

# FINAL EXAM

Luca Novello | SYD466V1A | Professor Navid Mohaghegh  
Software Analysis and Design - SYD466 Final Exam - 100 Marks  
Duration: 3 hours

---

## Question 1: Business Requirement Gathering for Future Mode of Operation (FMO) System (30 Marks)

- *Conduct a thorough business requirement analysis for a future mode of operation (FMO) system aimed at enhancing the stability and security of the current dealership website such as <https://www.autotrader.ca>. (15 Marks)*

### **Business Requirement Analysis:**

#### **Stakeholder Analysis:**

- **Buyers:** Want reliable, fast browsing, secure login/payment, verified listings.
- **Dealerships:** Require secure, intuitive platforms to list/manage vehicles.
- **Site Admins:** Need real-time monitoring, fraud detection, and performance logs.
- **IT Operations:** Require scalable, fault-tolerant architecture.
- **Third-party Partners:** Integrate with financing, insurance, and ad platforms.

#### **Current System:**

- Frequent downtime under high traffic
- Outdated user authentication and encryption methods
- Lack of real-time fraud detection and audit trail
- Poor admin control over listed content

#### **Business Requirements:**

- Ensure high availability and performance
- Enforce end-to-end encryption and OAuth2 authentication
- Integrate fraud detection
- Real-time logging and admin moderation tools
- Role-based access control for different user types

#### **Success Criteria:**

- More consistent uptimes
- faster average page load time
- lower fraudulent post ratio
- Full system adoption across all dealerships

- *Produce a Requirements Specification Document and a Feature Roadmap. (15 Marks)*
- *Consider the depth of requirement analysis, the inclusion of diverse stakeholder perspectives, the logic behind feature prioritization, and the quality of documentation.*

### **Requirements Specification Document:**

- **Functional Requirements:**
  - Vehicle browsing and filtering
  - Dealership user management
  - Secure checkout/payment integration
  - Real-time alerts for moderation
  - Analytics and reporting dashboards
- **Non-Functional Requirements:**
  - Security (OAuth2, SSL/TLS)
  - Availability (Cloud failover, CDN support)
  - Performance (Auto-scaling, load balancing)
  - Compliance (PIPEDA, PCI-DSS)

### **Feature Roadmap:**

- 1. Phase 1 - Core Stability & Security (Month 1–2)**
    - Implement new cloud-based hosting with monitoring
    - Add OAuth2 login and TLS 1.3 encryption
  - 2. Phase 2 - Admin Tools & Moderation (Month 3–4)**
    - Real-time content moderation and audit logs
    - Admin role dashboard with permissions
  - 3. Phase 3 - Dealer Services & Integrations (Month 5–6)**
    - Dealer portal revamp
    - Third-party financing and insurance API integration
  - 4. Phase 4 - Fraud Detection & Analytics (Month 6–7)**
    - AI-driven fraud detection
    - Business insights dashboards for dealers/admins
-

## Question 2: Budgeting, Design, and Risk Analysis for the above FMO Migration (30 Marks)

- *Develop a high-level budget plan and project timeline, assuming hypothetical costs. The objective is to assess your consideration of different technologies, resources, and areas required for the migration. Create a high-level system design, perform a risk analysis, and prepare an RFP document for selecting implementation vendors. (15 Marks)*
- *Deliver a high-level Project Budget Document, Gantt Chart, High-Level Design Overviews, a Risk Analysis Report, and an RFP Document (15 Marks).*
- *Please note that briefly listing and explaining the considered areas is sufficient, as it's understood that producing these documents in detail is not feasible within a three-hour exam. Kindly start with a table of content of the main topics you like to cover and then expand on your answers as time allows.*

