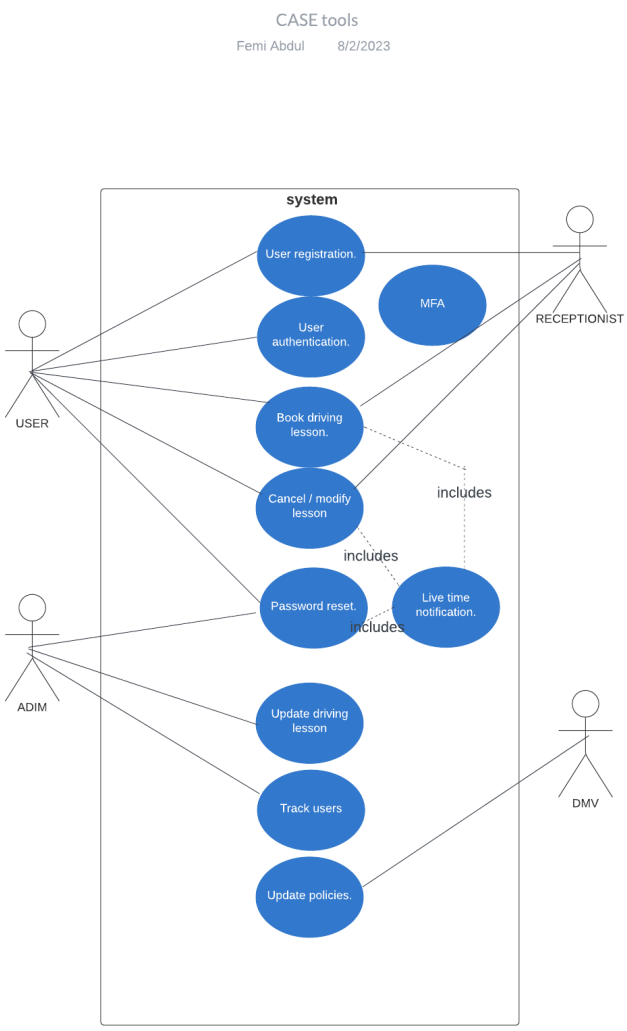
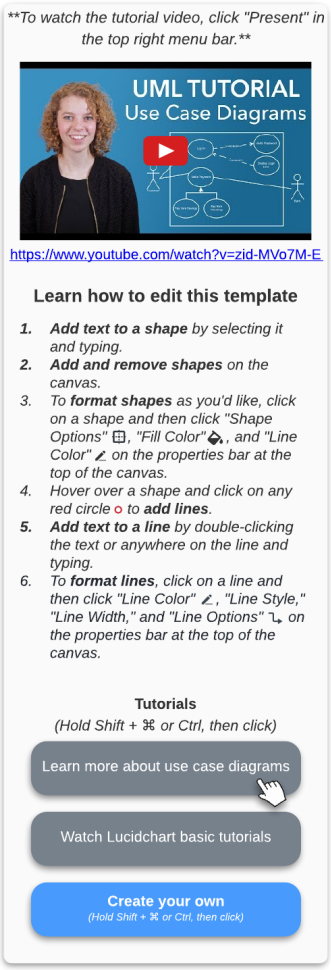
CS 255 System Design Document Template

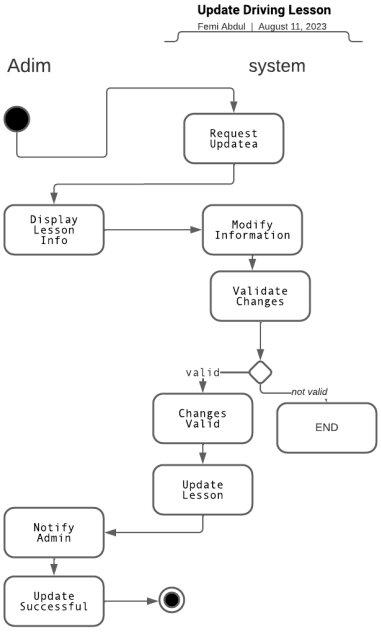
UML Diagrams

UML Use Case Diagram





UML Activity Diagrams



Learn how to edit this template

1. ***Add text to a shape*** *by selecting it and typing.*
2. ***Add and remove shapes*** *on the canvas.*
3. *To* ***format shapes*** *as you'd like, click on a shape and then click "Shape Options", "Fill Color", and "Line Color" on the properties bar at the top of the canvas.*
4. *Hover over a shape and click on any red circle to* ***add lines****.*
5. ***Add text to a line*** *by double-clicking the text or anywhere on the line and typing.*
6. *To* ***format lines****, click on a line and then click "Line Color", "Line Style," "Line Width," and "Line Options" on the properties bar at the top of the canvas.*

