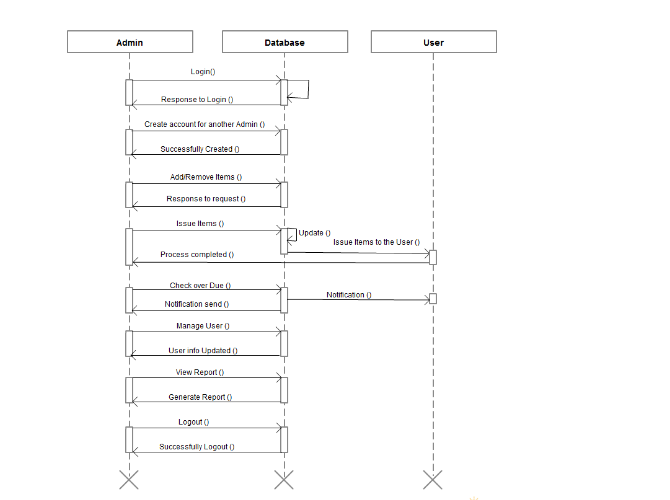
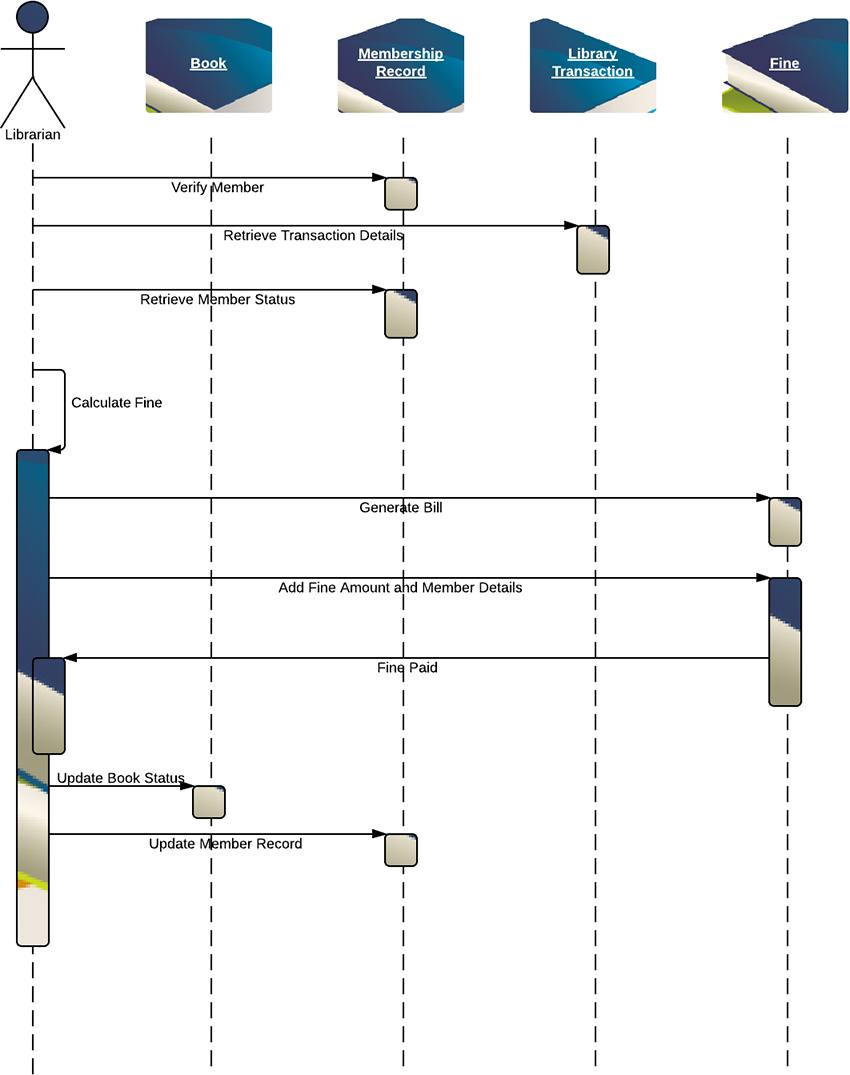
**Library Management System Activity Diagram**



