Vehicle *e*-Auction Final Phase

SUBMITTED TOWARDS THE

PARTIAL FULFILLMENT OF THE REQUIREMENTS OF

Mr.Dude e-Auction INC.

BY

William Wang

YuChen Zeng

Robert Beck

Michael Berezanich

JingRui Duan

Under The Guidance of

Prof. Wang-Chien Lee

DEPARTMENT OF COMPUTER SCIENCE

The Pennsylvania State University

University Park

Pennsylvania

Acknowledgments

It gives us great honor in presenting the Phase II project report on the team's progress in design and development of our **e-Auction** system**.** We would like to take this opportunity to thank the TAs and Prof. Wang-Chien Lee for their time and effort in advising and overseeing **Generic Team Name**'s progress. Without this kind patronage and guidance, Phases I & II would not have taken shape as it is today.

We are also grateful for all of the resources given to us from The Pennsylvania State University. The endless resources provided by the Department of Electrical Engineering and Computer Science facilitated the implementation and testing our design had gone through.

Without these indispensable faculties, suggestions, and lectures, **Generic Team Name** would not have been able to accomplish the milestones outline in this report. For this, we owe sincere gratefulness for all of the guidance and encouragement we received.

Phase II Highlights

This document is the Phase II report for the database-backed Vehicle e-Auction web application project. This Phase II report covers all the tasks reported in Phase I plus the following new tasks:

- Corrections and improvements based on the evaluations of Phase I report
- Database Schema Definition and Normalization
- SQL Statements to create tables and views
- Technologies selected to implement Vehicle e-Auction.
- MySQL scripts for populating test data

The goals of this Phase II report are concentrated on schema definitions, schema refinement and normalization, SQL statements to create tables and views, technology analysis and technologies selected to implement this Vehicle e-Auction system, and SQL script to populate test data.

Contents

List of Figures

List of Tables

1. *Introduction*

According to the United States Department of Commerce, the United States has one of the largest automotive markets in the world. Light vehicles sales alone reached 17.5 million units in 2016. Overall, the United States ranks the second largest market for vehicle sales. [1] As shown in Figure 1.1, automotive sales is experiencing a steady positive trend in sales. Furthermore, Figure 1.2 shows the rising percentage of e-commerce retail sales. It is very clear that Hilbert Dude's vision in building an online e-commerce for automotive sales is a great investment.

