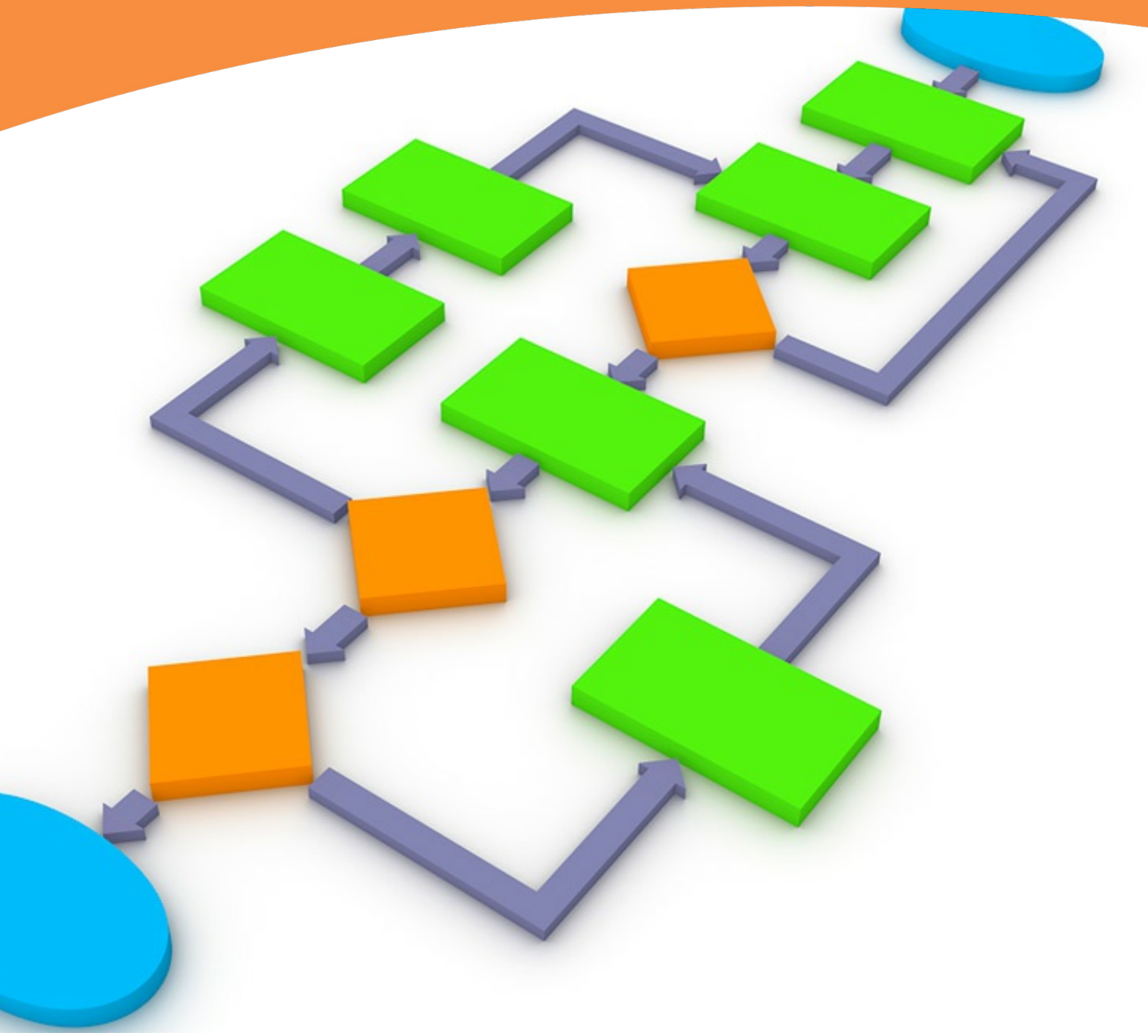
# Systems Analysis and Program Development

Mostafa Abd-ElHamid Atwa



bookboon
The eBook company

MOSTAFA ABD-ELHAMID ATWA

# SYSTEMS ANALYSIS AND PROGRAM DEVELOPMENT

Systems Analysis and Program Development
1st edition

ISBN 978-87-403-1836-4

# CONTENTS

# INTRODUCTION

Systems analysis is the process of designing and architecting a system before starting the implementation.

Systems analysis starts after gathering the requirements or making the requirements determination document from the user requesting any kind of systems whether it contains hardware or software or both of them with integration between components.

Systems analysis is giant process in the industry of software development. We can admit that without analyzing a system and documenting diagrams and role management, many of the systems may fail during the implementation process.

Throughout this book, we will learn what is systems analysis and program development which will be acting as the startup of a project using UML2.5, PHP.

# ABOUT THE AUTHOR

**Mostafa A. Hamid**

Information Systems Engineer
Systems Programmer
Massachusetts Institute of Technology Certified IOT Professional X
State University of New York at Potsdam Certified LPIC, RUP, CISSP and Certified in Java and PHP
Certified in Java from the American University in Cairo
Bachelor Degree Holder from Modern Academy for Computer Science and Management Technology

# ABOUT THE TECHNICAL AND OFFICIAL REVIEWER

Mohamed Abbarra

Technical Engineer at TechWorld
Bachelor or IS from The University of Hyderabad
**E-Mail:** mohamed.abarra.in@gmail.com

# 1 INTRODUCTION TO PROGRAMMING

Programming is a way to instruct the computer machine or any other machine and tell it what to do.

We use something called programming language to instruct the computer and tell it what to do.

The user of the output of the programming process also instructs the computer but using your program that you made using any programming language.

Programming language instructions are grouped together into something called functions, then the functions are grouped together into classes, then the classes are grouped together into packages or namespaces and the namespaces or directories reflect to directories where these files are located.

Organizing the packages or namespaces and functions with classes all together is the mission of the system architect and system analyst.

This process is the most headache in the head of creating a system because all of the implementation process is done with the aid and use of the analysis or the architecture the analyst or the architect is doing.

Some of the common methods which is well known in the programming and systems analysis and architecture is the MVC stands for Model-View-Controller and MVVM which is Model-View-View-Model, also there are other architectures that we will talk about in our next chapters throughout the book.

# 2 WHAT IS SYSTEM ANALYSIS?

System analysis is the process of guiding developers and programmers with database administrators and DevOps with the rest of the development team through the process of achieving business goals and delivering a piece of software that can satisfy customer needs.

The system analyst is the person who collects all the customer needs through the use of mockups or screens illustrating the system that can be drawn by the customer or by the analyst according to customer direction. Or through the use of text written by the customer as illustration of what each screen will be and the function of each and every one of these screens.

Also the outcome of the system and what the system will achieve.

Then the system analyst or the system architect draws the appropriate figures and diagrams to the programmers, database developers, and DevOps engineers throughout the process of implementation or development of the specified system. Also follows up back with the customers according to the customer needs with the modifications and existing work in progress being done by the developer to avoid system drop outs.

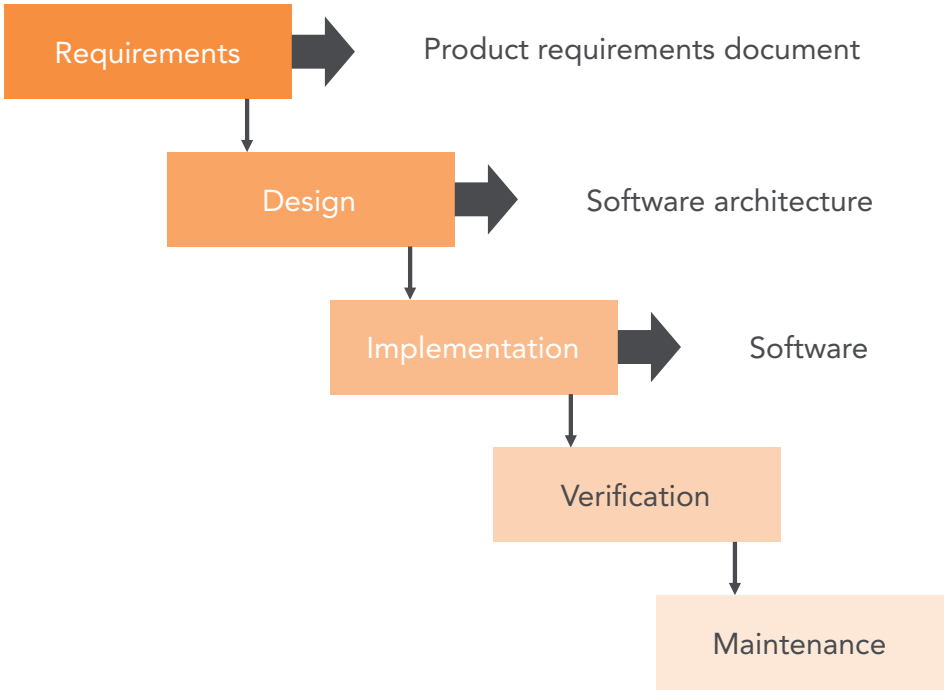
The diagrams differ from one method of development to another as follows:

Systems' **architects** often use one of five approaches in the software development life cycle:

1. Waterfall
2. Agile
3. Scrum
4. Lean
5. Iterative

Let us walk for some time throughout each and every one of these methods with graphs and illustrations as follows:

**The Waterfall**




The waterfall approach is a kind of software development as we can see in the figure starting from requirements determinations to design then implementation and verification and finally to the maintenance process.

