CS331 Project Final Phase Report

Team HaramBase
Shilei Lin, Joshua Folkerds, Matthew Kounniyom

Last updated Tuesday, December 13, 2016

HARAMBASE

THIS PROJECT IS DEDICATED TO OUR HERO, HARAMBE, THE GORILLA THAT WAS.

[PAGE LEFT INTENTIONALLY BLANK]

Table of Contents

PART I: Phase Summary

Section I. Task Decomposition

Name	Task Assigned and Completed
Shilei Lin	For this phase, I worked on the following functionalities: Search for Items, View Overall Commission Report, Show Item Info Bid on an Item, Show Items Bid On, Bid on, and View my Feedback. I also worked on CSS files, report, and debugging. Also, I worked on member and Item JavaBean. Additionally, I added the Search JavaBean in the last few days.
Matthew Kounniyom	For this phase, I was still sick, but I did my best to keep in contact and worked on the following functionalities: Login, View Sales Summary Report, and Update Profile. I also put most of the report together and found some bugs in the system. Also, I worked on Admin JavaBean.
Joshua Folkerds	For this phase, I completed the List my Items, Show Item Info, Show List of Bidders, and Add New Item functionalities. I also helped debug the system and helped with the report. Also, I worked on Bid JavaBean.

Table 1: Task Assigned and Completed.

Section II. Meeting Minutes

DATE	DETAILS
11/17	We got together and discussed the 12 functionalities we were to implement later. We divided them up into groups of 4. These groups are as follows. ➢ *Josh*: List my Items, Show Item Info, Show List of Bidders, Add New Item ➢ *Matt*: Login, View Sales Summary Report, Update Profile ➢ *Shilei*: Search for Items, View Overall Commission Report, Show Item Info Bid on an Item, Show Items Bid On, View my Feedback, Search Result, Bid on. ➢ Also, we decided to implement all the functionalities described in the project description no just the 12 required ones (more detailed in Part III).
11/20	We assembled as HaramBase and worked on our functionalities and joked around a bit. Also, we started to work on Javabeans and distributed 5 Javabeans to team members to implement them. “It is not bad at all”, said by Josh *Josh*: Bid Bean *Matt*: Admin Bean *Shilei*: Member and Item Bean
12/01	We have our first draft of Javabeans classes, and started testing them. To test the Javabeans we used the main method to drive the code. This was found out to be extremely helpful in debugging the Java classes. Also, Shilei added meaningful comments to Javabean methods.
12/04	We have our Javabeans ready. We quickly started writing JSP. We started with Admin functionalities. We got through all functionalities except the *adduser* functionality.
12/08	We continued integrating JSP pages with JavaBeans, and testing some part of the system. We also started some essential CSS design to make our Web Project sleeker and aesthetically pleasing. “We have finished 50% of the work!”, said by Shilei.
12/11	We got close to being done with our system. But with continued testing we found bugs and removed them accordingly. Shilei created the team logo for us with critique from his art major roommates. Also, 90% of the functionalities are proven working right now.
12/12-12/13	We finalized our tests and debugging. We finished the GABeS system as well as completed this report. “We got everything!”, Matthew yelled at us.

Table 2: Detailed Meeting Minutes

Section III. Tomcat URL

- The Tomcat URL is: http://tomcat.csbsju.edu/csci331/s1lin/gabes/index.jsp
- Here are some possible username and password combinations that can be used for demonstration and testing logins:

➢ **Testing Combos:**

TYPE	USERNAME	PASSWORD	BUYER	SELLER
Admin	admin	admin	-	-
Admin	manager	password	-	-
Member	slin	Lin	True	True
Member	irahal	Rahal	True	True
Member	Test	Test	True	True
Member	*Seller*	*Seller*	False	True
Member	*Buyer*	*Buyer*	True	False

*Table 3: Testing login Users. (PS: *means full functionality in Selling and buying)*

PART II: Reflection

Section I. Completed as required

1. **All the required 12 functionalities were completed.**
    - LOGIN for both Administrators and customers each of which must display the proper menu afterwards (Figures A1, A2, C1 and C2) --- 1 FUNCTIONALITY: login;
    - Administrator REPORTS (Figures A4 & A5) --- 2 FUNCTIONALITIES: view report 1, view report 2;
    - UPDATE PROFILE for customers (Figures C3) --- 1 FUNCTIONALITY: update profile;
    - SELLING MANAGEMENT (Figures C4, C5, C6, C7, in addition to Add Item) --- 4 FUNCTIONALITIES: list my items, show item info, show list of bidders, add new item;
    - BIDDING MANAGEMENT (Figures C8, C9, C10, C11, C6, & C12) --- 3 NEW FUNCTIONALITIES (4 in total): search for items, show item info (SAME ONE FROM SELLING MANAGEMENT), bid on an item, show items bid on;
    - VIEW MY FEEDBACK: 1 FUNCTIONALITY: View my feedback.