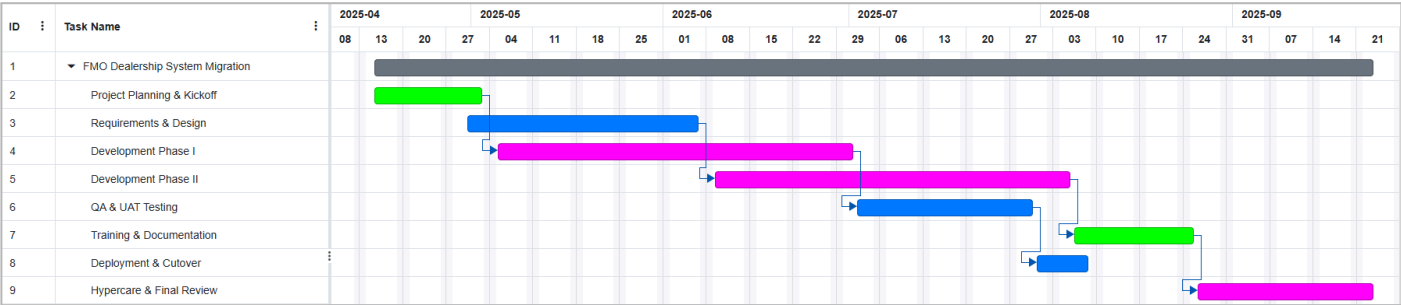
### **1. High-Level Budget Plan**

Category	Estimated Cost	Notes
Frontend/Backend Dev	\$120,000	React, Node.js, API integration
UI/UX Design	\$15,000	Responsive & accessible UI
Security Audit	\$10,000	Penetration testing, 3rd party audit
Cloud Hosting (Year 1)	\$10,000	AWS EC2, RDS, CloudFront
Project Management	\$20,000	Agile Scrum PM over 7 months
Contingency Buffer (10%)	\$17,500	Unexpected costs
<b>Total</b>	<b>\$192,500</b>	

## 2. Project Timeline

Phase	Duration	Deliverables
Planning & Requirement Gathering	2 weeks	BRD, stakeholder sign-off
Design & Prototyping	1 month	Wireframes, system design
Core Dev Phase I	6 weeks	Hosting setup, security modules
Dev Phase II	6 weeks	Admin tools, dashboards
QA & Testing	3 weeks	Unit, Integration, UAT
Deployment	1 week	Live system, DNS setup
Monitoring & Support	3 weeks	Feedback loop, bug fixes

