Project Proposal

Duke Shares Lunch

Paul Dellinger, AJ Eckmann, Josh Romine

Before each semester, each Duke student living on campus must purchase a food plan through Duke Dining, allotting them a set number of food points (\$1 = 1 food point). While a student always has the option to purchase more food points, there is currently no way to return them – students who don't spend all their points must watch them expire at the end of the spring semester. Often, students with excess points will end up purchasing meals for their friends (or dining out at the Washington Duke Inn (no longer allowed)) but still have points left over. Entire vending machines have been bought out in the last week of the spring semester, because it's better than letting the money disappear.

On the flip side are the students who purchase food plans that are too small for their needs and run out of food points early, finding themselves purchasing all of their meals with real money. In addition, students who reside off-campus (such as many seniors) are without a food plan and could stand to utilize others' excess food points.

Our proposed idea is an iOS application that would connect students with excess food points with students who have none or are running low. Instead of purchasing meals for just immediate friends, a student using our app would be able to broadcast their intention to eat at a Duke location, connecting them with the students who wish to eat at that location. As an incentive, the person with extra food points would specify a dollar: food point ratio. The person they connect with will be able to buy food at a fraction of the normal cost, and the person with extra food points will get some return on the money they would have otherwise lost completely. Once they meet, the app would facilitate sharing Venmo information – which, along with an iPhone, is a service most Duke students already have. In addition, we may collect some personal information when users sign up for the app, so they could specify if they wanted to eat with people who are the same year, in their major, or who live in the same dorm.

We intend to build the application using a MySQL database set up on our Google Cloud VM, connected to the application through PHP/PDO. We intend to write the iOS application using Swift. The database would be connected through a webservice that the iOS app would have access to.

There are numerous apps that use roughly the same concepts we are trying to achieve with our app. A phone app like Uber is looking to pair a driver (seller) with a rider (buyer). The driver can decide when and where he wants to drive and once a rider requests to ride, the driver then gets the opportunity to accept or decline the ride, and then the transaction is calculated and made from there. Our app is looking to do something very similar by pairing someone who is looking to sell food points, with someone who is looking to buy while monitoring the transaction in app. Uber has the advantage of being able to handle the transaction themselves, while we will need to use Venmo outside of the app, but confirm that the transaction actually occurred in app. This is a little more complicated on our end, but should not be too big of an issue.

We spoke with Professor Yang about our project idea after class. He said that he thought our idea would be feasible, but wanted us to think a little more about how we would handle the monetary transaction on Venmo. This is definitely one challenge that we will face when building our iOS, because as of now Venmo's API is no longer supported by new apps, so we would have to facilitate this transaction, without it being held on the application itself. Our solution to this is to hold venmo usernames in the database and giving this username to a buyer once the seller and buyer agree upon a meal and amount. After this, we can link the buyer to the Venmo of the seller so that they can pay the agreed upon amount. Finally, the seller can confirm on the app that they have received the payment and the transaction can be completed.

Tables:

User(<u>uid</u>, email, name, venmo, major, dorm)

ActiveSeller(<u>saleid</u>, uid, OrderTime, status, percent, location)

SellPreferences(uid, location, percent)

Purchase(<u>pid</u>, saleid, bid, price, approve, paid, p_description)

Meals(mid, price, location, description)

E/R Diagram: All arrows are meant to be rounded

