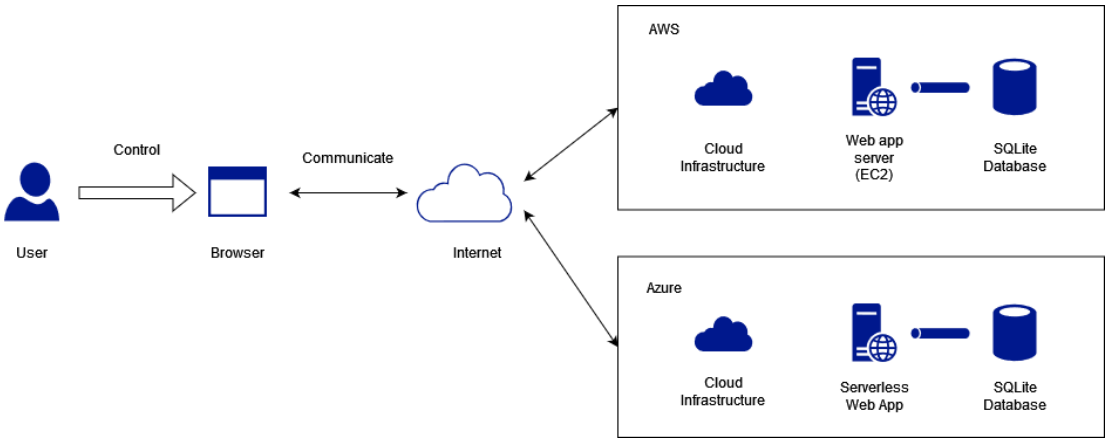
**CS673 Software Engineering**
**Team 4: Flea Market**
**Software Design Document**

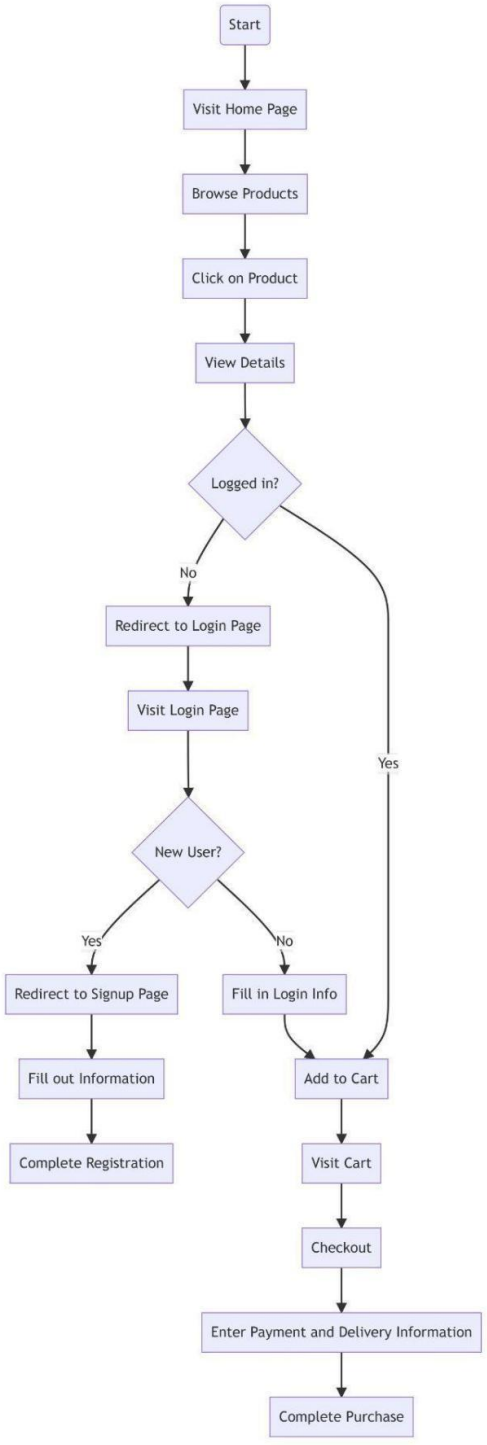
## 1. Introduction

This Software Design Document (SDD) delineates the architectural and design specifications for the development of a second-hand trading platform tailored for Boston University (BU) students. Envisioned as an online marketplace, the platform's primary objective is to foster a community-driven space where BU students can seamlessly engage in the buying and selling of second-hand items. While a diverse array of items can be traded. To ensure a smooth and trustworthy transactional experience, the system has been conceptualized to incorporate a plethora of features. These encompass item listings, shopping carts, secure payment gateways, and a user-centric interface that prioritizes intuitiveness and ease of navigation. This SDD will delve into the intricate details of the software's architecture, design considerations, user interface, and other pertinent components to provide a comprehensive overview of the envisaged system.