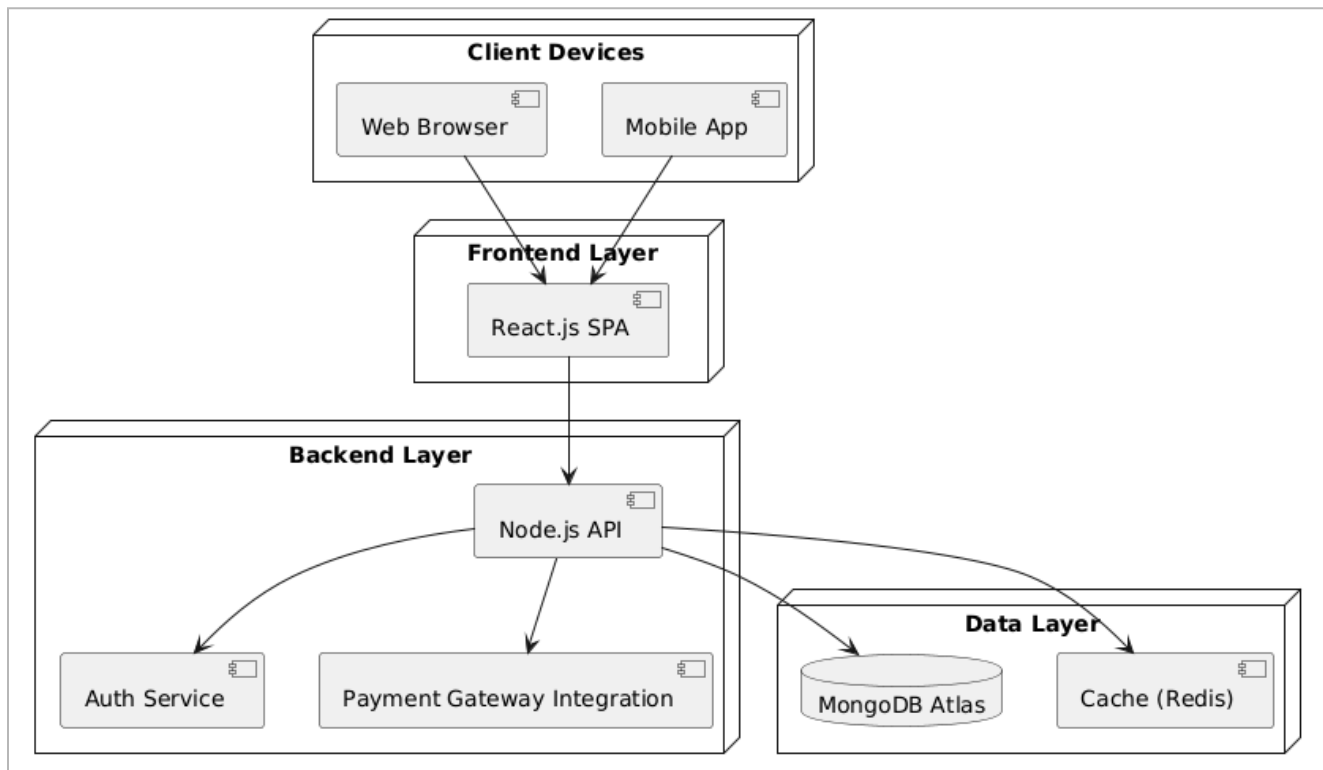
## Gantt Overview



### **3. High-Level System Design**

The system is designed using a modular, multi-tier architecture that separates concerns across client, frontend, backend, and data layers to support scalability, performance, and maintainability for the upgraded dealership platform.

- **Frontend:**
  - Built with React with Redux
  - Single Page Application (SPA) delivers a fast, responsive experience
  - UI rendering, routing, and API communication
- **Backend:**
  - Node.js (REST API)
  - Express
- **Database:**
  - PostgreSQL for storing structured data like users, listings, and messages.
  - Redis cache to optimize performance for high-traffic queries
- **Security:**
  - OAuth2
  - JWT tokens
  - HTTPS
- **Monitoring:**
  - New Relic for application-level performance tracking and error diagnostics
  - AWS CloudWatch to monitor infrastructure metrics, logs, and alerts



## **4. Risk Analysis Report**

Given the scope and complexity of upgrading the dealership platform to a secure, scalable cloud-based system, several risks have been identified. Each risk is evaluated based on its potential impact on system performance, delivery timelines, and stakeholder adoption.

Risk	Likelihood	Impact	Mitigation Strategy
Data Breach	Medium	High	End-to-end encryption, regular audits
Cloud Downtime	Low	High	Multi-zone deployment, auto failover
Scope Creep	High	Medium	Sprint planning, change control process
Staff Resistance	Medium	Medium	Training, early demos

## **5. RFP Document Overview**

### **Introduction:**

The goal of the Request for Proposal (RFP) is to identify and contract a qualified vendor capable of implementing the Future Mode of Operation (FMO) for a dealership platform. This includes modernizing the site architecture, ensuring security and scalability, and supporting features such as listing management, messaging, and reporting.

**Vendor Requirements:** To be considered, vendors must demonstrate:

- Proven experience in building secure, scalable web applications with robust authentication and data protection measures.
- Domain knowledge in dealership systems or similar eCommerce platforms, showing an understanding of listing-based marketplaces and user interactions.
- Full-stack development capabilities, covering frontend, backend, database, and API integration.
- Experience in cloud-based deployment and migration, including support for high availability and disaster recovery.
- Ability to provide comprehensive documentation, end-user training, and ongoing technical support.

### **Proposal Deliverables Should Include:**

- A detailed technical proposal outlining architecture, tools, and frameworks.
- Project timeline with key milestones and delivery dates.

- Cost breakdown across phases (design, development, testing, deployment).
- Team bios and relevant case studies.
- Support and warranty terms.

