TOY RENTAL SERVICE

Toy Class:

Retrieve List<toyId, List<RentalDetails>> to retrieve the previous details about the toy who rented it and all

PQ/Maps

toyHistory()

Create entry

Delete entry

UpdateEntry

Read/Access Entry

Customer Class:

Rental Date , Rental return date

Classes

Address

constructors(all param)

Entry new customers

constructor(id,pass)

Display their profile from db

rent()

Checks the toy availability and if present deposits the fee and rents the toy

Check is performed in map/set data structure of toy ids

Successful rent updation is done in a map

search

Search(with id/name)

Search filters based on toy type

Search based on toy type

return()

Deletes the entries from map and gets the security back

rentalProfile()

Displays the records of toys rented

personalProfile()

Displays the personal details of the customer

toString()

Admin

constructor(id,pass)

Admin constructor

addToy(all params)

Adds new toy ebtry

removeToy(toyId)

removeToy(toyName)

//removeToy(toyType)

Remove toy entries

UpdateToy(toyId)

UpdateToy(toyName)

` //UpdateToy(toyType)

getInventoryValue()

Gets the inventory worth

rentalIncome()

Gets the total price of rented amount

SearchToy(id)

SearchToy(Name)

SearchToy(ToyType)

Member Service

Login

Sign up

DataBases

Toy file

Used for CRUD ops

Customer file

Used in login & signup

ArrayList<Cutomer>

ArrayList<Toys>

ArrayList<ToyRentalDetails>

POJO

Intefaces

CustomerClass implements interfaces

Service layer

UI layer

Datastructures

Map for toyId to List<CustomerIDS>

Map for storing current rentals as toyID to <CustomerID,RentalID>

Customer class var: ArrayList<toyID> for Toys rented

Temporary ArrayLists for storing query results

UI idea

1. Select User Type
   1. Admin
   2. Customer

/////Checking ID Password

1. AdminUI
   1. Add toy
   2. View All toys
   3. Update Toys
   4. RemoveToy
   5. View Toy Rental history
   6. Search Toy etc
2. Customer UI
   1. Search for toy
   2. Rental History of Customer
   3. RentToy
   4. View Current toys which he has

PROJECT STRUCTURE

Main interface

AdminDashboard

Login

AdminService implements AdminServiceImpl

Admin obj:Toy

CustomerDashboard

SignIn

Signup

CustomerService implements CustomerServiceImpl

Customer obj

DatabaseOpsClass

CustomerDB

//AdminDB

Customer Interface

Customer

Toy