This model is commonly used in the software industry long time ago and still being widely used by most of the companies with the agile development method.

This model involves the use of the following diagrams:
Use Case Diagrams (Management and User Centric)
Business Model Diagrams (Management and User Centric)
Data Flow Diagrams (DFD) (Management and User Centric)
Process Flow Diagrams (PFD) (Developer and Technical Centric)
Entity Relationship Diagram (ERD) (Developer and Technical Centric)
Structured English (Developer and Technical Centric)
Pseudo Code (Developer and Technical Centric)
Documentation (Developer, Technical and User and Management Centric)
UML Database Diagrams (Developer and Technical Centric)

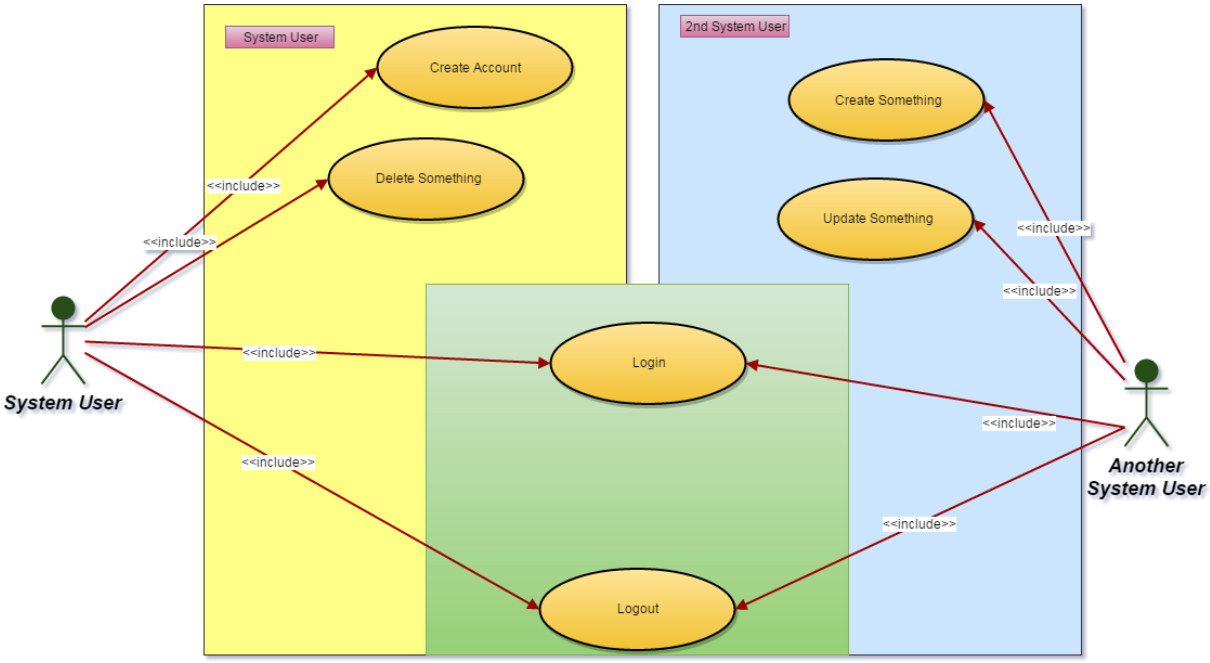
These are the main diagrams and methods as a system analyst to work with developers. Let us walk through all of these methods and how these diagrams are drawn as follows:

# 3 UML 2.5

**Use Case Diagrams**

Is a kind of a diagram that will show the reader how each use of the system will be and how will be interacting with other uses and of course the way the user will interact with the system.

Here is an example diagram of a use case drawn using http://www.draw.io as follows:



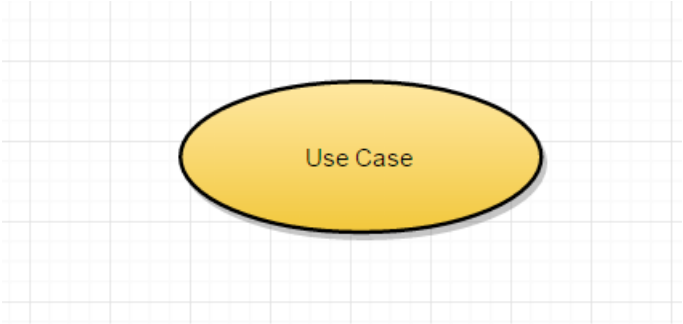

This is an example of a diagram that shows how 2 users interact with a system. As we can see in the previous diagram is that we have 3 main sections:

The first section contains the system user who will have the capabilities of login, logout, delete something, and create account privileges.

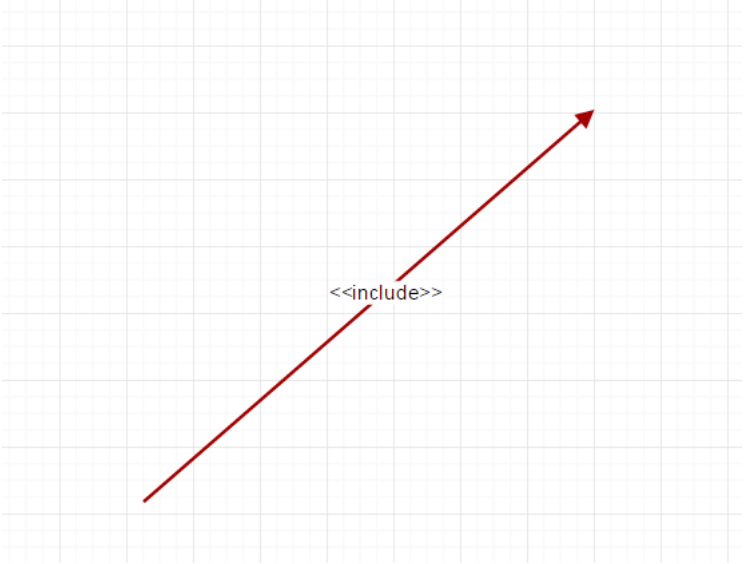
The second section in this diagram is 2nd system user who will have the capabilities of create something and update something.

The third section in this diagram is the intersection of both of the previous sections as both users will be intersecting in doing 2 use cases: login and logout.

The use case in the previous diagram represents an action into the system the user will perform and the symbol is:




The arrow with the <<include>> shows that the specified user will interact with the use case and the symbol is:



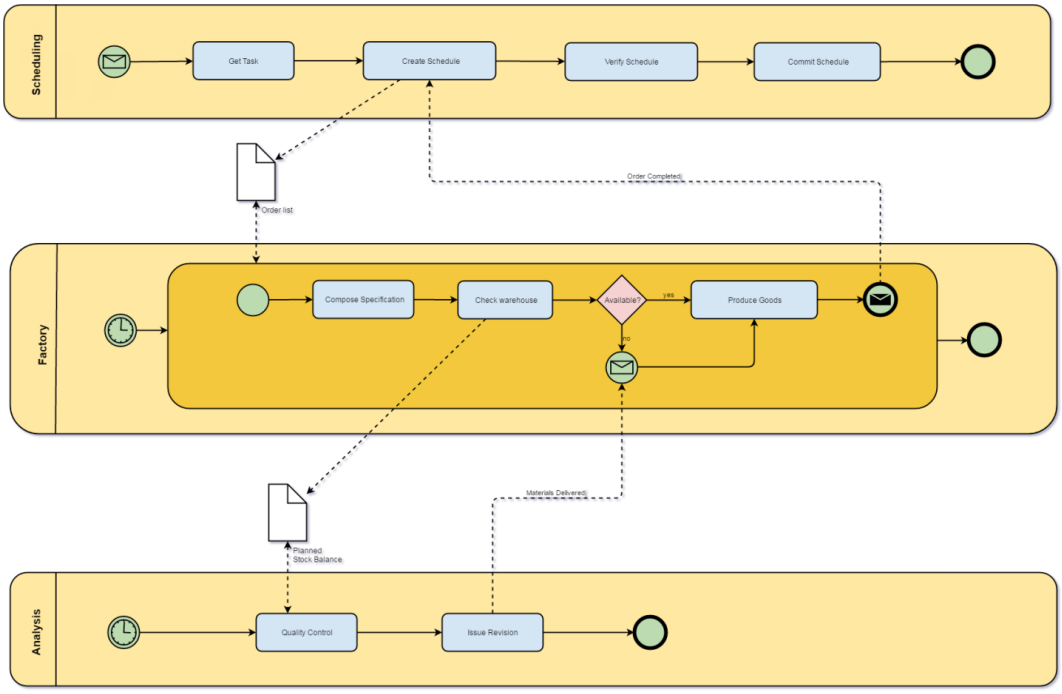

There are 2 main text indication types while working with UML 2.5 in general: include and extends.

If you feel while working with your UML diagram prior to version 2.2 that a use case is extended from another use case, then we use the <<extends>> text indication type, but if we have a separate use case that does not depend on another use case or not extended from the other use case, then we call text indicator: <<include>>

Also, you can change the text indicator type as you want to make it clear for the developer that will use your diagram any text of your choice.

