**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

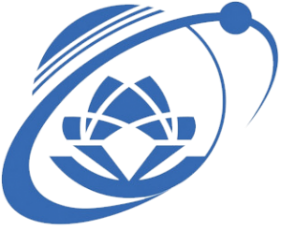
**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA CÔNG NGHỆ PHẦN MỀM**

**BÁO CÁO CUỐI KỲ**

**Phương pháp phát triển phầm mềm hướng đối tượng – SE100.M12.PMCL**

**PHẦN MỀM QUẢN LÍ CỬA HÀNG CHO THUÊ BĂNG ĐĨA**



Giảng viên hướng dẫn: Lê Thanh Trọng

Danh sách thành viên nhóm:

Nguyễn Trọng Tính - 19521024

Trần Hoàng Gia Bảo - 19521258

Phan Tiến Lào - 19521747

Table of contents

[I.](file:///D:\download\Báo%20cáo%20cuối%20kì%20quản%20lí%20đề%20tài%20nghiên%20cứu%20kh.docx#_heading=h.gjdgxs) General introduction 5

[1.](file:///D:\download\Báo%20cáo%20cuối%20kì%20quản%20lí%20đề%20tài%20nghiên%20cứu%20kh.docx#_heading=h.30j0zll) Generality 5

[1.1](file:///D:\download\Báo%20cáo%20cuối%20kì%20quản%20lí%20đề%20tài%20nghiên%20cứu%20kh.docx#_heading=h.1fob9te) Problem 5

[1.2](file:///D:\download\Báo%20cáo%20cuối%20kì%20quản%20lí%20đề%20tài%20nghiên%20cứu%20kh.docx#_heading=h.3znysh7) Solution 5

[2.](file:///D:\download\Báo%20cáo%20cuối%20kì%20quản%20lí%20đề%20tài%20nghiên%20cứu%20kh.docx#_heading=h.2et92p0) Technologies 6

[∙](file:///D:\download\Báo%20cáo%20cuối%20kì%20quản%20lí%20đề%20tài%20nghiên%20cứu%20kh.docx#_heading=h.tyjcwt) Windows Form (C#) 6

[∙](file:///D:\download\Báo%20cáo%20cuối%20kì%20quản%20lí%20đề%20tài%20nghiên%20cứu%20kh.docx#_heading=h.3dy6vkm) SQL Server 6

[3.](file:///D:\download\Báo%20cáo%20cuối%20kì%20quản%20lí%20đề%20tài%20nghiên%20cứu%20kh.docx#_heading=h.1t3h5sf) Design model 7

[II.](file:///D:\download\Báo%20cáo%20cuối%20kì%20quản%20lí%20đề%20tài%20nghiên%20cứu%20kh.docx#_heading=h.4d34og8) Detailed analysis 8

[1.](#_heading=h.2s8eyo1) Use Case Diagram 8

[2.](#_heading=h.3j2qqm3) Actors List 15

[3.](#_heading=h.3j2qqm3) Use Case List 15

[4.](#_heading=h.3j2qqm3) Use Case Specification 15

[4.1 Specification of Usecase “Login” 18](#_heading=h.2xcytpi)

[4.2 Specification of Usecase “Rent Video” 18](#_heading=h.2xcytpi)

[4.3 Specification of Usecase “Return Video” 18](#_heading=h.2xcytpi)

[4.4 Specification of Usecase “Enter Discs” 18](#_heading=h.2xcytpi)

[4.4 Specification of Usecase “Make a report” 18](#_heading=h.2xcytpi)

[6.](#_heading=h.1y810tw) Database Design 17

6.1 L[ist of classes and relationships 23](#_heading=h.23ckvvd)

6.2 Class in details [23](#_heading=h.23ckvvd)

[7.](#_heading=h.1y810tw) Class Diagram 17

[8.](#_heading=h.4i7ojhp) Activity Diagram 18

8.1 [Login 18](#_heading=h.2xcytpi)

8.2 [SignIn 18](#_heading=h.1ci93xb)

8.3 [Rent/Buy Request 18](#_heading=h.1ci93xb)

8.4 [Manage Stocks 18](#_heading=h.1ci93xb)

8.5 [Account management 18](#_heading=h.1ci93xb)

8.6 [Report 18](#_heading=h.1ci93xb)

8.7 [Rent/Buy 18](#_heading=h.1ci93xb)

8.8 [Yout cart 18](#_heading=h.1ci93xb)

8.9 [History 18](#_heading=h.1ci93xb)

[9.](#_heading=h.3o7alnk) Sequence Diagram 23

1. [Staff 18](#_heading=h.1ci93xb)

[Rent/Buy request 18](#_heading=h.1ci93xb)

