# Content

[Content 1](#_Toc454294414)

[Introduction 2](#_Toc454294415)

[Project Introduction 2](#_Toc454294416)

[Implementation Schema 2](#_Toc454294417)

[Tools used during the development 2](#_Toc454294418)

[Project Explanation 2](#_Toc454294419)

[Code Explanation 2](#_Toc454294420)

[Conclusions 2](#_Toc454294421)

[Appendix 3](#_Toc454294422)

[Report requirements 3](#_Toc454294423)

[First of all the layout 3](#_Toc454294424)

[Requirements 3](#_Toc454294425)

# Introduction

Explain what the Project is about and the final purpose of this course

Our company “**Radius Software Inc.**” have been contacted to design a Bank Management System that controls the operations between all the machines present in the system. It means

let us to design the project by UML.( The Unified Modeling Language is a standard language for writing software blueprints.The UML may be used to visualize, specify, construct, and document the artifacts of a software-intensive system). In fact UML is just a tool to solution problem, we yneed to complete is the establishment of the model, making both sides are satisfied, and then make a detailed process diagram, then according to this model to achieve the project.

# Project Introduction

Provide a brief introduction about the project you have to develop

* **PC Employees** : this computer run an application that perform the following operations :

Current account operations (withdrawal, deposit, top-up, transfer)

Stock option operations (buy, sell)

* **Booking System** : it is ad-hoc hardware running an application that executes the following operations:

Withdrawal

National Stock options

International Stock Options

* **Database System** : application that interacts with the relational database to perform the following operations :

Create new customer

Current Account Balance

Current Account Bank statement

Stock options file

* **ATM** : application that implements the following operations:

Withdrawal

Deposit

Top-up phone

# Implementation Schema

(1) **Use case diagram:**

Describes a functional unit provided by the system. In a visual way to understand the functional requirements of the system, "Actor" and the relationship between the use cases within the system**.**  In this case, the participant "bank depositor" and the ATM machine. The simplified ATM machine only withdraws money, deposits top up and other functions. The rest of the features do not elaborate.This diagram is based on the following elements: ***System*, *Actor*, *Use Case***

(2) **Class Diagram:**

    Describes the static structure of the system. Logical class, implementation class, implementation class is the programmer to deal with the entity. The class uses a rectangle containing three parts on the class diagram, as shown in Figure 2. The top part shows the name of the class, the middle part contains the properties of the class, and the bottom part contains the class operation (or "method").

        In this case, many individual accounts form the account library, the account has the account type, account number, balance three properties. Many bank depositors form a vault. The ATM system contains a number of ATM machines. Bank account and ATM machine two categories which contains attributes, which operations, their visibility and operation of the return type, the number of parameters, the type of parameters from the class diagram are at a glance. Through the class diagram can not only make the designer clearly express their own design intent, but also help to organize their own ideas, enrich and optimize their own design.

(3) **Sequence diagram:**

       Describes a detailed process for a specific use case (or part of a use case). It is almost self-describing, and shows the call between the different objects in the process, but also can be very detailed display of different calls to different objects. The sequence diagram has two dimensions: the vertical dimension displays the sequence of messages / calls in the chronological order of occurrence; the horizontal dimension displays the instance of the object to which the message was sent.

