

# DRIVEMASTER



DANIEL MATEO MONTOYA GONZÁLEZ - 2020200098  
DAVID GERARDO DIAZ GOMEZ - 20201020087  
CRISTIAN ARTURO PARRA GONZALEZ - 20201578102



# Introduction to DriveMaster

Innovation in driving education



**DRIVEMASTER**

# DriveMaster: Problem and Business Needs

## CURRENT PROBLEM

- Manual Paper Registration.
- Spreadsheet Tracking.
- Manual Certificate Generation.

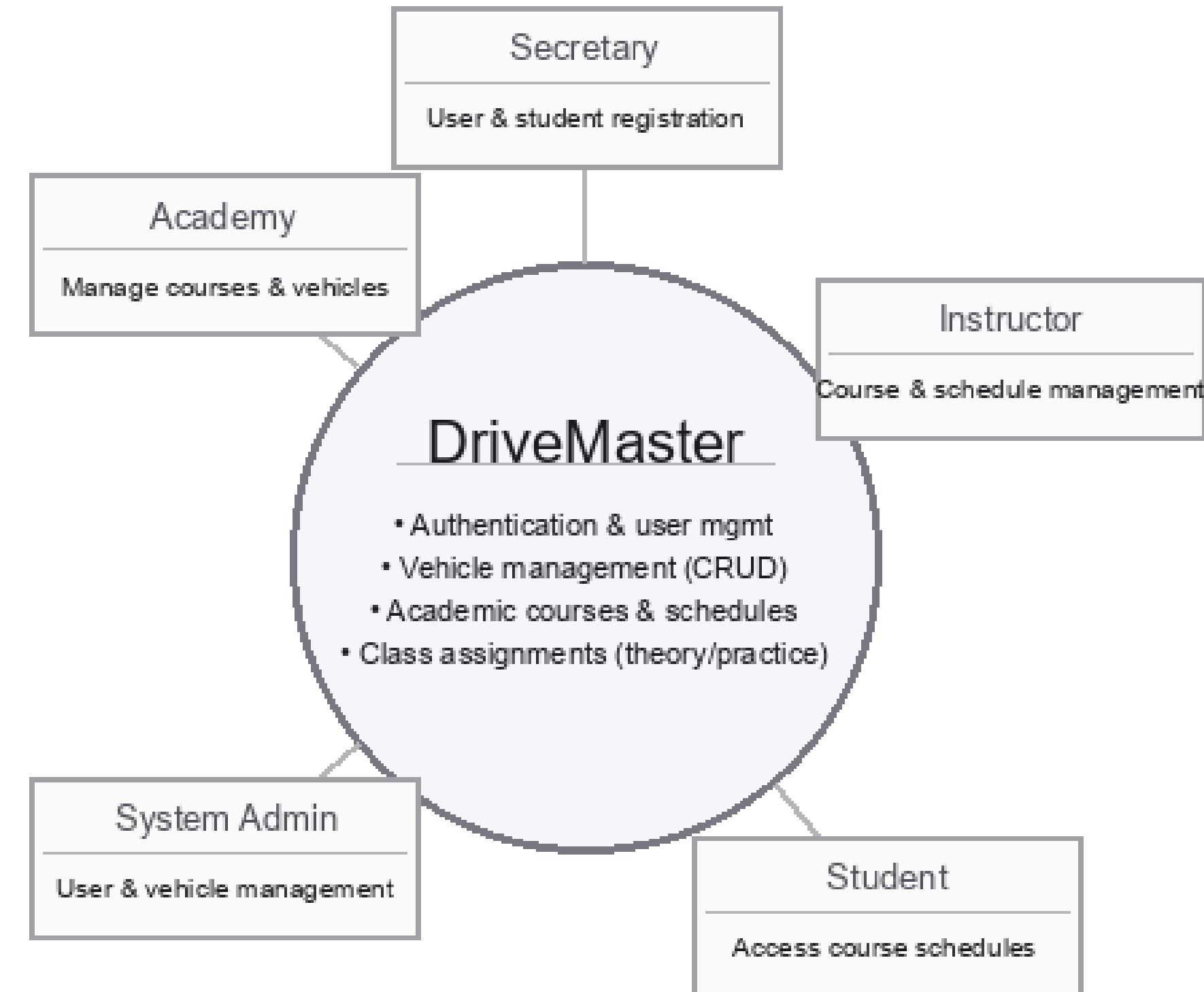
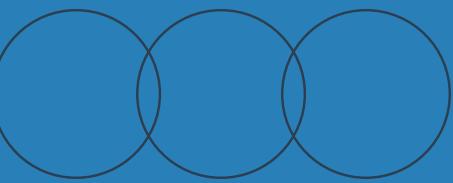
## IDENTIFY PROBLEMS

- Frequent human errors
- Document loss
- Regulatory non-compliance
- Excessive management time