**Evaluation Criteria:** Proposals will be evaluated based on the following weighted factors:

- **Technical Fit (30%):** How well the proposed solution aligns with system requirements and architectural vision.
  - **Security Expertise (25%):** Demonstrated ability to implement secure user authentication, data protection, and secure integrations.
  - **Timeline Feasibility (20%):** Realistic scheduling and adherence to delivery milestones.
  - **Total Cost (15%):** Overall value for money, including transparency of pricing.
  - **Post-launch Support (10%):** Availability of support staff, SLAs, and maintenance terms.
- 

### Question 3: System Design and Implementation (40 Marks)

- *Finalize the System Design Specifications (SDS) with UML (Unified Modeling Language) and ERD (Entity-Relationship Diagram) diagrams for the above. Outline test cases and a deployment plan, including recovery and rollback plans. Deliver an SDS document, TDD (Test-Driven Development) Plan, Deployment Plan, and Recovery and Rollback Documentation. Briefly listing and explaining the considered areas is sufficient, given the time constraints of the exam.*
- *Please make sure your UMLs and ERDs are matching what you covered in question 1 and 2.*

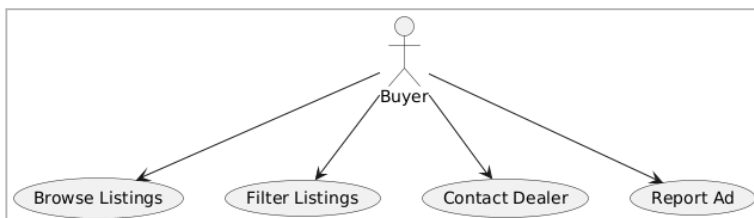
#### 1. SYSTEM DESIGN SPECIFICATION (SDS)

- **User Roles:** Buyer, Dealer, Admin
- **Core Modules:** Auth, Listings, Moderation, Analytics
- **Security:** OAuth2 login, rate limiting, logs
- **Scalability:** Load balancer, CDN, horizontally scalable backend

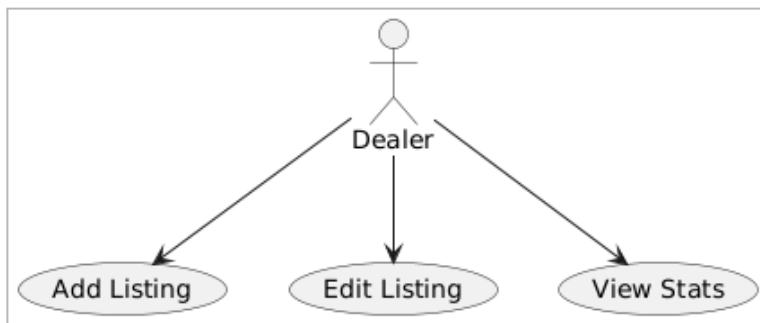
#### 2. UML DIAGRAMS

##### Use Case Diagram:

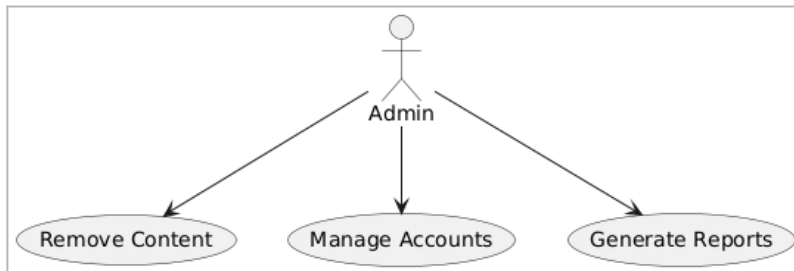
- **Buyer:** Browse, filter, contact dealer, report ad