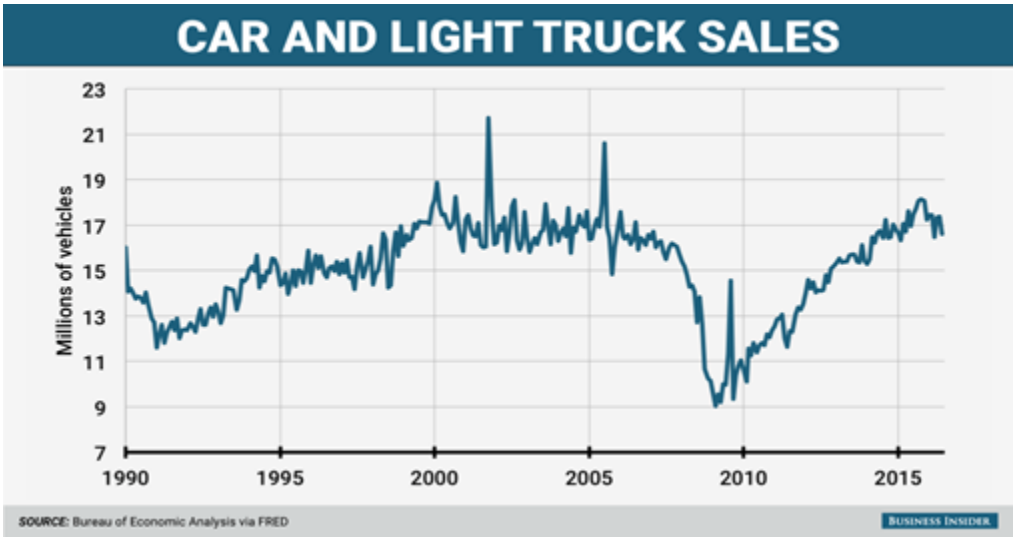


Figure 1.1: Cars and Light Truck Sales

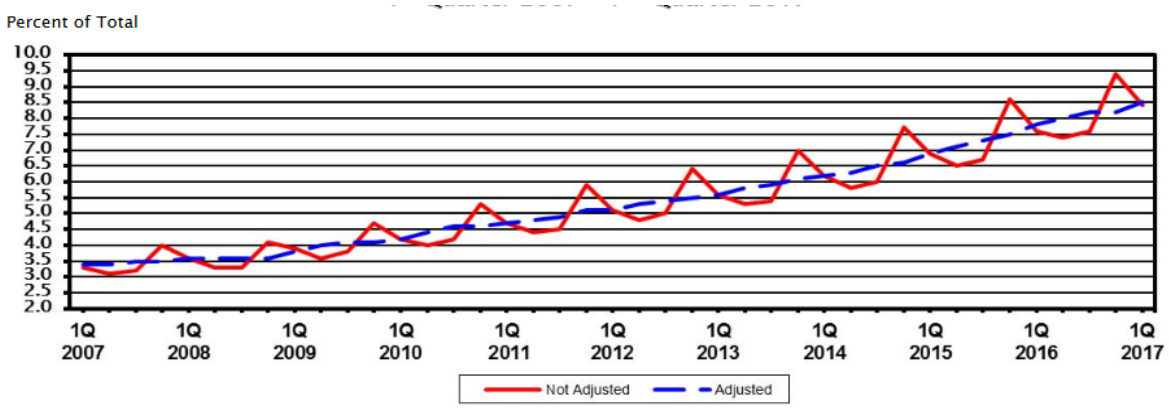


Figure 1.2: The rising Percentage of Online Retail Sales

After this Vehicle e-Auction web application is fully tested, the final product will be delivered to the investor Mr. Hilbert Dude.

Section 2 provides requirements analysis in terms of features and functionalities analysis. Section 3 gives the conceptual database design for the back-end database. Section 4 gives the database schema definitions. Section 5 provides the SQL statements for creating tables and views. Section 6 gives a detailed analysis about the technologies selected to implement this Vehicle e-Auction system. Section 7 provides a MySQL script for populating test data. This project report is an ongoing document. It will be fully completed during the final Phase.

2. *Requirement Analysis*

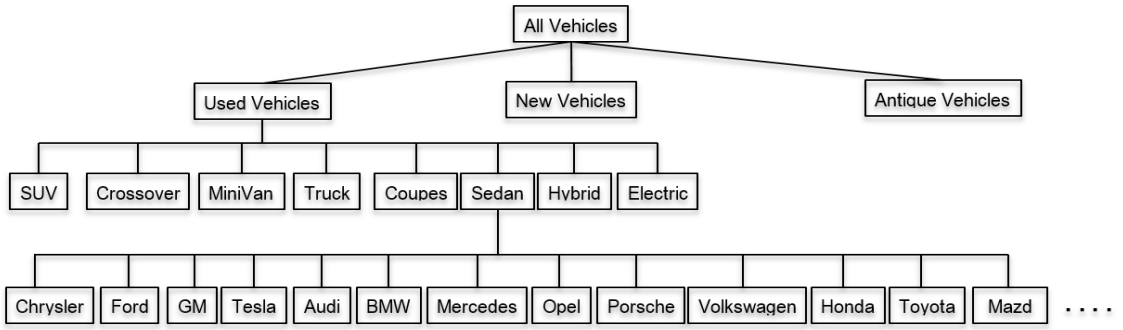
After the several communicating with Mr. Dude, the project team has a solid understanding about his vision in e-commerce and his expectations for the Vehicle e-Auction system. The team has performed a thorough requirements analysis. This section provides detailed descriptions for the features and functionalities of the Vehicle e-Auction system. The conceptual database design of the back-end database will be given in section 3.

2.1 Auction Items

Automobiles will be sourced from the users themselves. The website allows registered users (detailed in Section 2.2) to post their automobiles for cataloging. Accompanying each product entry will be short descriptions limited by 400 character length, personal URLs providing more specifics, photos, and location (state). Even though this system is primarily an auction website, the website needs to be able to homogenize itself within the automotive community allowing users to be able to connect with each other. This facilitates automotive shows and socials which furthers the network of users within the **e-Auction** community. Thus, the website does not mandate all postings to become auctionable items. However, if the registered user decides to list a product to be auctioned, a reserve price, buy it now price, auction

period, and shipping location is attached to the seller's vehicle. In more detail, the reserve price is a price set by the seller which determines the threshold when the seller is obligated to sell at the auctioned price. In other words, if the highest auction price at auction closing does not meet or exceed the reserve price the seller is not obligated to sell the product.

