College of Saint Benedict & Saint John’s University

Computer Science Department

GABeS

Phase 4

Team Potatoes

Grant Boyer, Kyle Olson, Tom Husen

[This page has been intentionally left blank]

Table of Contents

Database Design ……………………………………………………………………………………………………………………... 4

EER Diagram ………………………………………………………………………………………………………………... 4

Relational Map Diagram ...…………………………………………………………………………………………... 5

Physical Database Design ………………………………………………………………………………………………………… 6

System Functionalities ……………………………………………………………………………………………. 7-10

SQL Code Components ………………………………………………………………………………………………………….. 11

Triggers ……………………………………………………………………………………………………………………… 11

Stored Functions ………………………………………………………………………………………………………… 12

Stored Procedures ………………………………………………………………………………………………… 13-14

Views …………………………………………………………………………………………………………………………. 15 Sequences …………………………………………………………………………………………………………………. 16

Issues Faced During Phase .………………………………………………………………………………………………….… 17

Task Decomposition …………………………………………………………………………………………………………….… 18

Meeting Minutes …………..…………………………………………………………………………………………………….… 19

November 8, 2016 ..……………………………………………………………………………………………………. 19

November 13, 2016 ……………………………………………………………………………………………………. 20

November 14, 2016 ……………………………………………………………………………………………………. 21

Appendix …………..………………………………………………………………………………………………………………..… 22

A ..………………………………………………………………………………………………………………………………. 22

B ..………………………………………………………………………………………………………………………………. 23

C ..…………………………………………………………………………………………………………………………. 24-26

Phase Summary

The way our team approached this phase was through both division of labor yet simultaneously maintaining close collaboration throughout. We began similar to how we began the third phase by creating a spreadsheet (Appendix A) of everything that needed to get done, who was going to be responsible for what, and how things would work together.

Working our way through the system description document, as well as our own documentation we created a list of all functionalities we would need to complete to make the best system possible. Using a draft format, we all laid claim to tasks we wanted to take on, and also tried to align tasks that would be similar (i.e. Admin Reports) to one person. This promoted consistency and saved us from having to re-invent the wheel. For a more detailed task decomposition please see the *Task Decomposition* page included in this report on page XX.

Grant focused primarily on the feedback aspects of the system (both viewing and leaving of feedback) in addition to the reports for admin users. Using aggregate functions and advanced SQL statements, the database is queried for information about sales, commissions, and feedback for all sales processed through our system. Averages are calculated, records are sorted into groups, and all of this information is presented in a very useful and efficient manner. Additionally, when leaving feedback for a transaction, a check is included to prevent feedback from being left multiple times. Once feedback is left for a transaction rather than a *Leave Feedback* button appearing a thank you message lets the user know they have already left feedback. Once feedback is left for someone, they can now view that information from their *My GABeS* menu. Add one more functionality – probably from Kyle since he has 6

Kyle focused mostly on item and bidding facets of the system. Key functionalities such as show item info or bid on an item were all part of Kyle’s contribution to our final product. Additional aspects including list my items, show list of bidders, show items bid on, and show items won were all also included in Kyle’s work for this phase. Using sophisticated integrity checks, Kyle’s bidding system prevents any bids being submitted that would be considered invalid. His pages also have detailed instructions so users can quickly and easily understand how the bidding works to ensure they have a successful bidding experience. His item reports also contain a wealth of information and provide quick access to relevant actions such as bidding on an item or getting even more information.

Tom worked mostly on user-side functionalities as well as ‘outlier’ functionalities. Essential operations such as login or add new item were completed by Tom in addition to updating a user profile, adding a new user, and search. One of the unique and standout features of our site is the excellent user interface. During the login functionality, the credentials of the user are established and follow them throughout their visit to our system. The system is smart enough to know when a user is an admin and will include the additional *Admin Options* only when appropriate. The search is also unique in that you can search using any combination of the fields, you are not limited to only certain combinations. May add stuff about search bar if it works.

Reflection

In the fourth and final phase of our database systems project we were able to accomplish so much more than we ever anticipated when we began this project more than three months ago. From the beginning steps of trying to understand what this ‘GABeS’ system was all about, to designing a fully functional and beautiful website of our own we have learned and accomplished so much.

In our final submission, we have completed all of the base requirements as well as several additional ones.

One of the few things we omitted was the use of two distinct login pages. We approached this project from the view of a potential client or end user who would end up using this website on a regular basis. What we decided was it is very important for an admin to be able to accomplish things a standard user cannot, but it is also important that an admin is able to interact with the system in the same ways a standard user can as well. With this in mind what we did was incorporate the admin functionalities into a menu drop-down from our main menu bar. A variable within the *User* JavaBean establishes whether the currently logged in user is an admin or not, and only displays accordingly. By omitting distinct login pages, we were able to make our system more efficient by eliminating redundant pages and simultaneously significantly improving the end users experience when interacting with the site – both as an admin or as a standard user.

In addition to the extra GABeS functionalities we added, which will be discussed later in this report, we also added several elements to our website to improve the quality of our final product. One such feature is related to leaving feedback for an item that you have won. Once you win an auction, the newly won item will move from *Items I’ve Bid On* to *Item’s I’ve Bought*. When visiting the second page there is a button for you to add feedback about the transaction. One thing our group added to this page was that once a user has left feedback about a transaction, rather than seeing a button to leave feedback, they will see a message thanking them for already leaving feedback about this transaction. This ensures no conflicts within the database, as well as allows the user to quickly and easily see which transactions they have and have not left feedback for.