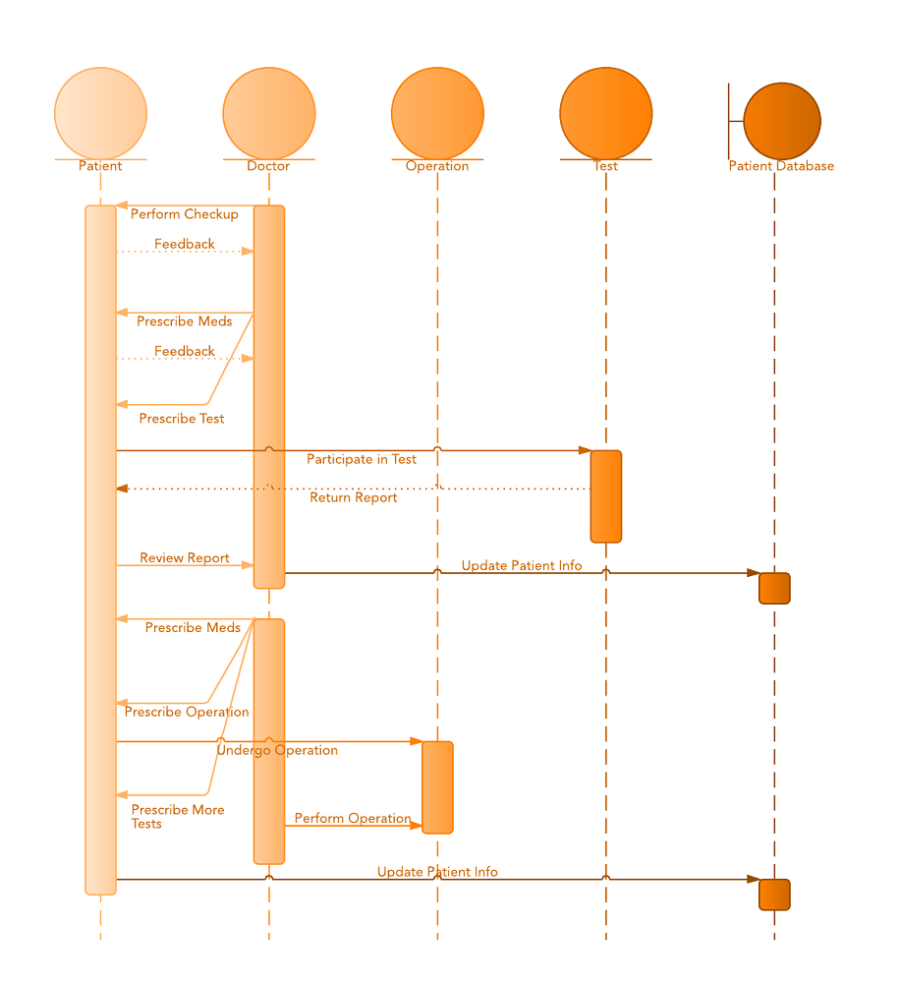
## Hospital Management System Activity Diagram



## Library Management System Sequence Diagram



## Hospital Management System Sequence Diagram



## [Writing a Software Requirements Specification Document](http://www.onedesk.com/2014/02/writing-a-software-requirements-specification-document/)

### What is a Software Requirements Specification Document?

Many developers choose to work with a software requirements specification document as it typically contains the following:

– A complete description of the software’s purpose and functionality  
– Details as to how the software will perform in terms of speed, response time, availability, portability, maintainability, recovery speed and more  
– Use cases of how users will use the software  
– The definition of how the application with interact with other hardware and program  
– Non-functional requirements (e.g: performance engineering requirements, quality standards, or design constraints)

### Why is it important?

A SRS allows developers to be clear on the goals of the software and on what they should focus on. Furthermore, it allows them to:

– Save time on communication  
– Minimize development efforts  
– Gives the customer feedback  
– Eliminate task duplication  
– Facilitate the transfer to new users or to new machines  
– Breaks problems down into parts  
– Serves as the main document to verify the validation and testing processes  
– Referring to past SRS documents helps identify deficiencies and process flaws.