### Business Model Diagrams

Below is an example of a business model diagram shows that how a factory-business analysis-scheduling flow of data as follows:



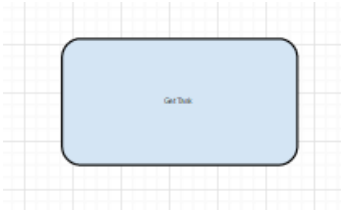
Now, let us see the items of this diagram as follows:

Timer indicates that there is a scheduled job to do and looks like follows:

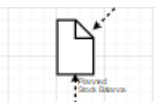
Next, we have the message symbol which indicates that there is a message should be sent to a specific user as follows:

Next, we have a big container that groups 2 or more processes or actions together as follows:

Next, we have the process or action which indicates a simple or complicated action that can be divided into one or more simpler actions as follows:

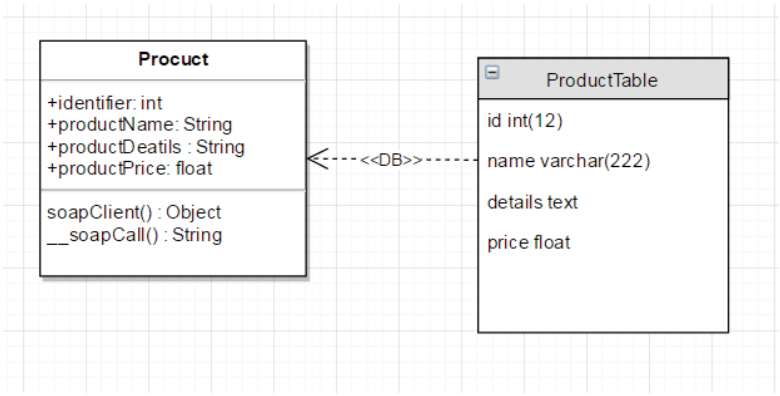


Next, we have the document icon which indicates the document to be sent to the user as follows:



After all, this is a kind of a diagram called business model diagram or business process diagram which covers the main concepts of how a system behaves and how the users of the system will be interacting with the system using input and output details.

Next, we will get to know how to model the database using UML 2.5.




As we can see in the previous class diagram, we find that the product class on the left side is a class in PHP as an example that contains 4 class members or class variables and 2 functions. The class variables are reflected into our database table on the right side as 4 database table entities and the datatypes may differ from String (in our class on the left) to Varchar or text on the right.

Let us go ahead to move on to our next diagram type in UML 2.5 which will be the activity in the method of swim lane diagram.

As we can see in the previous diagram, we have the following items

Start which is the start of an activity in a single lane block or a single thread of a program (threading, single threading and multi-threading) indicates that this should be a start of an activity.

The next item is process looks like follows:


The process is used to represent an action that the system will do.

The following item is flow aggregator that represents the flow aggregation to be into the same destination.

The next item will be the decision as follows:

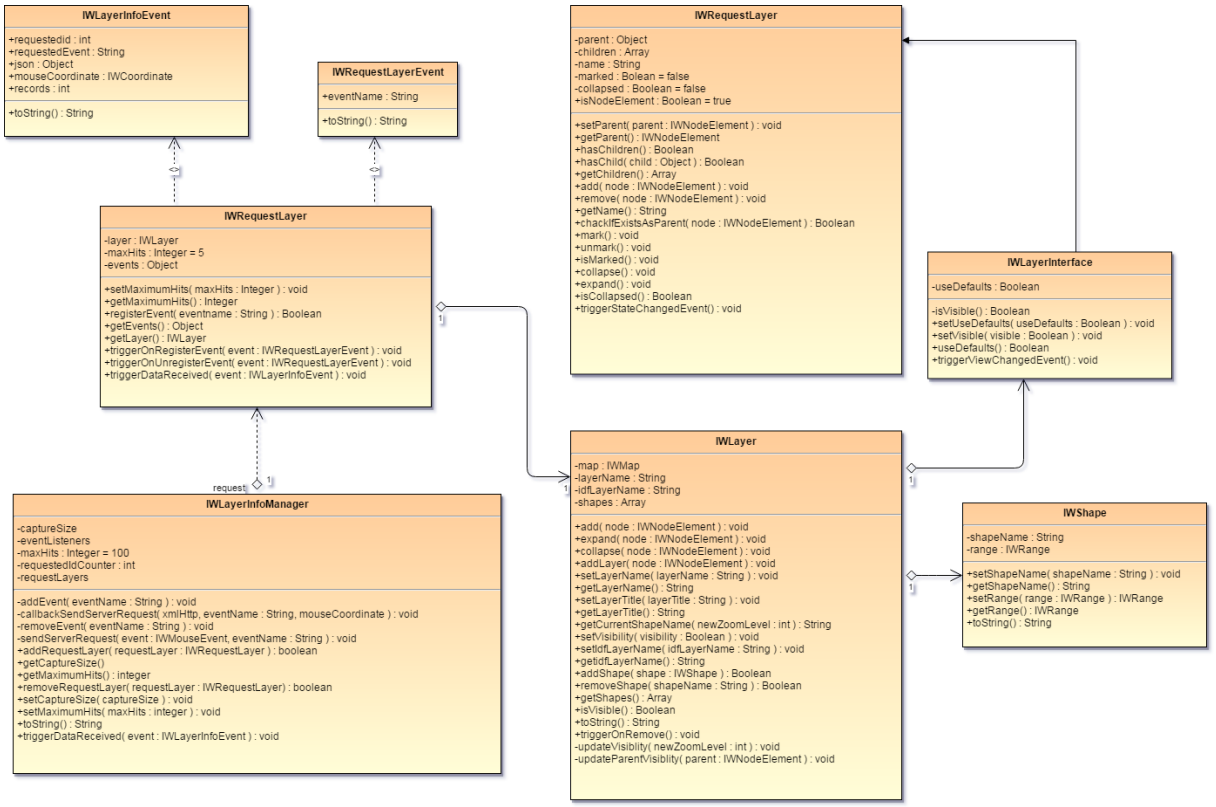

The decision represents if an action will happen or not or another action will happen.

The next item will be end which represents the end of an activity or group of activities.

We usually use 1 start item and 1 end item in the whole diagram but it can be using more than one start item but not more than 1 end item that ends all activities in the diagram.

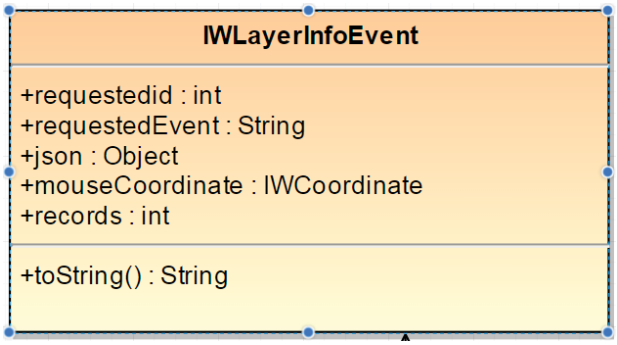
The term swim lane represents groups of actions in one lane block groups them all together.

The next diagram we will talk about in the next section will be the class diagram which will be as follows:




As we can see in the previous class diagram, we have the following items in details:

This block is called a class and let us see how we can make the class items in details as follows:




The **IWLayerInfoEvent** represents the class name that you will create in your code, and next to it, the class members (class variables) **requestedid: int** followed by : (colon) sign and the data type as int or String representation, then at the end of class block, we add the functions that will be in the class like the **toString(): string** (function), then comes in the end the **+** or - signs at the beginning of each line represents the public or private or protected function signature which will be useful in information hiding.

The next item to talk about is:

Relation between classes as aggregation, extension, or include.

The main relations that are most popular for use are <<extends>> and <<include>>

This way, we have covered most of the topics in UML 2.5 and in the next section, we will move on to the next section which will be design patterns and the latest patterns with PHP7 as an example to apply those patterns.

Next, we move on to our next approach which will be the

# 4 INTRODUCTION TO DESIGN PATTERNS AND THE LATEST PATTERNS WITH PHP7 AS AN EXAMPLE

Design patterns are code organization and security together grouped in to form a way of organization to your code inside an application.

