Software Design Specification Document

DecalPro

Team Members: Nick Daniel, Elijah Pearce, Anthony Corona, Jake Valdez

System Description

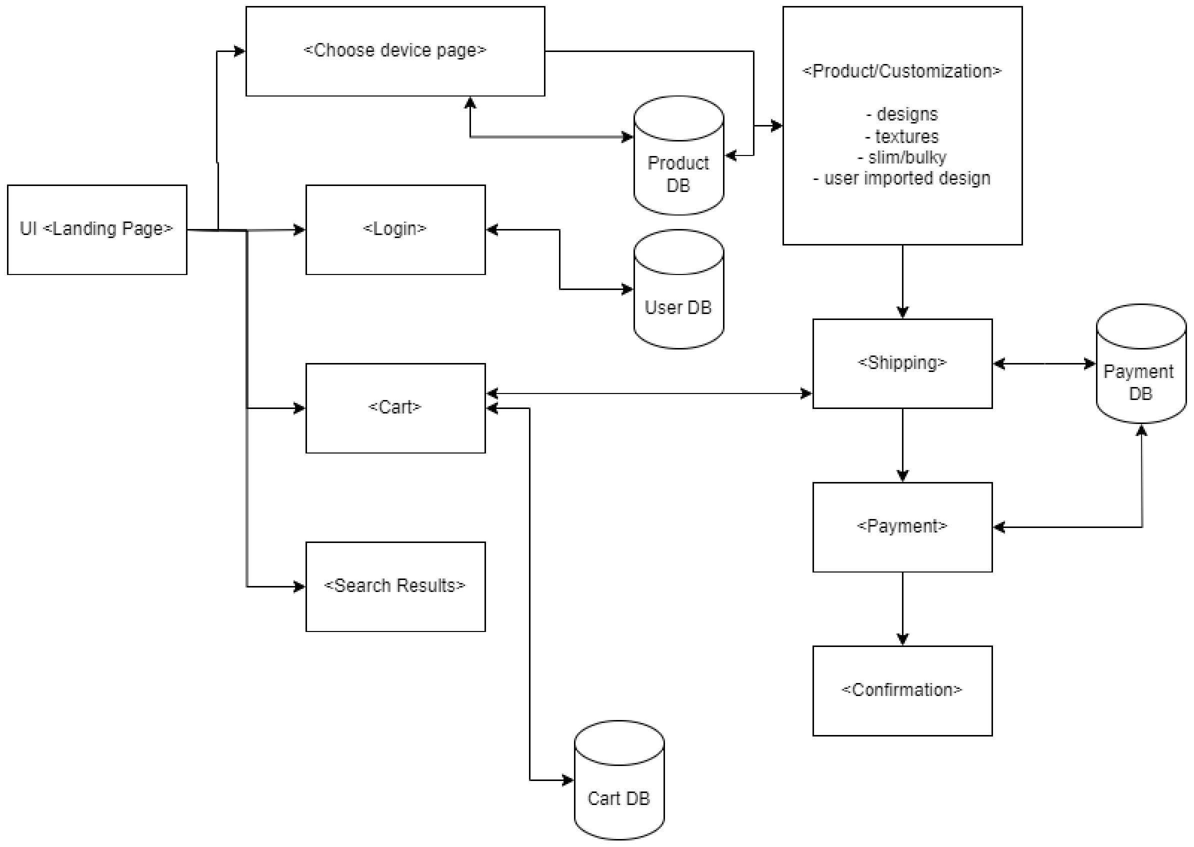
Brief Overview of System

The DecalPro website is an e-commerce platform dedicated to selling a variety of decal products for electronic devices. It provides a user-friendly interface for customers to browse, select, and purchase decals for laptops, smartphones, tablets, gaming consoles, and other electronic devices. The system also includes features such as user registration, order management, payment processing, and product reviews.

