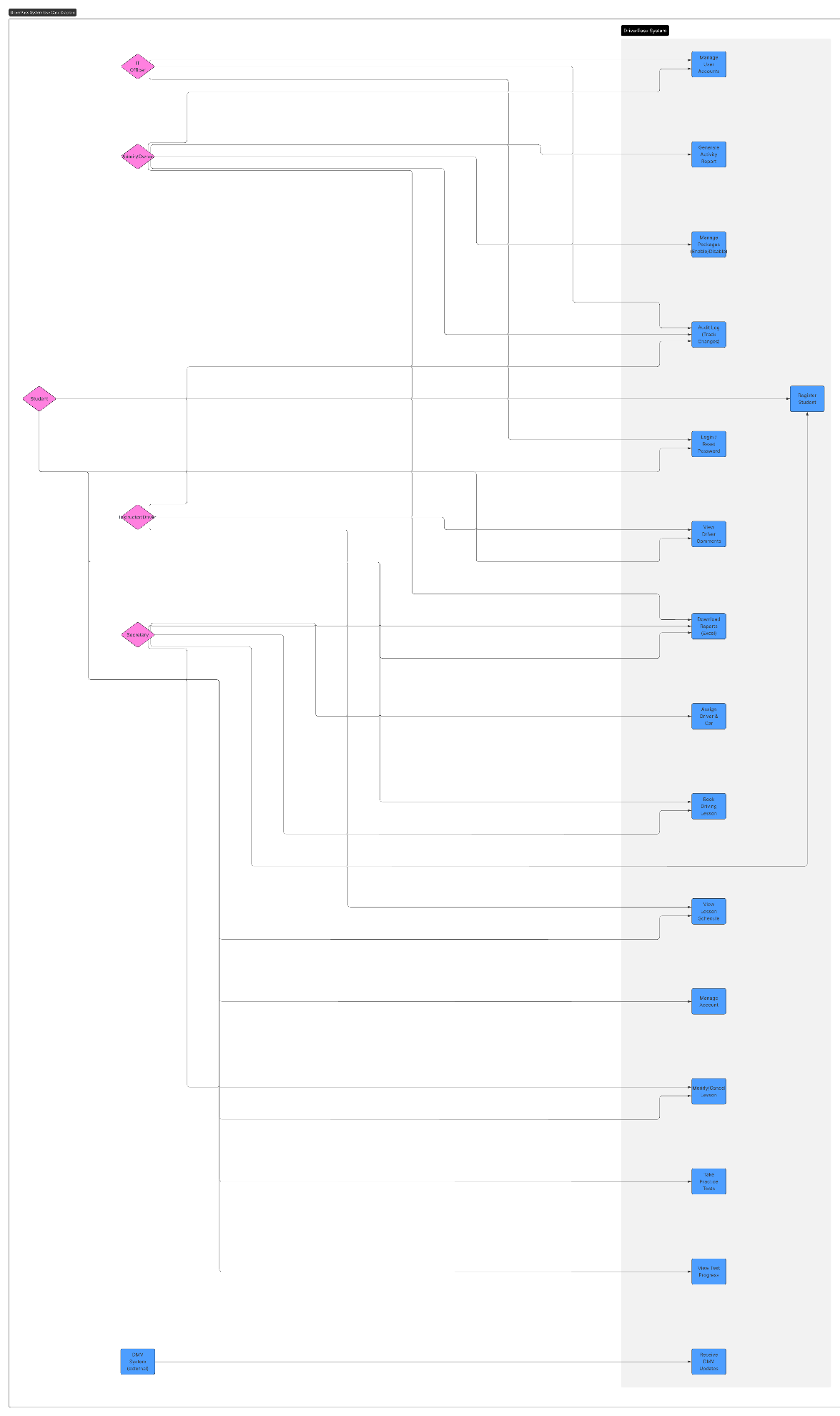
# CS 255 System Design Document Template

## UML Diagrams

### UML Use Case Diagram



### UML Activity Diagrams

A group of squares with black dots

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

### UML Sequence Diagram

A screenshot of a computer

AI-generated content may be incorrect.

### UML Class Diagram

A screenshot of a computer

AI-generated content may be incorrect.

## Technical Requirements

**1) Architecture Overview**

* **Cloud-hosted, web-based, multi-tier system** (presentation, API, data). Users access from desktop or mobile browsers; no on-prem servers required.
* **Primary modules**: Authentication/RBAC, Scheduling & Reservations, Driver & Car Assignment, Practice Tests & Scoring, DMV Content Sync, Reporting/Exports, Notifications.

**2) Hosting & Hardware**

* **Compute**: Auto scaled stateless web/API services (e.g., AWS EC2/App Runner or Azure App Service).
* **Database**: Managed relational DB (AWS RDS / Azure Database for PostgreSQL/MySQL).
* **Object storage**: For exports, backups (S3/Blob Storage).
* **Content delivery (optional)**: CDN for static assets.
* **Environment tiers**: dev, test, prod; isolated VPC/VNet.

