User Experience Optimization for Mobile Commerce Application:

Design Document

Team Members:
Arthur Pang
Joshua Conner
Nicholas Pallares

Revision 1.5 - 2/23/12

Table of Contents

1. Introduction

Put simply, there's still no fast, easy way to transfer money digitally from a mobile device.

Hermes Commerce Inc. (HCI) was founded, by our client Joshua Cross, to develop a mobile payment system that facilitates peer-to-peer (P2P) payments and consumer-merchant transactions in a quick, easy, and reliable manner. This project was started due to our current credit card system having flaws in dealing with fraud that hurts both merchants and consumers. By using HCI software, merchants are able to drive sales by hyper-locally targeting customers with advertisements and coupons. Customers also benefit from using HCI software because it gives them a convenient way to support local businesses and save money.

Our goal is to develop a graphical user interface (GUI) for the Hermes Commerce Application, on both iPhone and Android. Our GUI will have simple layouts and easy to use controls while allowing full access to all services provide by Hermes. Our implementation progress will follow 3 stages of rapid prototyping for iPhone, 2 stages of usability testing between prototyping, and finally 1 stage of platform migration from iPhone to Android.

Our project is required to have a home menu where users can select between different service options. Service options that are required are sending money to peers, managing pending transactions, and viewing transaction history. Optional service options such as Local Ads and QR codes are likely to be implemented as well but not required due to Hermes needing to make a decision on how those functions will work.

2. Architecture Overview

The HCI payment platform can be broken down into three components:

1. Merchant Application
2. HCI Server
3. Customer Application

The Merchant Application is responsible for posting transactions to the server and generating QR codes containing transaction data. The server performs CRUD operations for users and transactions, and also authenticates users. The Customer Application captures images with a camera, processes the images with a QR code reader, checks the QR code for a HCI signature, creates transactions, and fetches transaction data from the server.

2.1 Payment Process

The payment process can be summarized in 10 steps from actions by the Merchant to services by the HCI Server to tasks by Customer and back to the HCI Server for final checks.

Merchant

1. Create new Transaction
2. Display QR Code
3. POST Transaction to HCI Server

HCI Server

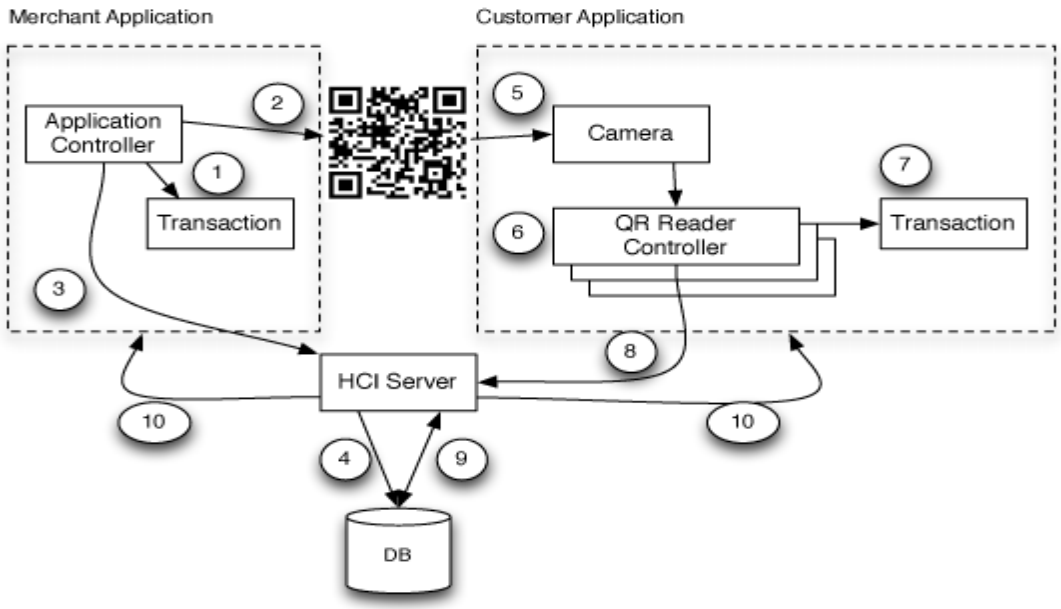
4. Create new Transaction

Customer

5. Capture Image
6. Process QR Code, check for HCI signature
7. Create new Transaction
8. PUT Transaction to HCI Server

HCI Server

9. Edit Transaction
10. Send response (success/error) to Merchant and Client



3. Component Descriptions

Each screen or service mentioned has a view controller associated with it and is indicated by the number next to it in parentheses. This key references the index of the modules in section 3.1

The Customer Application begins with the login screen (1) where the user enters their HCI username and password. Using these parameters, the app connects to the HCI server to authenticate the user, and redirects them to the app's home menu (2).

The home menu has one small window at the top displaying basic info about the user. The home menu also has a menu consisting of buttons showing the user access to five different functions. These functions are: send money (3), view pending transactions (4), view transaction history (5), and view local coupons (6), scan a QR code (7). Each service will redirect the user to a different view, associated with the task, where their will be guided through completing their desired action in an orderly and natural process flow.