[Manage Stock 18](#_heading=h.1ci93xb)

[Account management 18](#_heading=h.1ci93xb)

[Report 18](#_heading=h.1ci93xb)

1. [User 18](#_heading=h.1ci93xb)

[Rent/Buy 18](#_heading=h.1ci93xb)

[Your cart 18](#_heading=h.1ci93xb)

[History 18](#_heading=h.1ci93xb)

[10.](#_heading=h.3o7alnk) State Diagram 23

10.1 L[ogin 23](#_heading=h.23ckvvd)

10.2 Create account [23](#_heading=h.23ckvvd)

10.3 Import discs [23](#_heading=h.23ckvvd)

10.4 Payment [23](#_heading=h.23ckvvd)

[11.](#_heading=h.4f1mdlm) UI 27

6.1Login 27

[7.](#_heading=h.111kx3o) Installation instruction 32

[8.](#_heading=h.3l18frh) Tasksheet 33

[9.](#_heading=h.206ipza) Summary 34

[10.](#_heading=h.4k668n3) Related document 34

[11.](#_heading=h.2zbgiuw) Comments and grades 35

# General introduction

## General

### Problem

The demand for transporting and trading discs to serve the entertainment of watching movies and listening to music is increasing day by day. However, the management of buying, selling and lending discs at disc stores is still done on paper, books or simple spreadsheets. Such management will take a lot of time but not manage all and accurately. Therefore, the video rental store management software will solve those problems, make buying, selling and lending in stores easier and more developed.

Benefits of owning a record store management software:

- Save time managing, recording and looking up information.

- Save storage space.

- Manage an intuitive.

- Improve the work efficiency of store staff.

Switching from manual management to computer software management is a trend when technology is developing stronger and more accessible than ever.

### Problem solving

Build a video rental store management software that has all the basic features along with a simple and easy-to-see interface. The software includes two main users: people who want to rent tapes and store employees, with each object having separate functions.

Functions to serve the tenants of tapes:

- Buy/borrow tapes

- Search for tapes you want to buy/borrow

- Manage orders (purchased or borrowed)

Employee service functions:

- Manage imported/sold/rented tapes

- Manage user information / disks

- Search for tapes

- Export report

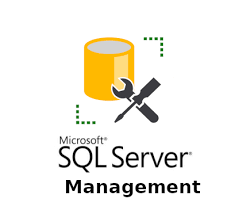
## Technologies

### Windows Form (C#)



Windows Forms is a free and open source graphical class library included as part of the Microsoft .NET Framework or Mono Framework that provides a foundation for writing other applications for PC, Laptop and tablet.

### SQL Server



**Microsoft SQL Server** is a relational database management system developed by Microsoft. As a database server, it is a software product whose primary function is to store and retrieve data required by other software applications. Can run on the same computer or on another computer on the network (including the Internet).

Why use Microsoft SQL Server:

- Free.

- Good user experience.

- Wide selection of add-ins.

- Easy to install.

## Design model

3-layer model consists of 3 main parts:

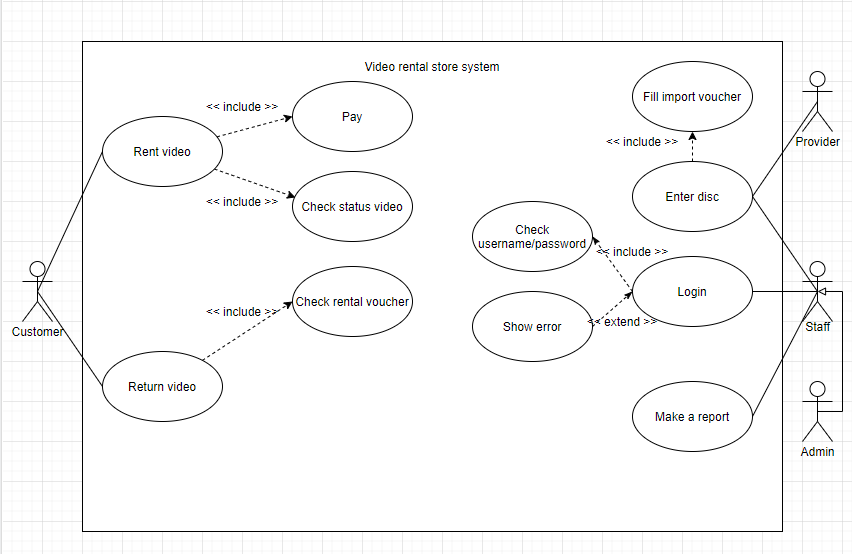
- Presentation Layer (GUI): This layer has the main task of communicating with the user. It includes interface components (win form, web form, ...) and performs tasks such as inputting data, displaying data, checking data correctness before calling the Business Logic Layer (BLL).