### How to Write a Software Requirements Specifications Document

There is no standard way of writing a requirements specifications document, but here are a few guidelines:

**Create an SRS outline**  
If you do not already have an SRS template, there are many you can find on the web. Use a template to create an outline for you SRS doc. Modify it to suit your organization’s needs.

SRS outlines vary, depending on the organization and their processes. Some may be simple, while others are more detailed and complex.

**Here is an example of a simple SRS outline:**  
1. Purpose  
2. Scope  
3. System Overview  
4. References  
5. Definitions  
6. Use Cases  
7. Functional requirements  
8. Non-functional requirements

I got this outline from [this website](http://www.stellman-greene.com/aspm/content/view/35/41/). Check it out to see it in more detail.

Once outlined, the SRS is ready to be written. Here are some tips to writing an SRS:

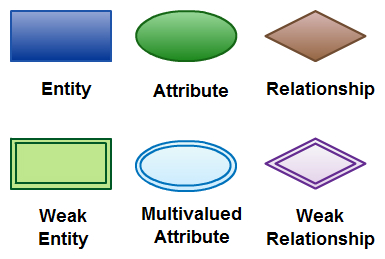
**Choose the best person to write it**  
The writer should have superior communication skills. The purpose of the SRS to make everyone understand the specifications. Anything that is unclear or miscommunicated can lead to not-so-great consequences. Many suggest having technical writers involved in the requirements specification process helps in preventing miscommunications. They are more skilled writers than developers, and they have an air for precision and clarity. Technical writers know how to gather and process the right information; they also know how to convey customer requirements.

**Make things visual**  
A picture can save 1000 words. Include graphics such as tables and charts to communicate your ideas better.

**Don’t over-document**  
Avoid including things that may not need to be documented. SRS documents may get a bit long, so avoid packing in unnecessary information.

**Keep an online version of the SRS and keep updating**  
As your tasks progress and if your staff and process changes, the SRS will need to be updated. For this reason, keeping a virtual version will help keep the whole team on the same page every time a change is made.

### ER Diagram Symbols and Notations

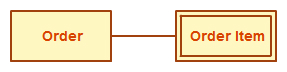


There are three basic elements in an ER Diagram: entity, attribute, relationship. There are more elements which are based on the main elements. They are weak entity, multivalued attribute, derived attribute, weak relationship and recursive relationship. Cardinality and ordinality are two other notations used in ER diagrams to further define relationships.

**Entity**

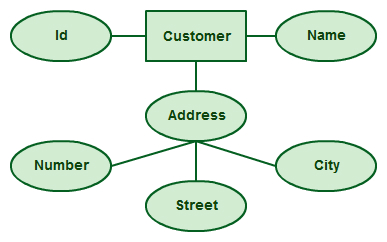
An entity can be a person, place, event, or object that is relevant to a given system. For example, a school system may include students, teachers, major courses, subjects, fees, and other items. Entities are represented in ER diagrams by a rectangle and named using singular nouns.

**Weak Entity :** A weak entity is an entity that depends on the existence of another entity. In more technical terms it can defined as an entity that cannot be identified by its own attributes. It uses a foreign key combined with its attributed to form the primary key. An entity like order item is  a good example for this. The order item will be meaningless without an order so it depends on the existence of order.



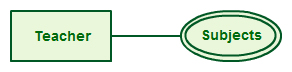
Weak Entity Example in ER diagrams

**Attribute :** An attribute is a property, trait, or characteristic of an entity, relationship, or another attribute. For example, the attribute Inventory Item Name is an attribute of the entity Inventory Item. An entity can have as many attributes as necessary. Meanwhile, attributes can also have their own specific attributes. For example, the attribute “customer address” can have the attributes number, street, city, and state. These are called composite attributes. Note that some top level ER diagrams do not show attributes for the sake of simplicity. In those that do, however, attributes are represented by oval shapes.