Some of the common design patterns and architectures are:

1. Micro Services
2. Multi-Threading
3. Single-Threading
4. Factory
5. Singleton
6. Model-View-Controller (MVC)
7. Model-View-View-Model (MVVM)

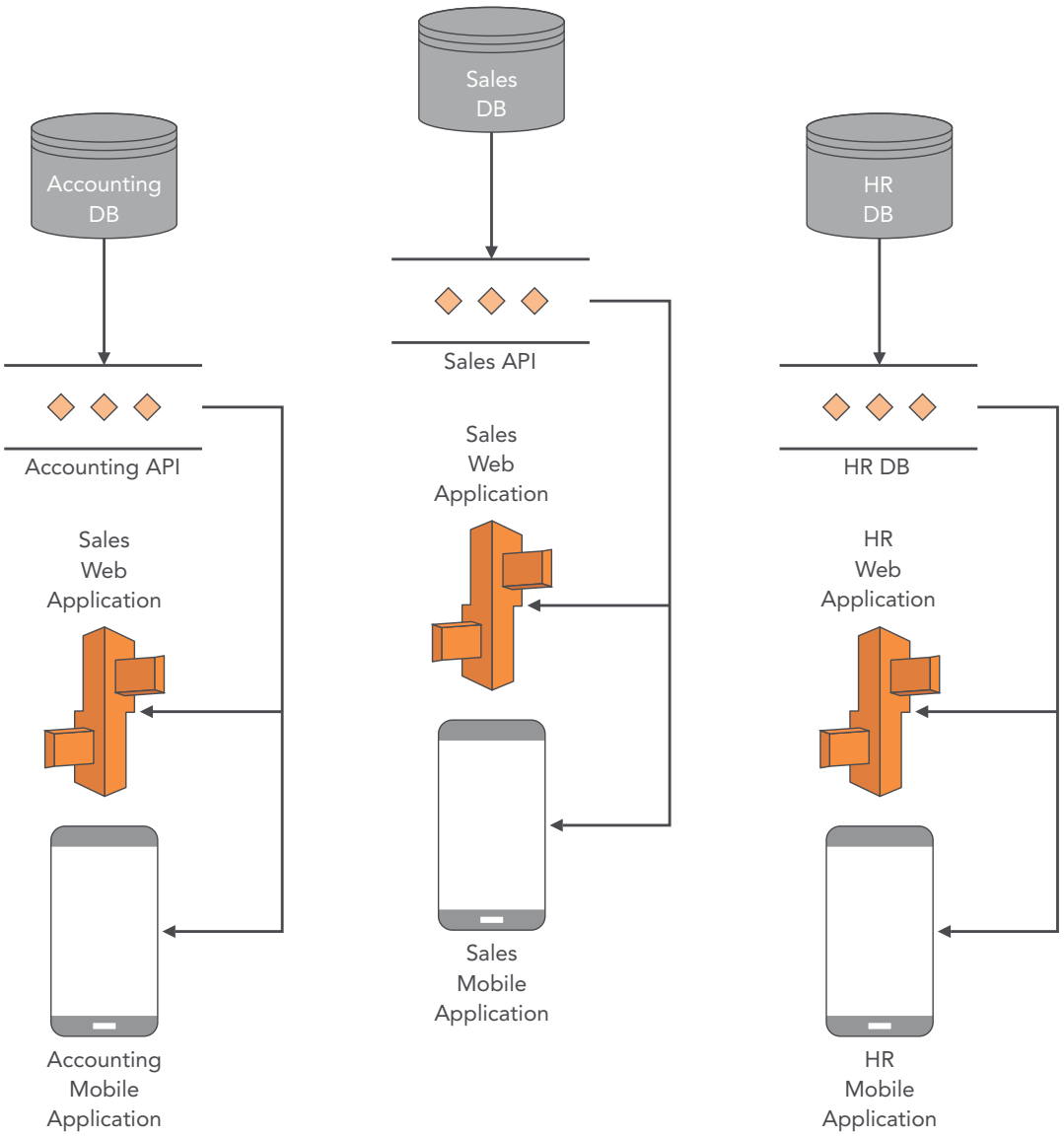
Let us describe each and every one of these patterns.

**Micro Services**

Micro services is a kind or software architecture that permits the architect of truncating the system processes into smaller chunks. Each chunk is called a micro service.

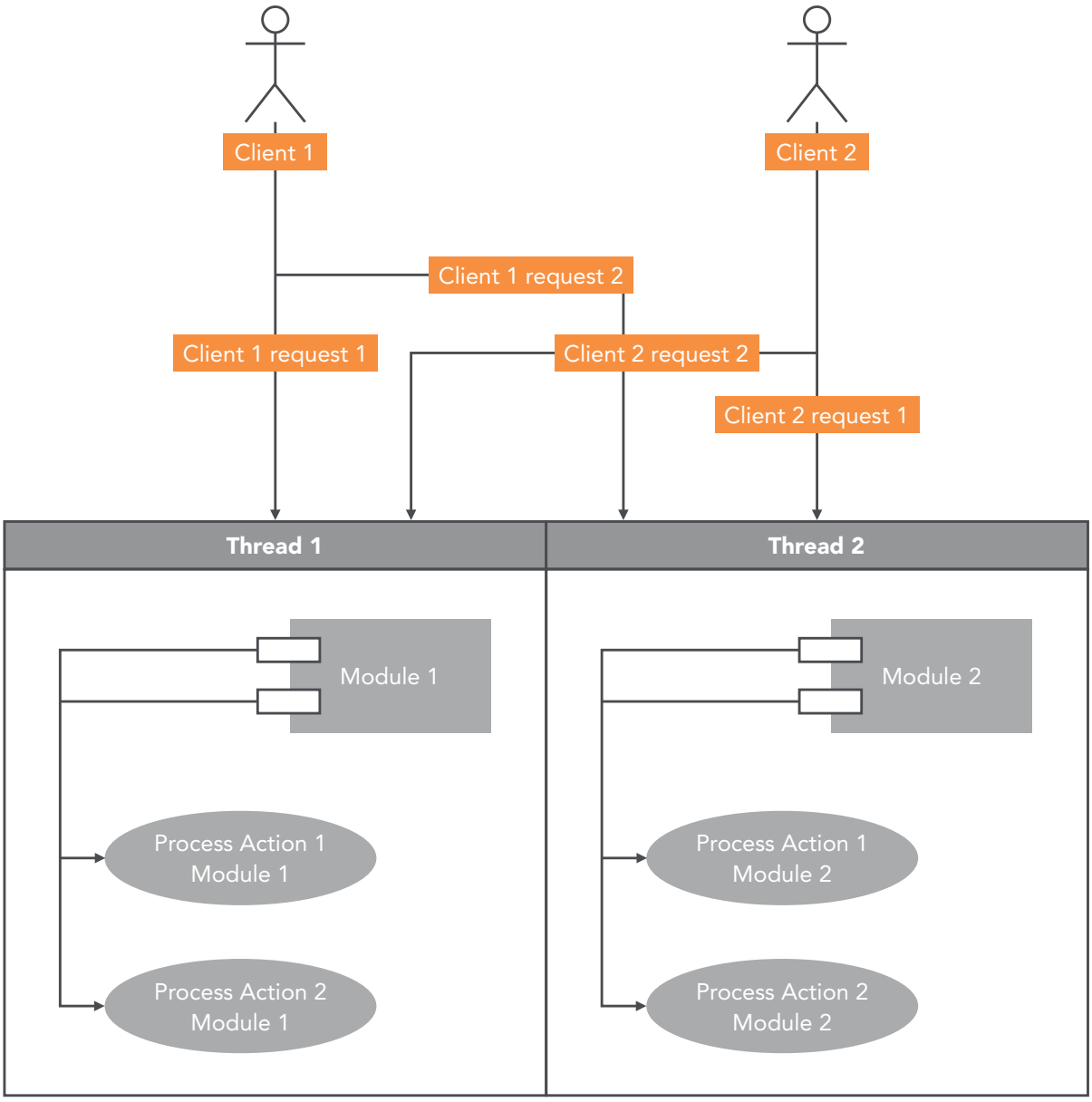
The micro service is a kind of process or part of a process into the system representing an action or group of actions together of the same kind.

Example architecture of a micro service is:



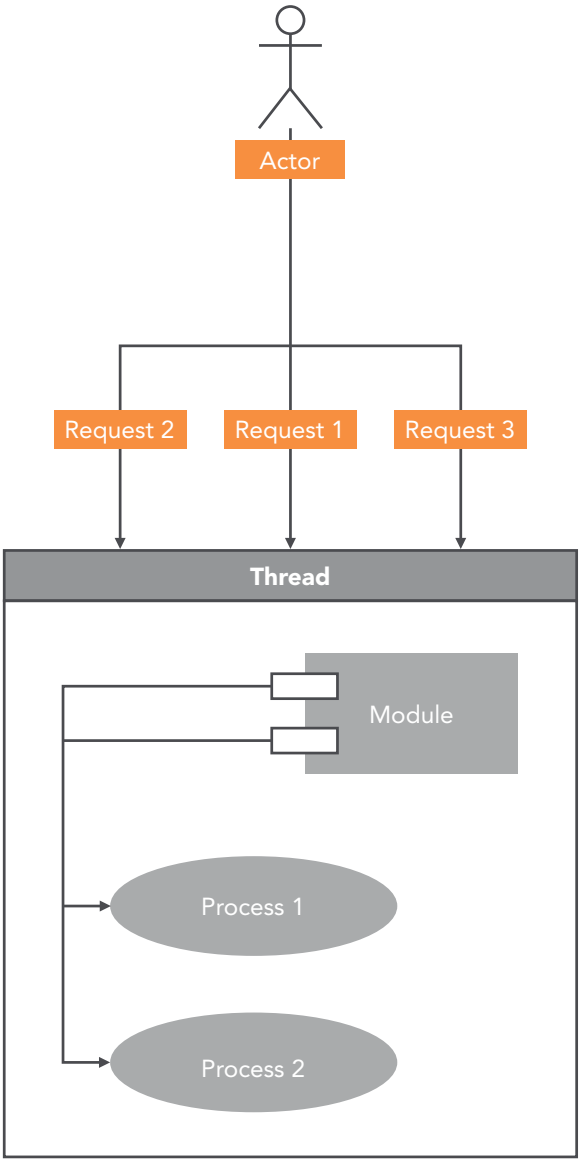

Here is an example of a micro service architecture diagram shows that 3 databases then integration with web and mobile applications from sales database as an example to sales web application, then to the mobile application as sinks receiving and sending to the server and API.

