Team 7

Project Phase 3 Requirements & UI Design

Janusz Nosek jnosek@hawk.iit.edu +1 (312) 351-9912

Phase Manager

Design

Documentation

Version Manager

Configuration Manager

Integration

Swathi Shenoy sshenoy6@hawk.iit.edu

Tests/Vaidation

Ernesto Garcia Grabwosky egarci10@hawk.iit.edu

Repository Information:

git clone git@cs442.ddns.net:~/team7\_proj\_cs442.git

(Please send configuration manager your public key)

Ticket and code tracker (redmine)

<http://cs442.ddns.net:3000>

(Credentials were distributed in phase 2)

1. Requirements (15 points) 1

1.1. Requirements and features list 1

1.2. UI Flow 4

1.3. Design Diagrams 5

1.4. APIs and class list 7

3. Elevator Pitch - Predemo (15 points) 13

Done on 11/14/2015 13

4. Use cases from end user point of view (+5 points) 13

4.1. UML Use Case Diagram 14

# Requirements (15 points)

## Requirements and features list

The project requirements and features have been reduced since phase 1 due to the loss of teammates, time, and programming ability of teammates.

Below are the requirements and feature stories.

A user should be able to list an advertisement of an item for sale, exchange or free

A user should be able to take a picture of their item to be sold, exchanged, or given away

A user should be able to view pictures taken using the platform

A user should be able to log into the system with their iit account

A user should only be able to log into the system if they have a picture

A user should only be able to log into the system if they have a non-default picture

A user should be able to buy/solicit the purchase of the item from the community

An item listing should contain a picture, title, description and price

A user should be able to take a picture of an item with their phone or tabled

A user should be able to use a picture from their gallery to list an item.