*Rationale*: Owner wants access “from any computer or mobile device” and not to “deal with backup and security,” favoring cloud hosting.

**3) Core Software Stack**

* **Backend**: Java 17 + Spring Boot (or Python 3.11 + Django/FastAPI) with REST/JSON.
* **Frontend**: Server-rendered HTML or SPA (React) consuming REST APIs.
* **Database**: PostgreSQL (preferred) or MySQL.
* **ORM**: JPA/Hibernate (Java) or Django ORM/SQLAlchemy (Python).
* **Background jobs**: Worker service (e.g., AWS SQS/Azure Queue + consumer) for notifications & DMV sync.
* **Exports**: CSV/XLSX generation (Apache POI / Pandas + openpyxl).

**4) Integrations**

* **DMV updates**: Pull or webhook endpoint to ingest new rules/policies/sample questions; version practice tests, record last sync timestamp, and flag applied updates.
* **Email/SMS**: Send confirmations for bookings and password resets (SES/SendGrid/Twilio).
* **(Optional) Payments**: Gateway (e.g., Stripe) if package purchase flows are added later.
* **Excel/CSV downloads**: Support “download the reports and some information” for offline review.

**5) Security & Access Control**

* **RBAC** with roles: Student, Instructor/Driver, Secretary, Admin/Owner, IT Officer.
* **Account admin**: IT Officer can reset accounts and block access.
* **AuthN/AuthZ**: Session/JWT, role checks on every endpoint; **password reset** flow.
* **Auditability**: Immutable **Activity Log** recording who created/modified/canceled reservations; who graded/submitted tests; when packages were enabled/disabled; and key account actions. Support printable activity reports.
* **Data protection**: TLS 1.2+, at-rest encryption (DB/storage KMS), hashed passwords (bcrypt/argon2).
* **Least privilege**: Separate app and admin DB creds; secrets manager.
* **MFA (recommended)** for staff roles (Admin/IT/Secretary).

**6) Scheduling & Lessons**

* **Business rules**:
  + Each lesson is **2 hours**; enforce end = start + 2h.
  + Reservation must **assign a specific driver and car** and prevent double-booking of either.
  + Create, modify, cancel with full audit trail and conflict checks.
* **Availability engine**: Checks instructor & vehicle calendars and student’s requested slot, returns alternatives if unavailable.
* **Pickup/Drop-off** captured with reservation (must allow “same location” rule).

**7) Practice Tests & Learning**

* **Test catalog**: Practice tests with questions and versions; **status** tracking per student (Not Taken, In Progress, Failed, Passed) with **time taken** and **score**.
* **Attempts**: Resume in-progress or start new; submit → grade → update progress & logs.
* **DMV compliance**: Apply DMV updates to keep questions/rules current; notify staff on changes.

**8) Reporting & Exports**

* **Activity report** (who did what, when) for accountability and dispute resolution; printable.
* **Student progress** report (tests taken, time, scores, status).
* **Lesson schedule** reports (driver/car utilization).
* **Export formats**: CSV/XLSX for Excel as requested by the owner.

**9) Notifications**

* **Email/SMS**:
  + Booking confirmation/cancellation/reschedule notices.
  + Password resets and account events.
  + (Optional) Reminders before lessons; updates when DMV content changes.

**10) Nonfunctional Requirements / Quality Attributes**

* **Availability**: 99.5% monthly (target) for the client-facing app.
* **Performance**: P95 < 500 ms for standard reads; scheduling/search P95 < 1.5 s under normal load.
* **Scalability**: Horizontal scaling for stateless tiers; DB read replicas if necessary.
* **Reliability**: Daily encrypted backups; 14–30 day retention; tested restores.
* **Privacy**: Limit access to PII; data minimization; retention policy for logs/attempts.
* **Usability**: Web UI consistent with the owner’s sketched screens (progress table with test name, time taken, score, status).

**11) Observability & Operations**

* **Monitoring**: Application metrics (latency, errors), DB health, queue depth.
* **Logging**: Centralized structured logs; separate **audit** log stream.
* **Alerts**: On availability, error rate spikes, failed DMV syncs, failed notification sends.
* **Environment management**: IaC (Terraform/CloudFormation/Bicep); CI/CD with automated tests and database migrations.

**12) Data Model (anchor to UML)**

* Entities: User (+roles), Student, Instructor, Lesson, Car, Package, PracticeTest, Question, TestAttempt, ActivityLog, DMVUpdate. (See Class Diagram section.)
* Keys: UUIDs for all primary entities; foreign keys for associations.
* Constraints: unique (email, plate); non-overlapping schedule per instructor/car.

**13) Assumptions & Constraints**

* **Internet required** for booking and DMV sync; offline only for viewing previously exported reports.
* **Future package customization** by non-developers is out of scope for v1 (owner can **enable/disable** packages).
* DMV integration availability may affect how quickly new rules appear in the app.