DriverPass Business Requirements Document

Client: DriverPass
Course: CS 255 – System Analysis and Design
Deliverable: Project One – Business Requirements Document

System Components and Design

Purpose

Design a secure, cloud-first web system that enables DriverPass customers to take online practice exams, enroll in training packages, and schedule on-the-road lessons; and enables DriverPass staff (owner, IT admin, secretary, trainers) to manage users, packages, resources (cars/drivers), and audit activity.

System Background

DriverPass identified a market gap: over 65% of DMV test-takers fail when relying only on past tests. The company will offer online practice tests and on-road instruction with configurable packages. The system must support online self-service scheduling, role-based access, and detailed audit trails.

Objectives and Goals

- Provide online practice exams with progress tracking (name, time taken, score, status).
- Enable customers to register, purchase packages, and schedule 2-hour lessons with specific cars/drivers.
- Allow staff to create/modify/cancel reservations and view audit history of who did what and when.
- Expose administrative tools for IT to reset/block accounts and manage roles/rights.
- Integrate with DMV updates to receive rules/policy/question changes and notify staff.
- Operate reliably via browser on desktop and mobile with secure cloud hosting.

Requirements

Nonfunctional Requirements

Performance Requirements

• Web-based application with responsive UI; typical page loads ≤ 2 seconds under normal load.

• Availability target 99.5%+ monthly; nightly incremental backups and point-in-time database recovery.

• Content updates (e.g., DMV changes) applied within 2 business days of notification.

Platform Constraints

• Cloud-hosted web stack (e.g., Linux, Nginx/Apache, app tier, relational DB).

• Modern browsers (Chrome, Edge, Safari, Firefox) and mobile browsers supported.

• Relational database for transactional integrity (packages, reservations, users, audit logs).

Accuracy and Precision

• Unique accounts distinguished by verified email; roles: Owner, IT Admin, Secretary, Instructor, Customer.

• Strong server-side validation (dates, times, pickup = drop-off).

• Comprehensive audit logging for CRUD on reservations, users, payments, content.

Adaptability

• Admin UI to enable/disable packages and edit content without code changes.

• Role management UI (assign/revoke roles) with least-privilege defaults.

• Configuration-driven package definitions to add/retire offerings in future releases.

Security

• Enforce TLS (HTTPS), salted password hashing, MFA option for staff accounts.

• Account lockout on brute-force attempts; secure password reset via email token.

• PCI-aware payment processing; do not store CVV; tokenize cards via a compliant gateway.

• Row-level authorization checks and input sanitization; regular vulnerability patching.

Functional Requirements

- The system shall validate user credentials and assign role-based access on login.
- The system shall allow customers to create accounts, reset passwords, and update profiles.
- The system shall let customers purchase training packages and view entitlements.
- The system shall schedule 2-hour driving lessons by date/time, car, and instructor with conflict checks.
- The system shall allow customers and the secretary to create/modify/cancel reservations.

- The system shall display test progress (test name, time taken, score, status: not taken/in progress/failed/passed).
- The system shall record driver notes per lesson and show start/end times and comments.
- The system shall maintain audit logs for reservation lifecycle events and administrative actions.
- The system shall allow the owner/IT admin to enable/disable packages.
- The system shall integrate with a DMV update feed or admin import to keep rules/questions current and notify staff.
- The system shall generate downloadable reports (CSV/Excel) for management review.

User Interface

• Customers: mobile/desktop browser access to register, buy packages, schedule/cancel lessons, take practice tests, view progress.

• Secretary: desktop browser access to enter caller registrations, schedule/cancel lessons, look up customers, print day sheets.

• Instructors: view assigned lessons, log start/end times, enter driver comments.

• Owner: dashboards, financials, reports, package toggles, audit trails.

• IT Admin: user/role management, password resets/blocks, system configuration.

Assumptions

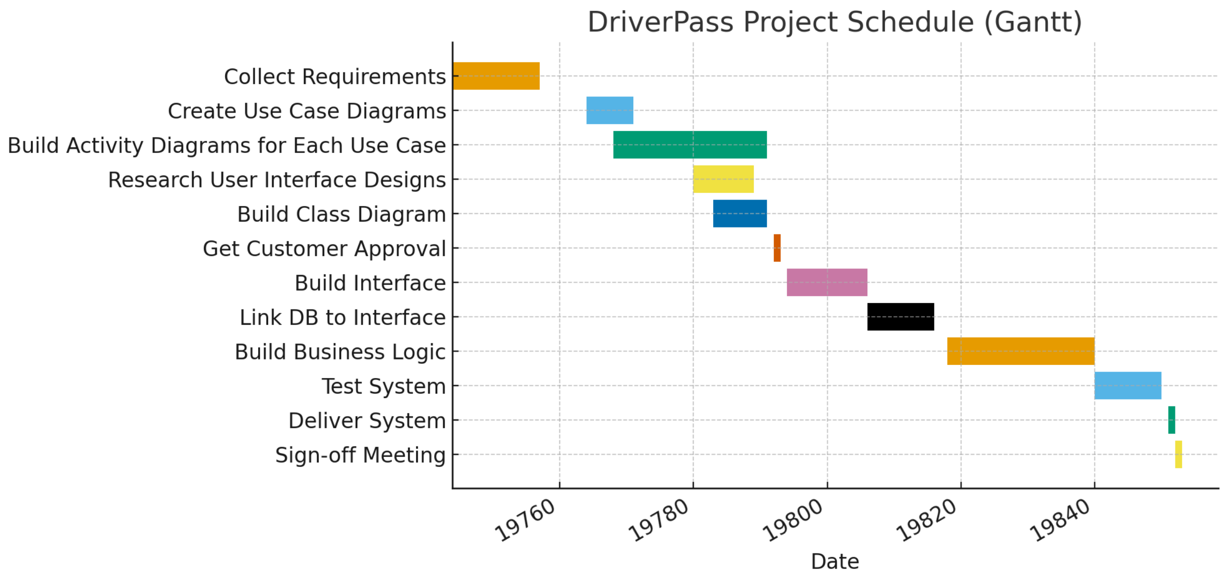
- DMV provides a legal mechanism to distribute content updates (file feed, API, or manual import).
- Payment processing will use a third-party PCI-compliant gateway; DriverPass will not store card CVV.
- Each car is assigned to one instructor at a time for scheduling simplicity.
- Time zone and holiday calendars align with DriverPass's local operations.

Limitations

- Offline editing is not supported to prevent data divergence; reporting extracts can be downloaded for offline viewing.
- Initial release supports enabling/disabling packages but not arbitrary module addition without development effort.
- Integration with DMV depends on available interfaces and may require manual updates initially.
- Budget/scope limits advanced analytics and mobile native apps in v1; web responsive only.

Schedule (Gantt Snapshot)

The following image visualizes the schedule described in the interview transcript. A Lucidchart version can be created following the included tutorial, and a screenshot can replace this chart for submission.


DriverPass Project Schedule (Gantt)
Collect Requirements
Create Use Case Diagrams
Build Activity Diagrams for Each Use Case
Research User Interface Designs
Build Class Diagram
Get Customer Approval
Build Interface
Link DB to Interface
Build Business Logic
Test System
Deliver System
Sign-off Meeting
Date