A snapshot of our redmine feature list:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 17 | Feature | New | Normal | Add connection system for remote database | Janusz Nosek |
| 16 | Feature | New | Normal | Investigate and create fragment flow from clicking on item list to description | Janusz Nosek |
| 12 | Feature | New | Normal | Create a list entry fragment | Janusz Nosek |
| 38 | Feature | In Progress | Normal | Create watchlist | Janusz Nosek |
| 37 | Feature | In Progress | Normal | Design remote database | Ernesto Garcia |
| 21 | Feature | In Progress | Normal | Integration of login system and account creation | Janusz Nosek |
| 13 | Feature | In Progress | Normal | Create the search fragment | Janusz Nosek |
| 10 | Feature | In Progress | Normal | Create item picture lightbox fragment | Janusz Nosek |
| 8 | Feature | In Progress | Normal | Create picture removal fragment | Ernesto Garcia |
| 1 | Feature | In Progress | Normal | Find dynamic database solution with SSL | Janusz Nosek |
| 43 | Feature | Feedback | Low | Create UPC datasource lookup | Janusz Nosek |
| 42 | Feature | Feedback | Low | Create ISBN lookup data source | Janusz Nosek |
| 39 | Feature | Feedback | Normal | Implement barcode IO encoding and decoding system. | Janusz Nosek |
| 119 | Feature | Closed | Normal | implement user picture md5 comparison | Janusz Nosek |
| 118 | Feature | Closed | Normal | implement default user | Janusz Nosek |
| 117 | Feature | Closed | Normal | implement backend description retrieval | Janusz Nosek |
| 116 | Feature | Closed | Normal | implement item listing to item description listing | Janusz Nosek |
| 115 | Feature | Closed | Normal | implement cursor adapter in item adapter | Ernesto Garcia |
| 114 | Feature | Closed | Normal | upgrade email to 254 | Janusz Nosek |
| 113 | Feature | Closed | Normal | change date to long for better sorting | Janusz Nosek |
| 112 | Feature | Closed | Normal | add db subconfiguration mechanism | Janusz Nosek |
| 111 | Feature | Closed | Normal | add price to backend functions | Janusz Nosek |
| 110 | Feature | Closed | Normal | add price to database | Janusz Nosek |
| 109 | Feature | Closed | Normal | add listener to item entry fragment for listing component | Janusz Nosek |
| 108 | Feature | Closed | Normal | resolve the gallery activity | Janusz Nosek |
| 107 | Feature | Closed | Normal | resolve the camera activiy | Janusz Nosek |
| 106 | Feature | Closed | Normal | add date picker to item entry fragment | Swathi Shenoy |
| 105 | Feature | Closed | Normal | Add image to item entry fragment | Janusz Nosek |
| 104 | Feature | Closed | Normal | create item listing callback system | Janusz Nosek |
| 103 | Feature | Closed | Normal | create secondary scaling function | Janusz Nosek |
| 102 | Feature | Closed | Normal | create bicubic scale fuction for efficient bitmap decoding | Janusz Nosek |
| 101 | Feature | Closed | Normal | Create bitmap thumbnail generator | Janusz Nosek |
| 100 | Feature | Closed | Normal | create bitmap decoding system | Janusz Nosek |
| 99 | Feature | Closed | Normal | create bitmap encoding system | Janusz Nosek |
| 98 | Feature | Closed | Normal | create backend to view users | Janusz Nosek |
| 97 | Feature | Closed | Normal | create backend to view items | Janusz Nosek |
| 96 | Feature | Closed | Normal | create backend to get invidual user | Janusz Nosek |
| 95 | Feature | Closed | Normal | Create backend test harness | Janusz Nosek |
| 94 | Feature | Closed | Normal | Create backend to insert user | Janusz Nosek |
| 93 | Feature | Closed | Normal | Create backend to list most recent item | Ernesto Garcia |
| 57 | Feature | Closed | Normal | Integrate the Sell Item Init and Content Description Display | Swathi Shenoy |
| 54 | Feature | Closed | Normal | Add a bypass user | Ernesto Garcia |
| 48 | Feature | Closed | Normal | Creating Sell Fragment Layout | Swathi Shenoy |
| 46 | Feature | Closed | Normal | Test item switching | Janusz Nosek |
| 45 | Feature | Closed | Normal | Test horizontal/vertical orientation | Janusz Nosek |
| 44 | Feature | Closed | Normal | Test basic google login | Janusz Nosek |
| 41 | Feature | Closed | Normal | Create database adapter | Swathi Shenoy |
| 40 | Feature | Closed | Normal | Implement item entry workflow | Ernesto Garcia |
| 36 | Feature | Closed | Normal | Create base login activity using google login | Janusz Nosek |
| 35 | Feature | Closed | Normal | Transfer IIT fonts | Janusz Nosek |
| 34 | Feature | Closed | Normal | Transfer iit color schemes | Janusz Nosek |
| 33 | Feature | Closed | Normal | Create clear/x icon | Janusz Nosek |
| 32 | Feature | Closed | Normal | Create refresh icon | Janusz Nosek |
| 31 | Feature | Closed | Normal | Create No-Picture placeholder icon | Janusz Nosek |
| 30 | Feature | Closed | Normal | Create spyglass icon | Janusz Nosek |
| 29 | Feature | Closed | Normal | Create arrow icon | Janusz Nosek |
| 28 | Feature | Closed | Normal | Create login icon | Janusz Nosek |
| 27 | Feature | Closed | Normal | Create icon | Janusz Nosek |
| 22 | Feature | Closed | In Progress | Retrieve and show top 100 items. | Ernesto Garcia |
| 20 | Feature | Closed | Normal | Set up Mysql Database | Janusz Nosek |
| 19 | Feature | Closed | Normal | Implement student rank major/rank retrieval system | Janusz Nosek |
| 18 | Feature | Closed | Normal | Add local database system | Ernesto Garcia |
| 15 | Feature | Closed | Normal | Investigate clicking system for recyclerView | Janusz Nosek |
| 11 | Feature | Closed | Normal | Create description removal fragment | Swathi Shenoy |
| 9 | Feature | Closed | Normal | Create bottom button fragment for item description | Janusz Nosek |
| 7 | Feature | Closed | Normal | Create contact seller fragment | Ernesto Garcia |
| 6 | Feature | Closed | Normal | Create a dialog box for picture selection when creating item | Ernesto Garcia |
| 5 | Feature | Closed | Normal | Create item entry fragment | Janusz Nosek |
| 4 | Feature | Closed | Normal | Create category list item fragment | Swathi Shenoy |
| 3 | Feature | Closed | Normal | Create list of categories | Swathi Shenoy |
| 2 | Feature | Closed | Normal | Create item description fragment | Janusz Nosek |

## UI Flow

## Design Diagrams

## APIs and class list

com.android.support:appcompat-v7:23.1.0

com.android.support:support-v4:23.1.0

com.android.support:design:23.1.0

com.android.support:recyclerview-v7:23.1.0

com.google.android.gms:play-services-identity:8.1.0

com.google.android.gms:play-services-plus:8.1.0

org.jsoup:jsoup:1.8.3

com.google.code.gson:gson:2.3

com.loopj.android:android-async-http:1.4.9

com.google.zxing:core:3.2.1

com.google.zxing:android-core:3.2.1

com.google.zxing:android-integration:3.2.1

org.jfree:jfreesvg:3.0

net.glxn.qrgen:android:2.0

Of special note, google play services identity is used for login as well as the plus api.