Les us continue with the next kind of architecture which is **multi-threading** and the next diagram will show you an example of the multi-threading architecture is:




Next, we continue to see the next architecture which is the single threading.

The following diagram shows how the single threading is and how it can be used to work with decent applications:




The previous diagram show how the architecture works on single threading mechanism and how to use a single thread.

Next, we will learn about factory design pattern, and the next diagram shows how the factory design pattern is applied to PHP7 as follows:

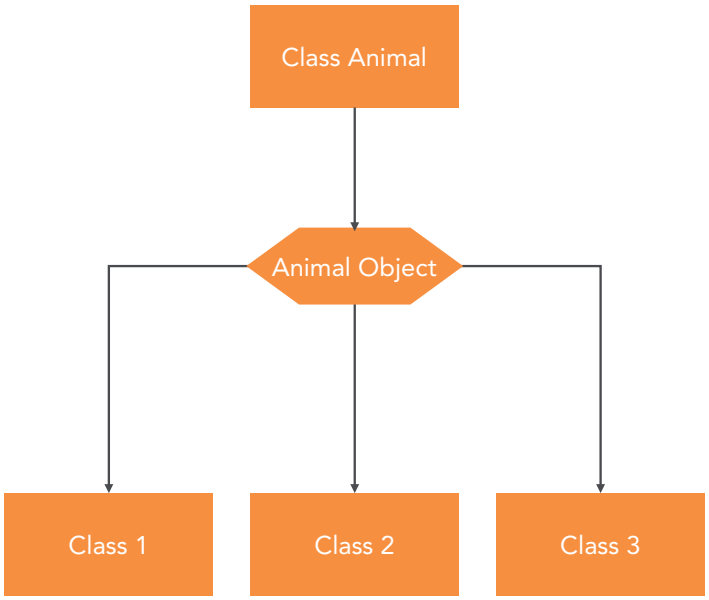
```php
<?php

interfaceAnimal{
publicfunctiongetName() : string;
}

interfaceElephant{
publicfunctiongetRealName() : string;
}

class AnimalimplementsAnimal{
    publicfunctiongetName() : \Elephant{
        returnnew\Elephant();
    }
}

class ElephantimplementsElephant{
    publicfunctiongetRealName() : string{
        return'Elephant\'s Real Name is PHP';
    }
}

$factory=new\Animal();
$factory_object=$factory->getName();
print $factory_object->getRealName();
```

As we saw in our previous example, we saw our implementation code in PHP7, we had 2 classes, one of them is class Animal and the other one is Elephant.

From our previous implementation, we created an Animal object and we called both functions using the same object without constructors such as getName() and getRealName() functions.

The purpose of the factory design pattern is how to call a function from a class using an object of another class and the other class we are using is instantiating the class we are calling its function, so both functions of the 2 classes are used from the same object.

Next, we will have the next design pattern, which is the singleton design pattern. The next diagram shows how the design pattern works:




The singleton design pattern allows only one instance of the class to be created and the class to be instantiated only once and not more than that.

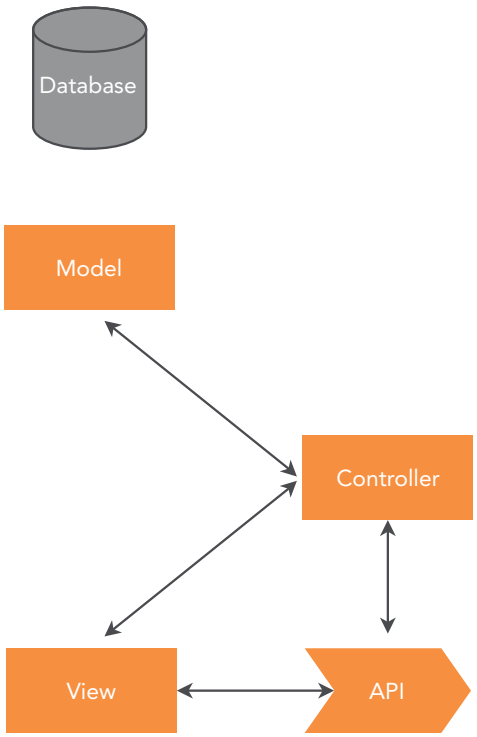
Only one object of the class animal shared to all classes and if you create another object from the class animal, you will get an error in your code.

The next implementation shows how the singleton

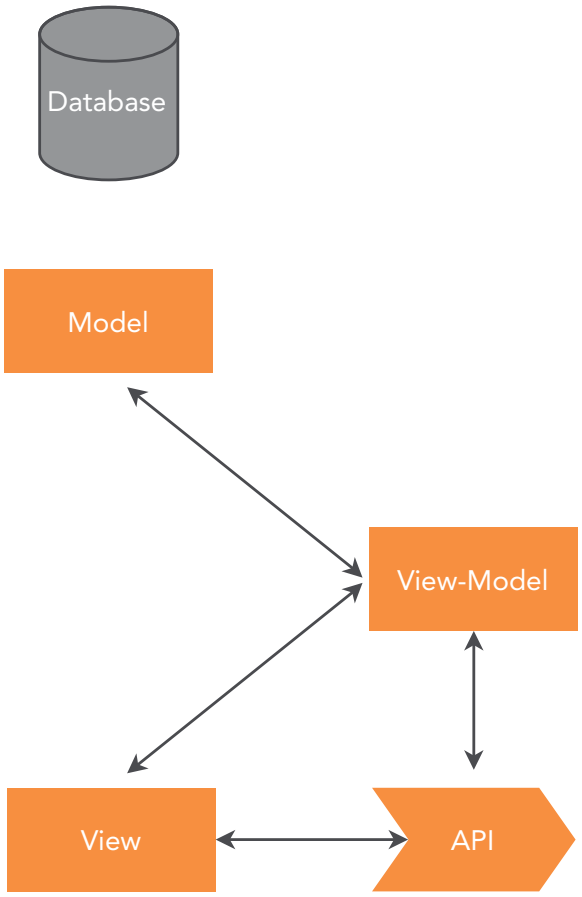
```
<?php

classGod{
        public static $instance;
        public static function __getInstance() \God{
                $instance = __FUNCTION__;
                if(!isset(self::$instance)){
                        self::$instance= new \self;
                }
                return self::$instance;
        }
}
```

Next, we move on to the next design pattern which is the model view controller which is applying the Model as the business login database connector and database manipulation, the controller for validation and routing, and the view contains the user interactive controls, and the rest of the user experience modules.

The following diagram shows you how to work with the model view controller design pattern which is one of the most popular design patterns ever as follows:



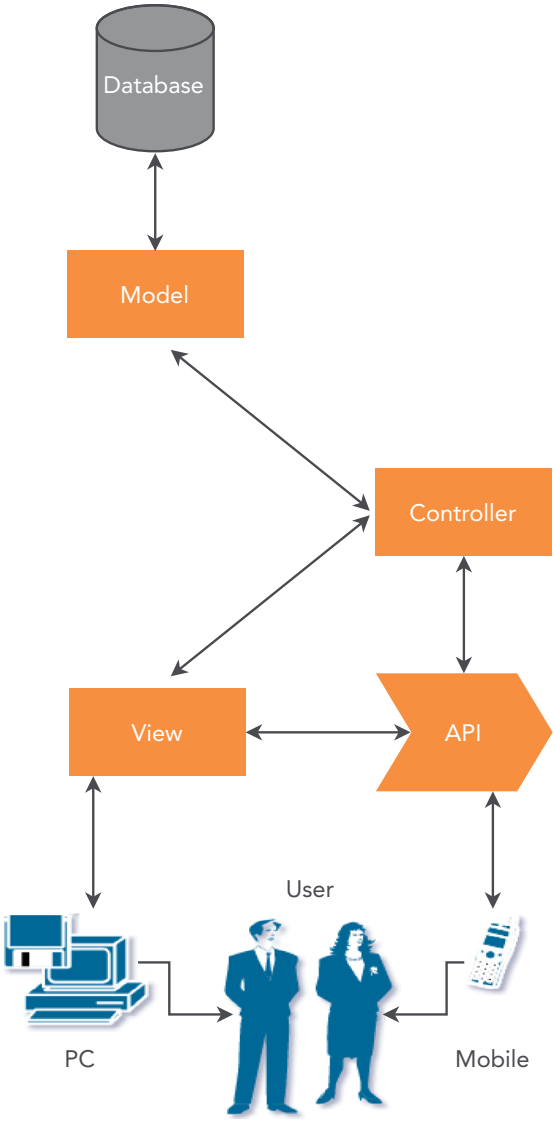
Now, let us go ahead and continue to the last pattern from within our book, we will work with MVVM or the model-view-view-model design pattern and the following diagram will show how this pattern is working:



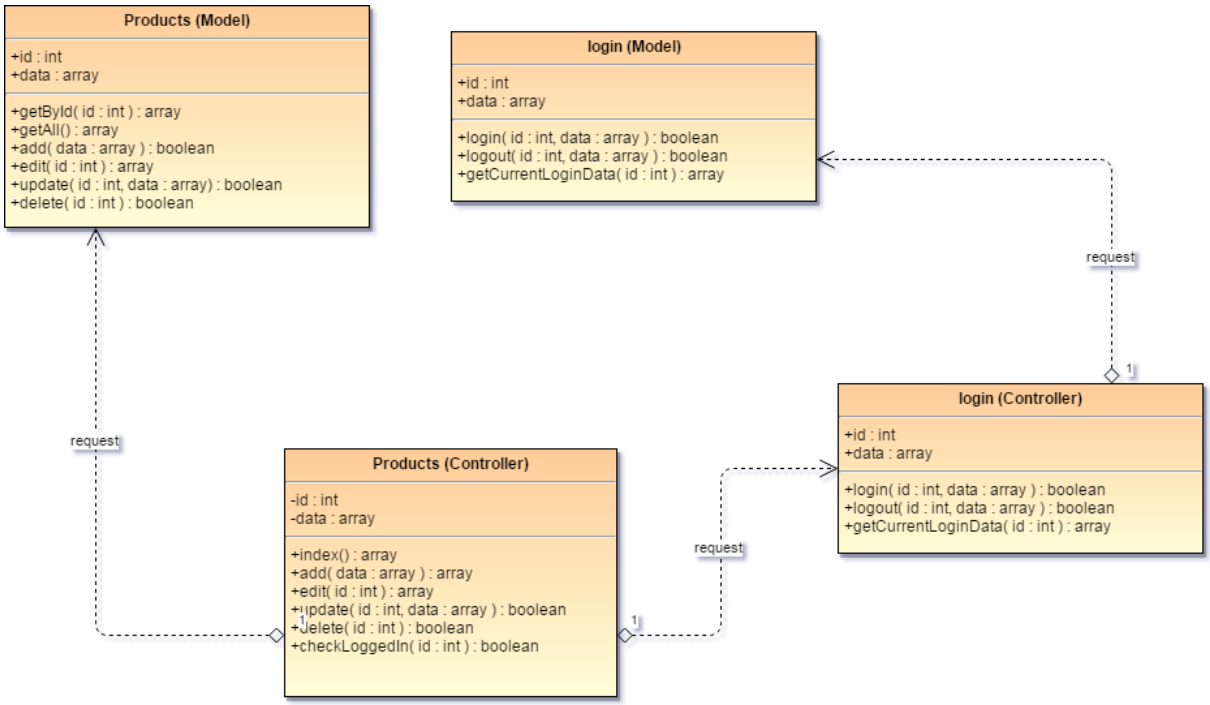

The diagram shows that the Model has all the database heavy lifting, then the view-model shows the connection between the view and the user interaction, to the model with the database, and comes the API as another alternative to the view in case integrating with another or external entity.

# 5 SYSTEM ANALYSIS AND ARCHITECTURE (UML 2.5) FOR A SIMPLE WEBSITE

To make a simple website, we need to look at the following diagram:



Next, we will go through the architectural technical diagram as follows:




The following UML diagram shows that there are 2 models called Products and Login. The login Model has 3 functions, login, logout, and getCurrentLoginData. The same model also contains 2 members, id which is an integer and data which is an array.

The 3 functions in the Login Model are used for the main authentication-authorization behavior to check if the user is logged in or not, login, and logout behaviors.

Next, we have the login controller which has the same functions and members as the login model.

Also we have the second model which is the products model and contains the basic CRUD functionality which consist of add, edit, update and delete, and the index function contains the select.

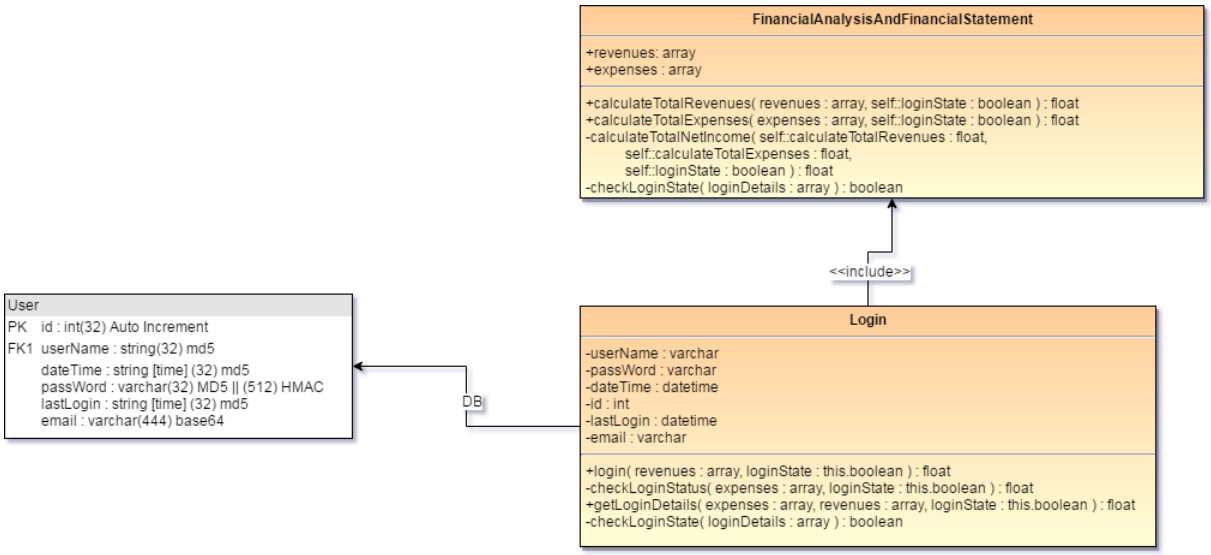
And finally, we have the products controller that has the same functions as the products model with the same functionalities.

Next, we have the relationships between classes represented as request with arrow.

# 6 SYSTEM ANALYSIS AND ARCHITECTURE (UML 2.5) FOR A SIMPLE ACCOUNTING SYSTEM

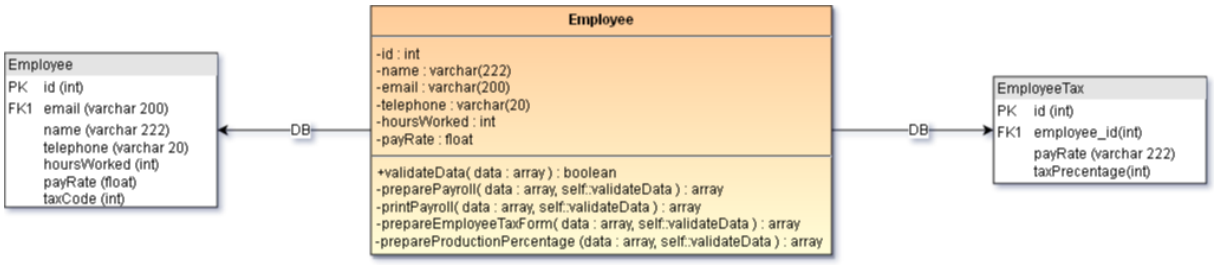
We will go through the analysis and design of an accounting information system in UML 2.5. Note that this kind of a system is somehow complicated. It only contains data grids and some calculations, but it is like a maze you must go through and finish until the end and all of the entries are connected together.

The following diagram shows the financial and budget with login part of a system as follows:



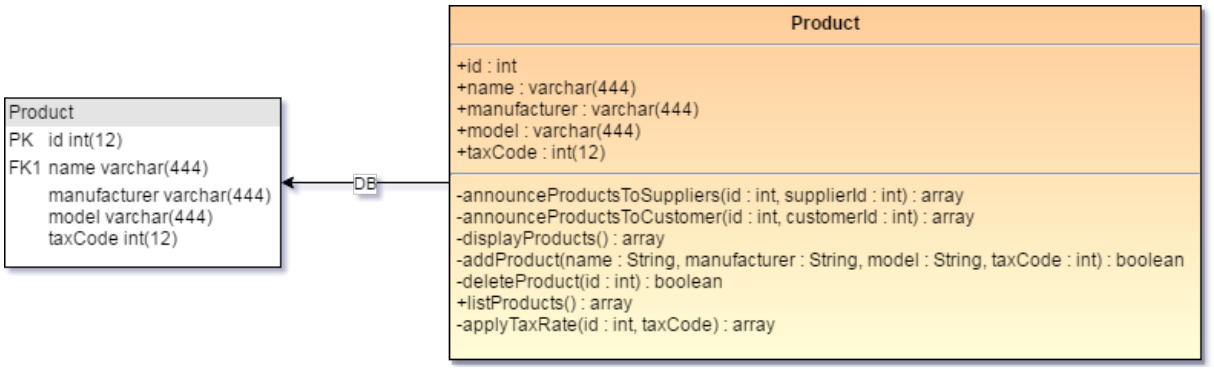

The previous diagram shows 2 classes named login and "FinancialAnalysisAndFinancialStatement" including most of the functions required to perform the tasks that display the financial statement and "Login" which also contains the database table called use and the required fields to be identical to the Login class with its operations.

The next diagram shows you the details of the tree of the accounting system as follows:




The previous diagram shows the employee class and the employee table with its fields, and class members and the functions associated with the preparation, calculations, and printing the payroll on a cheque, a pay slip or a bank transfer statement.

