BookShare UW

Vision

The Problem

Textbooks are expensive. The UW bookstore is abusive on refunds. Finding someone to buy your textbooks is difficult.

Our Product

We propose building a platform that allows CSE Students to share books in a fair and simple manner.

Mechanics Of Sharing

The idea is straight forward. Fair means each student that has used a book should pay the same amount.

This leads to the following payment strategy:

Student1: Acquires textbook. Pays for full price.
Student2: Acquires textbook. Reimburses Student1 50% of the original cost.
Student3: Acquires textbook. Reimburses [Student1 16.67%, Student2 16.67%] of the original cost.
Student4: Acquires textbook. Reimburses [Student1 8.34%, Student2 8.34%, Student3 8.34%] of the original cost.
…

At each step n, every student has effectively only paid 1/n of the books price.
More Sharing, More Saving!

Target Market

For the purpose of the class and bootstrapping the product we will focus on UW CSE students sharing textbooks.

Why Do This?

There is no good, simple solution at UW for sharing textbooks. This project is an opportunity to bring people together, to create a network that is fair and useful, and to learn a lot doing it.

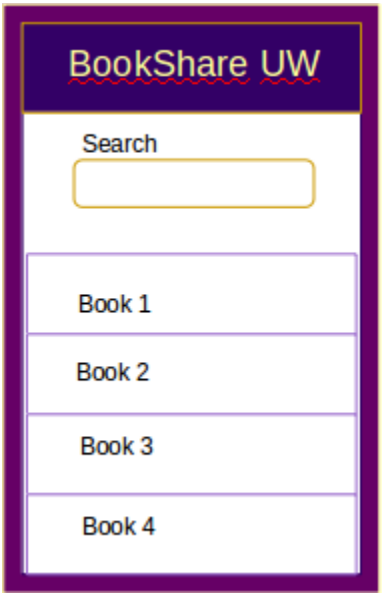
Competition

We face competition form Facebook book exchanges, and the UW bookstore. However, Facebook's interface is lacking, and there is no underlying structure to their payments, and the UW bookstore is expensive. We also face the possibility that students don't buy books anymore, but given that the UW bookstore still exists, this is unlikely.

Software Architecture

Overview
The finished product will follow the conventional organization for this type type of application. It will consist of a database that holds the state, a user interface, and an API that allows the state to be examined and updated. Our system will also interface with various APIs that provide peer to peer payments, book purchases and shipments, and possibly on demand delivery.

User Interface
Students should be able to use the product though a phone application, either IOS or Android. As a minimum viable product, the application should provide users a way to look up the available books, pay, and get reimbursed. We can then add a postmates-like delivery service to increase the intrinsic value of the application.



Database
We can use one of the typical DBMS, like PostgreSQL or MySQL.

API
To be specified.

Payment System
We will need to a robust peer to peer payment system. This is likely beyond the scope of the class so we will use the Stripe API.

Why is this Interesting?
This is an opportunity to gain experience in application development, e-commerce systems, and various APIs. At the end we should have a commercially viable product that can be used by our peers to share the burden of textbook purchases.

Suggested Tools
Android or/and IOS platforms for user interface. Stripe for payments. Amazon for book purchases, or we could satisfy orders by going to the UW bookstore ourselves. Maybe Postmates API for on demand delivery.

Challenges and Risks
The biggest challenge to a successful delivery of our product is learning the tools. A large part of the project is integrating different APIs into a single cohesive user experience. We will have to learn the different APIs, and we will need to learn the application development platforms.