ZXing is used of the barcoding if they are part of the final product.

qrGen is used in conjunction with ZXing.

jfreeSVG is used for vector graphic manipulation in conjunction with qrGen

Gson is used for backend processing

Android support, design, and recyclerview are new topics and not covered in class. They allow for real time and dynamic list update mechanisms.

.

└── edu

└── iit

└── cs442

└── team7

└── iitbazaar

├── BazaarActivity.java

├── CircleImageView.java

├── FontManager.java

├── FuturaView.java

├── IITBazaar.java

├── Item.java

├── ItemAdapter.java

├── ItemNumberListener.java

├── ItemRetrievalListener.java

├── LoginActivity.java

├── MenuKeys.java

├── MoneyValueFilter.java

├── PhotoListener.java

├── UserRetrievalListener.java

├── common

│   ├── DBKeys.java

│   ├── DepartmentKeys.java

│   ├── FontConstants.java

│   ├── LevelKeys.java

│   ├── MysqlDataSizes.java

│   └── RankResultCode.java

├── database

│   ├── CustomDBAdapter.java

│   └── DBController.java

├── fragments

│   ├── BuyFragment.java

│   ├── BuyMenuFragment.java

│   ├── BuySellButtonFragment.java

│   ├── CategoryAdapter.java

│   ├── CategoryFragment.java

│   ├── ContactSellerContactFragment.java

│   ├── ContactSellerFeedbackFragment.java

│   ├── ContactSellerFragment.java

│   ├── ContactSellerItemsFragment.java

│   ├── ContactSellerNameFragment.java

│   ├── ContactSellerPictureFragment.java

│   ├── DescriptionRemovalFragment.java

│   ├── ItemDescriptionButtonFragment.java

│   ├── ItemDescriptionContentFragment.java

│   ├── ItemDescriptionFragment.java

│   ├── ItemEntryFragment.java

│   ├── PictureEntryDialogBoxFragment.java

│   ├── RecyclerViewCategoryListFragment.java

│   ├── RecyclerViewItemListFragment.java

│   ├── RemovePictureButtonFragment.java

│   ├── RemovePictureFragment.java

│   ├── SellFragment.java

│   ├── SellItemInitEntryFragment.java

│   ├── SellMenuFragment.java

│   ├── SellNavKeys.java

│   ├── SellServiceInitEntryFragment.java

│   └── TestFragment.java

├── rankrequester

│   ├── Harness.java

│   ├── HarnessHelper.java

│   ├── PhoneBookKeys.java

│   └── RankRequesterService.java

└── user

└── User.java

10 directories, 54 files

├── config.php

├── db\_connect.php

├── db\_functions.php

├── get\_categories.php

├── get\_item.php

├── get\_item\_description.php

├── get\_items\_after\_date.php

├── get\_user.php

├── insert\_item.php

├── insert\_user.php

├── test\_get\_item.php

├── test\_get\_user.php

├── view\_categories.php

├── view\_items.php

├── view\_users.php

└── view\_watchlist.php

0 directories, 16 files

1. **Test cases and test report (15 points)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tracker | Status | Subject | Assignee | Description |
| Test | New | Sell screen activities: In the selling item details screen, select ending date previous to current date and press list item. | Swathi Shenoy | It throws a error pop up prompting the user to enter a date later than current date |
| Test | Closed | Sell screen activities: Enter all the details correctly in the sellign item details screen and click on List Item | Swathi Shenoy | It lists the item entered with all the details and gives option to Follow, Buy or Share it. |
| Test | Closed | Sell screen activities In the selling item details screen, leave the description blank and press list item. | Swathi Shenoy | Expected: It throws a error pop up prompting the user to enter the description |
| Test | Closed | Sell screen activities In the selling item details screen, leave the price blank and press list item | Swathi Shenoy | Expected: It throws a error pop up prompting the user to enter the price |
| Test | Closed | Sell screen activities In the selling item details screen, leave the image blank and press list item. | Swathi Shenoy | Expected: It throws a error pop up prompting the user to upload an image |
| Test | Closed | Sell screen activities Enter the title of item to be sold and click on Enter. | Swathi Shenoy | Expected: It navigates to next screen where further details about the item to be sold is entered. |
| Test | Closed | Buy screen activities Check the overall layout of the page . | Swathi Shenoy | The background color and theme must be consistent with the rest of the app.  The radio tabs on the top of the page must be aligned correctly. |
| Test | Closed | Buy screen activities | Swathi Shenoy | Check the listing dataCheck the alignment and value of ending date in the listing.It must be later than current date |
| Test | Closed | Buy screen activities | Swathi Shenoy |  |
| Test | Closed | Buy screen activities | Swathi Shenoy | Check the listing data  Check the alignment and text of price in the listing |
| Test | Closed | Buy screen activities: | Swathi Shenoy | Check the listing data  Check the alignment and text of item name in the listing |
| Test | Closed | Buy screen activities | Swathi Shenoy | Check the listing data  Check the image displayed on the extreme left side of the item |
| Test | Closed | Buy screen activities User logs in and it is successful |  | Expected: It lands in the main page and displays the list of item which can be bought |