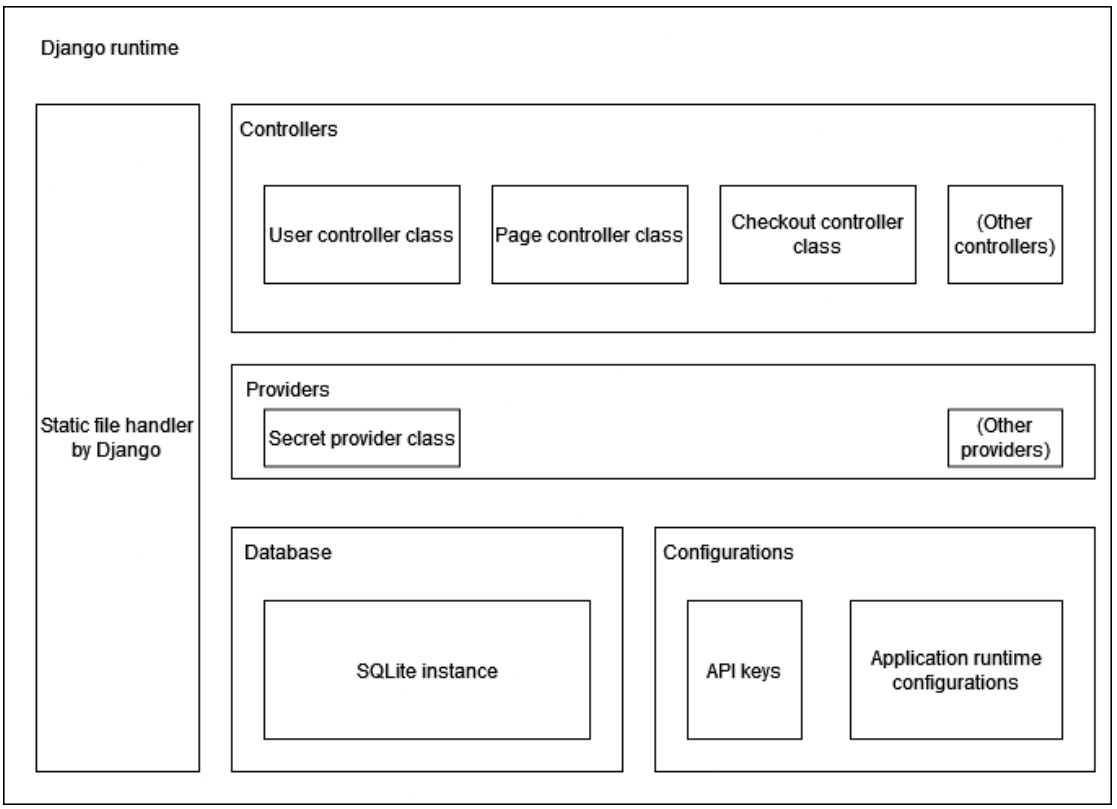
## 2. Software Architecture (Tianpei)