Software Architecture Overview

This section will contain a Software Architecture Diagram alongside a UML Class Diagram. The main elements of the website and how they interact are shown in the software architecture diagram. It gives a quick overview of the system's architecture. The essential classes, their attributes, and operations will be shown in the UML class diagram, providing a thorough understanding of the internal organization of the product.

Software Architecture Diagram



Landing Page

The homepage of the website. On this page, the user has links to login/register, to the user's own cart, the search bar, the gallery, and the choose device page.

Choose Device Page

The Choose Device Page link is available on the landing page. It is a page that has all of the devices that we provide cases for. When you click on a device on the page, it will link to a results page that contains links for all of the products available for said device.

Login Page

This is a page that the user is redirected to from the landing page. On the Login page, the user may enter their Username/email and Password or register an account for the website. The reason we would want a Login page is to help users who repeatedly use the website to make purchases faster by saving their personal information as an account in the UserDatabase.

Cart Page

This page is used to record and show the user what they plan on purchasing as well as the accumulated price of all the items they have purchased. The User may add or take away products from their cart as they want.

Search Results Page

The result page is the page that the user will be linked to after they use the search bar or the Choose Device Page. The page will contain all of the available products that fit the search criteria of the user's search input. If no product can be found that matches the search, “No result” will be displayed on the page.

Product Database

The database that stores all of the cases being sold, their names, prices, descriptions, and what device each case is compatible with.

User Database

This is the server that holds the information of each registered account for the website. Each account has a username, email, and password. This information helps assign a user to a shipping address and payment information. This server gets the information from the Login page.

Cart Database

This is the database that will be responsible for maintaining the activity of user’s carts. It will hold the items every user has inside their cart prior to checkout and will be used by cart page.

Products / Customization Page

This page holds a list of the products and allows the user to change the designs, textures, and import designs for the products.

Designs

This field is dedicated to showing the customer basic colors or default designs so that they can choose one for any case. This page can be accessed from the Product/Customization page.

Textures

This field is dedicated to showing the customer all the available textures and that they can choose one for any case. This page can be accessed from the Product/Customization page.

Slim/Bulky

This field is dedicated to showing the customer all the available slim and bulky cases and that they can choose one for any device. This page can be accessed from the Product/Customization page.

User-imported design

This field is dedicated to allowing the customer to import a JPEG picture so that they can put it on any device. This page can be accessed from the Product/Customization page.

Shipping Page

This page allows the user to enter the desired address that they want the product delivered to.

Payment Page

This is the page that allows the user to enter their payment information into the page in order to purchase the products.