2. **Create all needed (fully documented & tested) Javabeans for components listed in above.**
    - The detailed of implementations are listed below:

JavaBean Classes	Functionalities Implement	Use View	Use Prepared Statement	Use Callable Statement
Admin	A1,A2,A3,A4, A5	Yes	Yes	Yes
Member	C1,C2,C3,C4,C 5,C8,C9,C10	No	Yes	Yes
Item	C6,C7,C12,Ad d Item	No	Yes	Yes
Bid	C11	No	No	Yes
Feedback	C14,View my feedback	No	Yes	Yes
Search	C12, C13	No	No	Yes

Table 4: Lists of JavaBeans and use of advance SQL or statements.

3. **Connect your HTML Web-based user interface to the JavaBeans using JSP and test your system.**
    - We have total of 33 files (include **.jsp:27, *.css:5, *.png:1*) for implementing the Web-based user interface;
    - The browser can properly open the web-based user interface.

4. **The writing of this nice and detailed report.**

Section II. Modified and Completed.

1. **For views, we have created 3 new Views and update 1 View:**
   - Since not all seller has feedback, we decided not to use `WHERE`-Clause because that will not find the correct number of records.
   - Therefore, we use *Natural Left Join(NLJ)* for two tables in many views especially the one Feedback with Item. Since Item must have a unique ID and a non-null sellerID, we join two table can give us the item along with the rating of current seller by sellerID.
   - Even though some sellers do not have any feedback, *NLJ* will keep those entries in the new table. This way we can get the correct information of total commission .
   - Here is detailed implementation:

COMPONENTS: VIEWS

NO	Description	Code
1	**NEW:** Total_Commission_View This new View sums up all the Commissions grouped by UserID.	`CREATE OR REPLACE VIEW Total_Commission_View AS SELECT SV.UserID as userID, HM.Uname as User_Name, HM.Fname as First_Name, HM.Lname as Last_Name, HM.Email as Email, SUM(SV.COMM) AS COMMISSIONS FROM Harambase_MEMBER HM, ITEMID_SELLERID_VIEW SV WHERE HM.UserID = SV.USERID Group By SV.UserID, HM.Uname, HM.Fname, HM.Lname, HM.Email ORDER BY SV.USERID;`
2	**NEW:** ITEMID_SELLERID_VIEW This new View Joins Sales_summary_report with Item by nature left join to get the sum of commission grouped sellerID.	`CREATE OR REPLACE VIEW ITEMID_SELLERID_VIEW AS SELECT SELLERID AS USERID, SUM(COMMISSION) AS COMM FROM (SALES_SUMMARY_REPORT NATURAL LEFT JOIN HARAMBASE_ITEM) GROUP BY SELLERID, ITEMID ORDER BY SELLERID;`
3	**NEW:** RATING_BY_ITEM_VIEW This new View Joins Feeback with Item by nature left join to get the average of seller rating grouped sellerID.	`CREATE OR REPLACE VIEW RATING_BY_ITEM_VIEW AS SELECT SELLERID AS USERID, AVG(OVERALLRATING) AS RATE FROM (HARAMBASE_FEEDBACK NATURAL LEFT JOIN HARAMBASE_ITEM) GROUP BY SELLERID ORDER BY SELLERID;`
4	**NEW This View has been updated.**	`CREATE OR REPLACE VIEW Overall_Commission_View AS SELECT USERID AS USERID, User_Name as User_Name, First_Name as First_Name, Last_Name as Last_Name, Email as Email, RATE AS Seller_Rating, COMMISSIONS AS COMMISSIONS FROM (Total_Commission_View NATURAL LEFT JOIN RATING_BY_ITEM_VIEW) Group By USERID, User_Name, First_Name, Last_Name, Email, RATE, COMMISSIONS ORDER BY USERID;`