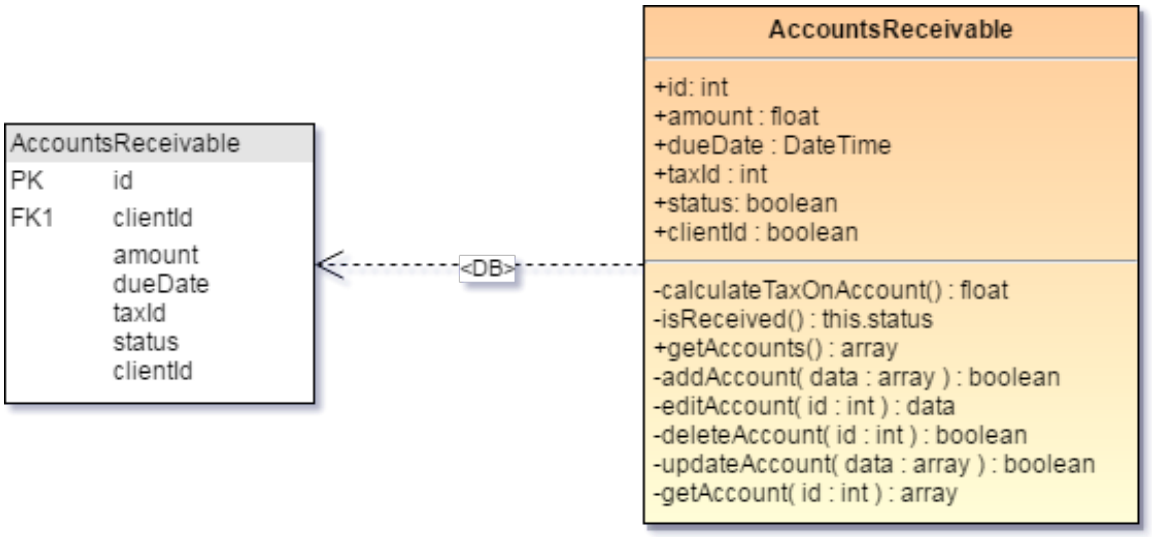
Let us go ahead and move next, the next section will be the tax and the calculation of the tax on the income which involves the existence of products/services and employee tax as we saw in the previous portion of our diagram and in the next section we will see the product/service diagram portion as follows:




The previous diagram shows the members of the class product with the product database table.

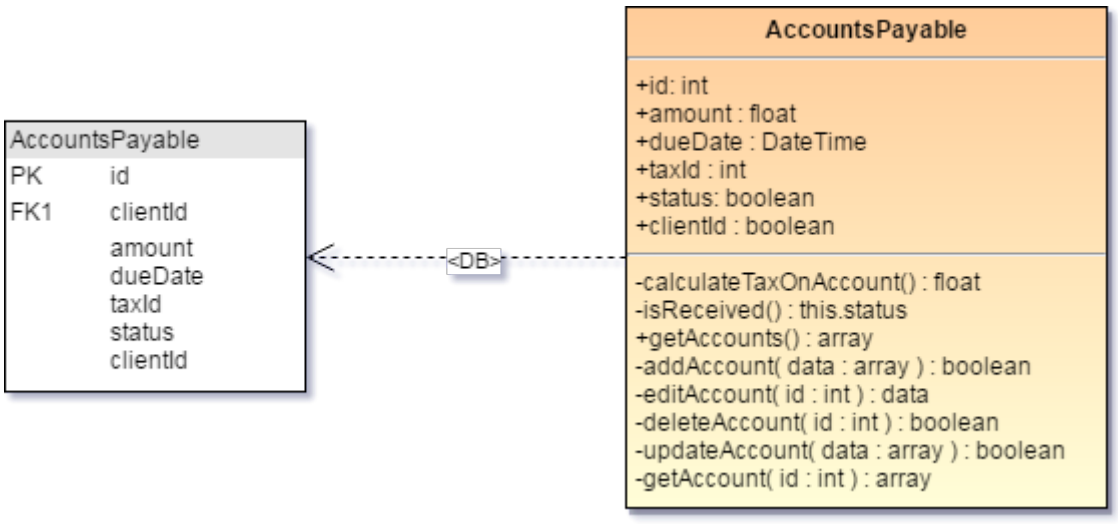
We see the operation as functions at the end of the class square containing all the functionality the class will perform.

The next section will be the accounts receivable and the following is the diagram belongs to it as follows:



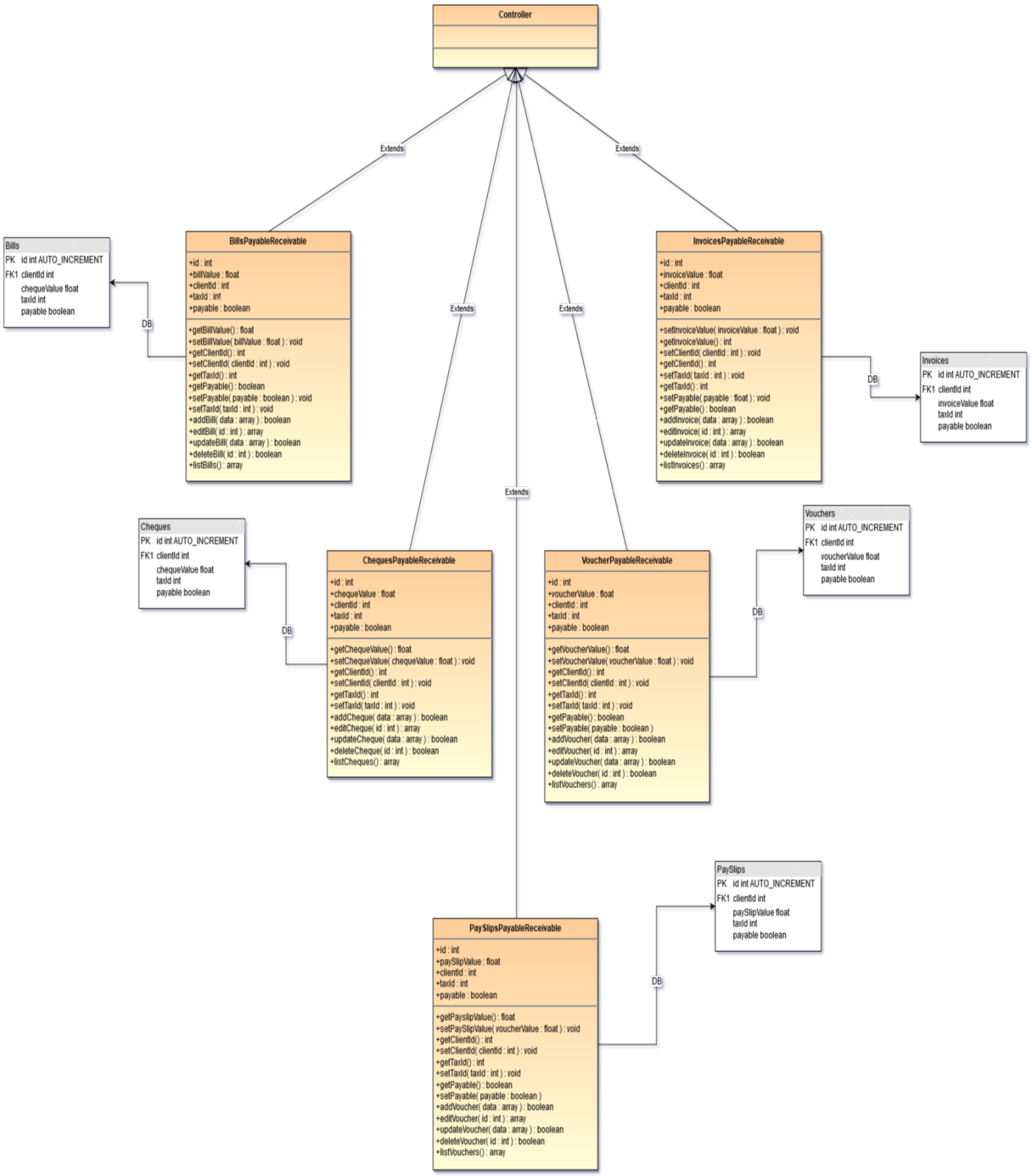

The accounts receivable section is where your clients' accounts will be paid and when with due date and status linked to your client table.

The next section will show you a diagram of the accounts payable. The accounts payable section will be as follows:




The previous diagram shows how the accounts payable is and its table which is the same structure as the accounts receivable table and class but they must be separated as deductive or additive amounts that must be in a separate processing areas.

The next diagram shows bills, invoices, vouchers, pay slips, bank accounts, and cheques and all of them have debit and credit types means payable and receivable.




This way, we have completed the architecture of a simple accounting information system and comes next, the whole enterprise resource planning system.

# 7 SYSTEM ANALYSIS AND ARCHITECTURE (UML 2.5) FOR A SIMPLE ENTERPRISE RESOURCE PLANNING SYSTEM

The definition of **E**nterprise **R**esource **P**lanning system is working completely with all of the company departments according to the company regulations.