| Test | Closed | Side menu and navigation Go to Categories screen and click on Buy tab on the top | Swathi Shenoy | It navigates to Buy screen |
| Test | Closed | Side menu and navigation - Go to Categories screen and click on Sell tab on the top | Swathi Shenoy | It navigates to Sell screen |
| Test | Closed | Side menu and navigation Go to Sell screen and click on Selling Item tab on the top | Swathi Shenoy | It displays the list of items being sold by the user |
| Test | Closed | Side menu and navigation - Go to Sell screen and click on Buy tab on the top | Swathi Shenoy | It navigates to Buy screen |
| Test | Closed | Side menu and navigation Go to Buy screen and Click on Category tab on the top | Swathi Shenoy | It navigates to Category screen listing the categories of items available. |
| Test | Closed | Side menu and navigation Go to Buy screen and Click on Ending tab on the top | Swathi Shenoy | It navigates to the buy screen |
| Test | Closed | Side menu and navigation Go to Buy screen and Click on Sell tab on the top | Swathi Shenoy | It navigates to Sell screen |
| Test | Closed | Side menu and navigation Click on the Disconnect option in the side menu | Swathi Shenoy | It disconnects the user session and logs out of the app |
| Test | Closed | Side menu and navigation Click on the Logout option in the side menu | Swathi Shenoy | It logs out of the app |
| Test | Closed | Side menu and navigation Click on Home option in the side menu | Swathi Shenoy | It navigates to the buy screen |
| Test | Closed | Side menu and navigation Login to the app and click on the hamburger icon | Swathi Shenoy | It shows the list of navigation options |
| Test | Closed | Login screen activities User logs in and wants to register with his hawk ID to use the app | Swathi Shenoy | The app opens and prompts the user to add new google account with which he/she can login. |
| Test | Closed | Login screen activities User clicks on IIT Bazaar app and she/he is not connected to the internet | Swathi Shenoy | The app throws an error message stating that there is no internet connection and user should check this and reopen the app to Login screen activities User clicks on IIT Bazaar app and she/he is not connected to the internet |
| Test | Closed | Login screen activities User clicks on IIT Bazaar app and she/he is connected to the internet | Swathi Shenoy | The app home page loads fine and prompts user for the login creds |
| Test | Closed | Login screen activities User enters his/her hawk ID and clicks on Login(invalid user ID) | Swathi Shenoy | The backend validates that the user name is invalid and throws an error message |
| Test | Closed | Login screen activities User enters his/her hawk ID and clicks on Login(Valid user ID) | Swathi Shenoy | The backend validates that the user name is a valid and gets navigated to the item listing view |
| Test | Closed | Login screen activities: Go to menu and click on the IIT Bazaar app. |  | The app opens and prompts the user to login |
| Test | Closed | Login Activities for IIT Bazaar app | Swathi Shenoy | \*Test case Description\*:Go to menu and click on the IIT Bazaar app.   \*Expected output\*: The app opens and prompts the user to login |

# Elevator Pitch - Predemo (15 points)

## Done on 11/14/2015

# Use cases from end user point of view (+5 points)

The target user of this application is a member of the IIT community and to enable that community to facilitate the exchange of goods and service for money or otherwise. A predecessor of this system was also called the IIT Bazaar. This system is restricted to IIT students only to keep the safety and

A selling/listing users would list an item on the network and this item would be available for all to see.

The selling use must include a picture, a short description and a price. The price can be 0.00, meaning free or they are willing to trade. As is standard practice during the end of a semester, people will throw out many of their belonging. This can be put to a better cause by offering the chance for other student to obtain these items for either free, money or a trade for something of equal value but small and easier to transport.

A buying/soliciting user would look for items they want and click the buy button, which would message selling/listing user. Negotiation and meeting would be done through the already available email and the emails are structured in a way that allows filtration.

Listing on this system provides a safe environment for students, staff and faculty to exchange goods and services. As with services like craigslist, meeting in person in public places such as the MTCC is recommended, viewing the item before purchase is also recommended. Also, most importantly, money is an instrument that can be exchanged for goods, but they can also be given away or exchanged/bartered for something else.

This system mainly provides a solicitation platform for classified advertisements. Unlike a bulletin board, the scope is much narrower and specifically requires photos and items to be exchanged.

See additional use case diagram on next page

## UML Use Case Diagram