- Business Logic Layer (BLL): This layer is divided into two tasks: This is the place to meet the data manipulation requirements of the GUI layer, process the main data source from the Presentation Layer before transmitting to the Data Access Layer and stored in the database management system. This is also a place to check constraints, data integrity and validity, perform calculations and process business requirements, before returning the results to Presentation Layer.

- Data Access Layer (DAL): This layer has the function of communicating with the DBMS such as performing tasks related to data storage and query (search, add, delete, edit, ...).

# Detailed analysis

## Use Case Diagram

**

# Actor List

|  |  |  |
| --- | --- | --- |
| .No | Name of Actor | Short description/Notes |
| 1 | Customer | Customer who rent videos |
| 2 | Staff | Personal staff who is working in a video store |
| 3 | Admin | Manage the shop |
| 4 | Provider | Person who provide disc for shop |

# Use-case List

|  |  |  |
| --- | --- | --- |
| STT | Name of Use-case | Short description/Notes |
| 1 | Login | Login to use features of video rental store system |
| 2 | Rent video | Customer rent videos which they want |
| 3 | Return video | Customer want to return the disc |
| 4 | Pay | Make a payment after rent video |
| 5 | Check status video | Check video available or sold out |
| 6 | Check rental voucher | Check information customer who rent this video |
| 7 | Enter disc | Import new disc into store system by staff |
| 8 | Fill import voucher | Staff fill this voucher when import new disc to store |
| 9 | Check username/password | Check username, password when login |
| 10 | Show error | Show error message when login failed |
| 11 | Make a report | Staff send a report to admin |

# Use-case Specification

## 4.1 Specification of Use-case “Login”

### Short description

* Staff or Admin login into system

### Flow of events

#### Basic flow

* Input username and password on the login page
* Press login button
* System authenticate
* Allow login

#### Alternative flow

* None

### Special requirements

* Before login must have register account

### Pre-condition

* The system is at ready state
* The database is at ready state
* Must have valid account

### Post-condition

* Log in successfully and can use the feature of the system

### Extend Points

* Show the error message when username or password incorrect

## 4.2 Specification of Use-case “Rent Video”

### Short description

* Customer rent video from the store

### Flow of events

#### Basic flow

* Customer fill the rental voucher
* Staff check the status of video
* If video available, customer can rent

#### Alternative flow

* None

### Special requirements

* Video must have status available

### Pre-condition

* Customer fill rental voucher

### Post-condition

* If video available, customer can rent or wait for a while until video available

### Extend Points

* None

## 4.3 Specification of Use-case “Return video”

### Short description

* Customer return rent disc

### Flow of events

#### Basic flow

* Customer give rental voucher for staff check the validity of it
* Staff will compare the rental voucher to its data in system database
* System will make a bill and cashier will proceed the payment
* Write down transaction in database

#### Alternative flow

* None

### Special requirements

* Customer must have rental voucher

### Pre-condition

* Customer must have rental voucher

### Post-condition

* Customer return video and make a payment

### Extend Points

* None

## 4.4 Specification of Use-case “Enter disc”

### Short description

* Staff import new disc into store/system

### Flow of events

#### Basic flow

* Staff give the provider import voucher
* If video available, provider give disc to store and bill
* Staff will enter disc to system

#### Alternative flow

* None

### Special requirements

* Staff must fill import voucher and video must available from provider

### Pre-condition

* Staff fill import voucher

### Post-condition

* If video available, provider will give it and bill to store

### Extend Points

* None

## 4.5 Specification of Use-case “Make a report”

### Short description

* Staff statistic on the number of disc sold

### Flow of events

#### Basic flow

* Staff statistic on the number of disc sold
* Staff send it to admin

#### Alternative flow

* None

### Special requirements

* None

### Pre-condition

* The system is at ready state

### Post-condition

* Staff send report to admin

### Extend Points

* None

## Database Design

## List of classes and relationships

|  |  |  |  |
| --- | --- | --- | --- |
| .No | Class/Relationship | Type | Note |
| 1 | Account |  | Username, password of account |
| 2 | Video |  | Information about video |
| 3 | VideoCategory |  | Type of video |
| 4 | ImportVoucher |  | Staff use this to import new disc |
| 5 | Provider |  | The person who provide disc to store |
| 6 | Bill |  | Information about video which customer rent |
| 7 | BillInfo |  | Detail information of bill |
| 8 | RentalVoucher |  | Customer fill this when rental video |