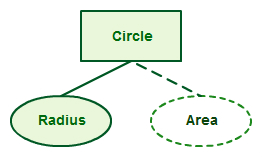
Attributes in ER diagrams, note that an attribute can have its own attributes ( composite attribute )

**Multivalued Attribute :** If an attribute can have more than one value it is called an multivalued attribute. It is important to note that this is different to an attribute having its own attributes. For example a teacher entity can have multiple subject values.

[](http://static2.creately.com/blog/wp-content/uploads/2012/03/Multivalued-Attribute-ER-Diagrams.jpeg)

Example of a multivalued attribute

**Derived Attribute :** An attribute based on another attribute. This is found rarely in ER diagrams. For example for a circle the area can be derived from the radius.

[](http://static1.creately.com/blog/wp-content/uploads/2012/03/Derived-Attribute-ER-Diagrams.jpeg)

Derived Attribute in ER diagrams

**Relationship:** A relationship describes how entities interact. For example, the entity “carpenter” may be related to the entity “table” by the relationship “builds” or “makes”. Relationships are represented by diamond shapes and are labeled using verbs.

[](http://static2.creately.com/blog/wp-content/uploads/2012/03/Relationship-ER-diagrams.jpeg)

Using Relationships in Entity Relationship Diagrams

# E-R Diagram of Library Management System

E-R Diagram for library management system

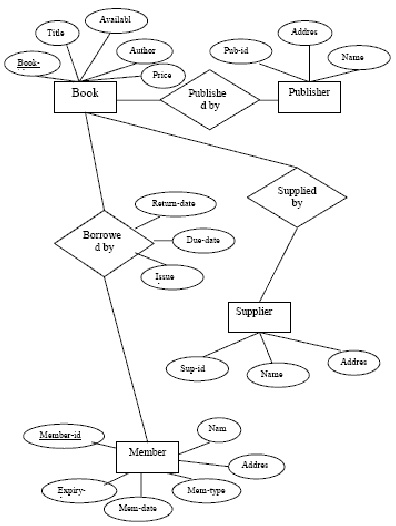
In the library Management system, the following entities and attributes can be identified.

•Book -the set all the books in the library. Each book has a Book-id, Title, Author, Price,and Available (y or n) as its attributes.

•Member -the set all the library members. The member is described by the attributesMember\_id, Name, Street, City, Zip\_code, Mem\_type, Mem\_date (date of membership),Expiry\_date.

•Publisher -the set of all the publishers of the books. Attributes of this entity are Pub\_id,Name, Street, City, and Zip\_code.

•Supplier -the set of all the Suppliers of the books. Attributes of this entity are Sup\_id,Name, Street, City, and Zip\_code.Assumptions: a publisher publishes a book. Supplier supplies book to library. Members borrowthe book (only issue).



**Block oriented elements**

**Definition:** The short definition is that block-level elements are elements that create blocks or large groupings of text.

block-level elements have some specific distinctions from inline elements:

* block-level elements generally can contain text, data, inline elements, or other block-level elements.
* block-level elements generally begin new lines of text.
* block-level elements inherit directionality information differently from inline elements.

**Block-level Elements**

A block-level element always starts on a new line and takes up the full width available (stretches out to the left and right as far as it can).

The <div> element is a block-level element.

Examples of block-level elements:

* <div>
* <h1> - <h6>
* <p>
* <form>

**Inline Elements**

An inline element does not start on a new line and only takes up as much width as necessary.

This is an inline <span> element inside a paragraph.

Examples of inline elements:

* <span>
* <a>
* <img>

**The <div> Element**

The <div> element is a **block-level element** that is often used as a container for other HTML elements.

The <div> element has no required attributes, but **style** and **class** are common.