2.1.2 Categories



Note: To avoid crowded diagram, only partial of the category tree is drawn

Since **e-Auction** provides a platform for peer to peer commerce. A predefined set of categories will be in place to facilitate users in cataloging their vehicles. Since this system focuses primarily with automobile transactions, the root of the catalog are automobiles. Shown in Figure 2.1 is a breakdown of the categories.

2.2 Users

The target audience for this website are automobile owners, buyers, and dealers (detailed in Section 2.2.2) as well. Aforementioned, **e-Auction** promotes not only the transaction aspect of the automobile community but also the community and its social aspects. Thus unregistered users are allowed access to this community but are limited to only browsing(detailed in Section 2.4) and searching(detailed in Section 2.5). In order to take advantage of the full sets of features the **e-Auction** community has to offer, the user must become a registered user (detailed in Section 2.2.1).

2.2.1 Registered Users

In order to become a registered user, one must register an account. Each account is identified by a user name and authenticated by a password. Additionally, an account consists of an email address, name, home address (street, city, state, and zip), phone number, bank account/s (account number, routing number, and bank name), age, gender, annual income, and auto insurance policy information(policy number, insurance company name, insurance company code). To protect the vehicle during auction transaction, auto insurance must be verified when an user wins an auction. After confirmation of a successful registration, the user will be able to list vehicles to become cataloged, bid on auctions, and sell their listings to become auctioned. Additionally, the user will be allowed to edit a personal bio page to provide a background. To cater to the community aspect, registered users will have the privilege to post and respond to discussions within the forums section of the website. Only

registered users can bid on auctions.

2.2.2 Car Dealer Companies

Car Dealer Companies which express interested in using this platform must also register as a user. However, only after confirmation of the existence and proper licensing of the respective dealership may companies become active members. Upon successful confirmation, all dealership accounts are identified by username and authenticated by password likewise to registered users. However, dealership accounts are maintained with information about company name, main point of contact, address, phone number, revenue, dealership categories(ie. Luxury, Exotic, Brand, etc.) and dealership age. These accounts inherit the same privilege as normal registered users; however, all listings must be posted as auctionable items.

2.3 Ratings

All registered users within their bio page contains a mandatory public rating and reviews section. This system is used to prevent scams and item fraud within the **e-Auction** community. In order for a registered user to be eligible to leave ratings on another user's bio page, the reviewer must have participated in a past reviewed user's auction. The system is comprised of a scale out of ten accompanied by a short explanation with a 250 character limit.

2.4 Browsing

Mentioned in Section 2.2, guests (unregistered users) only have the privilege to browse **e-Auction**. Thus guests and users will be able to traverse through the category tree to view the catalog of vehicles within the database. Within each level of the category tree, the user is presented with a description of the category as well as a summary of all items within that category. As the user goes deeper and deeper into the category tree, listings become much more specific and less broad thus limiting the resulting summaries. However, guests will not be able to bid(detailed in Section 2.6) on a vehicle they are browsing. Allowing guests to browse **e-Auction** helps attract new users and potentially gain more customers.

2.5 Searching

All users are able to search the system for specific vehicles. To search, users are able to input keywords or select conditions. For example, conditions may be as specific as a combination of brand, model, condition, and year range. After inputting the search parameters, a list of vehicles which satisfies the search will be listed for the user to browse. The searching method provides a much more detailed manner for users to browse for products. As mentioned in the previous sections, guests are still restricted to bid on vehicles they are browsing.

2.6 Bidding

Only registered users and non-dealers may place bids on items which are not theirs during the designated auction period. Since there is a reserve price for each item auctioned, the starting price will begin at 0 dollars. A user may only bid prices which are 5% above the current highest bid price and are not restricted in the number of bids. The bid price restrictions promotes competitive prices for items. After each change in highest price, all users who participated in the auction will be notified of the new price as well as who leads the auction in highest price. When the auction period is over, all participating users will be notified of the winner and winning price; additionally, contact information is sent to both the buyer and seller. Furthermore, the auction will transition into the delivery (detailed in Section 2.9) phase.

