

KAWS Customs

By: Amber Nelson,
Kellen Brown,
Walker Bryan



Summary Of Application



What we do:

- KAWS Customs sells and designs custom shoes for our online marketplace
- Our team has designed a program to allow our users to access and interact with the KAWS Customs online marketplace



Our Implementation:

- The marketplace allows a customer to sign up, navigate the store, and select from many options to find and buy shoes
- Customers also have the option to design their own shoes once they enter the marketplace



Overall value:

- Our program provides a realistic set of features, a seamless UX, and advanced accessibility for our customers.

Classes

Shoe

- The Shoe class contains fields and methods used in customizing shoe designs and constructing new Shoe objects

Running Shoe

- The RunningShoe class is invoked to create RunningShoe objects that extend from the Shoe class, but with unique fields

Basketball Shoe

- The BasketballShoe class is invoked to create BasketballShoe objects that extend from the Shoe class, but with unique fields

Online User

- The OnlineUser class contains fields and methods used to edit user account details, such as username, password, order history, and account balance

Store

- The Store class contains the functionality for our store interaction end-to-end scenario, including the collections, variables, and methods

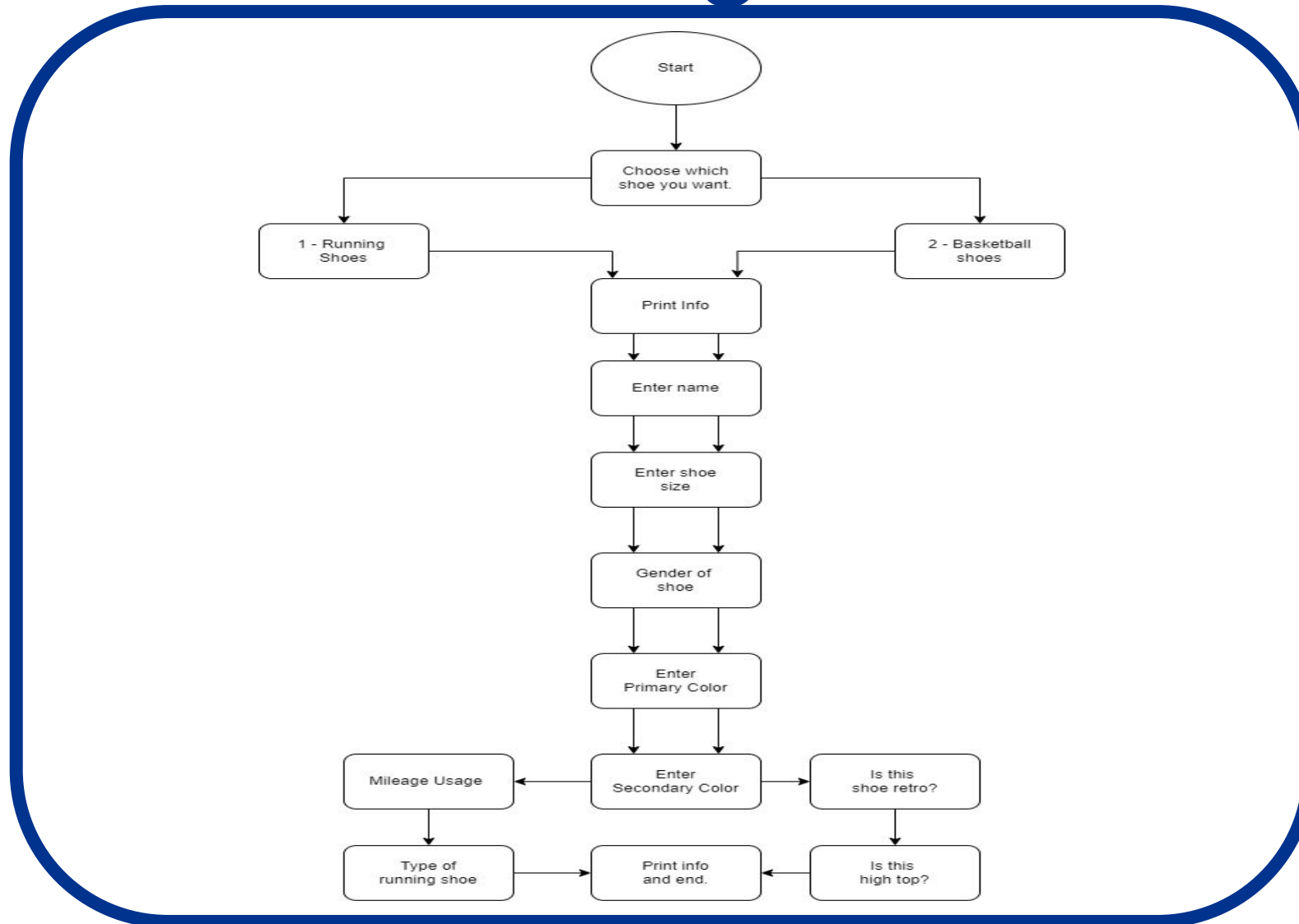
Design

- This class contains the design end-to-end scenario, and contains instructions and methods to allow a user to design a shoe