When used together with CSS, the <div> element can be used to style blocks of content:

<html>

<body>

<div style="background-color:black; color:white; padding:20px;">

<h2>London</h2>

<p>London is the capital city of England. It is the most populous city in the United Kingdom, with a metropolitan area of over 13 million inhabitants.</p>

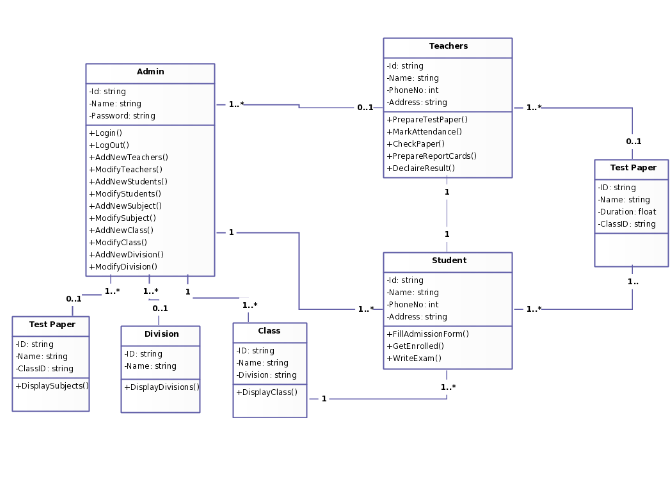
<p>Standing on the River Thames, London has been a major settlement for two millennia, its history going back to its founding by the Romans, who named it Londinium.</p>

</div>

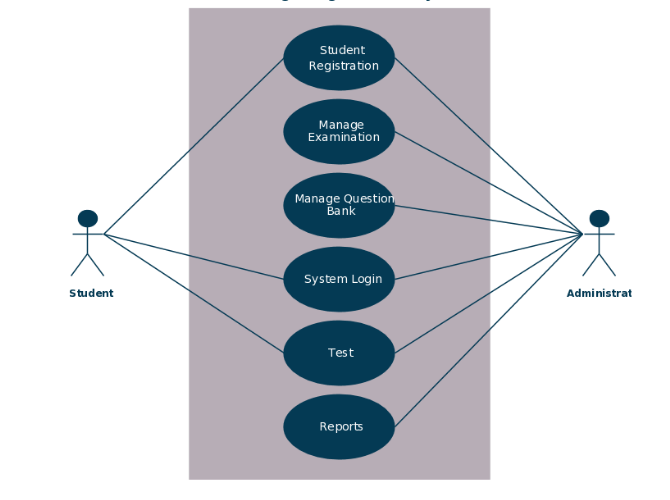
</body>

</html>

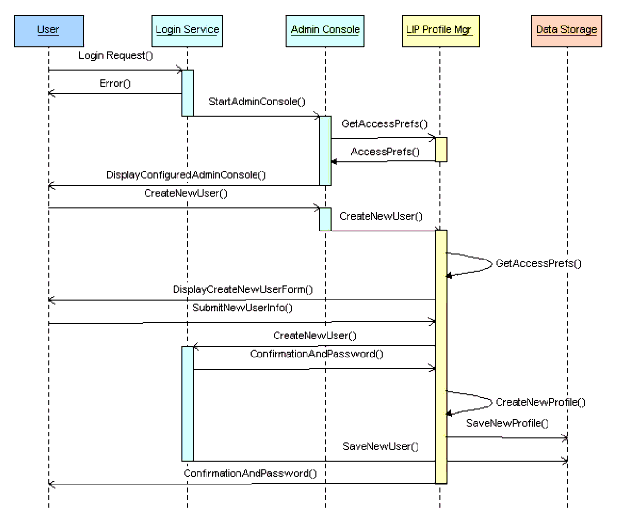
# Class Diagram for College Management System



