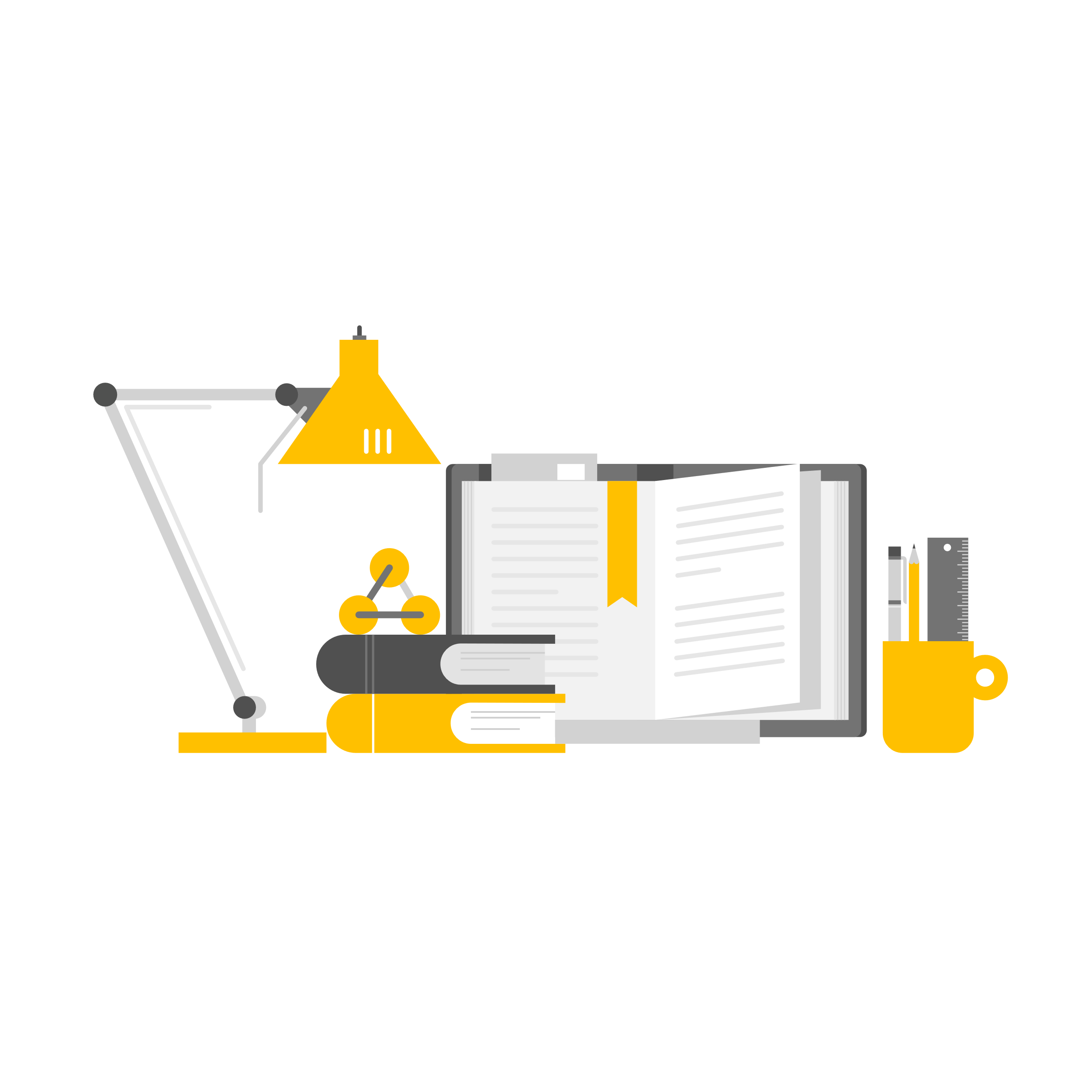
Logo

Description automatically generated

Course:

Data structure & Algorithm

**Synthesis Report**

**Part 3**

**Applications**

Guide teachers

Lê Ngọc Thành

Tạ Việt Phương

Phạm Trọng Nghĩa

Academic year

2020-2021

*Thank you*

**Introduction members:**

|  |  |
| --- | --- |
| **Full name** | **Student ID** |
| Nguyễn Quốc Huy | 20127518 |
| Lại Minh Thông | 20127635 |
| Nguyễn Lê Hoàng Thông | 20127078 |
| Trần Anh Kiệt | 20127545 |
| Đào Đại Hải | 20127016 |

Contents

Contents

[Contents 4](#_Toc82075395)

[Application 1: Buy movie tickets 5](#_Toc82075396)

[I. Problem 5](#_Toc82075397)

# APPLICATION 1: Buy movie tickets

### I. Problem

* **A movie theater** wants a program to **sell tickets** and manage customer information who are members of them. They will keep customer information such as ID, name, and number of points that customers accumulate after purchasing tickets. These points can be exchanged for discounts.
* If the customer is not a member, we can register them as a member or not.
* Accumulated points can only be applied to customers who are already members
* Assumption that we have 3 types of ticket:

|  |  |
| --- | --- |
| Price of ticket | Point |
| LOW: 60000đ | 1 |
| NORMAL: 80000đ | 2 |
| HIGH: 100000đ | 3 |
| *10 points = 50000đ (discount)* | |

### II. Solve problem

We will organize the program in a structured way including Person, Hashtable, Seat and Cinema.

1. **Person structure**

This structure will keep customer information who are members of cinema including ID, name, and accumulated point. Notice that the customer’s ID will be encoded to security.

1. **Hashtable**

* This structure will store all customers as a hash table. The ID in the encrypted form will be hashed to become the key stored in the hashtable. For collision handling, we use double hashing in this problem.
* The hash table's data will be read and written to the file.

1. **Seat**

Every seat in the movie theater has a price, status booked or not, and point that you will get after book it.

1. **Cinema**

This is the biggest structure that you can manage hashtable and seat. Its role:

* Reserve seat for customer: use array to store all seat of the cinema so we can apply **binary search** to find number seat for customer.
* Cancel seat: cancel seat that customer has already chosen
* Payment: make payments and accumulate points for cuscomer
* Discount:
  + If the customer's accumulated points meet the criteria for a discount, the discount will be applied to the bill immediately
  + If the customer has just registered as a member and has enough accumulated points, the discount will also be applied.
* Calculate sold tickets and revenue

# APPLICATION 2: Grocery Store

### I. Problem

* Create a **grocery store** sales app. In the store, we must manage and store product information such as **name**, **price**, and **quantity**.
* When importing and selling to customers, the application can adjust the quantity. We need to make a shopping cart for each customer so they can add and remove the products they want.
* We will proceed to pay the customer after they have selected the product.
* When a product is imported, we can adjust the price as well as the quantity on the store side.

### II. Solve Problem

We will structure the program by using **stock items**, **stock lists**, **binary search trees**, **baskets** (shopping carts), and **linked lists**.

1. ***Stock item***

Stock items will save product information such as name, price, quantity, and quantity in the cart but not yet paid for.

1. ***Stock list***

A stocklist is a list that keeps track of all the products in the store. When the application is active, the stock list will read the product data from the file and save it back to the file when finished.

1. ***Binary Search Tree***

To store products, we will use a binary search tree, which will speed up the search.

1. **Basket**

The shopping cart will save the products the customer has chosen as well as the quantity; the customer can add or remove products. We will charge all of the products in the cart for the customer to pay once they have made a selection.

1. **Linked List**

The products in the basket will be stored in a linked list. It is responsible for adding and removing items from the basket.