## Class in details

### *Account*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | AttributeName | Type | Constraints | Meaning/Note |
| 1 | UserName | String | private | Username login |
| 2 | DisplayName | string | private | Display name when use this account |
| 3 | PassWord | string | private | Password login |
| 4 | Type | int | Private | 1: Admin, 0: Staff |

### *VideoCategory*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | AttributeName | Type | Constraints | Meaning/Note |
| 1 | id | Int | Protected | Id of video category |
| 2 | name | string | protected | Type of video |

### *ImportVoucher*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | AttributeName | Type | Constraints | Meaning/Note |
| 1 | id | Int | private | Id this voucher |
| 2 | idProvider | Int | private | Id of provider who provide disc |
| 3 | idVideo | Int | private | Id of video |
| 4 | idCategory | int | Private | Id of video category |

### *Provider*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | AttributeName | Type | Constraints | Meaning/Note |
| 1 | id | int | protected | Id of provider |
| 2 | name | string | protected | Name of provider |

### *Bill*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | AttributeName | Type | Constraints | Meaning/Note |
| 1 | Id | Int | protected | ID of bill |
| 2 | DateCheckIn | Date | private | Date check in |
| 3 | DateCheckOut | Date | Private | Data check out |
| 4 | idVideo | Int | Private | Id of video |
| 5 | Deposits | Float | Private | Deposits when customer rental video |
| 6 | Status | Int | private | Available or sold out |

### *BillInfo*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | AttributeName | Type | Constraints | Meaning/Note |
| 1 | Id | Int | private | Id of bill info |
| 2 | idBill | Int | private | Id of bill |
| 3 | idVideo | Int | private | Id of video |
| 4 | count | Int | private | Count how much disc was sold |

### *Video*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | AttributeName | Type | Constraints | Meaning/Note |
| 1 | Id | Int | Protected | Id of video |
| 2 | Name | String | private | Name of video |
| 3 | idCategory | Int | private | Type of video |
| 4 | Price | Float | Private | Price of video |
| 5 | Status | Int | private | Available or sold out |
| 6 | Director | String | private | Director of movie |
| 7 | Actor | String | private | Actor of movie |
| 8 | Length | String | private | Length of movie |

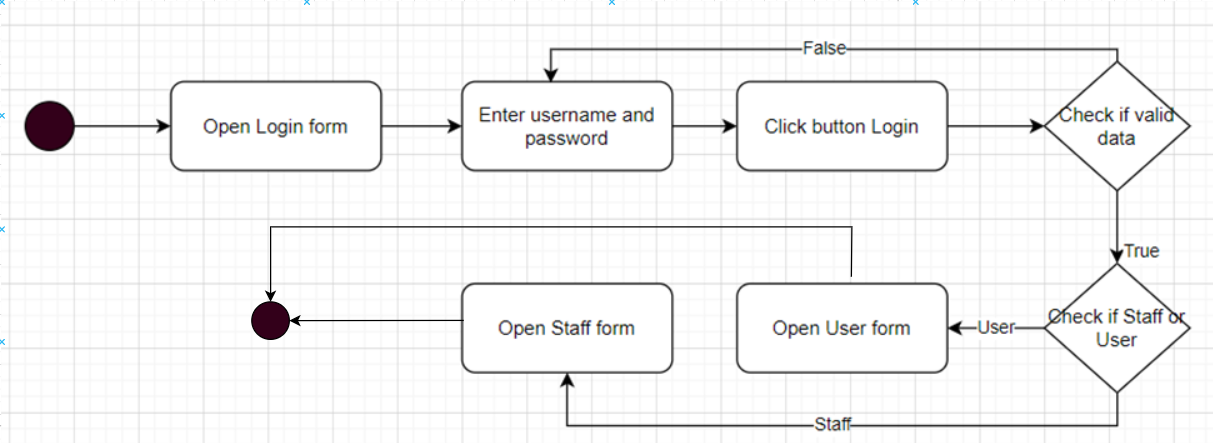
### *RentalVoucher*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | AttributeName | Type | Constraints | Meaning/Note |
| 1 | Id | Int | private | Id of this voucher |
| 2 | nameCustomer | String | private | Name of customer who rent this video |
| 3 | addressCustomer | String | private | Address of customer who rent this video |
| 4 | idVideo | Int | Private | Id of video |
| 5 | nameVideo | String | private | Name of video |
| 6 | Deposits | Float | private | Deposits of video |
| 7 | dateRent | Date | private | Time customer rent this video |