# College Enrollment System



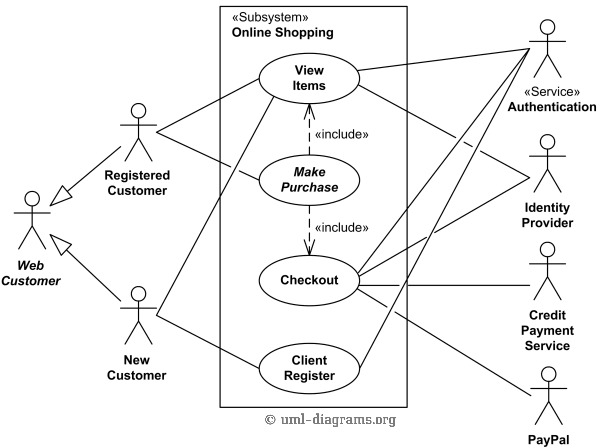
**Sequence diagram for college management system**



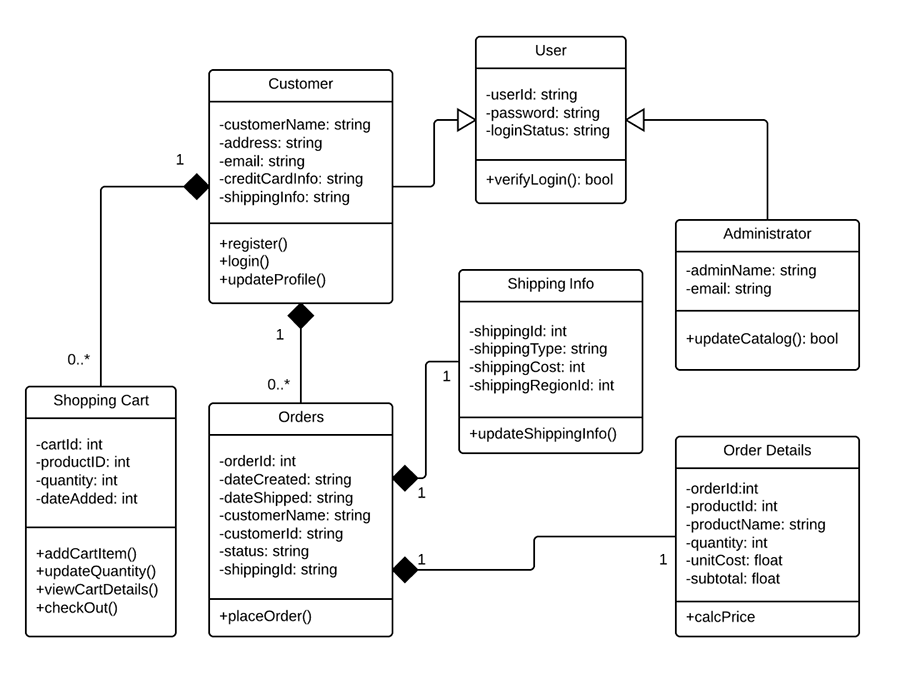
# Activity Diagram for Library Management System



# Use Case Diagram for Online Shopping



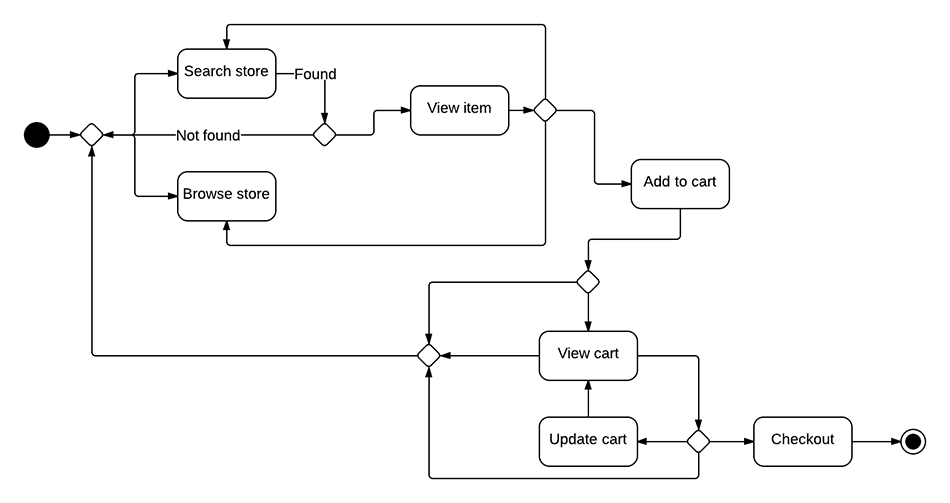
# Class Diagram for Online Shopping System