2.7 Buy It Now

Accompanying each auctioned vehicle, there is an option to purchase the vehicle at a "Buy It Now" price, determined by the seller. This feature allows users to skip the auction period and bidding processes attaining the vehicle at a premium. After confirming that a user is using the "Buy It Now" method instead of bidding, all bidders will be notified of the auction closing early, who won the vehicle, and the winning price ("Buy It Now" price). Contact information is sent to both the buyer and seller; the auction will transition into the delivery (detailed in Section 2.9) phase.

2.8 Auction Statistics

Weekly reports are made for each category of items. The reports summarize the statistics of the items. These reports help determine trend in value of categories not only by a small weekly view but also monthly, seasonal, and yearly. These statistics help **e-Auction** to determine trends in depreciation and inflation of value. For example, a winter coat may be cheaper in the summer than it is in the winter. This type of statistics help sellers as well as buyers to determine when to auction an item.

2.9 Delivery

The final and most important stage of an auction's life cycle is delivery. To ensure the successful movement of goods and transfer of money, **e-Auction** serves as the medium where both money transfer and vehicle shipping is conducted. At the end of an auction, the seller will have a 2-week window to transfer the vehicle to one of **e-Auction** holding sites. Likewise, the buyer's bank account will be notified to transfer the proper amount of money to **e-Auction** within the 2-week window as well. Once **e-Auction** confirms the arrival of both the funds as well as the vehicle within the window period, the vehicle will be transferred to the buyers address and the money will be deposited to the seller's bank account. On the other hand, if there is no successful confirmation of both the funds and vehicle arriving within the 2-week period. The transaction is deemed as void and items which **e-Auction** received will be forwarded back to the respective owners. Both parties will be notified whether the delivery was successful or voided. These information will be stored for a six month period

time after notifications are sent.

2.10 Reports to Telemarketers

Telemarketers will receive details of the users behaviors. Included in these reports will be the name, address, email, age, gender, and annual income. These information will be given privately without user knowledge. Thus when users register for accounts the telemarketing report information is required.

2.11 Mechanic Certifications

Sellers have the option to have their vehicle appraised and evaluated by one of **e-Auction** certified mechanics. The certified mechanic recommends users the current market price as well as a holistic vehicle report. These reports are attached with the VIN of the vehicles and are available for buyers to view. After an auction closes and transitions to the delivery phase, all vehicles will go through a holistic report from a certified mechanic on site. If the vehicle's description is deemed misleading or fraudulent by the certified mechanic, the transaction is deemed void and a 20% penalty is charged to the seller.

2.12 Deal Rating

With weekly auction reports, significant amount of data is gathered through the life-time of the website. This allows for holistic analysis of auctioned products. These statistics allows users to adjust their reserve and "Buy It Now" prices according to the value of their

product. Additionally, these statistics are displayed in the vehicle summary to foster competitive auctions. Buyers will be able to have a feel of which vehicle better suits their long term and short term budgets.

2.14 Discussions

To cater to the social aspect of the automobile enthusiast community, **e-Auction** provides a social discussion driven platform for registered users. As mentioned, guests may browse and search for relevant discussions but are restricted from posting and responding. The Discussion sections is separated into two parts: auction questions, and topics. Auction questions provides a platform for buyers to post questions about an auctioned vehicle as well as sellers to post responses and announcements. On the other hand, the topics portion of the discussions feature is a topic driven social platform which promotes users to ask general questions whether it be maintenance concerns, car ownership stories, etc.

2.15 Display Floor

The display floor allows users to proudly display their vehicle. The display floor accompanies each vehicle and provides a medium for users to upload images and videos which highlights the beauty and features of their vehicle. This allows buyers to be able to have a virtual tour of the vehicle as well as users to display their collection.

3. *Conceptual Database Design*

This section details the technical model for **e-Auction**'s database. The database will be modeled using the ER-Model, entity relationship model. The following subsections will describe the entities, relationships, and full diagram of **e-Auction**'s ER-Model. Additionally, each subsection will highlight the assumptions and constraints.

