for universities/agents: Track commissions, agreements, performance.

- **Alumni Network:** Post-enrollment module for feedback, referrals, and community connections.

- **Additional Features:**

  - **Application Management:** Automated workflows for university applications: Form filling assistance, document bundling, submission tracking to multiple institutions.

  - **Abroad Institutions Database:** Searchable database of global universities, programs, eligibility criteria, scholarships, and costs. Admin can update via CSV imports or APIs.

  - **Lead Automation:** Auto-follow-ups via email/SMS sequences; AI lead qualification to prioritize high-potential inquiries.

  - **Scholarship and Aid Research:** Tools for students/counselors to search and apply for financial aid, with tracking.

  - **Post-Arrival Support:** Dashboard section for tracking accommodation, insurance, and initial settlement tasks.

---

4. Non-Functional Requirements

4.1 Performance

- Response Time: <2 seconds for standard operations.

- Scalability: Handle 1,000 concurrent users; auto-scale via cloud (e.g., AWS).

- Uptime: 99.9% with redundancy.

4.2 Security

- Authentication: JWT-based (via Laravel Sanctum), multi-factor for admins.

- Data Protection: Encrypt sensitive data (e.g., passports, financials); GDPR/HIPAA compliant.

- Access Control: Role-Based Access Control (RBAC) with support for multi-roles.

- Audit Logs: Track all changes/actions, especially document uploads and role assignments.

4.3 Usability

- UI/UX: Intuitive, mobile-responsive; support multiple languages (English, local languages). Dynamic menus based on roles.

- Accessibility: WCAG 2.1 compliant.

4.4 Reliability and Maintenance

- Backup: Daily automated backups.

- Error Handling: Graceful failures with user-friendly messages.

- Updates: Modular design for easy patches.

---

5. Data Requirements

5.1 Data Model (High-Level Entities)

Use ER diagram in implementation; key entities (updated with role assignments):

Entity	Attributes Examples	Relationships
User	ID, Name, Email, Roles (array: e.g., ['Counselor', 'Accountant'])	Has_many: For backend users; Links to Student if applicable
Student	ID, Name, Email, Countries, Education Level, Passport, Academic Docs (JSON array), Travel Docs	Has_many: Enrollments, Payments, Applications, Visas
Course	ID, Name (e.g., IELTS), Description, Duration	Has_many: Classes
Class	ID, Course_ID, Teacher_ID, Schedule, Students_List	Belongs_to: Course, Teacher
Teacher	ID, Name, Expertise, Classes_Assigned	Belongs_to: User (if multi-role)
Payment	ID, Student_ID, Fee_Type, Amount, Status, Date	Belongs_to: Student
Inquiry	ID, Student_Details, Status, Assigned_Counselor	Converts_to: Student
Visa	ID, Student_ID, Country, Status, Documents (array), Interview_Date	Belongs_to: Student

| Application | ID, Student_ID, University_ID, Status, Submission_Date | Belongs_to: Student, University |

| University | ID, Name, Country, Programs (JSON), Eligibility | Has_many: Applications |

5.2 Data Storage

- Structured: MySQL for transactions and relational data.

- Unstructured: MongoDB for files and documents, with metadata indexing; cloud storage (e.g., AWS S3) for large files.

---

6. Integration and Interfaces

- **Internal:** RESTful APIs between modules (e.g., CRM notifies LMS on enrollment, Accounting on payments). API for role checks and menu generation, built with Laravel routing.

- **External:**

  - Payment Gateways: API for transactions (e.g., Stripe integration via Laravel packages).

  - Email/SMS: Services like SendGrid/Twilio (via Laravel Mail/Notifications).

  - Calendar: Google Calendar sync for schedules.

  - University APIs: For real-time program data (e.g., Common App integrations).

  - Embassy/Visa Portals: Where feasible for status checks.

- **APIs:** Expose endpoints for mobile app (future extension) using Laravel API resources.

---

7. Testing and Deployment

- **Testing:** Unit, Integration, System, UAT; cover 95% code, including multi-role scenarios. Use PHPUnit for backend, Jest for frontend.

- **Deployment:** CI/CD pipeline (e.g., GitHub Actions or Laravel Forge); staging/prod environments.

- **Hardware/Software:** Cloud-based (e.g., AWS or DigitalOcean); PHP 8.x, MySQL 8.x, Node.js for frontend builds.

---

8. Recommendations

Based on analysis of study abroad consultancy needs and industry standards:

- **AI Enhancements:** Implement chatbots for initial inquiries; predictive analytics for student success rates and lead conversion; AI document verification for authenticity (using PHP libraries like TensorFlow.php if needed).

- **Mobile App:** Native app for students/teachers to access LMS, track visas/applications on-the-go (e.g., using React Native).

- **Blockchain for Documents:** Secure, tamper-proof storage for certificates/visas to enhance trust with partners (integrate via PHP SDKs).

- **Sustainability Features:** Track carbon footprint of study abroad (e.g., flight estimates) for eco-conscious branding.

- **Expansion Potential:** Modular design to add HR module for staff management, e-commerce for study materials, or integration with immigration software.

- **Prioritization:** Start with CRM (including visa/application tracking) and Accounting for quick ROI, then LMS. Focus on compliance features for data-heavy processes like visas.

- **New Recommendations:** Implement multi-language support for global reach; analytics for counselor performance; integration with social media for lead generation. Test multi-role assignments thoroughly for permission conflicts. Use Laravel's Eloquent ORM for efficient MySQL interactions and Horizon for queue monitoring.