Payment Database

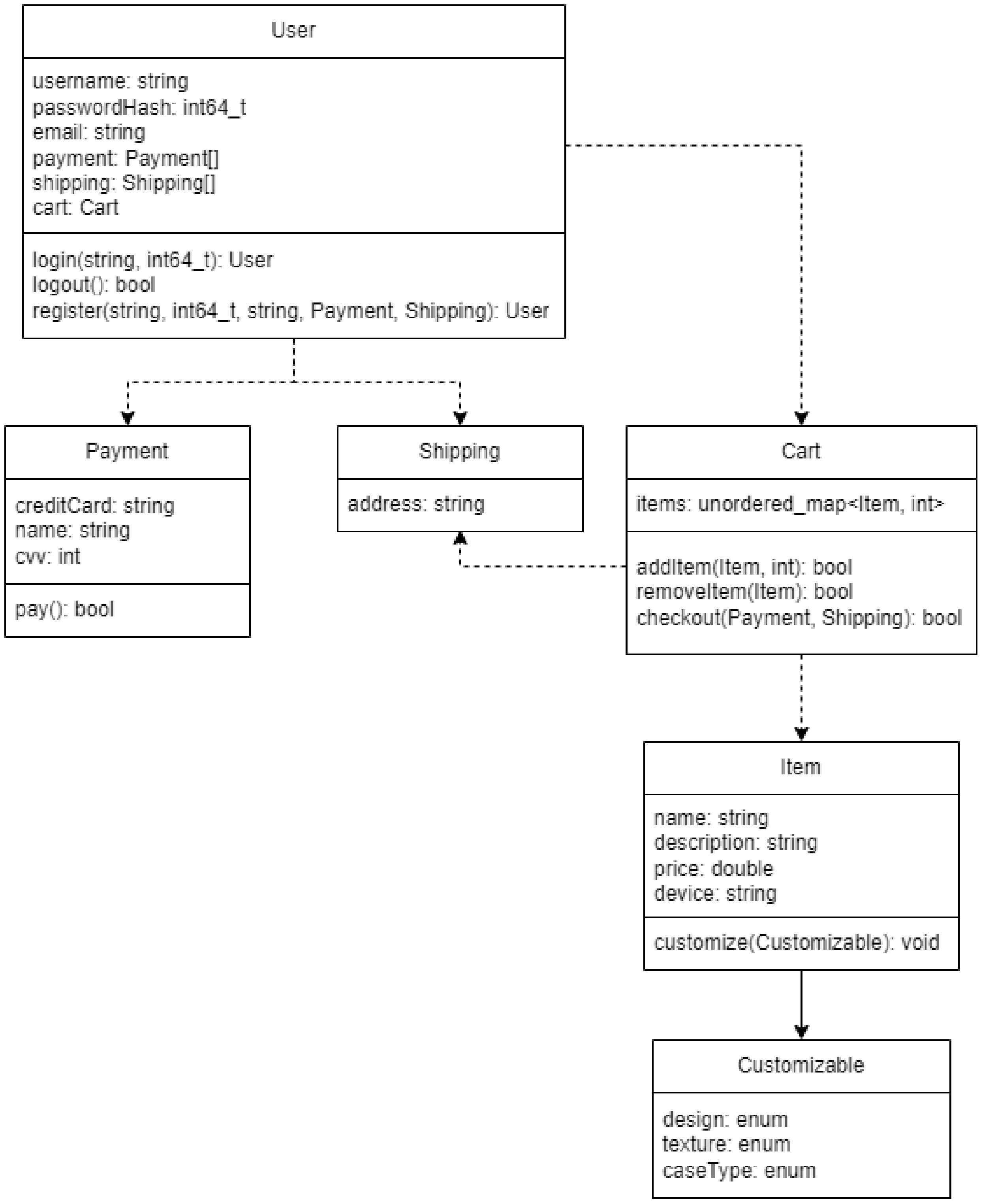
This is a database that holds the payment methods for registered users and temporarily holds it for guests.

Confirmation Page

This page shows the user all the products they wish to buy, the desired shipping address, the payment method, and the total price in order to make sure that everything is okay by the user.

UML Class Diagram

The UML Class Diagram below provides a visual representation of the major classes and their relationships within the backend.



User

The purpose of the user class is to initialize new users and allow them to log in to, log out of, and register accounts. This class is also utilized to manage the activity of a particular user. Their activity can span from modifying their cart, login credentials, payment options, or shipping options. This class comes equipped with the data necessary to store a user's login in a safe way. A user's username will be stored in a string whereas the password will be sent as a salted hashed from the user to allow for the safe storing of passwords. In the event a data leak occurs, the user's true password won't be exposed. Amongst the login credentials, there are instances of the following classes: Payment, Shipping, and Cart. Payment and Shipping instances are stored in a list to allow for multiple options from the user while the Cart instance exists as a single instance for every User. Function wise, this class can login, logout, and register a user.

Payment

This class, or more so struct, is a collection of data required to make a purchase. Our service only allows payments through direct credit card information. Therefore, this class contains the data required for such a purchase to be made. This is made into a class to allow the possibility of expanding payment options in the future. The user class will also store a list of payments, allowing the user to choose a preferred payment at checkout.

Shipping

Just like the payment class/struct, this collection of data will be the necessary information to ship products to users after a purchase has been made. The user class stores a list of this class to allow for multiple shipping addresses to be stored and chosen at checkout.

