CS322 Project Proposal

Names: Maddi Acton, James Petersen, Christina Melnic

Introduction

For our project, we plan on creating a customizable online shop similar to Etsy that would allow a user to purchase, create or personalize it if they choose to, purchase and choose shipping options, and also receive notifications about new products and alerts related to their orders. In more detail, our base unedited products will be created using the factory pattern. The decorator pattern will be used to customize and add to the base clothing items if a customer chooses to such as adding a custom print to a shirt, adding a design to a hat, etc. We will also use the strategy pattern to handle different shipping speeds and also to handle a variety of payment methods such as PayPal, Visa, etc. Finally, for our last design pattern we will use the observer pattern to notify users of new products and features and also to notify customers about their orders like order confirmations and shipping notifications.

Methodology

We will develop an online shop in Java that lets customers browse products, customize them, choose shipping options, make payments, and receive notifications. First, we will build the core system using the factory pattern. An abstract ProductFactory will declare a createProduct method. Concrete factories like ClothingFactory and AccessoryFactory can instantiate specific products like Hat or Mug, all of which implement a common Product interface.

Once the base products and factories exist, we can implement the decorator pattern to allow customization. An abstract ProductDecorator will wrap the core products. We will create implementations for decorators ColorDecorator and TextDecorator which will add color changes or custom text. Because each decorator wraps another Product, customers can stack multiple customizations. Next, the Strategy pattern will be implemented. We can define a

ShippingStrategy interface with implementations StandardShipping and FreeShipping. We will define a PaymentStrategy interface with PayPalStrategy and CreditCardStrategy. At runtime, the Order will hold references to the selected strategies and call their methods for cost calculation and processing. Finally, we will implement the Observer pattern for notifications. The observers will be registered such that they get notified when a product is ordered and once it is shipped. We can then work on creating a graphical user interface that calls necessary methods after the overall system is completed.

Implementation Timeline

We will be uploading our code on github. We will section off different strategies and have the bulk of our code finished by the weekend of April 20th. For the following week, we will integrate our code and debug it. Our presentation and final report will be finished by April 26th, so we have time to rehearse.