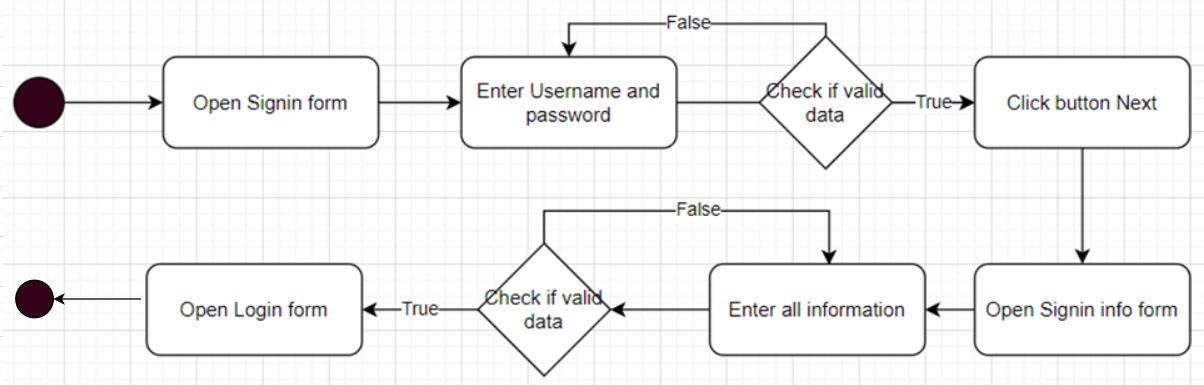
## Class Diagram

## Activity Diagram

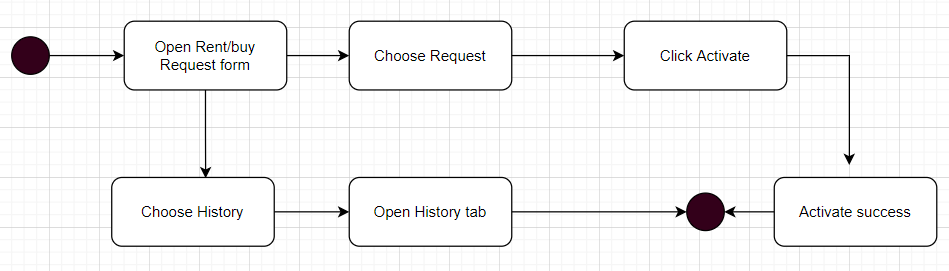
## Login



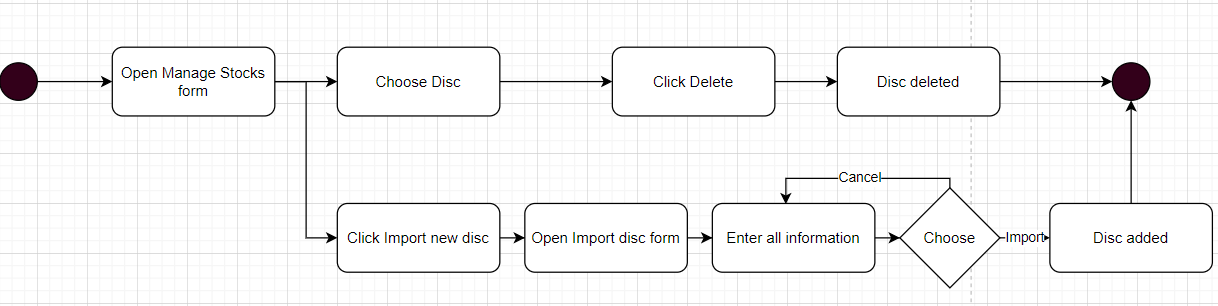
## 8.2 Sign In



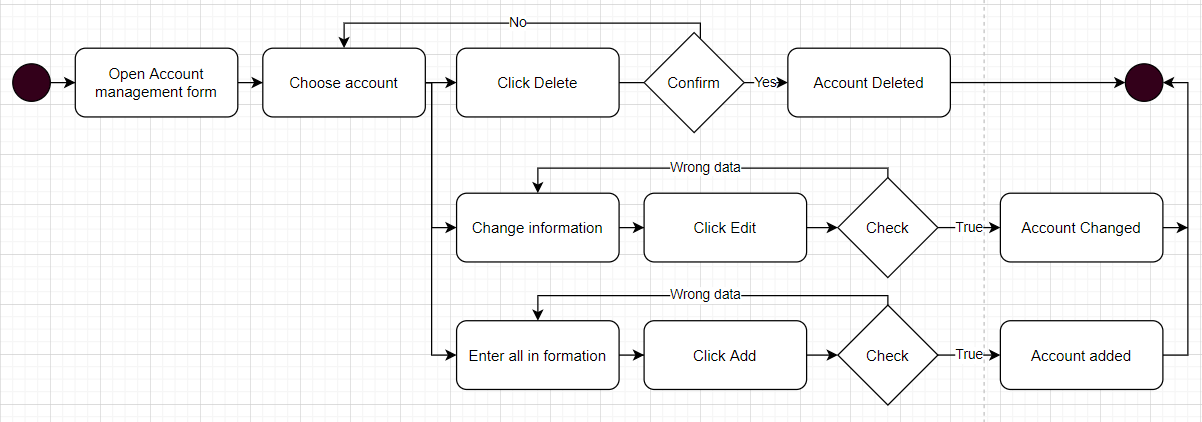
## 8.3 Rent/Buy request



## 8.4 Manage stocks

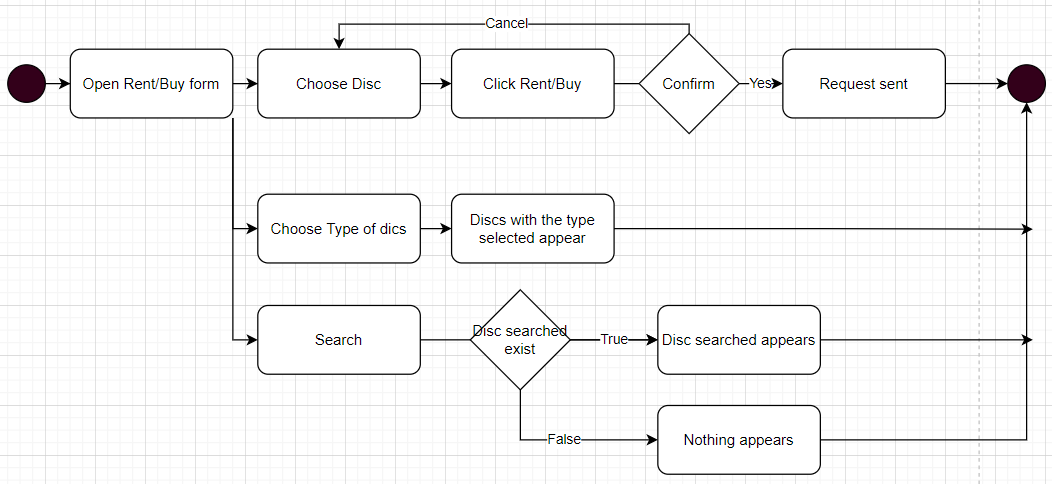


## 8.5 Account Management



## Report

## 8.7 Rent/Buy



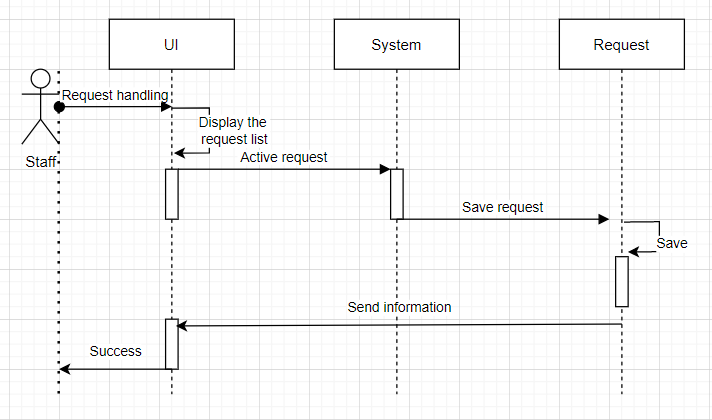
## 8.8 Your cart

## 8.9 History

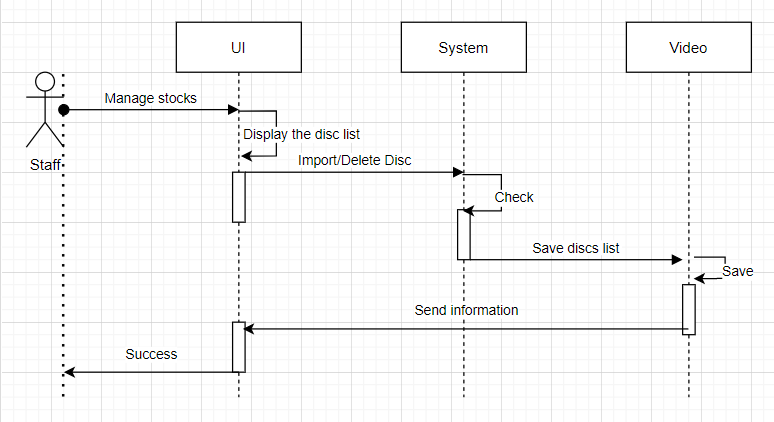