3.1 Component Functionality

1. LoginViewController
   - A picture of the company logo is showcased here
   - User enters their username and password into text fields
   - Authenticates user's login parameters with HCI server and redirects them if successful or stays here if denied

2. RootViewController
   - Contains one small window located at the top of the screen displaying basic

info about the user such as their full name and account balance
- Centered beneath the window is a 2x2 table menu with buttons for the send money, pending transactions, history, and local ads services
- Beneath the menu and aligned to the right is a QR scan menu service

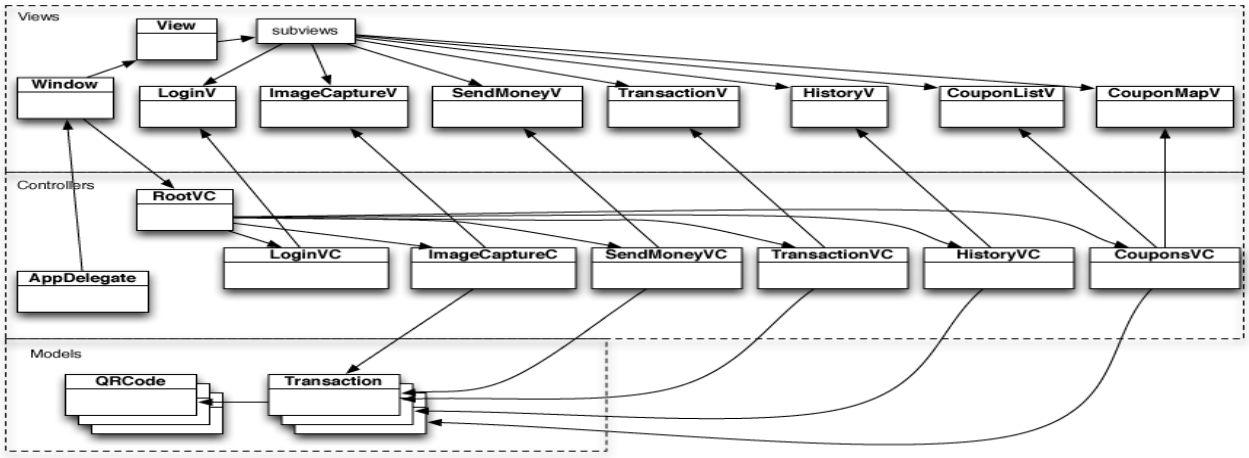
3. SendMoneyViewController
   - Displays contacts from address book with HCI accounts
   - Displays modal dialog for a new transaction
   - Create a new transaction and sends a POST request to the server

4. TransactionViewController
   - Fetches and displays pending transactions from server
   - Each transaction cell will have info about the sender's name and the balance needed to be paid to them

5. HistoryViewController
   - Fetches and displays paid transactions from server.
   - Each transaction cell will have info about the recipient's name and the amount paid to them

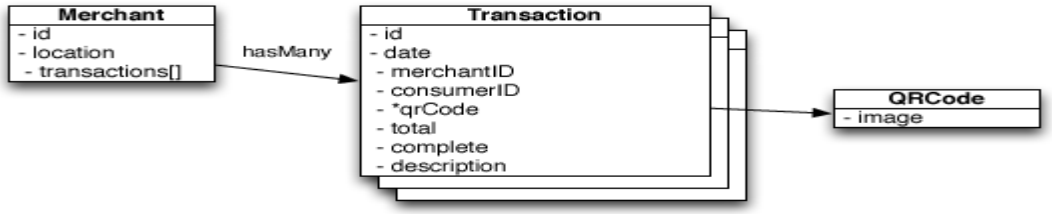
6. CouponViewController
   - Fetches and displays local coupons from server

7. ImageCaptureViewController
   - Captures images from camera
   - Processes QR code
   - Creates a new transaction and sends a PUT request to the server.



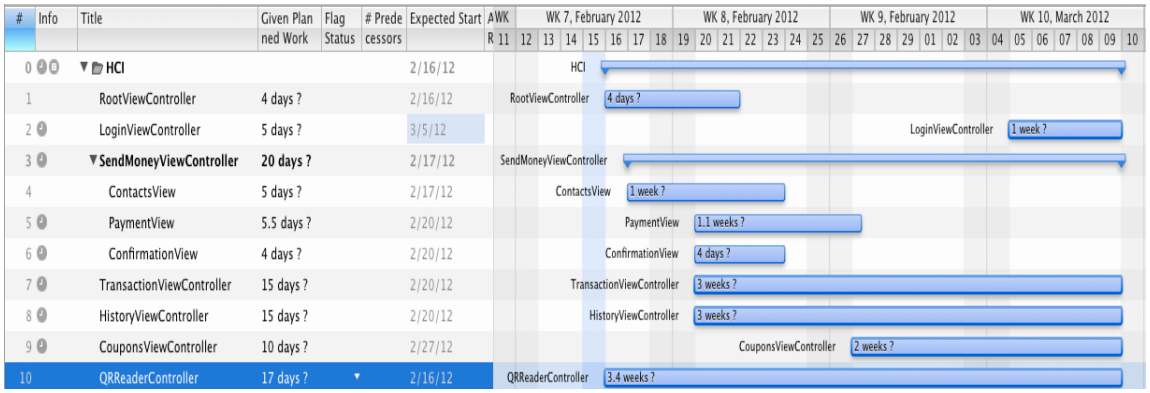
3.2 Model Descriptions

Merchants and Consumers have many transactions, and each transaction contains the following fields:



4. Implementation Timeline

The figure below illustrates our team's implementation timeline for the first iPhone application iteration.



4.1 Current Status

As of February 15th, 2012, the Customer Application has a skeletal class structure, including classes for reading and processing QR codes. Server-side functionality, user interface components, and models still need to be implemented.

RootViewController - Skeleton class setup
- UserInfoView - Not implemented

SendMoneyViewController - Skeleton class setup
- Contacts View - Not implemented
- Payment View - Not implemented
- Confirmation View - Not implemented

TransactionViewController - Skeleton class setup

HistoryViewController - Skeleton class setup

CouponsViewController - Skeleton class setup

QRReaderController - Skeleton class setup