Table 5: Added and Updated View

2. **For Functions, we have updated/modified 1 function:**
   - We found out the function `Harambase_getPrice_Func` did not return the second highest price among biddings. Thus, we realized we have encountered a bug. We modified the function and it works now.
   - Here is detailed implementation:

COMPONENTS: FUNCTIONS

NO	Description	Code

1	***Title:*** `GetPrice_Func`
	
	This is the trigger that will get the userID from the `USERSEQ` sequence before it gets insert into the database. It takes the input itemid and returns the next bid price for that item.
		`create or replace Function Harambase_getPrice_Func(iID int)`
		`Return int`
		`  AS`
		`    price int := 0;`
		`    numRows int := 0;`
		`    Begin`
		`        Select count(*) Into numRows From Harambase_BID B Where B.ItemID = iID;`
		`        If (numRows = 0) Then`
		`            RETURN 0;`
		`        ElsIf (numRows = 1) Then`
		`            Select I.StartPrice + 1 Into price From Harambase_Item I Where I.ItemID = iID;`
		`        ElsIf (numRows = 2) Then`
		`            Select bid into price`
		`            From (Select B.MaxBidLimit+1 bid, rownum rnum From Harambase_BID B Where B.ItemID = iID and rownum<=1 Order By B.MaxBidLimit DESC)`
		`            Where rnum >=1;`
		`        Else`
		`          Select bid into price`
		`          From (Select B.MaxBidLimit+1 bid, rownum rnum From Harambase_BID B Where B.ItemID = iID and rownum<=2 Order By B.MaxBidLimit DESC)`
		`          Where rnum >=2;`
		`        End If;`
		`        Return price;`
		`    End;`

Table 6: Modified Function

3. **For SQL, we have added some additional queries in JavaBean to make tables in *.jsp simpler to generates.**
   - In admin JavaBean, we add `getSubTotal()` and `getTotal()` two methods for us to use in generating the `sales_summary_report` table. The Method `getSubTotal()` will sum up sold price and commissions by category, and the method `getTotal()` will sum up all the sold price and commissions.
   - In Feedback JavaBean, we add `getNumOfRating()` and `getAvgOverall()` two methods for us to use in generating `view_my_ratings` table.
   - In Item JavaBean, we add `getBidItemInfo()`, `getAdditionalItemInfo()` and `getMaxBid()`. There are helping with getting Item information more thoroughly.

Section III. Omitted

We have omitted nothing.

Section IV. Added and Completed.

4. Added and completed (and briefly state why you added them)

Functionalities	Why added
Add user	This functionality is essential for the Harambase to easily add new test members into system. Instead of adding new users by adding new entries or regenerating the database, having this functionality can be considered as a shortcut.
View user	This functionality provides results of whether or not we have successfully add a new user or update a profile of a new member.
List of Bought	This is essential functionality for testing whether the selling system is working or not. When the auction ends, the winner will be chosen by the system and the person who wins the auction should know about it. This functionality provides this ability.
Leave feedback	This will provide another shortcut for testing serval functionalities such as Overall Commission Report.

Table6: Added functionalities not specified herein but specified in project description

5. More detailed about how we implemented please look in PART III Section I.

PART III: Highlights of HaramBase

ARGUMENT OF WHY WE DESERVE MORE THAN C

Section I. >3 GABeS features not required herein:

- **Admin Only Features:**
  - **View Users:** This feature is only available to the Admin. It is part of the *User Management System* that we have implemented which includes a list of all users (not admin) in detailed.
  - **Add User:** This feature is only available to the Admin. It is part of the *User Management System* that allows admin to add a new user in to Harambase. The add user functionality functions as follows: enter basic information (*username, first name, last name, email and password*), the user's id is generated automatically by triggers before executing the our add user procedure. (PS: the new users are set to be both buyer and seller).
- **Member Only Features:**
  - **List Items Bought:** This feature is only available to Members. It is under our *Buying Management Tools*, it only displays a list of the items that were bought by the current Member. How it functions is as follows: the method *getBoughtItems()* in our Member JavaBean file is called, this then returns a list of **distinct** item information that is being retrieved by checking the current userid with the id returned from the Getwinner_Func function (finds the Winner of current item) in Harambase for a match, and check if the item's status is "sold" through the GET_STATUS_FUNC in our Database.
  - **Rate Seller:** This feature is only available to Members when they have an item listed under their *List of Items Bought Menu*. It allows the Member to rate the seller of an item they have successfully bought. How it functions is as follows: the user enters their opinion of the following *Overall Rating, Item Quality, Delivery, Comments.* These values are passed added to the feedback item, which is then added to the database.