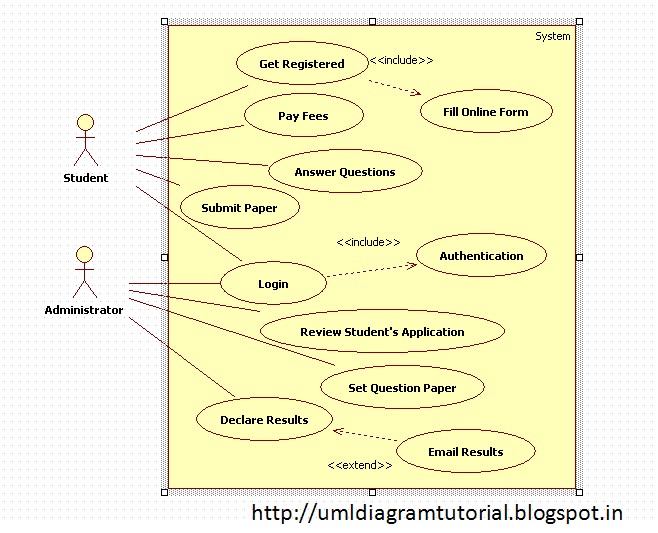
# Sequence Diagram for online shopping

# sequence diagram for online shopping system

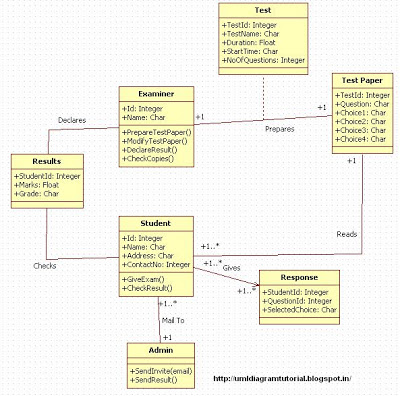
# Activity Diagram for Online Shopping System



### Online Examination - Use Case Diagram



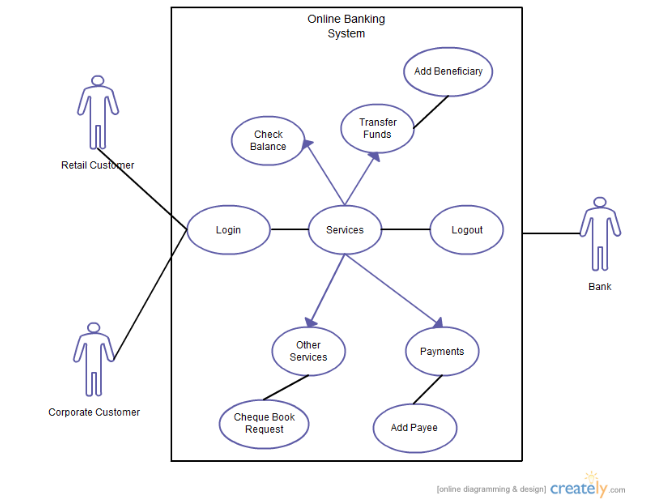
**Class Diagram for Online Examination System**