## Sequence Diagram

## Staff

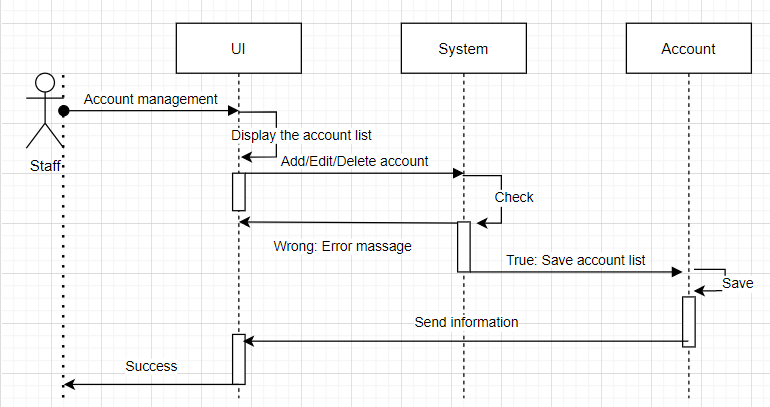
## Rent/Buy request



## Manage stocks



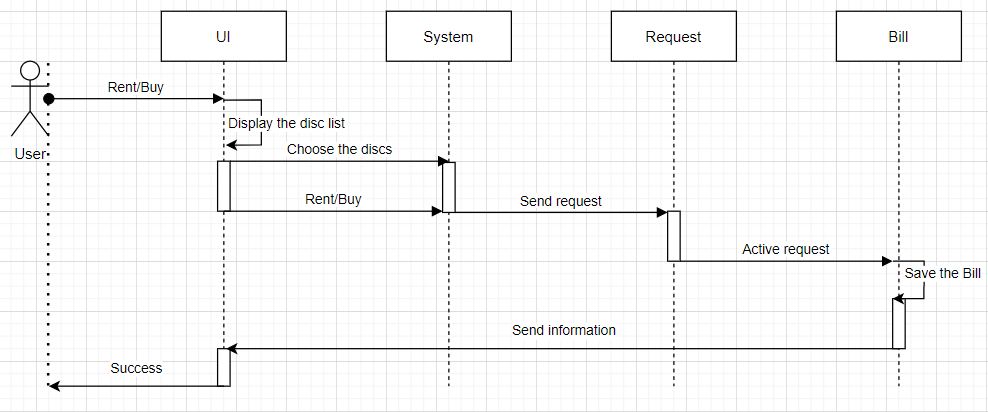
## Account management



## Report

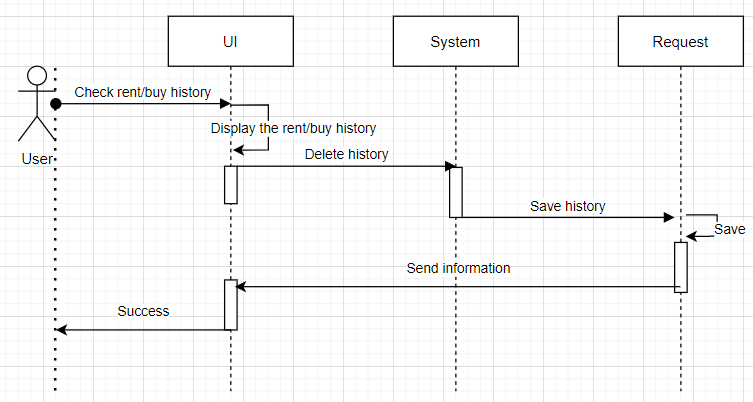
## User

## Rent/Buy



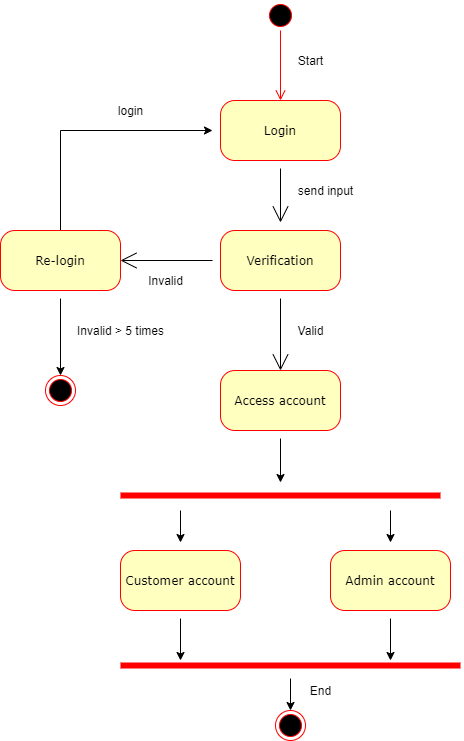
## Your cart

## History

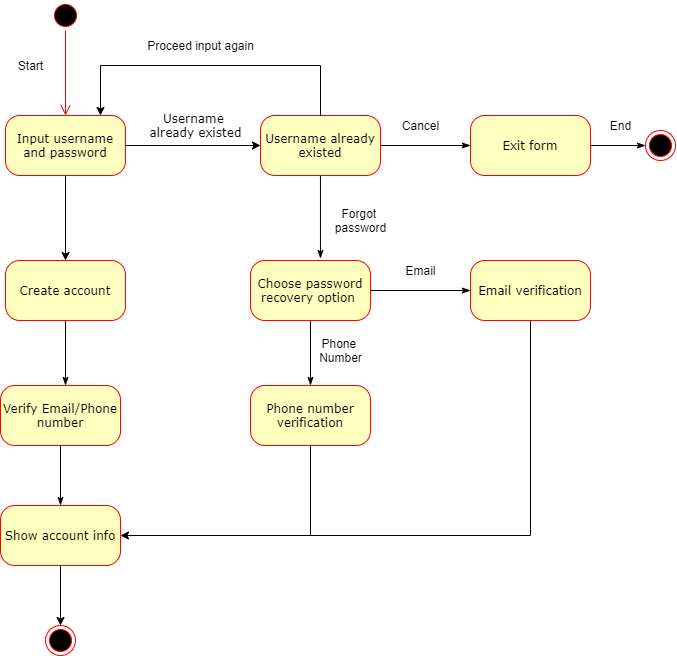


## State Diagram

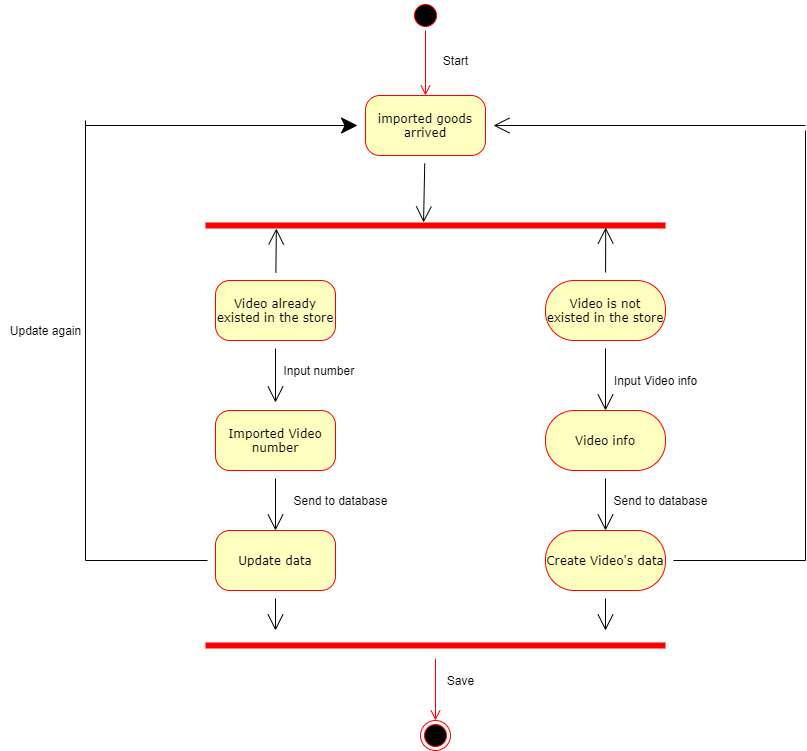