Cart

The cart class will be the primary class used to store the items a user has inside their cart. It can be used as a way to organize all their potential purchases before checkout. The only data contained in this class is a map of items a user has selected. A map is used for quick lookup times and the ability to increase/decrease quantity of a particular item. Amongst this, there are functions for adding and removing items from this block of data as well as a function to finally checkout which returns if the checkout was successful or not. As every user is equipped with the potential to purchase, this class will be a member of every user.

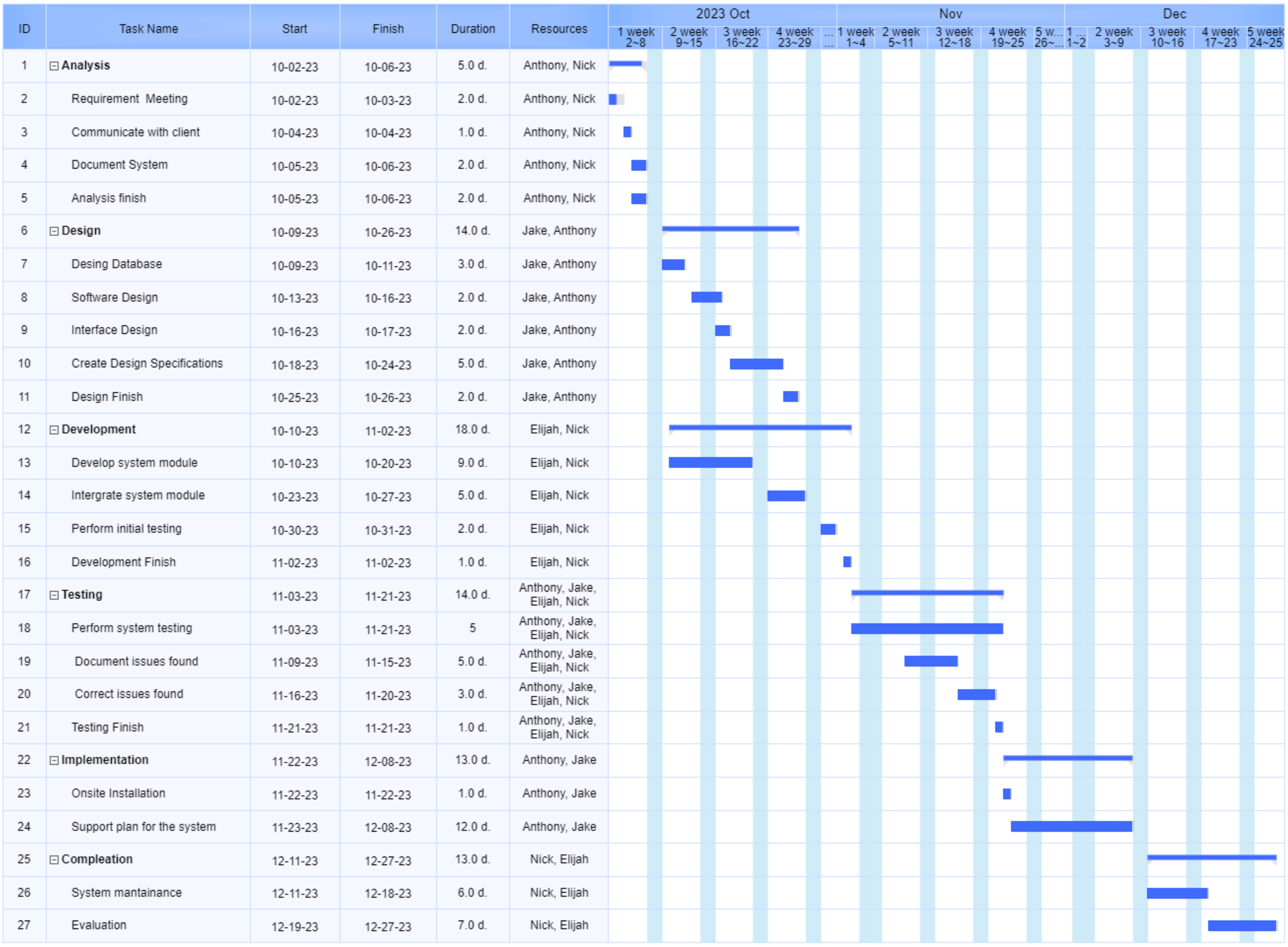
Item

This class, similar to a struct, contains the data that every item listed on the website will need. This data includes the name, description, price, and device type. On top of this, this class inherits the class Customizable as every Item we provide is considered customizable. This class also contains a customize function in which a Customizable class is passed. A function is used here to ensure that the requested customization is actually possible with the specific item calling it.

Customizable

This class allows for the further customization of specific items. As every item we sell allows for customization, this class will be inherited by the class Item. The possible types of customization include design, texture, and case type specific choices. These values are stored as enums as specific options for each. Inheritance is also used here to allow for the modularity object orientation provides; this modularity allows for the ease of changing/adding customizable features to items.

Development Plan and Timeline



ID	Task Name	Start	Finish	Duration	Resources