Tutorials

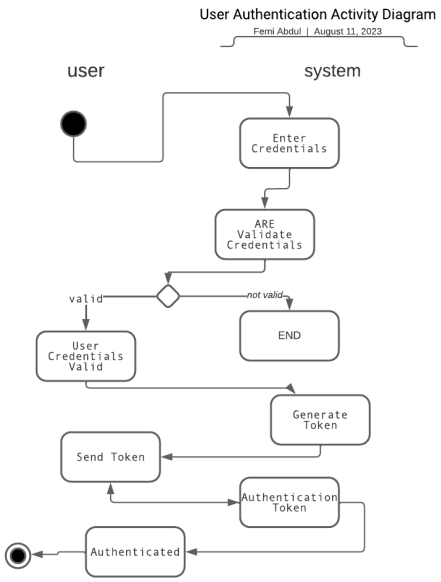
(Hold Shift + ⌘ or Ctrl, then click)

Read about UML diagram templates and examples

Visit our UML activity diagram page

Watch Lucidchart basic tutorials

See an example
(Hold Shift + ⌘ or Ctrl, then click)



Learn how to edit this template

1. ***Add text to a shape*** *by selecting it and typing.*
2. ***Add and remove shapes*** *on the canvas.*
3. *To* ***format shapes*** *as you'd like, click on a shape and then click "Shape Options", "Fill Color", and "Line Color" on the properties bar at the top of the canvas.*
4. *Hover over a shape and click on any red circle to* ***add lines****.*
5. ***Add text to a line*** *by double-clicking the text or anywhere on the line and typing.*
6. *To* ***format lines****, click on a line and then click "Line Color", "Line Style," "Line Width," and "Line Options" on the properties bar at the top of the canvas.*

Tutorials

(Hold Shift + ⌘ or Ctrl, then click)

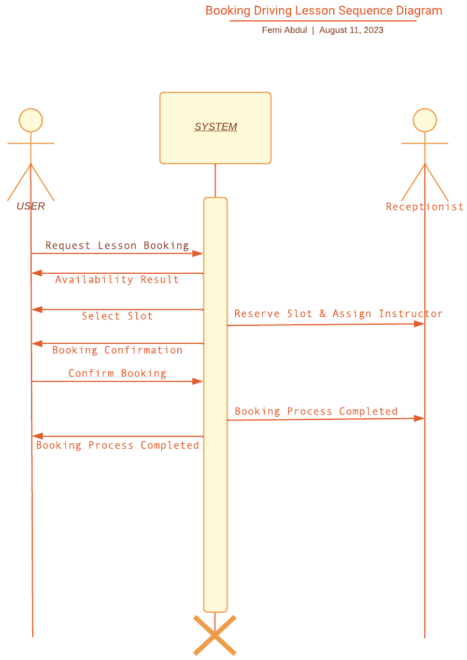
Read about UML diagram templates and examples

Visit our UML activity diagram page

Watch Lucidchart basic tutorials

See an example
(Hold Shift + ⌘ or Ctrl, then click)

UML Sequence Diagram



Learn how to edit this template

*To **manually create your sequence diagram**:*

1. ***Add text to a shape** by selecting it and typing.*
2. ***Add and remove shapes** on the canvas.*
3. *To **format shapes** as you'd like, click on a shape and then click "Shape Options", "Fill Color", and "Line Color" on the properties bar at the top of the canvas.*
4. *Hover over a shape and click on any red circle to **add lines**.*
5. ***Add text to a line** by double-clicking the text or anywhere on the line and typing.*
6. *To **format lines**, click on a line and then click "Line Color", "Line Style," "Line Width," and "Line Options" on the properties bar at the top of the canvas.*

*To **automatically create your sequence diagram**:*

1. *Click "</> Use Markup" in the UML Sequence shape library to the left.*
2. *Add your syntax.*
3. *Click "Build."*

Tutorials

(Hold Shift + ⌘ or Ctrl, then click)

