6/21/2019

Kyer Potts

30003389

Games Store Application Report

Programming II

Table of Contents

**No table of contents entries found.**

# Introduction

The Games store program is a simple data capture and retrieval program that will allow the user to save and view simple forms of business transaction related data. The user will also be able to generate new transaction records fluidly via the user interface, by selecting previously saved data from relevant fields and filling out the rest of the required fields. This will be a simple step by step process, which is inherently familiar with the design and layout of the user interface. Simplicity is key for this application and ease of use is prioritised over functionality. No option to remove or alter records will be provided, and no search or sorting functionality will be implemented into the program. Data populated into the list fields will be saved to three separate binary files. These files will be loaded when the application begins and saved when the application is closed. No further options to save or load data will be provided to the user. Minor additional functionality will be provided to clear fields in the form of a double click option assigned to the ProductID and CustomerID controls. This will clear all other relevant controls of user entered data, and any data added to those controls will not be saved to their respective list boxes.

All appropriate fields will need to be manually entered by the user to save data, except for three fields within the Transactions section of the application. These fields will be automatically populated upon selection of the appropriate items from both the Products and Customers lists. These fields have also been colour-coded to make identification of these fields easier for the user.  
All relevant controls within the program have had tooltips added with simple instructions on their respective uses.

## SDLC

An appropriate option of SDLC for a project of this size and scale would be the Waterfall model.

With clear predefined requirements and a low risk of scope creep, the Waterfall model is ideal for the development of a small-scale enterprise application with minimal functionality and interactive elements. The step by step focus of the waterfall model allows for a streamlined simplistic development approach, moving from stages upon completion until project completion and client handover. The Waterfall model consists of 5 stages; Requirement definition, Project Design, Project build and development, testing and finally deployment and maintenance.

(Sarycheva, 2019)



# Analysis

## Input

User data input consists of product information, customer information and transaction information. Additional input required from the user will consist of click functionality in the form of “Add” buttons to record the information to the relevant list box controls. Listbox controls will also require click input to highlight and focus listbox selections.

## Processes

* Binary write and save processes will be required on form load and form close.
* Add processes will be required to save user input to appropriate List<T> object.
* Display and clear processes will be required to print saved user input into appropriate listbox controls.
* Error processes to ensure that data entered by user is complete and correct.
* Item selection process to populate Transaction fields for core functionality
* Item selection process to select appropriate Item from Products and Customer list boxes required for core functionality.
* On double click process required to clear relevant Products and Customer fields of all user entered data.

## Output

* Separate binary files written for each appropriate List<T> object.
* Data from List<T> objects written to appropriate listbox controls.
* Automatic population of Transaction fields on Product and Customer selection from relevant list boxes
* Automatic population of Product and Customer fields when relevant Transaction item is selected from list box.

# Project Plan

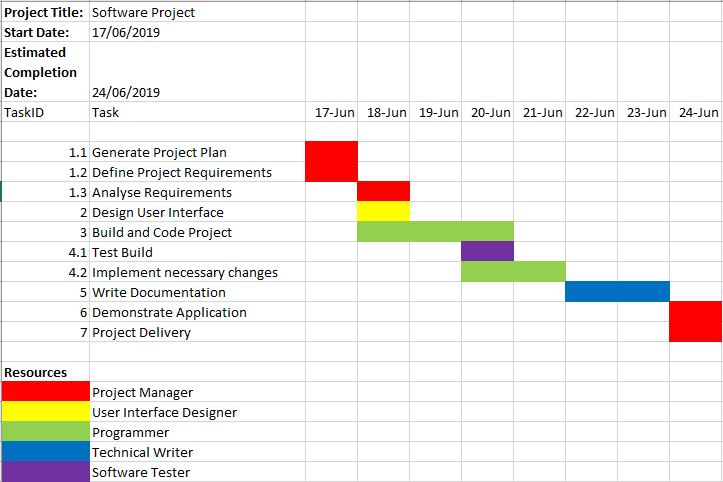
## Required Tasks

* 1.1: Generate project plan. – Project Manager
* 1.2: Define project requirements – Project Manager
* 1.3: Analyse Requirements – Project Manager, User Interface Designer, Programmer
* 2: Design User Interface – User Interface Designer
* 3: Build and code project – Programmer
* 4.1: Test build – Software Tester
* 4.2: Implement necessary changes – Programmer
* 5: Write Documentation – Technical Writer
* 6: Demonstrate Application – Project Manager
* 7: Project Delivery – Project Manager