## 3.Business Logic(Flow chart)

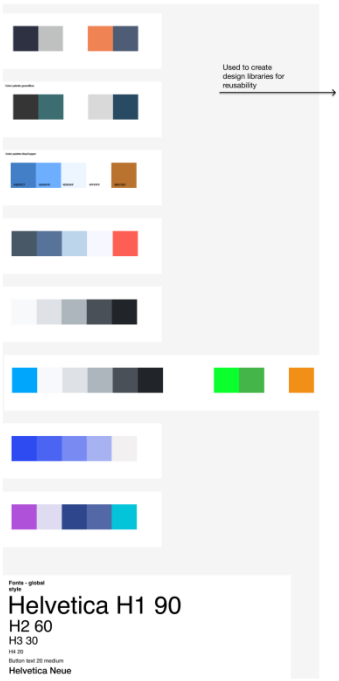



## 4.Class Diagram (Tianpei)

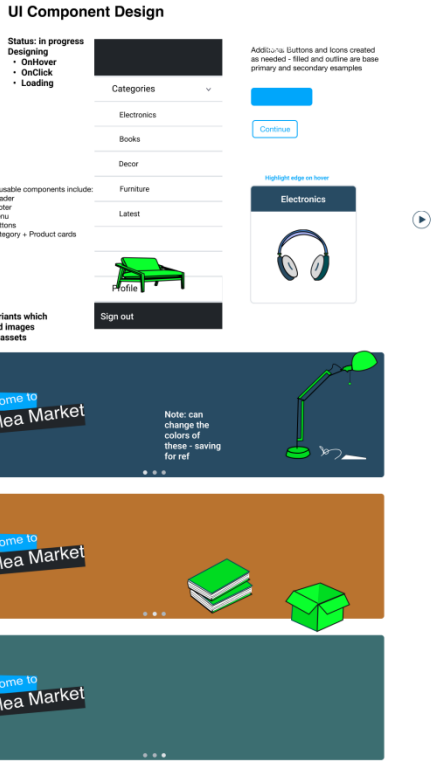



## 5. UI Design (Jennifer)

## Color themes + Design Inspiration

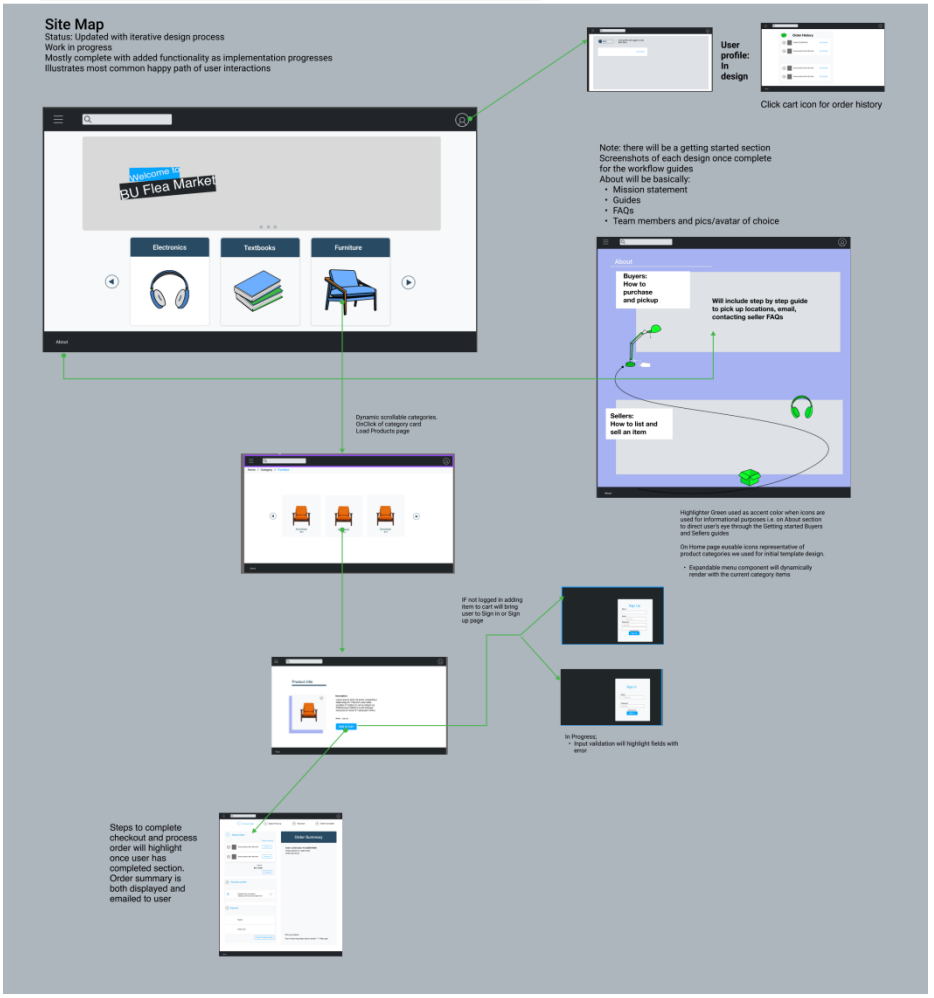



## UI Component Design

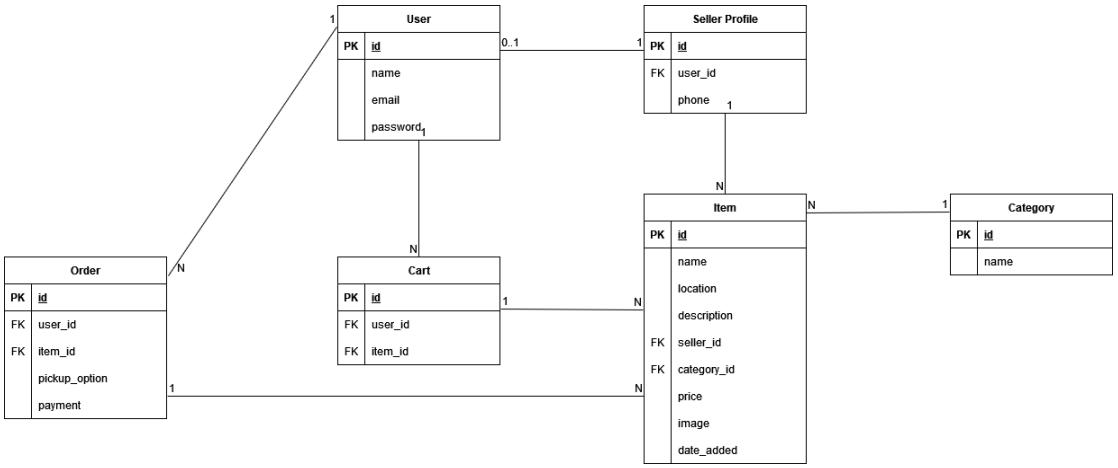



## Site Map

Status: Updated with iterative design process
Work in progress
Mostly complete with added functionality as implementation progresses
Illustrates most common happy path of user interactions



## 6. Database Design (Tianpei)




## 7. Design Pattern Overview

In the architectural foundation of the BU students' second-hand trading platform, we employ the revered MVC (Model-View-Controller) design pattern, a proven methodology for structuring interactive applications. The pattern facilitates a clear separation of concerns, ensuring the application remains modular, maintainable, and scalable.

**Model:** Powered by Django, this component manages data and core business logic, ensuring transactions are accurate and secure.

**View:** Using React, the View renders a responsive and intuitive user interface, tailored to showcase furniture and bedding prominently.

**Controller:** This bridges the Model and View, directing data flow and handling user interactions for a seamless experience.

The combination of React and Django, structured around the MVC pattern, ensures the platform is efficient, user-friendly, and adheres to modern software design best practices.

## 8. Acceptance Criteria

User Registration and Authentication:

- Users can register using valid BU email addresses.
- Password complexity requirements are enforced.
- Users can log in and log out securely.

Item Listings:

- Sellers can list items with a title, description, price, category (e.g., furniture, bedding), and at least one image.
- Listings can be edited or removed by the respective sellers.
- Users can search for items using keywords, categories, or price ranges.

Shopping Cart:

- Users can add items to their shopping cart and view all items in the cart.
- Items in the cart can be removed or quantities adjusted.
- The total price is correctly calculated and displayed.

Secure Payments:

- Integration with a trusted payment gateway (like Stripe or PayPal).
- Users can successfully make payments for items.
- Transaction details are recorded securely.

User Interface:

- The platform's UI is responsive across devices (desktop, tablet, mobile).
- Key features, like item listings and shopping carts, are easily accessible.
- UI loads efficiently with minimal lag.

Integration:

- Seamless integration between the React frontend and Django backend.
- No broken links or features.