3.1 Entities

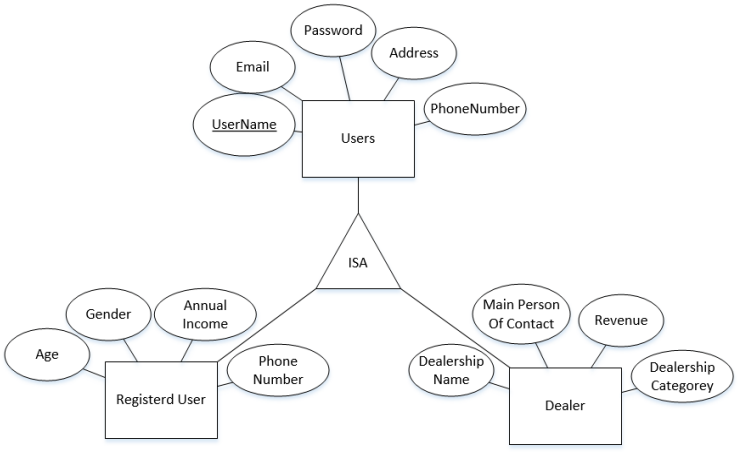


Figure 3.1: Users ISA Hierarchy

3.1.1 Users

The user entity is a set of all users with registered accounts within the **e-Auction** community. However, detailed within the requirement analysis, there is a separation between users. A user can can either be a Dealership or a registered user. Thus an ISA hierarchy will be implemented to differentiate the two entities. Thus the hierarchy can be described as follows: **User** is a **Dealer** or **User** is a **Registered User**. Figure 3.1 describes the ISA structure and entities in more detail.

Table 3.1: Users Attributes

	Attribute Name	**Primary Key**	**Foreign Key**
Users	userid	yes	
	password		
	email		
	address		
	phonenumber		

continued on next page

continued from previous page

Dealers	userid	yes	Users
	dealershipname		
	personofcontact		
	revenue		
	dealershipcategory		
RegisteredUsers	userid	yes	Users
	name		
	gender		
	age		
	income		
	insurancepolicyno		
	insurancecompanycode		

continued on next page

continued from previous page

	insurancecompanyname		

3.1.2 Vehicles

The vehicles entity is a set of all vehicles which users catalog. The vehicles entity set is identified by VIN (Vehicle Identification Number) and have attributes: category, make, model, year, mileage, description.

Table 3.2: Vehicles Attributes

Attribute Name	**Primary Key**	**Foreign Key**
vin	yes	
category		
make		
model		
year		
mileage		
description		

3.1.3 Bank Accounts

Since **e-Auction** directly deals with money transfers, many users may have allocated money in various accounts. To accommodate this, bank account information is designed as an entity set. This allows users to enter multiple bank accounts to pay for items they have won. Additionally, each Bank Account must be owned by an user.

Attribute Name	**Primary Key**	**Foreign Key**
account number	partial	
routing number		
bank name		

Table 3.3: Bank Account Attributes

3.1.4 Reviews

Each user can post a review for auctions the user was involved in. The buyer can post a review to rank the seller and the seller also can post a review to rank the buyer. The reviews entity set is identified by the review ID; other attributes: date, auction ID, ranking, review.

Attribute Name	**Primary Key**	**Foreign Key**
reviewid	yes	
reviewerid		Users
sellerid		Users
actionid		Auctions
rating		
comments		

Table 3.4: Reviews Attributes

3.1.5 Auctions

Auctions entity is the set of all auctions. This entity is identified by auctionId and has attributes: VIN, end date, reserve price, buy it now price.

Attribute Name	**Primary Key**	**Foreign Key**
auctionid	yes	
vin		Vehicles
sellerid		Users
start date		
end date		
reserve price		
buy now price		

Table 3.5: Auctions Attributes

3.1.6 Bids

Users can place any number of valid bids on any day. The Bids entity set is identified by the bid id and have attributes: bidder username, auction ID, price, timestamp.

Attribute Name	**Primary Key**	**Foreign Key**
bidid	yes	
bidderid		Users
amount		
timestamp		

Table 3.6: Bids Attributes

3.1.7 Transactions

Transactions entity set records all valid transactions. Transaction entity set is identified by transaction id, seller username, buyer username, VIN, price, date.

Attribute Name	**Primary Key**	**Foreign Key**
transactionid	yes	
bidid		Bids
auctionid		Auctions
transaction state		
transaction date		

Table 3.7: Transactions Attributes

3.1.8 Discussions

The discussions entity is part of the social aspect of **e-Auction**'s car community. This is broken down by ISA hierarchy with auction questions entity and topics entity. Auction Questions is for both seller and buyers to post messages. Buyers can post questions and comments to the seller and seller can answer questions. Topics are for all users to discuss about miscellaneous topics.