Section II. A highly sophisticated Web interface with CSS

- This can be refuted by looking at each Web-Page, which is paired with a corresponding CSS file. The CSS code is extensive and intricate, some highlights are the dropdown menu and team logo that are persistent features for users. We will demonstrate login page and 2 pages in the following table from admin and members total of 5 pages.

DESCRIPTION	SCREENSHOTS
LOGINPAGES	
1.Index.jsp This is the login page for our data base.	

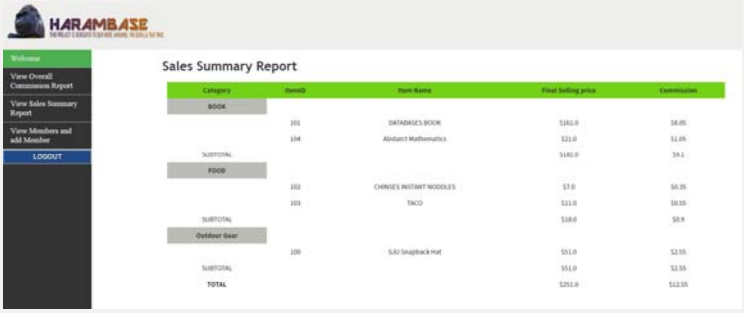
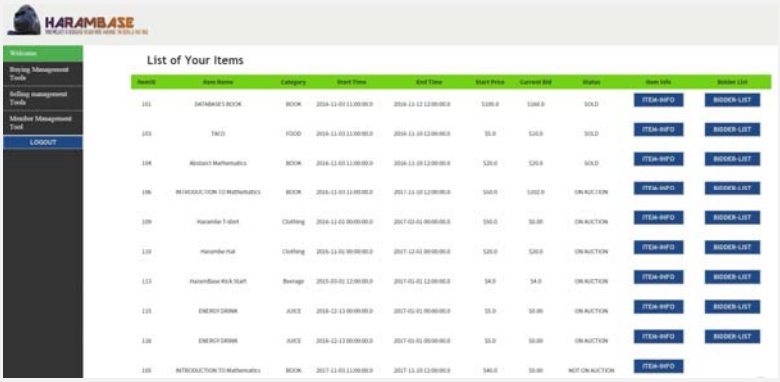
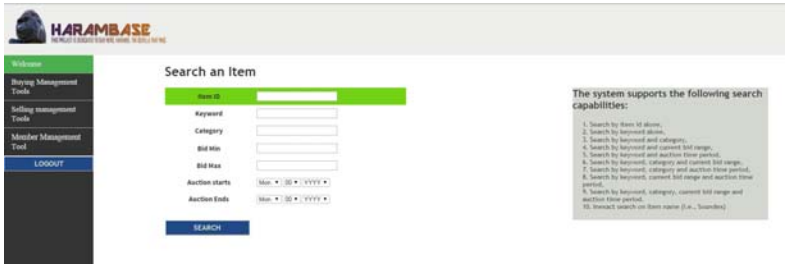
TWO ADMIN PAGES	
2.Welcomadmin.jsp This is the welcome page for admins.	
3.SalesReport.jsp This page shows Sales Summary Report. View Report 1.	
TWO MEMBER PAGES	
4.Listitem.jsp This page shows list of the items that have been listed by current member.	
5.Search.jsp This page will allow members to search for an item for bidding.	

Table3: Sophisticated web interface with CSS

Section III. Adequate error handling and security checks

- A user (admin and member) is not able to visit any pages without being logged in. If the current user is no longer logged in, then the user will be sent back to the login page.
- Bid on item will not be submitted unless
  - Current logged in member is not current winner;
  - Submitted max bid limit is higher than the item start price.
- Search result will omit the items that have been sold, or items that are listed by current logged in member or items currently that are not on auction.
- Add item will not be submitted unless:
  - The auction start time will be later than current timestamp;
  - The auction end time will be later than current timestamp and later than the start time;
  - The start price is not 0 or negative;
- The item information is not changeable once it is listed EVEN not on auction.