Subject System’s Introduction

Subject System : DART (viDeo gAme and song RenTal System)

DART is a software for a video game and music rental system. The company has previously been using the old paper and pen to record the data about games, personnel, and transactions etc. With rentable items being added to the system every day and increasing number of customers, the company decides to digitalize their system by creating a simple Java-based console application. As a result, they create a software system called "DART '' and distribute it to various stores of their franchise.

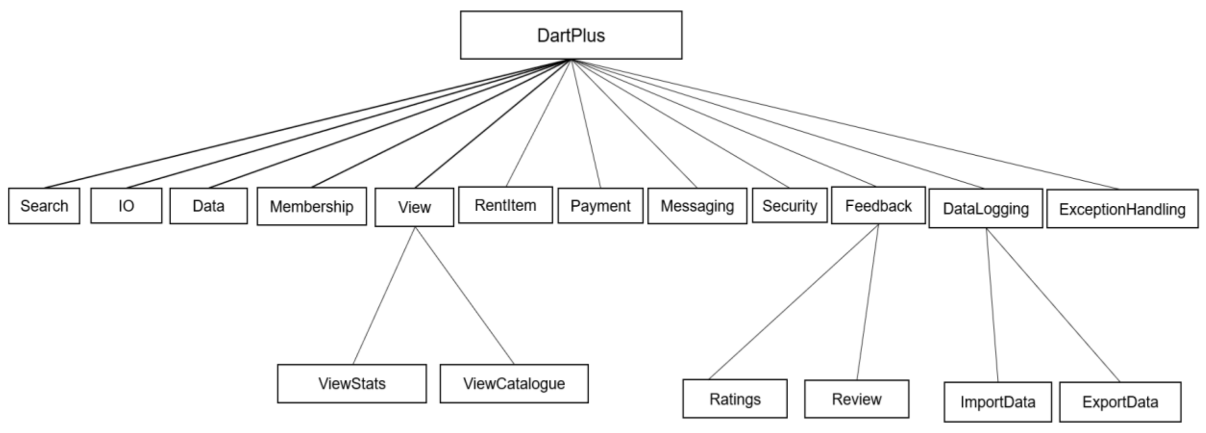
DART can be used by three different types of users: Managers, Employees, and Customers. Managers are responsible for the high-level tasks, such as adding, editing, and removing employees. Employees are responsible for registering customers, upgrading, or downgrading customer-memberships, and adding items to the inventory etc. Both managers and employees can view different statistics to monitor the sales in their respective stores, for example most profitable item, rental frequency of each item, and best reviewed item etc. Customers can interact with the system to rent video games and songs, as well as to interact with each other, and give feedback on the items they rented.

The specifications of the system (including the feature model and feature descriptions) are given below.

DARTPlus

DARTPlus has been created as an extension to DARTBasic with new features. The feature model of DARTPlus is shown along with the new feature descriptions for the features.

Feature Model for DartPlus



Search: All users can browse through the rentable items. SearchItem allows users to find a particular item given an ID. Users can also find items based on their titles, genre, and year of release. Lastly, users can filter items by item type (video game, song). **Location:** Customer.java, Employee.java, Manager.java.

IO: Short for "Input Output"., is an abstract feature comprising sub-features providing user interaction. Comprises all the menus for different users of the system. Presents a list of all options of activities a user can perform, as well as the option to exit in case the user has completed the interaction. Also verifies that users only enter valid options and prompts users to enter a valid option if they provide an invalid value. Responsible for delegating the responsibilities to the right class based on the provided input. Also responsible for handling user input. User input can be of different data types (number, string, character), and used for different purposes (entering password, selecting from the menu options). Checks if the input follows the required format and conforms to other criterion. Re-asks the user for input in case a wrong value is added, or otherwise throws an exception. **Location:** IO

Data: Deals with data management in the system. Managers use the feature to manage the data about employees (registering and removing employees). Employees use it to manage customers (registering and removing them), and to manage rentable items (adding and removing). **Location:** Employee.java, Manager.java

Membership: Allows customers to request upgrades in their memberships (silver, gold, platinum). Customers receive more benefits with each increasing membership level. The feature also stores all pending membership requests and allows employees to accept or refuse them. Employee.java, Customer.java. Security:

View: Abstract feature allowing users to view different things in the system.

ViewStats: Intended for Manager and Employees, the feature calculates and shows different types of stats related to the system. A few examples are average ratings of a rentable item (video game or song), renting frequency of an item, and most profitable item from the catalogue to display to the manager and employees. The feature is also used by employees to view all registered customers, and for managers to view all registered employees. Location: Manager.java, Employee.java ViewCatalogue: Intended for all users. Shows all registered video games and songs to customers, managers, and employees. The views can be generated after filtering as well (e.g., showing songs released in a specific year). **Location:** Manager.java, Employee.java

ViewCatalogue: Intended for all users. Shows all registered video games and songs to customers, managers and employees. The views can be generated after filtering as well (e.g., showing songs released in a specific year). **Location:** Employee.java, Customer.java.

RentItem: Allows customers to rent video games and songs. Responsible for checking if the item is in stock, as well as if the customer has exceeded his/her maximum limit of rentable items. For customers with different priority levels (regular customer, silver member, gold member, platinum member), the maximum limit of items rented at one point varies. **Location:** Customer.java

Payment: Allows customers to return and pay for the rented items. As a way to reward the loyal members, the store gives credits, which can be used to rent items in the future. Customers of different priority levels (silver, gold, platinum) get different amount of credits after returning every item (1,2, and 3 respectively). Customers can use the accumulated credits to pay for items, 5 credits each. The feature also allows customers to redeem discounts (10%, 15%, 25%) depending on their priority level (silver, gold, platinum). Location: Customer.java

Feedback: Abstract feature allows customers to provide critique on their rented items.

Ratings: Customers can rate games and songs they rented. Ratings range between 1 and 5, the higher the better. Items can only be reviewed by customers upon returning them. **Location:** Customer.java

Review: Allows customers to provide textual feedback on their rented items. Reviewing is optional. Reviews can only be provided by customers upon returning a rented item. **Location:** Customer.java

ExceptionHandling: Adds exceptions in all the methods dealing with user input, and throws exceptions against invalid inputs, for example, NameEmptyException (customer name not provided during registration) and NegativeSalaryException (salary of less than 0 added for an employee by the manager). Location: Utilities

Messaging: Allows customers to send messages to each other. Sets an empty inbox against each customer, which is filled by the messages sent by other customers. Customers can read the messages, which changes their status from unread to read. They can also respond to messages and remove messages from their inbox. **Location:** Message.Java, Controller.java, Customer.java, CustomerSilver.java, CustomerGold.java, CustomerPlatinum.java, Employee.java

Security: Allows employees to add passwords for customers when registering them. Also responsible for checking if the entered passwords at the time of logging in are correct but enhanced to automatically generate unique passwords for each new registered customer. **Location:** Tools.java.

DataLogging: Abstract feature allowing managers to read data from files and write data back to files.

ImportData: Managers can import data into the system. Specifically, they can add employees, games, songs, and customers from text files into the system. **Location:** Manager.java, Controller.java

ExportData: Manager can export data to files. Specifically, they can write the data of the registered employees, games, songs, and customers in text files. **Location:** Manager.java, Controller.java