## Login



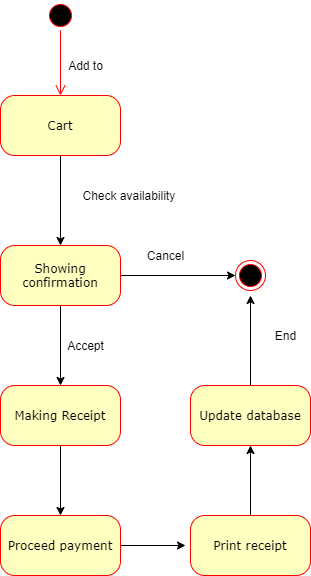
## Create account



## Import discs



## Payment



## UI

# Installation instructions

# Task Sheet

# Summary

At the end of learning OOAD, the group learns how to build a management software, from database design, interface design, to testing and handling. logic often encountered when building a management software. The product developed by the team has an easy-to-see, intuitive, no-frills interface, and has all the necessary key functions.

However, there are still many development directions we have not been able to do, such as the software can only run on one device. The team plans in the future to implement device-to-device communication, to make the software practical.

# Related documents

https://github.com/GiaBao2001zz/VideoRentalStore.git

# Comments and grades

Thank you for teaching my group last semester. Wish you have a happy and joyful 2022 Tet holiday with family and friends.