## Required Resources

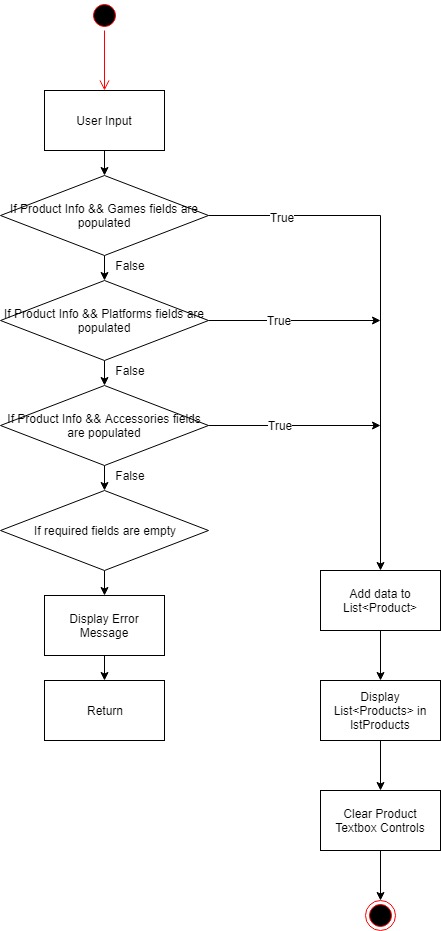
* Computer
* Workspace
* Desk
* Visual Studio
* Microsoft Office
* UML document editor
* Data backup