Attribute Name	**Primary Key**	**Foreign Key**
discussionid	yes	
discussion		
timestamp		

Table 3.8: Discussions Attributes

3.1.9 Display Floor

Display floor allows users to upload pictures and videos of the vehicle. Users can browse the pictures and watch videos. The display floor entity set is identified by VIN, and have attributes: title, url.

Attribute Name	Primary Key	Foreign Key
displayid	yes	
userid		Users
vin		Vehicles

Table 3.9: Display Floor Attributes

3.1.10 Events

Events entity is another part of the community aspect. Events are user created and are identified by eventID with attributes: event name, date, address, and category.

Attribute Name	Primary Key	Foreign Key
eventid	yes	
eventdate		
location		
description		

Table 3.10: Events Attributes

3.1.11 Media

Media entity is a set of all media files users uploads. Media is identified by fileID and has attributes: type, name, date.

Table 3.11: Media Attributes

Attribute Name	**Primary Key**	**Foreign Key**
mediaid	yes	
displayid		Displayfloors
filetype		
filename		

3.2 Relationships

3.2.1 MoneyIn

The MoneyIn relationship set establishes the ownership relation between users and bank accounts. Each bank account must owned by exactly one user, a key constraint and total participation constraint is applied to the Bank Accounts entity set. Figure 3.2 describes this entity in detail.

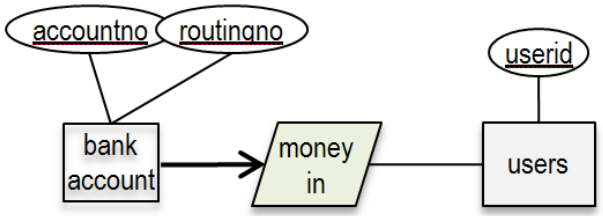


Figure 3.2: MoneyIn Relationship

3.2.2 Owns

e-Auction allows registered users to input vehicles to catalog within its database. Thus a relationship arises between registered users and vehicles. This relationship does not mean that users are selling the vehicle. This relationship essentially allows proud owners to display their collection of vehicles to the **e-Auction** community. Within this relationship, a key constraint is placed onto the vehicles set. These constraints are placed because a vehicle can only be listed by one owner. Figure 3.3 illustrates the relationship set between users and vehicles.

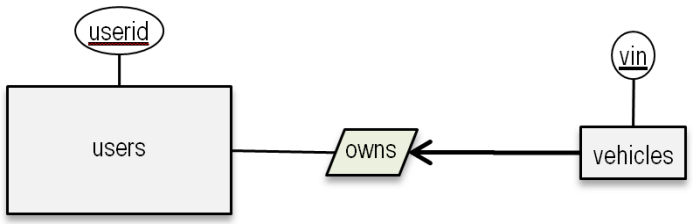


Figure 3.3: Owns Relationship

3.2.3 List for Auction

The main focus of **e-Auction** is to foster a platform for peer to peer commerce. Thus the list for auction relationship associates the users entity set to vehicles entity set. A key constraint is placed upon the auctions set due to the fact that every auction can only have one seller.

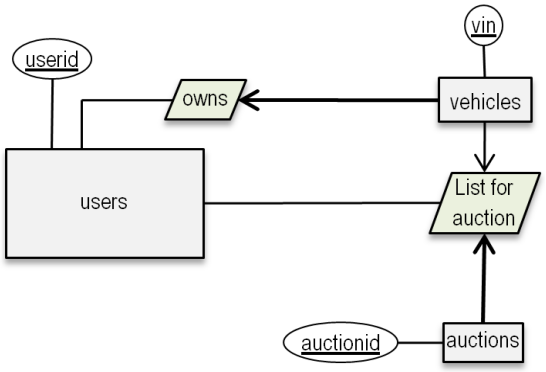


Figure 3.4: List for Auction Relationship

3.2.4 Rates

Users are allowed to rate other users who have participated in an auction together. Since the system is designed for users to rate users, this relationship is ternary involving a two way association of users entity set and an association with reviews entity set.