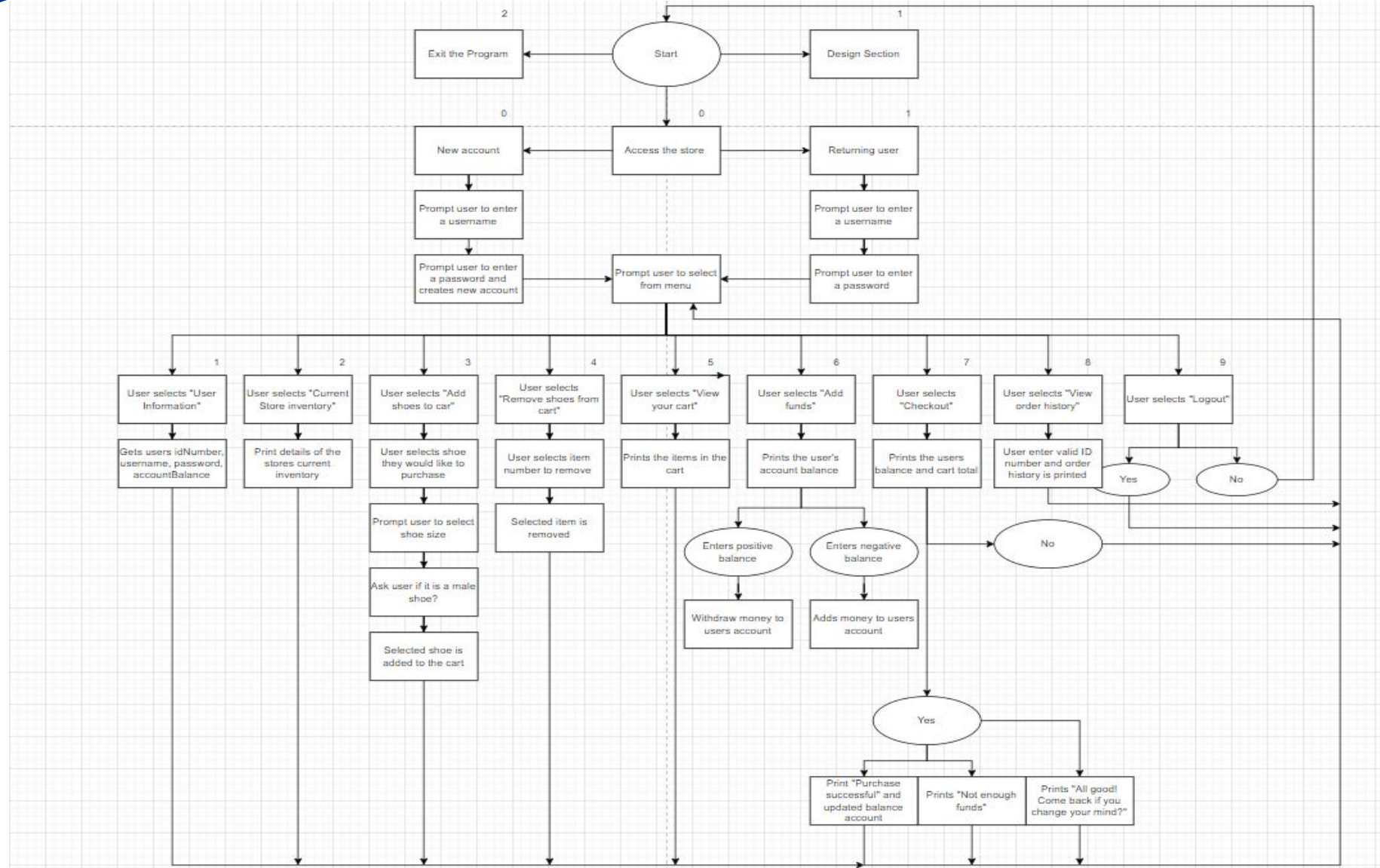
KAWS Customs

- Implementation class where our team showcases our methods and end-to-end scenarios

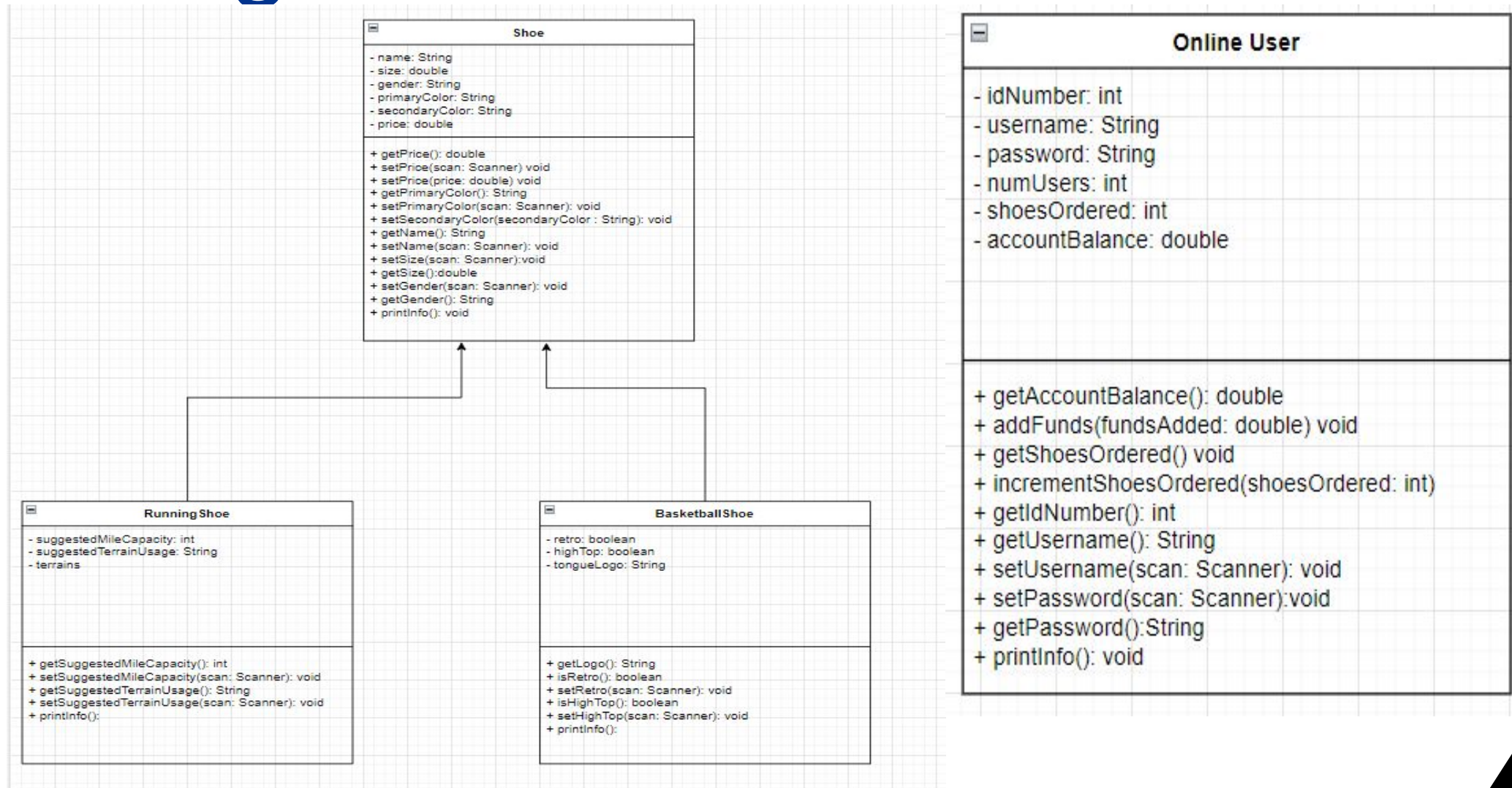
Flowchart 1: Design



Flowchart 2: Store



UML Diagram



Collections

ArrayList



- In our Store class we implemented a private static ArrayList named **userDatabase**
- This ArrayList is used to store the registered users of KAWS Customs

```
for (int i = 0; i < userDatabase.size(); i++) {
    if (userName.equals(userDatabase.get(i).getUsername())) {
        usernameFound = true;
        System.out.println("Enter your password:");
        String password = scan.next();
        password = password.trim();
        if (userDatabase.get(i).getPassword().equals(password)) {
            System.out.println("Successfully logged in.");
            currentUserInStore = userDatabase.get(i);
            correctPassword = true;
            break;
        }
    }
}
```

HashMap

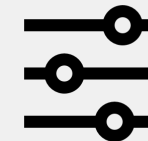


- In our Store class we implemented a private HashMap named **orderHistory**
- This HashMap is used to store the order history of registered users

```
public void viewOrderHistory(Scanner scan) {
    System.out.println("Please enter your ID number:");
    int idNumber = scan.nextInt();

    if (orderHistory.containsKey(idNumber) && idNumber == currentUserInStore.getIdNumber()) {
        System.out.println("Your account has spent $" + orderHistory.get(idNumber));
    }
    else if (idNumber != currentUserInStore.getIdNumber()) {
        System.out.println("Not your account ID nosy guy.");
    }
}
```

HashSet



- In our Shoe class we implemented a private HashSet called **colorList**
- This HashSet is used to hold a set of colors that users may customize their shoes with

```
colorList = new HashSet<String>();
Collections.addAll(colorList, "Maraschino", "Cayenne", "Maroon", "Grape",
    "Salmon", "Tangerine", "Banana", "Lime", "Pine", "Teal", "Turquoise", "Black", "White", "Silver",
    "Gold");
}
```

Array



- In our Store class we implemented a private static final Array called **storePresets[]**
- This Array is used to store our preset Shoe objects

```
private static final Shoe[] storePresets =
    {defaultShoe1, defaultShoe2, defaultShoe3, defaultShoe4};
```

Conclusion

- The KAWS Customs online store that our team has presented features a comprehensive criteria of methods, fields, and overall functionality that enable it to be implemented as comprehensive online marketplace to best serve our customers.

Questions?