Watch a tutorial on how to create sequence diagrams automatically

Read about our automatic UML sequence markup

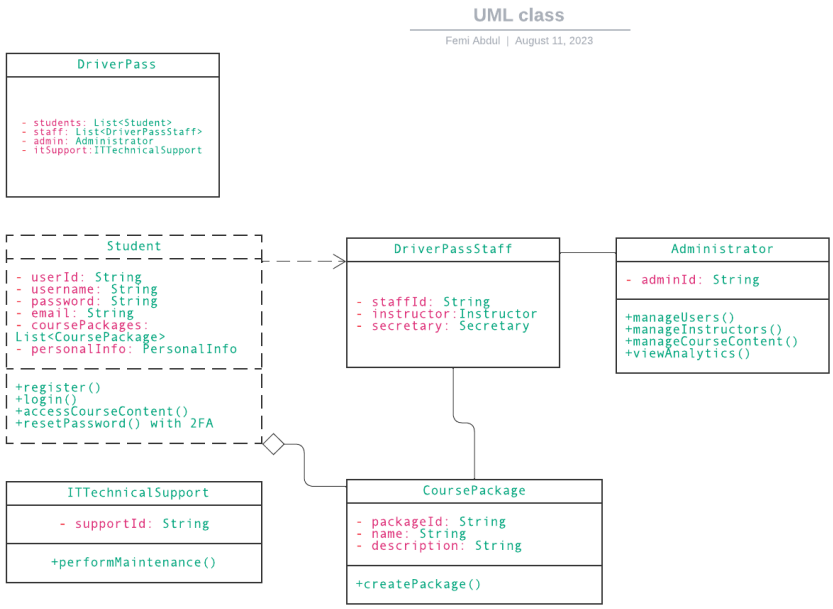
Watch a tutorial on how to make UML sequence diagrams

Watch Lucidchart basic tutorials

See an example

(Hold Shift + ⌘ or Ctrl, then click)

UML Class Diagram



Technical Requirements

Technical Requirements:

1. Hardware Requirements:

- Web Servers: To host the online platform and serve content to users.

- Database Servers: To store user data, practice exams, instructor profiles, and session information.

- Mobile Devices: The system should be accessible and functional on various mobile devices.

- Network Infrastructure: Reliable and high-speed internet connectivity to ensure seamless user interactions.

2. Software Requirements:

- Operating Systems: The system should be compatible with major operating systems like Windows, macOS, iOS, and Android.

- Web Browsers: Support for modern web browsers such as Chrome, Firefox, Safari, and Edge.

- Database Management System: To manage user data, exam records, instructor profiles, and session details.

- Programming Languages: The development of the system may involve languages like Java, JavaScript, HTML, and CSS.

- Server Software: Web server software (e.g., Apache, Nginx) and database management software (e.g., MySQL, PostgreSQL).

3. Tools and Frameworks:

- Front-End Frameworks: Such as React, Angular, or Vue.js for building user interfaces.

- Back-End Frameworks: Frameworks like Node.js, Django, or Ruby on Rails for handling server-side logic.

- Version Control: Tools like Git for collaborative development and source code management.

- Testing Frameworks: To ensure the quality and functionality of the system through automated testing.

- Security Tools: For implementing encryption, securing data, and protecting against potential vulnerabilities.

4. Infrastructure Requirements:

- Cloud Hosting: Consideration of cloud services like Amazon Web Services (AWS), Microsoft Azure, or Google Cloud Platform for scalability and reliability.

- Data Storage: Cloud-based storage solutions for managing user data, practice exams, and other system resources.

- Load Balancing: If needed, implement load balancing mechanisms to distribute user traffic evenly across servers.

- Security Measures: Implementation of firewalls, intrusion detection systems, and regular security audits to safeguard user data.

5. User Authentication and Security:

- Secure Authentication Mechanism: Implementation of a secure authentication process with strong password policies.

- SSL Encryption: SSL certificates to secure data exchange between clients and servers, ensuring privacy.

- Account Lockout: Mechanism to temporarily lock accounts after a certain number of failed login attempts.

- Password Recovery: Procedure to help users recover access to their accounts if they forget their passwords.

6. User Interface and Accessibility:

- Responsive Design: Designing the user interface to be responsive and adaptable to various screen sizes.

- Cross-Browser Compatibility: Ensuring that the user interface functions consistently across different web browsers.

- Mobile App Development (Optional): Consideration of developing mobile applications for iOS and Android platforms for enhanced user experience.

7. Performance and Scalability:

- Scalability Strategy: Implementation of strategies to accommodate a growing user base without compromising performance.

- Caching Mechanisms: Use of caching to improve response times and reduce server load.

- Load Testing: Conducting load testing to ensure the system can handle a large number of simultaneous users.

8. Backup and Recovery:

- Regular Backups: Scheduled backups of the database and critical system components to prevent data loss.

- Disaster Recovery Plan: A plan for recovering data and system functionality in case of unforeseen incidents.

9. Maintenance and Updates:

- Version Control: Implementing version control for the system's source code to manage changes and updates.

- Regular Updates: Continuous updates to ensure practice exams remain accurate, and the system's security is maintained.

These technical requirements are derived from the Business Requirements Document's functional and non-functional requirements. Both administrators and users will have access to secure, efficient, and user-friendly equipment, software, and infrastructure for the proposed system to meet the project's objectives and goals.

]