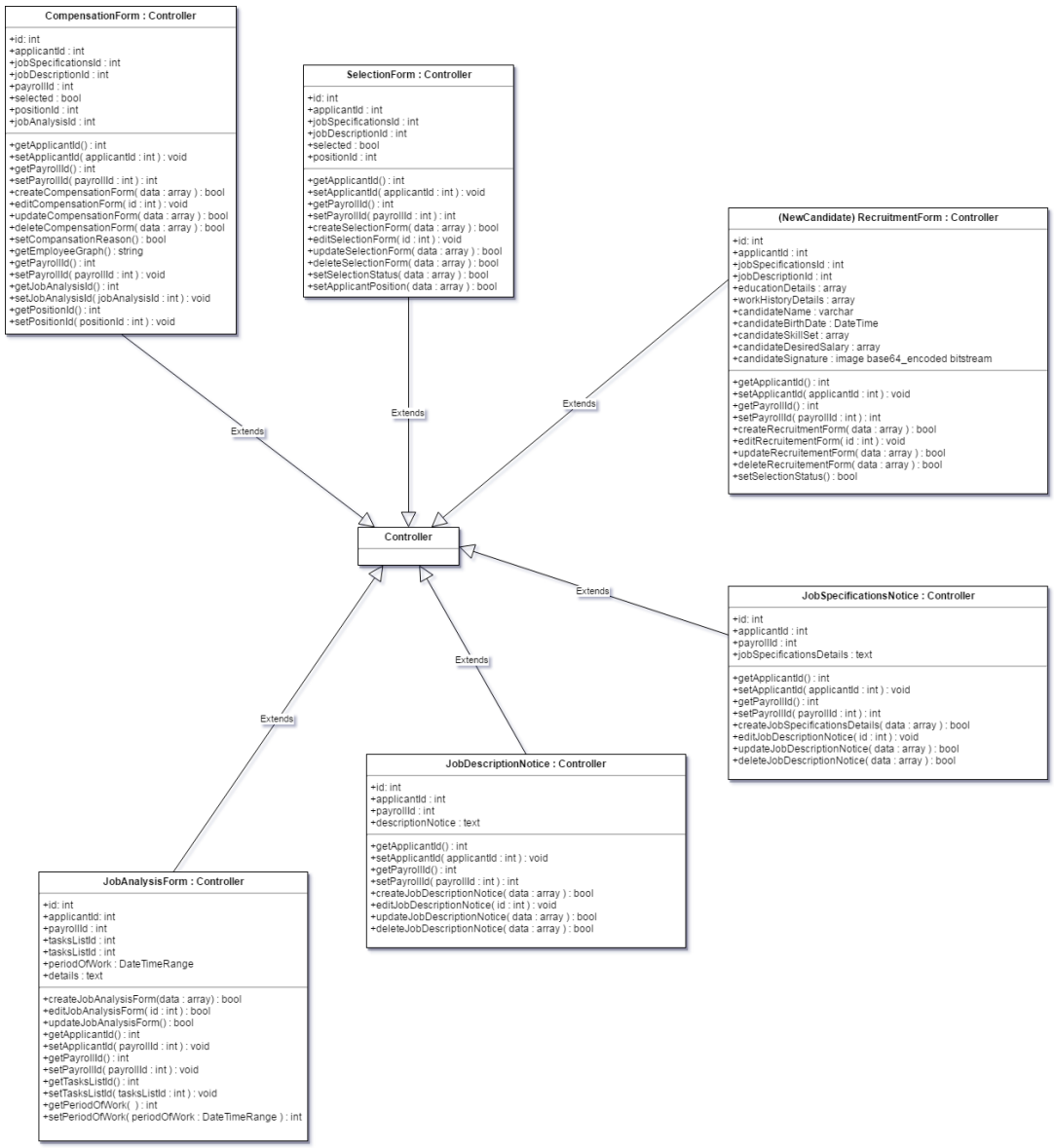
The enterprise resource planning usually consist of accounting and budgeting, payroll processing, that we covered in the previous chapter, then the human resources which we will cover and analytics with performance appraisal of employees and system.

Let us first determine the Human Resources (HR) process and the following diagram shows how the human resources department will be represented in class diagrams:

An example of the listing of the activities of the HR department can be:

1. Job Analysis Form
2. Job Descriptions From
3. Job Specifications Form
4. Recruitment Form
5. Selection Form
6. Compensation Form
7. Training and Development Form
8. Performance Appraisal Form
9. Health, Safety and Security and Insurance Form

The class diagram will be as follows:



We can see that in the previous diagram the following class list:

Compensation

Selection

Recruitment

Description

Specification

Analysis

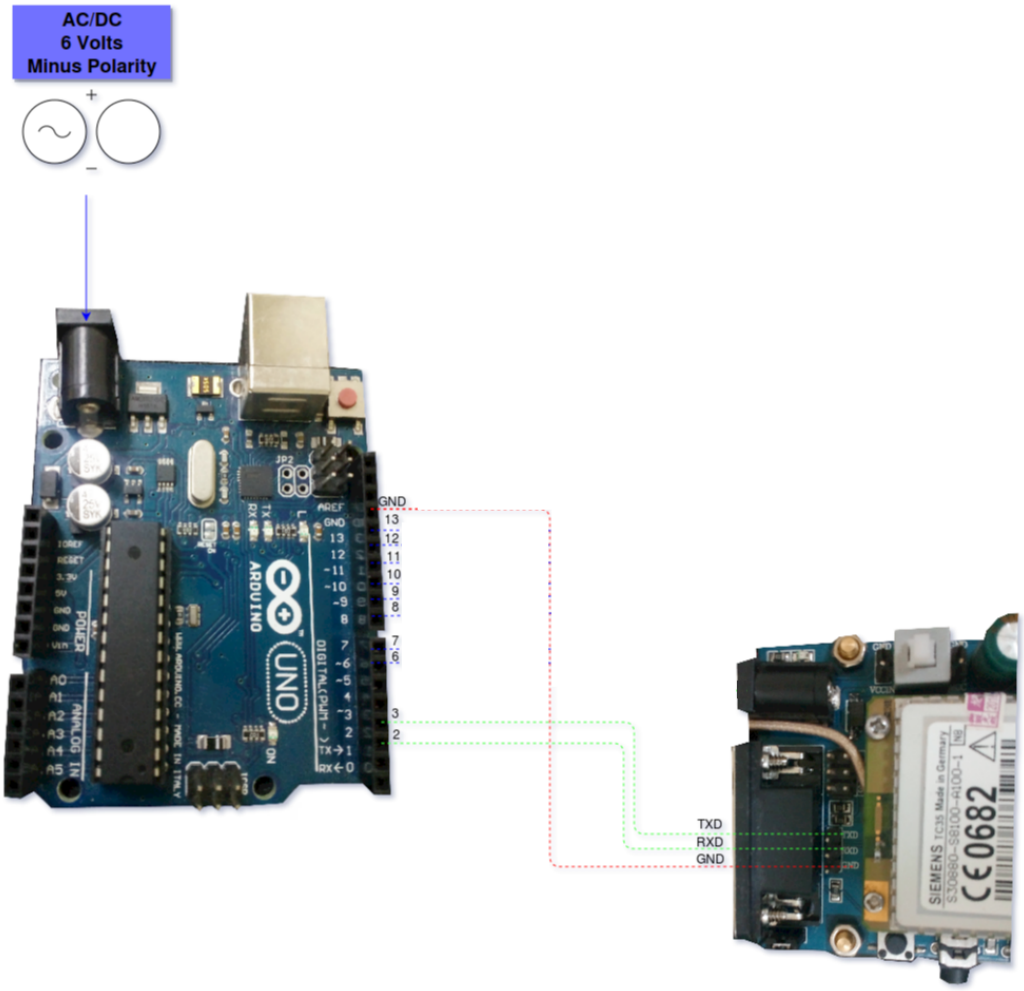
And there are 3 more topics that are not mandatory in the rest of the HR responsibilities and may or may not exist in an organization. This is why we did not mention them in the previous diagram.

The next section will be about hardware diagramming and schematics in Arduino UNO and Siemens TC35 Connection.

# 8 SYSTEM ANALYSIS AND ARCHITECTURE (SCHEMATIC) FOR A SIMPLE ARDUINO UNO – SIEMENS TC35 GSM CONNECTION

We will work in the next section with a SysML diagram and a Schematic Diagram.

The following will be the schematic diagram that will show us the connection using wiring of the Arduino UNO and the Siemens TC35 as follows:



The previous diagram show how the connection between the Siemens TC35 and using TX, RX and GND connection.

The GND connection is in red color, the RX and TX are in green.

This way, we finished most of the system analysis and program development features throughout this book.

# REFERENCES

https://martinfowler.com/books/uml.html

https://www.amazon.com/Teach-Yourself-Hours-Complete-Starter/dp/067232640X/ref=la_B001IU2V30_1_4?s=books&ie=UTF8&qid=1493896746&sr=1-4

https://www.jumia.com.eg/jumia-books-uml-distilled-a-brief-guide-to-the-standard-object-mode-451228.html

Thank you so much for your interest in reading for BookBoon.com

Further details, please contact bookboon.com team.

My Linkedin:
o

My Github:
https://github.com/mostafaahamid?tab=repositories

My E-Mails:
mostafa.hamid@arabintelco.co.uk
mostafa.hamid@arabintelco.com

Contacting BookBoon.com Team is through:
http://bookboon.com