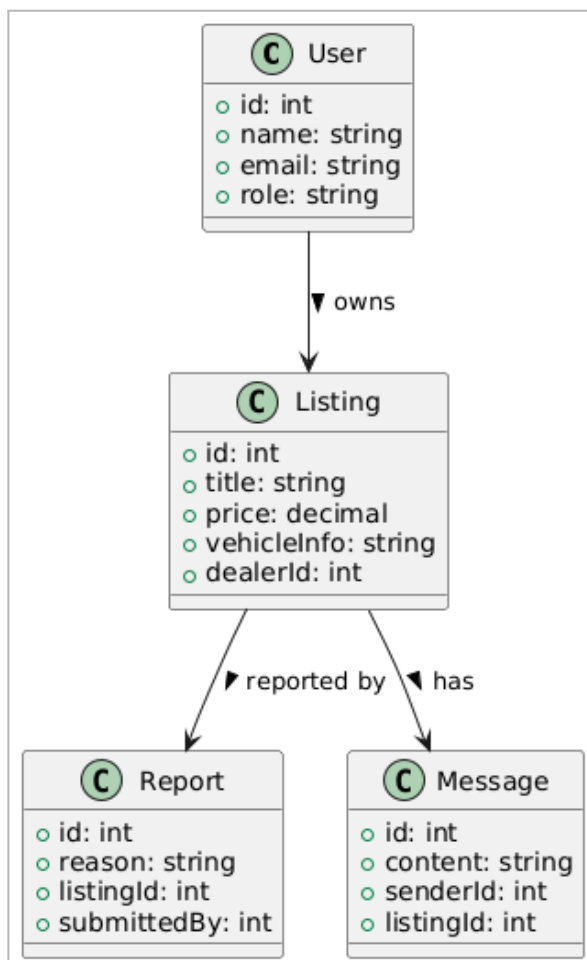
- **Dealer:** Add listing, view stats, edit listing