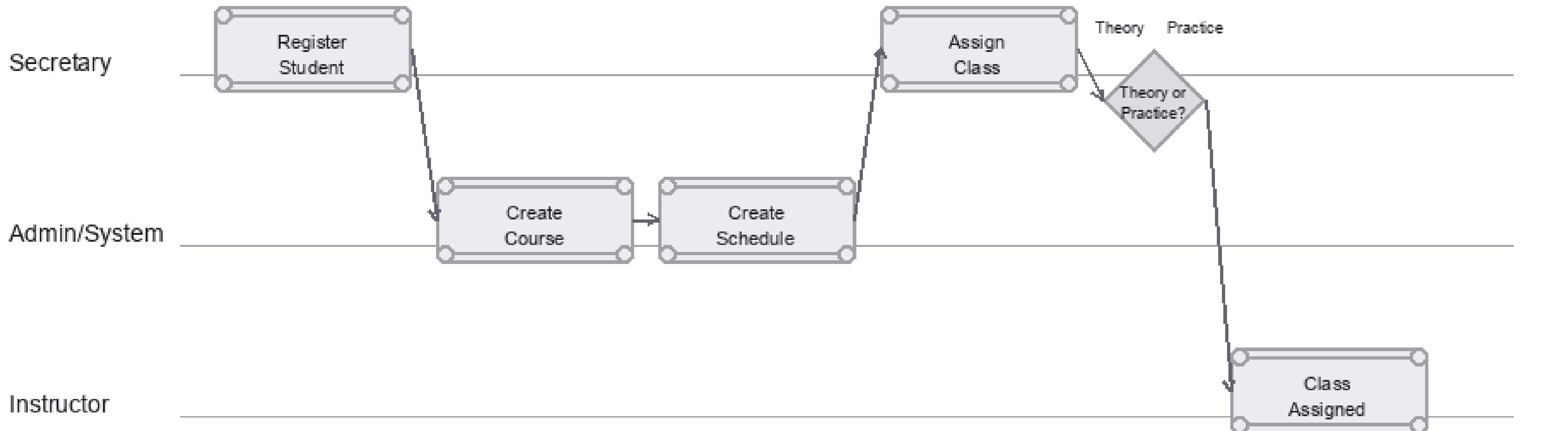
## NEEDS

- Centralize student registration
- Automate hour tracking (theoretical & practical)
- Comply with Colombian regulations (licenses A1-C3)
- Generate certificates automatically
- SaaS model for multiple academies

# DriveMaster:Final Product and Stakeholders



# DriveMaster: Key Business Process



# Key Technical Decisions

Technology	Purpose	Reason
<b>Java Spring Boot</b>	Authentication & role management	Enterprise-grade security and robust JWT handling
<b>Python FastAPI</b>	Business logic service	Fast development, high performance, automatic validation
<b>Angular</b>	Web frontend	Full-featured TypeScript framework, modular and maintainable
<b>MySQL + MongoDB</b>	Hybrid persistence	MySQL for ACID authentication data; MongoDB for flexible operational data
<b>Docker + Compose</b>	Containerization & orchestration	Portability, reproducible environments, simplified modular deployment

# Agile Roadmap (Timeline)

Sprint	Deliverable
<b>Sprints 1-2</b>	MVP: Authentication and basic registration
<b>Sprints 3-4</b>	Student and instructor management
<b>Sprints 5-6</b>	Tracking of training hours and vehicle assignments
<b>Sprints 7-8</b>	Certificate generation module
<b>Sprints 9-10</b>	Containerization + CI/CD pipeline
<b>Sprints 11-12</b>	Performance testing and final adjustments

# Implementation & Results

- **Testing Strategy**

- Unit Tests:
  - Java → JUnit + Mockito
  - Python → Pytest
- Integration Tests: Postman workflows
- Performance Tests: JMeter load scenarios

- **CI/CD Pipeline (GitHub Actions)**

- Commit → Automated tests → Build → Containerized deployment

- **Test Coverage**

- Java: 85% (authentication, JWT, roles)
- Python: 78% (scheduling, sessions, vehicles)
- Automated pipeline running successfully
- JMeter validated all core endpoints with >90% success rate
- Full platform containerized and ready for deployment

# Key Conclusions

- DriveMaster effectively addresses real operational challenges in driving academies by automating student tracking, scheduling, instructor assignment, and compliance with regulatory training hours.
- The microservices architecture (Spring Boot + FastAPI) enables independent scalability, fault isolation, and maintainability across the platform.
- The agile methodology with short sprints allowed continuous delivery of functional increments, ensuring fast iteration and validated progress.
- Containerization (Docker) and CI/CD pipelines ensure consistent deployments, reproducible environments, and reliable system integration.

THANKS FOR THE ATTENTION  
INVITATION