## Gantt Chart

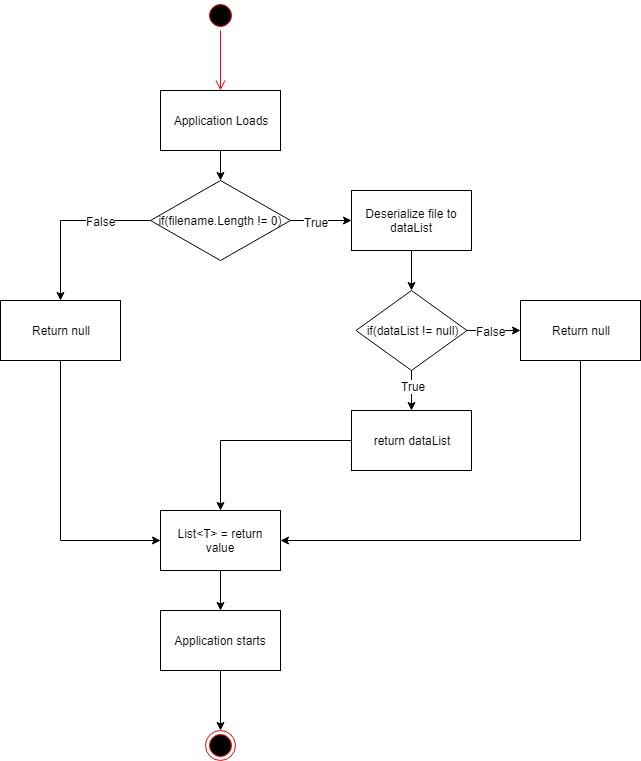


# Design

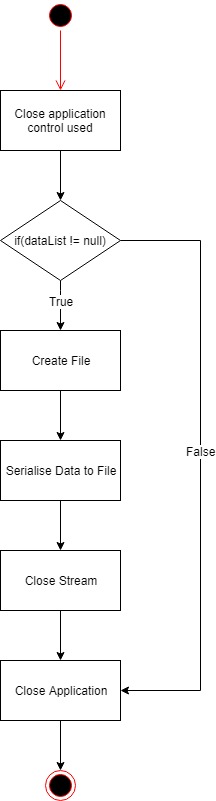
## Product Add Button



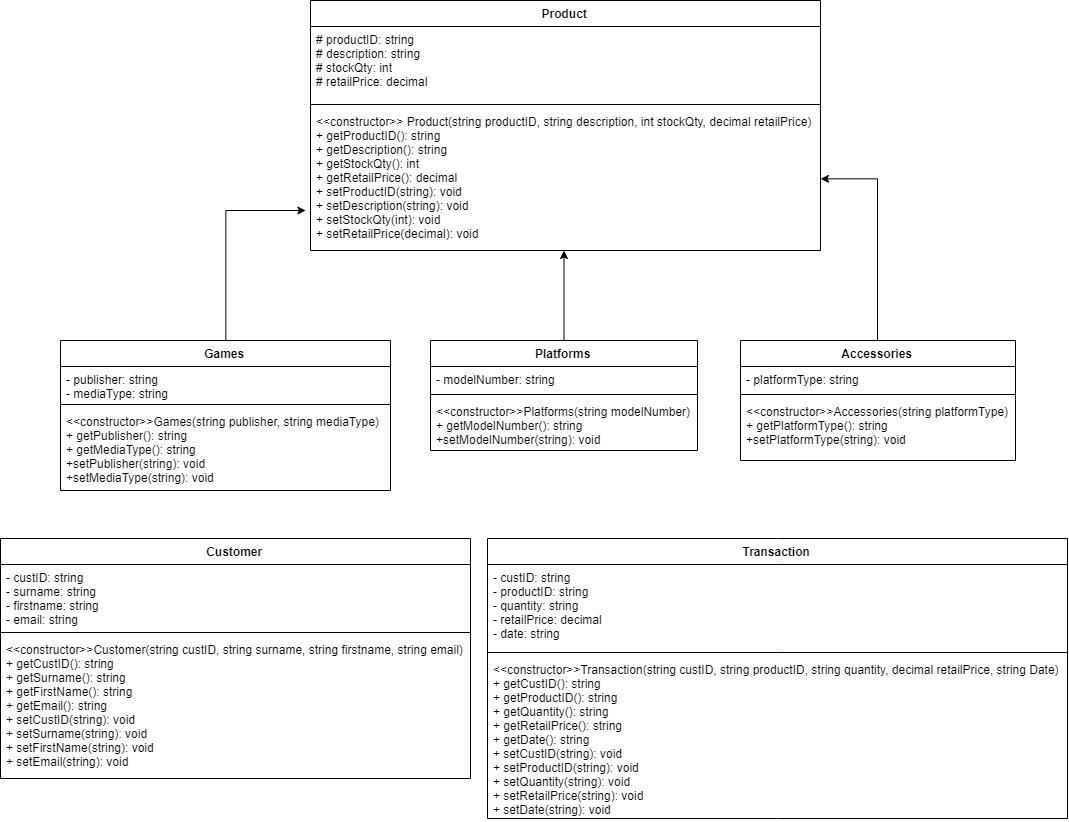
## Read Binary File



## Write Binary File



## Class Diagram



# Test Data and Evidence

|  |  |  |  |
| --- | --- | --- | --- |
| Test ID | Description | Expected Outcome | Outcome |
| 1 | Add Game to List<T> | Data Added to List<Product>, displayed in Products list box |  |
| 2 | Add Platform to List<T> | Data added to List<Product>, displayed in Products list box |  |
| 3 | Add Accessory to List<T> | Data added to List<Product>, displayed in Products list box |  |
| 4 | Error message returned when textbox controls incorrectly populated | Error message is displayed, data is not added to List<Product>, data is not displayed in products list box |  |
| 5 | Add Customer to List<T> | Data added to List<Customer>, displayed in Customers list box |  |
| 6 | Add default customer to List<T> | Yes/No message box is displayed, if clicked yes, default customer will be created |  |
| 7 | Populate Transactions fields with listbox Index selections | Appropriate fields are populated in Transactions textbox controls |  |
| 8 | Error message displayed because Transaction fields are incomplete | Error message displayed, no data added to List<Transaction>, data is not displayed in Transaction list box |  |
| 9 | Add Transaction to List<T> | Data added to List<Transaction>, displayed in Transactions list box |  |
| 10 | Select relevant Product and Customer records when Transaction Item is selected | Relevant records are selected, and All data is populated into the relevant Products and Customers fields |  |
| 11 | Products textbox controls are cleared when ProductID field is double clicked | Relevant fields are cleared |  |
| 12 | Customers textbox controls are cleared when CustID field is double clicked | Relevant fields are cleared |  |
| 13 | Data is saved upon program close | 3 .dat files are created for each respective List<T> |  |
| 14 | Data is loaded upon program open | Data is loaded successfully |  |

# Program Code

//Method to clear Product textbox controls

public void ClearProdTbs()

//Method to clear Customer textbox controls

public void ClearCustTbs()

//Method to clear Transactions textbox controls

public void ClearTransTbs()

//Method used to display products in listbox

public void DisplayProducts()

//Method used to display customers in listbox

public void DisplayCustomers()

//Method used to display transactions in listbox

public void DisplayTransactions()

//Serialisation save method

public void SaveData<T>(string filename, List<T> dataList)

//Deserialisation open method

public List<T> OpenData<T>(string filename)