1	⊟ **Analysis**	10-02-23	10-06-23	5.0 d.	Anthony, Nick
2	Requirement Meeting	10-02-23	10-03-23	2.0 d.	Anthony, Nick
3	Communicate with client	10-04-23	10-04-23	1.0 d.	Anthony, Nick
4	Document System	10-05-23	10-06-23	2.0 d.	Anthony, Nick
5	Analysis finish	10-05-23	10-06-23	2.0 d.	Anthony, Nick
6	⊟ **Design**	10-09-23	10-26-23	14.0 d.	Jake, Anthony
7	Desing Database	10-09-23	10-11-23	3.0 d.	Jake, Anthony
8	Software Design	10-13-23	10-16-23	2.0 d.	Jake, Anthony
9	Interface Design	10-16-23	10-17-23	2.0 d.	Jake, Anthony
10	Create Design Specifications	10-18-23	10-24-23	5.0 d.	Jake, Anthony
11	Design Finish	10-25-23	10-26-23	2.0 d.	Jake, Anthony
12	⊟ **Development**	10-10-23	11-02-23	18.0 d.	Elijah, Nick
13	Develop system module	10-10-23	10-20-23	9.0 d.	Elijah, Nick
14	Intergrate system module	10-23-23	10-27-23	5.0 d.	Elijah, Nick
15	Perform initial testing	10-30-23	10-31-23	2.0 d.	Elijah, Nick
16	Development Finish	11-02-23	11-02-23	1.0 d.	Elijah, Nick
17	⊟ **Testing**	11-03-23	11-21-23	14.0 d.	Anthony, Jake, Elijah, Nick
18	Perform system testing	11-03-23	11-21-23	5	Anthony, Jake, Elijah, Nick
19	Document issues found	11-09-23	11-15-23	5.0 d.	Anthony, Jake, Elijah, Nick
20	Correct issues found	11-16-23	11-20-23	3.0 d.	Anthony, Jake, Elijah, Nick
21	Testing Finish	11-21-23	11-21-23	1.0 d.	Anthony, Jake, Elijah, Nick
22	⊟ **Implementation**	11-22-23	12-08-23	13.0 d.	Anthony, Jake
23	Onsite Installation	11-22-23	11-22-23	1.0 d.	Anthony, Jake
24	Support plan for the system	11-23-23	12-08-23	12.0 d.	Anthony, Jake
25	⊟ **Compleation**	12-11-23	12-27-23	13.0 d.	Nick, Elijah
26	System mantainance	12-11-23	12-18-23	6.0 d.	Nick, Elijah
27	Evaluation	12-19-23	12-27-23	7.0 d.	Nick, Elijah

Planned start date: September 18, 2023.
Planned completion date: December 27, 2023.
Total design period: 101 days.