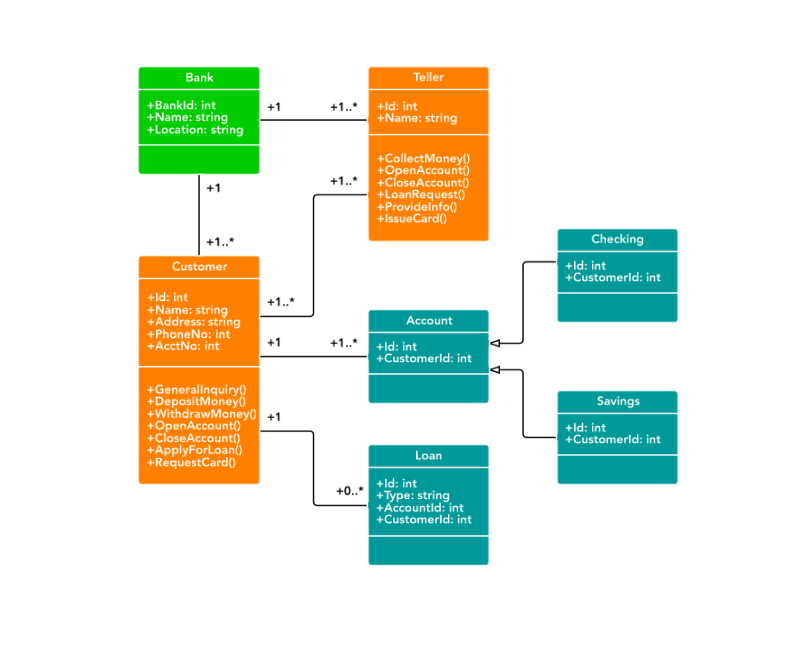
# Sequence diagram for online examination system

# http://4.bp.blogspot.com/-FIzvx15lykc/ULCvi_qXNiI/AAAAAAAAAcc/VkOb_gKLfNc/s1600/OnlineExaminationSystem-SequenceDiagram.jpeg

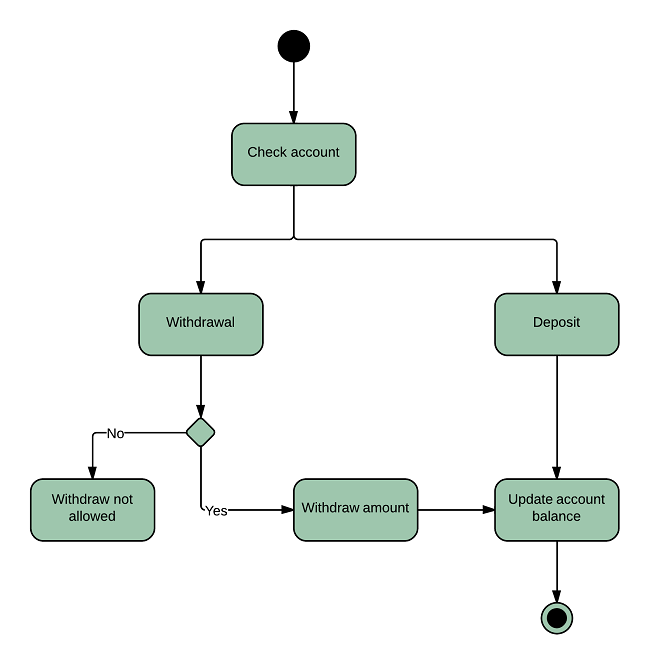
# Use Case Diagram for Banking System



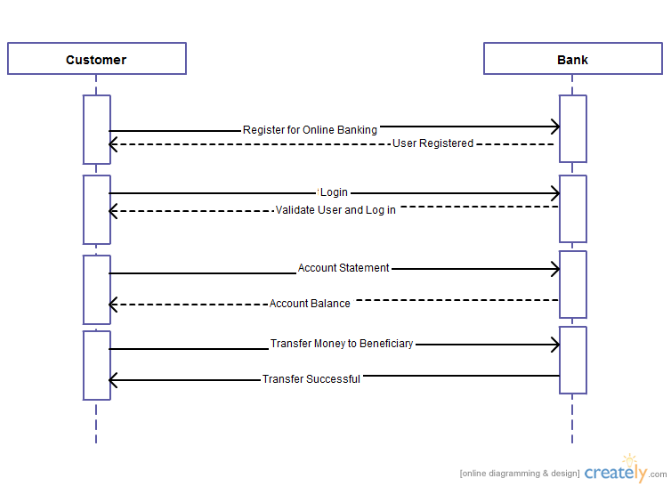
# Class Diagram for Banking System



# Activity Diagram for Banking System



# Sequence Diagram for Banking System

****