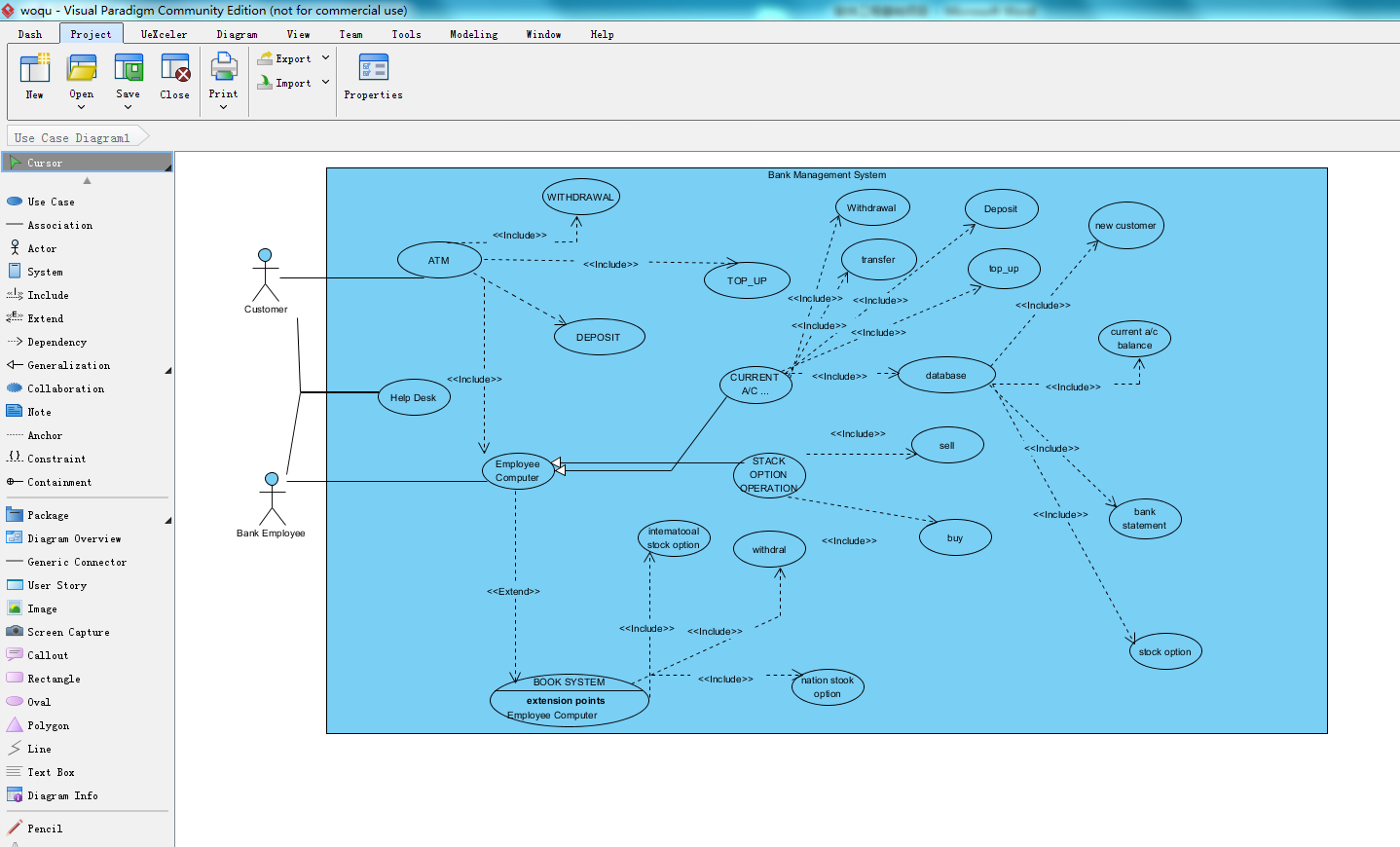
        This example describes in time the flow of information when the customer withdraws money from the ATM, and the sequence diagram focuses on the time sequence of message transfer between the objects.

## Tools used during the development

1. Visual Paradigm (VP-UML) is a UML CASE Tool supporting UML 2, SysML and Business Process Modeling Notation (BPMN) from the Object Management Group (OMG). In addition to modeling support, it provides report generation and code engineering capabilities including code generation. It can reverse engineer diagrams from code, and provide round-trip engineering for various programming languages. Tool2 - description
2. [Processon](http://www.processon.com) . ProcessOn is an online mapping tool aggregation platform, it can be online painting flow chart, mind mapping, UI prototype, UML, network topology, organizational chart, etc. You do not have to worry about downloading and updating the problem, regardless of Mac or Windows, a browser can anytime, anywhere to play creative, planning work

# Project Explanation

Explain the implementation logic for your project, that is, provide a clear and simple description about HOW you solved the project.



1-1 Use case diagram

# Code Explanation

Provide the “diagrams” for your project along with the corresponding detailed description .

# Conclusions

Provide your personal conclusions about the project.

In the course of learning, we learned the basic operation of Visual Paradigm, realized how to implement a specific UML diagram of the project. Let us do the bank system diagram directly was bit difficult, Before we build the model, we review the notes and PPT, review the basic elements of several diagram and. The most important thing is that everyone has their own ideas, after a lot of discussion we make these ideas transfer to one. Then I can see that the teamwork makes work complex but perfect.

# Appendix

Insert any reference to web sites used during your project

# <https://en.wikipedia.org/wiki/Visual_Paradigm_for_UML>

<http://blog.csdn.net/giianhui/article/details/5513133>

<https://wenku.baidu.com/view/5b93daefa45177232e60a23a.html>

<http://www.processon.com>