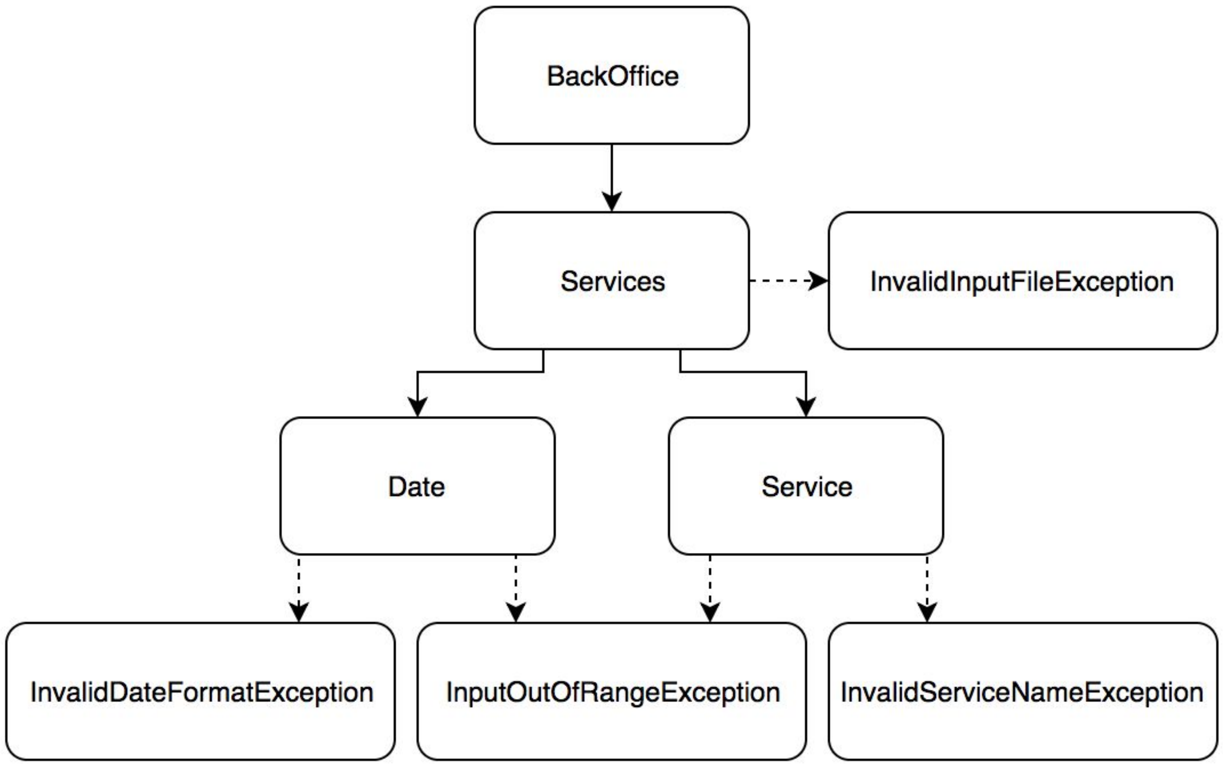
CISC 327 Assignment 4
Design Document

The Spice Girls (Group 13)

Natasha Djurdjevic
Allan Legemaate
Megan McClure
Irena Dunjic

Dependancy Diagram:


BackOffice
Services
InvalidInputFileException
Date
Service
InvalidDateFormatException
InputOutOfRangeException
InvalidServiceNameException

Classes:

BackOffice : main class that contains the main method (deals with loading input files into the program, and writing output files)

Methods:

- **void** main(String[] args) : starting point for program

Date : date object class, holds all the information for a date (including year, month, and day)

Methods:

- Date(int year, int month, int day)
    - date constructor, calls setYear, setMonth, and setDay methods. if all successful, the date is set
- Date(String date)
    - date constructor, calls setYear, setMonth, and setDay methods. After separating string in format YYYYMMDD. If all successful, the date is set
- **int** getYear()
    - returns year
- **int** getMonth()
    - returns month
- **int** getDay()
    - returns day
- **void** setYear(int year)
    - validates that the year is within range. If the year is valid (between 1980 and 2999 inclusive), sets year attribute
- **void** setMonth(int month)
    - validates that the month is within range. If the month is valid (between 1 and 12 inclusive), sets month attribute
- **void** setDay(int day)
    - validates that the day is within range. If the day is valid (between 1 and 31 inclusive), sets day attribute
- **String** toString()
    - Returns the date as a string formatted as YYYYMMDD

InputOutOfRangeException : exception class used to throw exception when a given input is out of range

Methods:

- InputOutOfRangeException(String message)
    - Throws exception with given message
    - Used in Date class for: Date constructor, setYear, setMonth, and setDay methods
    - Used in Service class for: Service constructor, setNumber, setCapacity, and setNumberSold methods

---

InvalidDateFormatException : exception class used to throw exception when the formatting for a date is incorrect

Methods:

- InvalidDateFormatException(String message)
    - Throws exception with given message
    - used in Date class for: Date constructor

InvalidInputFileException : exception class used to throw exception when the input file is invalid

Methods:

- InvalidInputFileException(String message)
    - Throws exception with given message
    - Used in Services class for: readCSF, and readTSF methods

---

InvalidServiceNameException : exception class used to throw exception when the given service name is invalid

Methods:

- InvalidServiceNameException(String message)
    - Throws Exception with given message
    - Used in Service class for: Service constructor, and setName methods

---

Service : service object, holds all the information for a service (including service number, service capacity, the current number of tickets sold for the service, the service name, and the service date)

Methods:

- Service(int number, int capacity, int numberSold, String name, Date date)
    - Service object constructor. calls setNumber, setCapacity, setNumberSold, setName, and setDate
- **int** getNumber()
    - Returns service number
- **int** getCapacity()
    - Returns service capacity
- **int** getNumberSold()
    - Returns number of tickets sold for service
- **String** getName()
    - Returns service name
- **Date** getDate()
    - Returns service date
- **void** setNumber(int number)
    - Checks if service number is valid, if valid, sets service number
- **void** setCapacity(int capacity)
    - Checks if service capacity is valid, if valid, sets service number
- **void** setNumberSold(int numberSold)
    - Checks if number of tickets sold is valid, if valid, sets number tickets sold
- **void** setName(String name)
    - Checks if service name is valid, if valid, sets service name
- **void** setDate(Date date)
    - Sets service date
- **String** toString()
    - Returns the service as a string formatted for the Central Services File (CSF)

Services : holds a list of Service objects that is read from the Central Services File (CSF), and will have each transaction that is read in from the transaction summary file applied to it. It can then create a String formatted for writing into the Updated Central Services File (CSF) once all the transactions have been completed.

Methods:

- **boolean** validatePath(String path)
    - Checks if file/directory exists
- **void** readCSF(String csfPath)
    - Reads Central Services File (CSF) into data structure
- **void** readTSF(String tsfPath)
    - Reads Transaction Summary File (TSF) into data structure
- **void** addService(Service newService)
    - Adds service to list

- **void** removeService(int serviceNumber)
    - Removes service from list
- **String** getVSFString()
    - Returns list of valid service numbers (generated from serviceList list)
- **String** getCSFString()
    - Returns the service as a string formatted for the Central Services File (CSF)
- **Service** findService(int serviceNum)
    - Looks through the serviceList and finds the required service object
- **void** sellTicket(Service service, int ticketNum)
    - Adds the given number of tickets to the given service object
- **void** cancelTicket(Service service, int ticketNum)
    - Cancels the given number of tickets from the given service object
- **void** changeTicket(Service originalService, Service destinationService, int ticketNum)
    - Changes the given number of tickets from the original service to the destination service, and returns an array containing the two updated Service objects