In the interest of creating the best user experience possible, we knew it was crucial to include functionalities such as *Leave Feedback*, *Add New User*, *and Update Item Info*. These are three important things to have in a system such as ours for several reasons. First, they allow the system to be easily managed, updated, and maintained. Without operations like these, manual inserts into the database would be required on many circumstances which leads to an increased likelihood of redundancy and poor data integrity. Additionally, as a user these functionalities are very basic when trying to use a bidding service such as ours. Without having these abilities, the scope of what you are able to accomplish is severely diminished.

Argument

As discussed previously in this report and demonstrated in our live demo, all 12 required functionalities are included and work flawlessly.

Additionally, included are 3 GABeS features from the description document, but not incorporated within the phase 4 requirements. These functionalities include: *Add Feedback, Add New User,* and *Edit Item Info*. All of these functionalities work as intended and were also demonstrated during the live demo. more

Also included in this report is the Normalization analysis for all tables within our database. more

As clearly relevant when interacting with our site, there is a sophisticated web interface front-end to our system that makes the user experience pleasant, efficient, and productive. All formatting completed via CSS. More

Additional Oracle features (Sequences, etc) are also included. JSP features. More

Security checks – bids, etc. more

Issues Faced

During this phase we faced many different issues stemming from SQL Developer not working properly to figuring out how on earth everything would be connected and come together. Ultimately, we were able to vanquish many of these issues and by not cutting corners and working our hardest to ensure our solutions were both efficient and forward thinking, we will be saving ourselves lots of work in the next phase as well as making our product more complete.

When we began working on this phase, we started by going through every website that we designed during Phase 1. For each webpage we discussed what exactly the purpose of that page was, and how we though to best accomplish that in terms of SQL implementation. Then we divided all of the tasks so each team member could work on their tasks before bringing them all together into final products. For some of these, issues arose when trying to implement our initial implementation idea so we had to bounce ideas off each other to determine what would actually be the best approach to the problems. This collaboration about our ‘separate’ parts of the project was immensely helpful as they often applied to more than one task which caused us to be consistent across the board when it came to the implementation.

Another problem we faced was a couple times when testing a SQL query on our team server, we experienced lag and sometimes crashing of our Horizon Client that was running on our personal computer (though not necessarily VMware’s fault). While there is nothing we could really do at the time of crashing, we agreed that if we were each to have a copy of the databases on each of our individual Oracle DB connections, we could test our queries/functions/procedures/etc. without these bugs happening. Once we determined that the process worked properly on our own DB connection we would sync it with the team connection.

When working on the Bid on Item functionality, we were planning on using a Scheduler to execute a stored procedure every 5 seconds or so because that seemed like the best way to do psudo-live checking of the time. When we tried creating the scheduler we ran into permissions issues and the Oracle server wouldn’t allow us to create or run the scheduler we needed. The code that we would have used is listed in Appendix A. We created a work around that does work but it doesn’t check every 5 seconds like we had hoped. Our hope is we can straighten out our scheduler and implement that in Phase 4.

Task Decomposition

Grant:

* Feedback JavaBean
* Admin Sales Summary
* Admin Commission Report
* View My Feedback
* Add New Feedback

Kyle:

* Bids JavaBean
* List My Items
* Show Item’s Info
* List of Items User Bid on
* List of Items User Won (Bought)
* List of all Items for Sale
* Edit Item’s Info
* Show List of Bidders for an Item
* Bid on an Item

Tom:

* User and Item JavaBean
* Add New User
* Add New Item
* Update Profile
* Login/Logout
* Search

All Team Members:

* Populated beans with methods pertaining to our respective functionalities
* Normalized the tables from our database
* Implemented an extremely user-friendly UI using in depth CSS/HTML

Team Potatoes Minutes

November 8, 2016

Meeting began at 3:45 am.

**In Attendance**:

* Grant Boyer
* Kyle Olson
* Thomas Husen

**All**:

* Discussed all of the elements that goes into this phase and the best approach for completing them
* Began creating database tables and populating them with sample data values
* Discussed our relational map and EER diagram as they relate to the database and began making appropriate changes

Meeting adjourned at 5:30 am.

Meeting Minutes

Team Potatoes Minutes

November 6, 2016

Meeting began at 4:30 pm.

**In Attendance**:

* Grant Boyer
* Kyle Olson
* Thomas Husen

**Tom**:

* Update EER and Relational Map based on our most recent decisions regarding our database

**All**:

* Review all webpages created during Phase 1 – generate spreadsheet breaking down each page to the tasks it needs to accomplish, how we think to best accomplish that task, comments about the task, and who is going to be responsible for completing that task – Appendix B
* Discuss any changes that are required to be made on the EER diagram or the Relational map

Meeting adjourned at 11:30 pm.

Team Potatoes Minutes

December 7, 2016

Meeting began at 9:00 pm.

**In Attendance**:

* Grant Boyer
* Kyle Olson
* Thomas Husen

**Grant**:

* Finishing touches on leave new feedback functionality
  + - Added restriction about reviewing a transaction multiple times
  + Began generating information for the phase report

**Kyle**:

* Continue work on editing an item’s info.
* Put finishing touches on bidding functionality

**Tom**:

* Continue work on search functionality
* Compile various versions and various files into our master repository

**All**:

* Continued testing the website, trying out many different scenarios to improve reliability and consistency throughout the entire user experience
* Assisted each other with any blockers they experienced with their assigned tasks
* Continued implementing CSS styling throughout the site

Meeting adjourned at 12:00 am.

Appendix A:

[Link to Full Spreadsheet](https://docs.google.com/spreadsheets/d/1pejnJdGgBSz0PK1PAWdoMoDXfXdx-fRefrvWnf6iepg/edit?usp=sharing)



Appendix B: