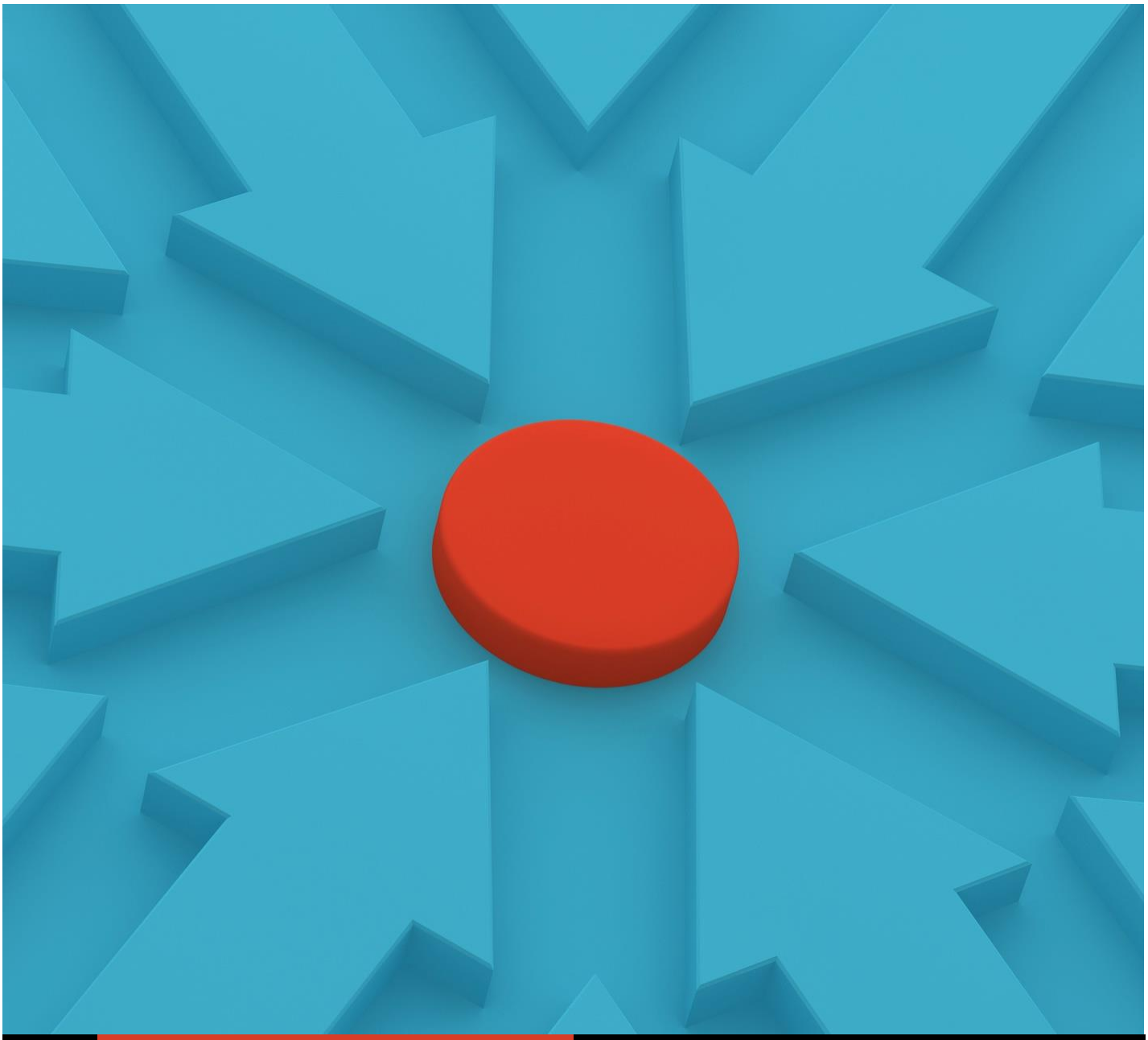


Class Model with Packages, Package Names, Goals, and Design Tradeoffs, Logical Data Flow and Physical Data Flow Diagrams.

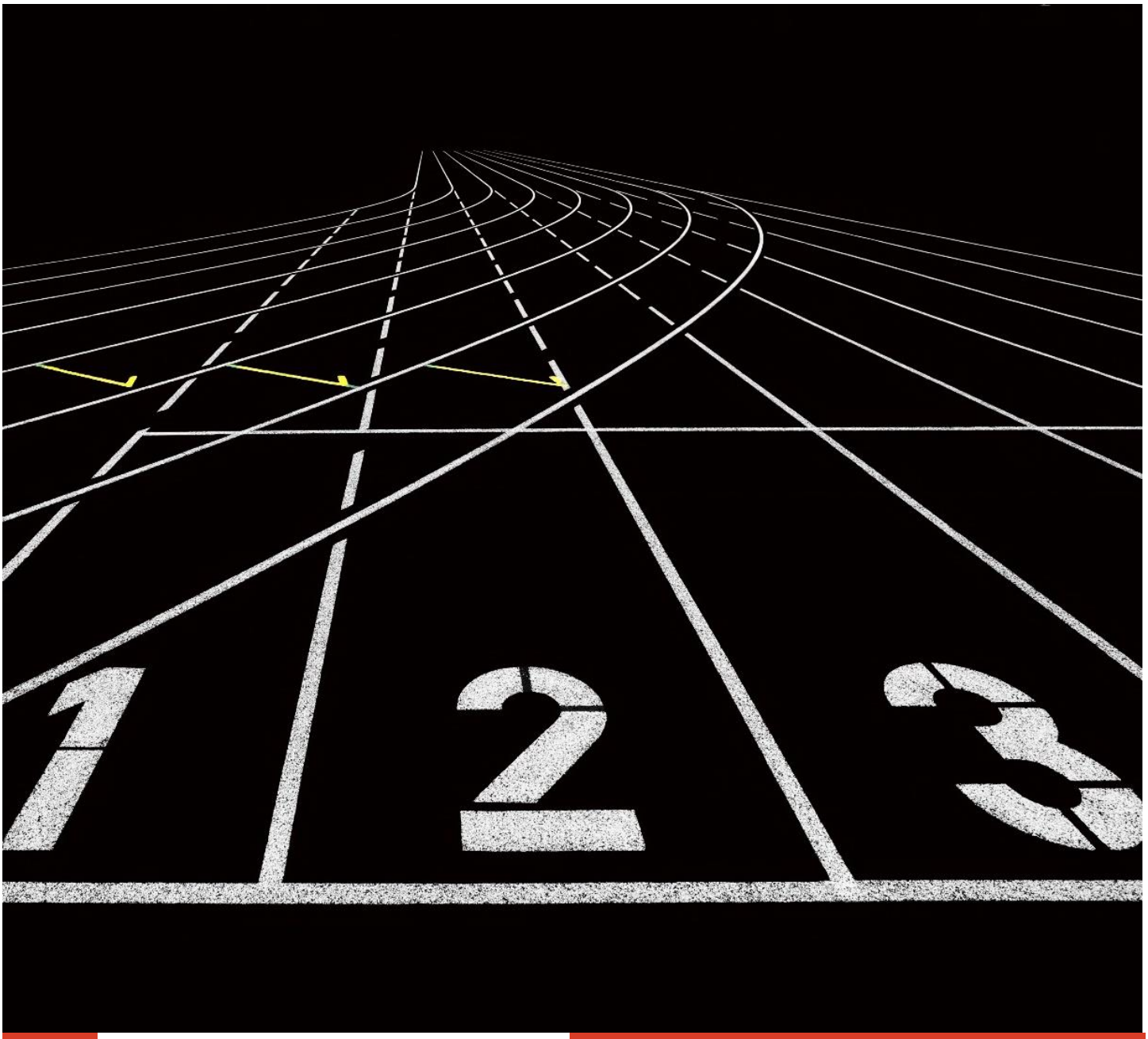
Abstract: What does the revised Class Model diagram look like with the incorporated packages? What are these packages' names, goals, and design tradeoffs? What does the Logical Data Flow diagram look like for the BestPurchase application? What does the Physical Data Flow diagram look like for the BestPurchase application?

Harrison Huston

CS682 – Assignment 5, February 21, 2023

TABLE OF CONTENTS

FIGURES



INTRODUCTION

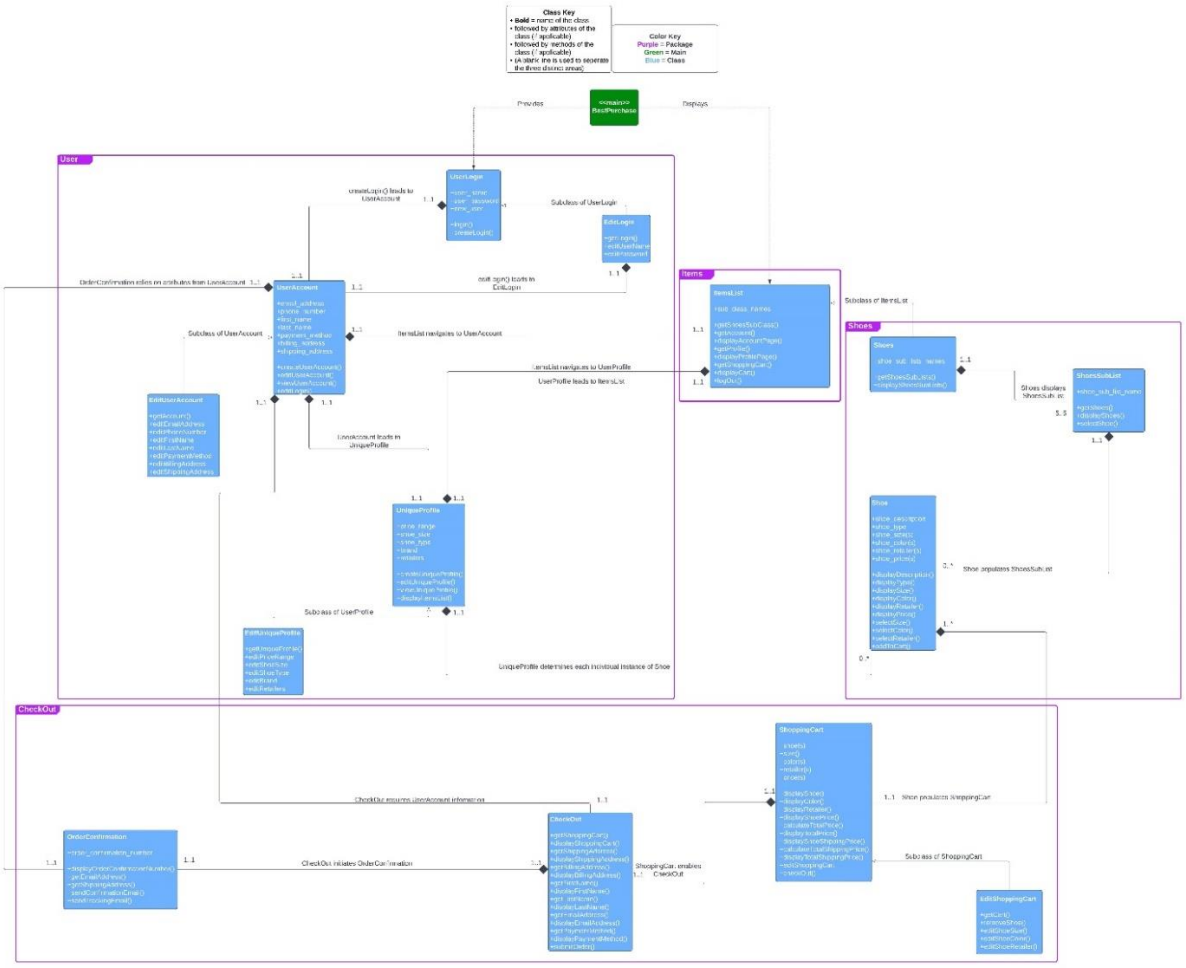
Assignment 5 focuses on the design goals and data flows for the BestPurchase application. The Class Model diagram is revised to include packages. The packages are described with their name, design goals, and design tradeoffs. The Logical Data Flow and Physical Data Flow diagrams are created and described.

5.1 CLASS MODEL

The below class diagram has been revised and updated following the Class Model diagram created in the previous paper. In *Appendix 1*, a portion of its description is included from that paper.

The updated version of the Class Model diagram below now includes four distinct packages: User, Items, Shoes, and Checkout. The “BestPurchase Package”, which acted as an overarching package for the design, has been removed to better accommodate the design and new packages. While the names of the classes and their relationships to other classes have not changed, the diagram spacing has been edited to better fit the new packages and their structures within the diagram. The encapsulation of all attributes of the classes have been changed from (~) package access, to (+) public access to better accommodate the new packages and access to these attributes. The method “viewUserAccount()” within the UniqueProfile class has been changed to “viewUniqueProfile()”, as this was a minor error in the original diagram. The EditLogIn class and getLogIn() method names were also edited to EditLogin and getLogin(). Lastly, the “Text Key” has been renamed to the “Class Key”, and the “Encapsulation Key” has been removed.

Note: The original Class Model diagram from the previous paper is located in *Appendix 2*.



(“Module 4,” 2023; Williams, 2023)

5.2 PACKAGES

- **Package name**:
  User

- **Design goals for this package**:
  The User package facilitates the process of creating a login, an account, and a unique profile for new users, as well as facilitating the login process and editing of the account and unique profile for current account holders.

- **Design tradeoffs for this package**:
  The User package consists of three classes UserLogin, UserAccount, and UniqueProfile, each of which has a subclass: EditLogin, EditUserAccount, and EditUniqueProfile. The classes primarily rely on one another to facilitate the above design goals for the package. Due to the structure of the package and their goals, there is a high level of cohesion and sufficiency in the package design. The package is also very flexible and can accommodate changes to requirements for logging in, the user account information and the unique profile details. When considering reusability for the package, it could easily be used in another application to facilitate similar goals related to the design goals of the package with small changes to some of the class structures.
  Due to the structure of the application, there does exist some small levels of coupling in the design. The User package does depend on the Items package from a navigation standpoint and moving from the Unique Profile to the Items List page and in turn, the Items package depends on the User package to navigate from the Items List to the User Profile page and the User Account page. The Shoes package is also dependent on the User package as it relates to the population of instances of Shoes based on the Unique Profile of the user. Lastly, the CheckOut package depends on the User package to gather information related to the Account Holder to facilitate the shipping, billing, and confirmation processes.

(“Module 5,” 2023; Williams, 2023)

- **Package name**:
  Items

- **Design goals for this package**:
  The Items package facilitates the displaying of sub lists of items to be sold through the BestPurchase application.

- **Design tradeoffs for this package**:

The Items package consists of a singular class, ItemsList. Due to this, the level of cohesion is extremely high as well as its sufficiency. This package was designed to increase the flexibility of the application. If the BestPurchase application were to eventually sell more items than just shoes, this package would be able to be used in conjunction with the new item's package, similar to how it is currently designed in relation to the Shoes package. With this in mind, the package also has a high level of reusability, with minor changes to the class, the package could act in a similar manner for any application requiring a list of items.

There does exist coupling in regard to this package. As mentioned, the Items package is dependent on the User package to navigate to both the Unique Profile and User Account pages, and the User package depends on the Items package to navigate from the Unique Profile to the Items List page. Lastly, the Shoes subclass, which is located in the Shoes package is the subclass of the ItemsList class, located in the Items package. While this is not ideal, this design structure allows for the ItemsList class to have various subclasses pertaining to different items that could be sold in the future. The higher degree of coupling based off this structure, allows for better flexibility and reusability of the Items package and is the major tradeoff in relation to this package.

("Module 5," 2023; Williams, 2023)

- **Package name**: Shoes

- **Design goals for this package**:

The Shoes package facilitates the process of selecting and viewing sub lists of shoes, as well as viewing the individual shoes that the sub lists contain.

- **Design tradeoffs for this package**:

The Shoes package consists of the Shoes subclass and the ShoesSubList and Shoe classes. The package was designed to have a high level of reusability in anticipation that the BestPurchase application could expand to sell more items. With the current package structure, it could be reused with edits to the classes in the creation of additional packages to accommodate new items to be sold through the BestPurchase application. With this in mind, it also is flexible, in that changes to the classes could allow it to handle various different items. The package also has a high level of sufficiency in its design in relation to the goals of the package and a high level of cohesion.

The package has three main areas of coupling. Firstly, the Shoes subclass within the Shoes package is the subclass of the previously mentioned ItemsList class which is contained within the Items package. The Shoes package is also depended on by the CheckOut package as it relates to populating the ShoppingCart class with instances of each Shoe. Lastly, the package does rely on the User package as it pertains to the instances of the Shoe class within the Shoe package. The UniqueProfile class within the User package, as well as the ShoesSubList class in the Shoe package, determine the instances of the Shoe class that will be shown to the user through the application.

("Module 5," 2023; Williams, 2023)

- **Package name**: CheckOut

- **Design goals for this package**: The CheckOut package facilitates the process of completing an order, including displaying the shopping cart, editing the shopping cart, submitting the order, and the creation of an order confirmation.

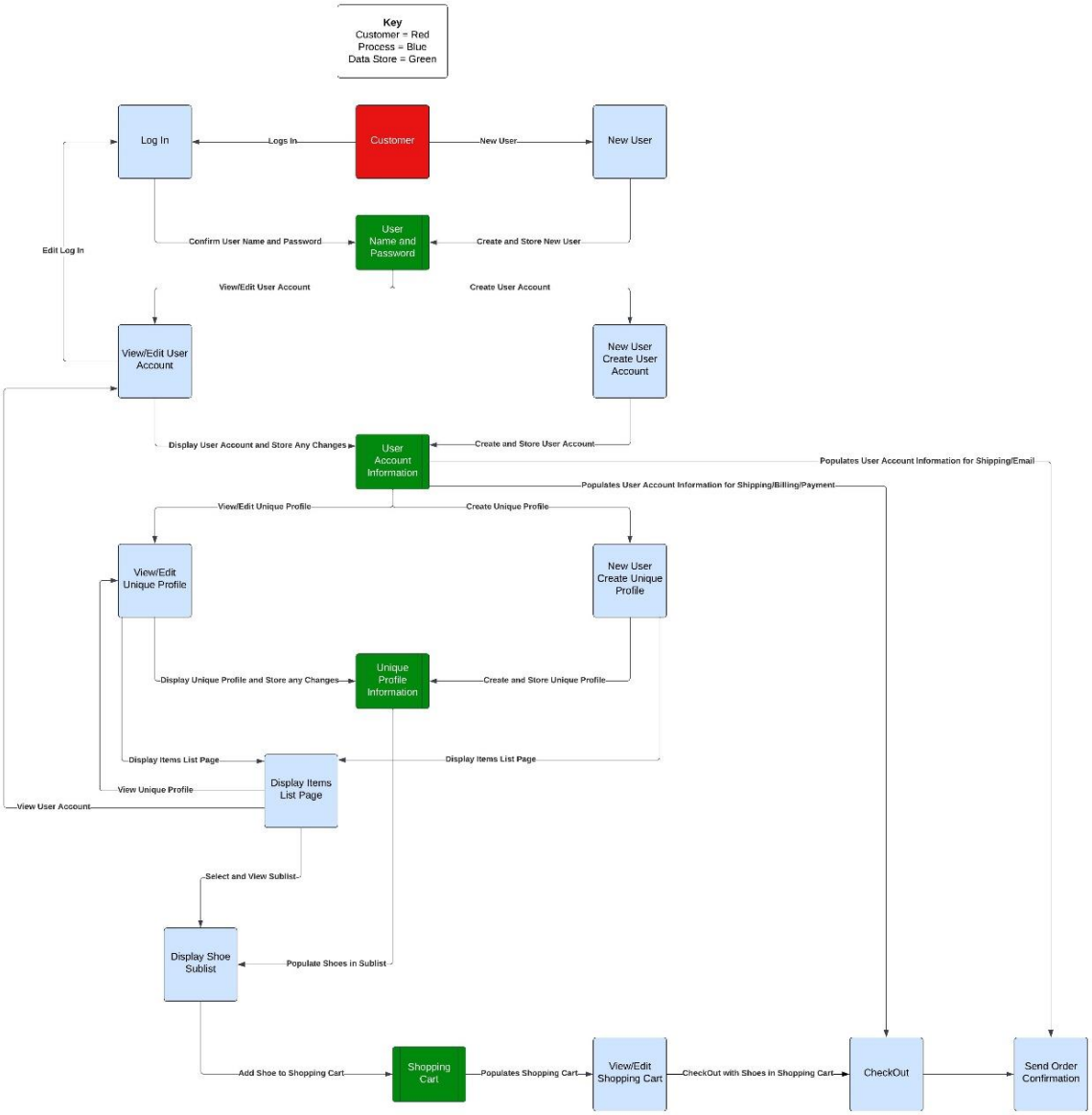
- **Design tradeoffs for this package**:
  The CheckOut package consists of three classes, ShoppingCart, CheckOut, and OrderConfirmation, as well as one subclass, EditShoppingCart. The package itself has a high level of reusability and could be used in other applications to facilitate the processes involved in completing an order. With minor changes to the classes within the package, the package is designed to have a high level of flexibility in accommodating new items to be potentially sold through the BestPurchase application in the future. The package has a high level of cohesion, with each class playing an integral part in fulfilling the design goals of the package.
  There are three instances of coupling as it pertains to the design of this package. Firstly, the CheckOut package interacts with the User package, more specifically, the UserAccount class within the User package. The CheckOut and OrderConfirmation classes within the CheckOut package rely on Account Holder information within the UserAccount class to help facilitate the check out and order confirmation processes. Lastly, the instances of the Shoe class within the Shoes package are what populates the ShoppingCart class within the CheckOut package.

(“Module 5,” 2023; Williams, 2023)

5.3 LOGICAL DATA FLOW DIAGRAM

The below Logical Data Flow diagram depicts the processes and flow of data through the BestPurchase application. Starting from the customer, there are separate paths pertaining to current and noncurrent Account Holders. One area that may not be obvious in its meaning, is on the left there are paths from Display Items List Page to both View/Edit Unique Profile, and View/Edit User Account. This would be for a case in which the user was to already be an Account Holder and is navigating from the Items List page to either of those pages to either view or edit them. Lastly, there is a key located at the top of the diagram to better understand its elements.

(Williams, 2023)

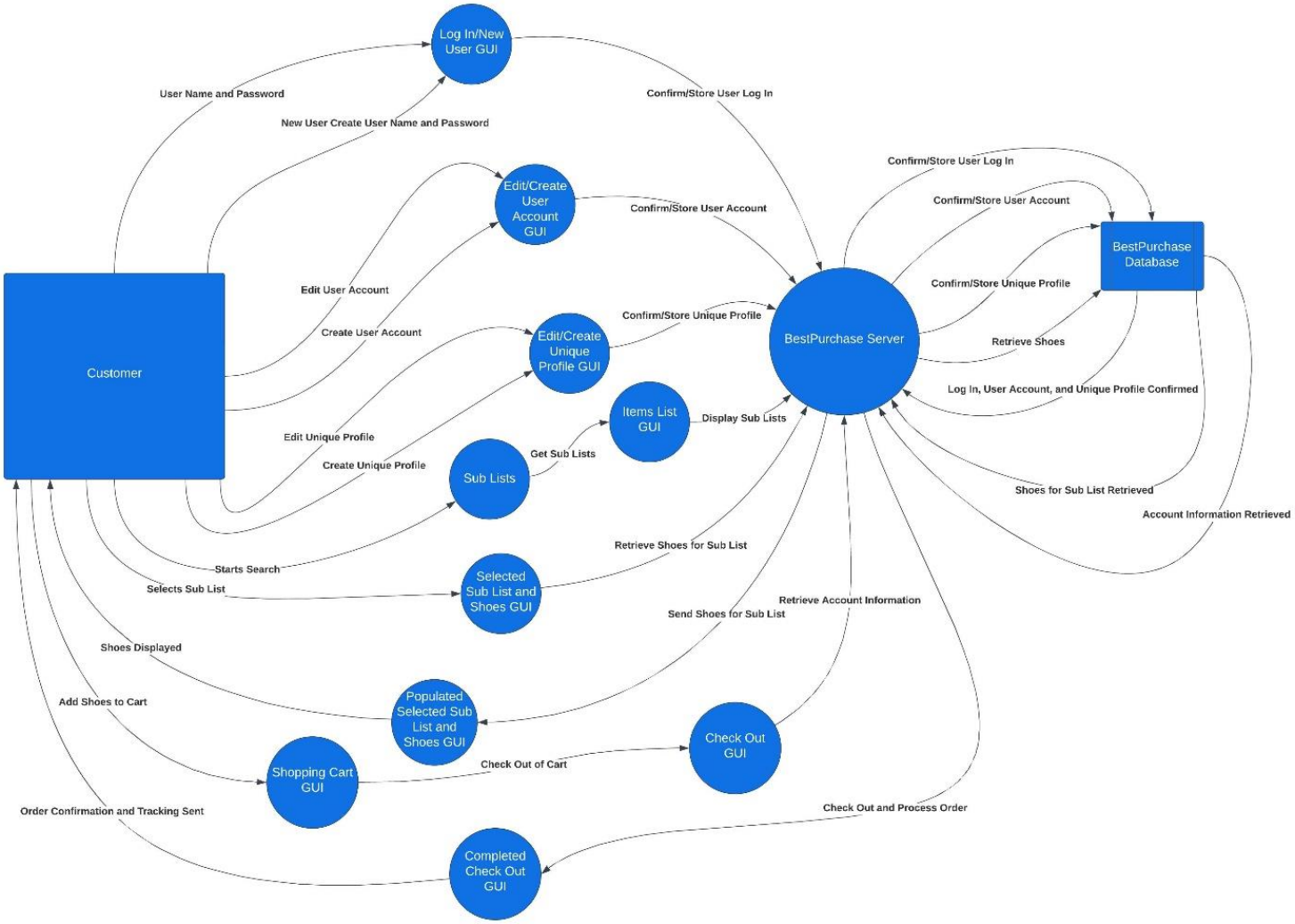


(Williams, 2023)

5.4 PHYSICAL DATA FLOW DIAGRAM

The below Physical Data Flow diagram depicts how the BestPurchase application is implemented and the relationship between the customer and the physical processing elements. The diagram depicts both new user and current account holder flows of data. In an effort to clarify the below diagram, if it is unclear, it is important to pay attention to the direction of the arrows and lines. Keep in mind, there are nine inputs into the BestPurchase Server. The BestPurchase server also has six outputs, four of which are to the BestPurchase Database. The BestPurchase Database has four inputs and three outputs. All processing elements are labeled, as well as each line to better understand the flow of data.

(Williams, 2023)



(Williams, 2023)

CONCLUSION

The Class Model diagram displays a revised version of the previous Class Model diagram and provides for the inclusion of four distinct packages; User, Items, Shoes, and CheckOut. The packages are described regarding their design goals and design tradeoffs in relation to the project. The Logical Data Flow diagram illustrates the processes and flow of data through the BestPurchase application, while the Physical Data Flow diagram depicts the relationship between the customer and the physical processing elements.

(Williams, 2023)

SUMMARY

In summary, the paper focuses on design goals and data flows for the BestPurchase application. It provides a Class Model diagram with the inclusion of packages. The packages are described in relation to the design goals and design tradeoffs. Lastly, the flow of data is depicted via the Logical Data Flow and Physical Data Flow diagrams.

(Williams, 2023)

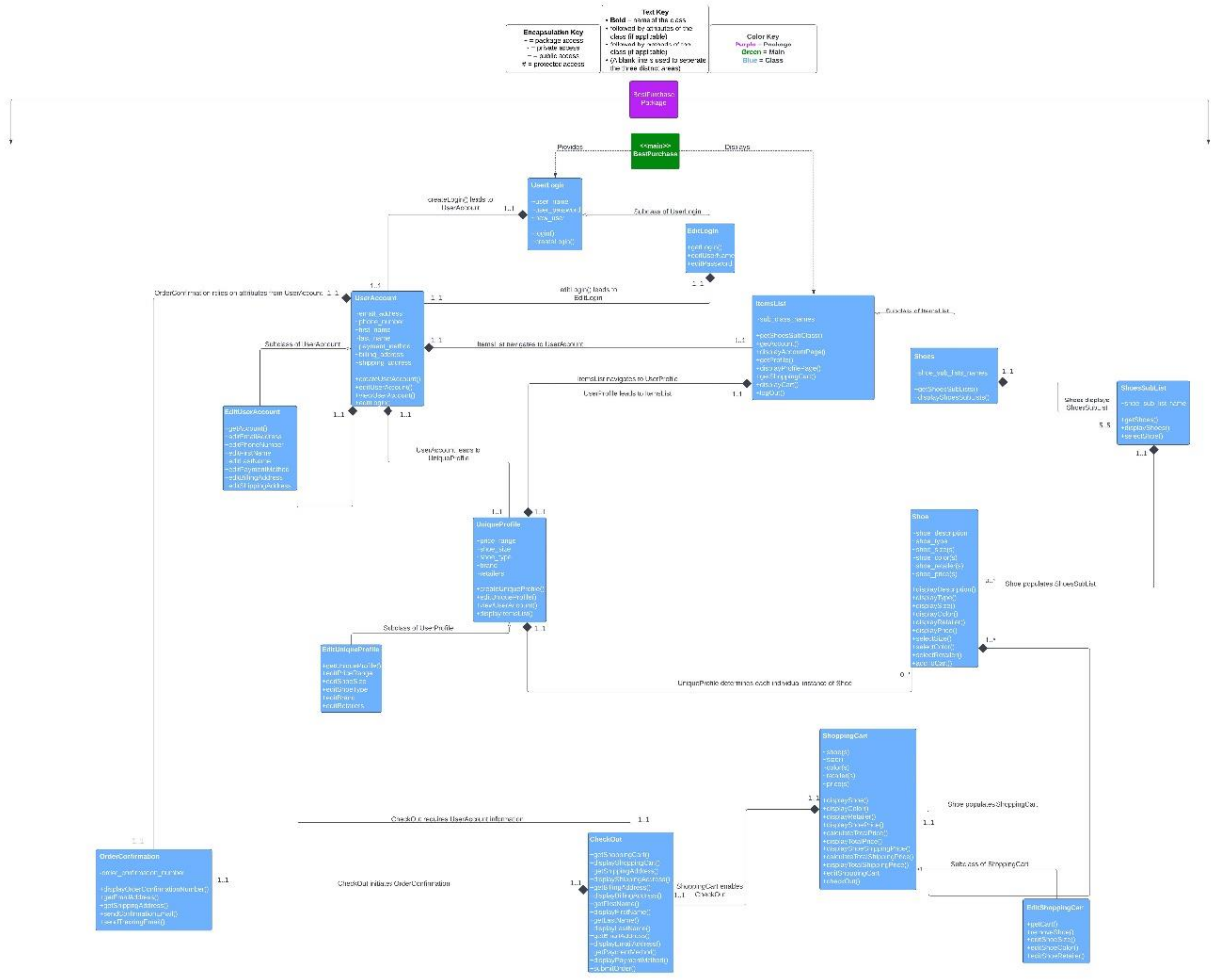
APPENDIX

Appendix 1:

The diagram itself depicts the classes in the system and their related attributes/methods. One key area to notice is the ShoesSubList class, which within the Shoes class contains five separate instances of the class, to accommodate the five different sub lists. Outside of this instance, the other classes relate to a single instance of this class. Another area of note is the instances of Shoes class that populate each ShoesSubList class and their relation to the UniqueProfile class. The instances are from zero to many, to accommodate a sub list that has no shoes that fit its criteria or a UniqueProfile that has zero instances of Shoes that match its criteria.

Lastly, there are keys at the top of the diagram to help better understand the diagram itself.

Appendix 2:



(“Module 4,” 2023; Williams, 2023)

REFERENCES

[1] Module 4: Object-Oriented Design and UML Modeling. (2023). In D. Williams (Ed.), *Review of Class Diagram Components* (pp. 16). Boston University Metropolitan College.

[2] Module 5: System Architectures. (2023). In D. Williams (Ed.), *Sufficiency of Designs* (pp. 6). Boston University Metropolitan College.

[3] Module 5: System Architectures. (2023). In D. Williams (Ed.), *The Goals of System Architectures and Designs* (pp. 5). Boston University Metropolitan College.

[4] Module 5: System Architectures. (2023). In D. Williams (Ed.), *The Other Design Goals* (pp. 7). Boston University Metropolitan College.

[5] Williams, D. (2023, January 22). *Supplementary Live Session Week 4* [PowerPoint slides]. Boston University Metropolitan College.

[6] Williams, D. (2023, February 11). *Supplementary Live Session Week 5* [PowerPoint slides]. Boston University Metropolitan College.