- **Admin:** Remove content, manage accounts, generate reports



### Class Diagram:



#### • **User:**

- id
- name
- email
- role

#### • **Listing:**

- id
- title
- price
- vehicleInfo
- dealerId

#### • **Report:**

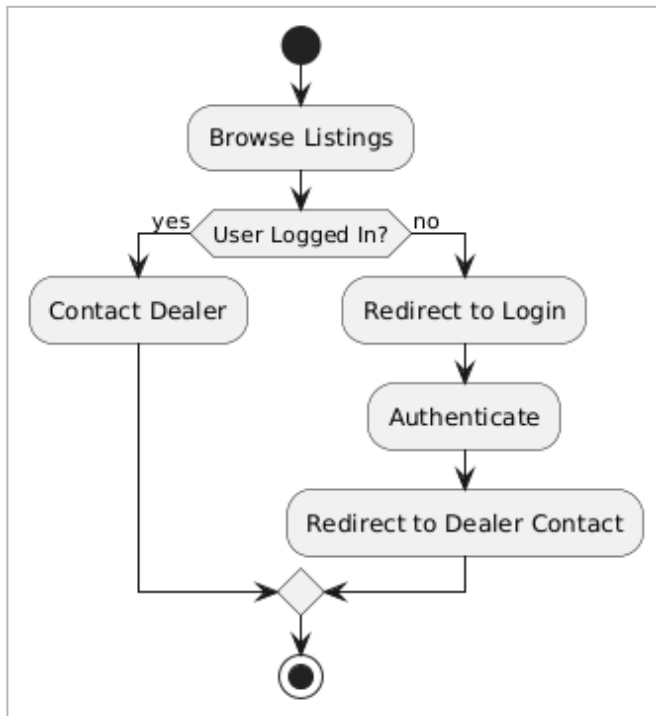
- id
- reason
- listingId
- submittedBy

#### • **Message:**

- id
- content
- senderId
- listingId



### Activity Diagram:

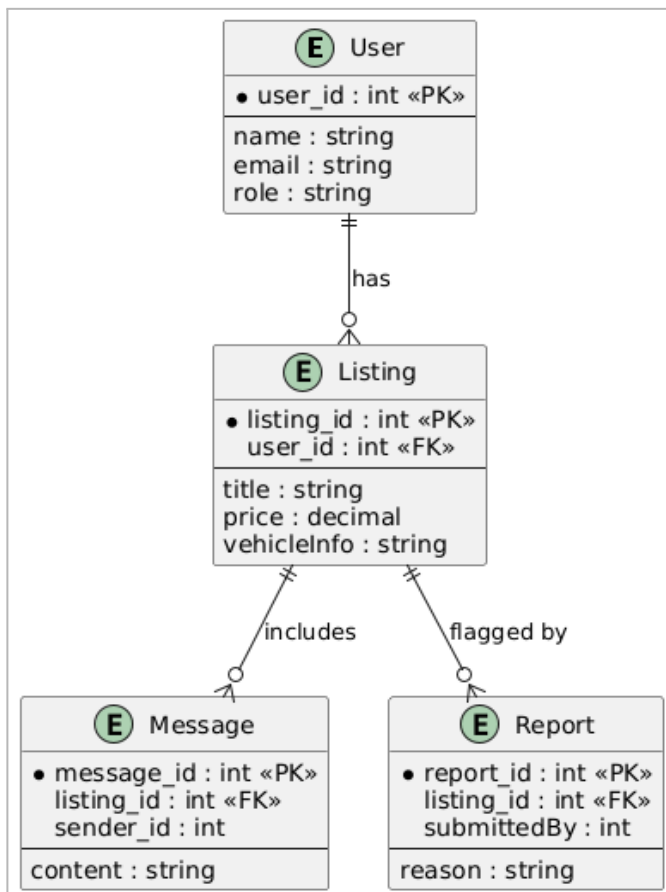


The diagram illustrates the process a user follows to contact a dealer on a vehicle listings website.

- The **User** browses listings.
- We check to see whether the user is logged in.
  - If the **user** is logged in, they can directly contact the **dealer**.
  - If not, they are redirected to the login page, complete authentication, and are then redirected to the **dealer** page.
- The process ends after the contact has been initiated.

The diagram shows both authenticated and unauthenticated user paths. It models a common access control pattern in web applications.

### 3. ENTITY-RELATIONSHIP DIAGRAM (ERD)



- **User** (user\_id PK) → **Listing** (listing\_id PK, user\_id FK)
- **Listing** → **Message** (message\_id PK, listing\_id FK)
- **Listing** → **Report** (report\_id PK, listing\_id FK)

## **4. TDD PLAN**

Each feature is tested before it is implemented, promoting high code quality and early defect detection.

- **Unit Tests:**
  - Login
  - Add listing
  - Validate input
  - Filter query
- **Integration Tests:**
  - Login → add listing → query → message flow
- **Acceptance Tests:**
  - Stakeholders test typical flows
- **Tools:**
  - Jest
  - Cypress
  - Postman
  - GitHub Actions

## **5. DEPLOYMENT PLAN**

The deployment strategy focuses on ensuring a smooth, secure, and scalable rollout of the new dealership platform with minimal disruption to users.

- **Infrastructure Setup:**
  - AWS EC2 for backend services
  - RDS for relational data storage
  - S3 for static asset hosting
  - CloudFront for global content delivery
- **CI/CD:**
  - GitHub Actions to automate builds, tests, and deployment
  - PM2 to manage Node.js application processes in production
- **Monitoring:**
  - New Relic for performance tracking and alerting
  - AWS CloudWatch logs for infrastructure-level monitoring
- **Phased Rollout:**
  - Begin with a soft launch targeting a small group of key dealers
  - Gradually expand deployment based on system stability and feedback

## **6. RECOVERY AND ROLLBACK PLAN**

To ensure high availability and data integrity, this structured recovery and rollback strategy is in place to address potential system failures or deployment issues.

- **Backup Strategy:**
    - Daily snapshots of application and data
    - RDS multi-AZ backups for automatic redundancy and failover
  - **Rollback:**
    - Git version tags to revert backend code to a stable release
    - Previous frontend builds stored in S3 for immediate restoration
  - **Disaster Plan:**
    - DNS failover to redirect traffic to a backup environment
    - Manual database restore procedures tested and documented
    - Email alert system to notify administrators of critical failures
-