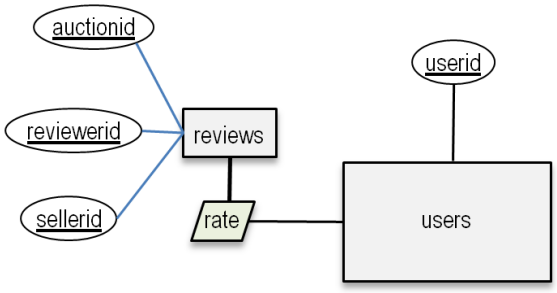


Figure 3.5: Rating Sellers Relationship

3.2.5 Place Bids

Registered users can place any number of valid bids on auctions before auction close. Place bids is a ternary relationship associating registered user entity, bids entity, and auction entity. Only registered users can place bids

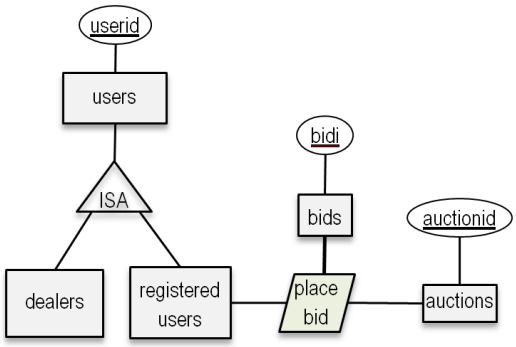


Figure 3.6: Place Bids

3.2.6 Closed

When a vehicle auction end date is reached, all bids for the vehicle will be checked to announce the winning bid if there is any. Closed associates bids entity and auctions entity with descriptive attribute validity.

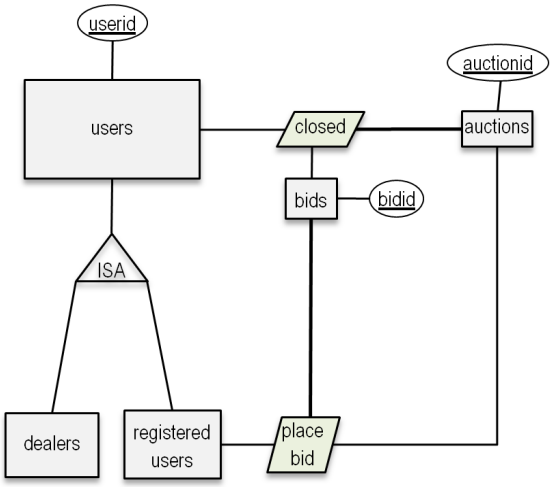


Figure 3.7: Closed Auction Relationship

3.2.7 Delivered

The delivered entity associates closed auctions and transactions. Figure 3.8 shows that after the vehicle is delivered, the transaction is completed.

3.2.8 Returned

To ensure buyer satisfaction, certified mechanics at **e-Auction** will examine the vehicle to certify that the seller description is valid. If the vehicle is deemed faulty, the vehicle is returned to the seller and a 20% penalty fee is applied, the transaction is deemed void. Additionally, if the either the money or vehicle did not arrive in the 2-week period, the received item will be returned and the transaction is deemed void. The returned relationship uses closed relationship as an aggregation and associates it with user entity and vehicle entity. Figure 3.8 illustrates a vehicle can be returned for a faulty transaction.

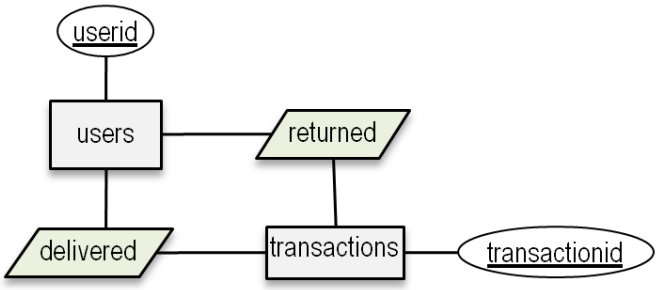


Figure 3.8: Vehicle Delivery and Return

3.2.9 Uploads Media

A seller can upload multiple pictures and videos of the vehicle to the display floor. Uploads media relationship associates user entity with media entity.

3.2.10 Belongs To

The display floor contains media files for users to browse. Belongs To relationship associates media entity with display floor entity.

3.2.11 Post Discussion

Users can post discussions. Post Discussion relationship associates users entity and the discussions entity.

3.2.12 Create Event

Users can create events for the community to see. Create Event relationship associates user entity and event entity.

3.2.13 Attending Event

Users can attend events listed within the community. Attending Event relationship associates users entity and event entity.

3.3 All Together

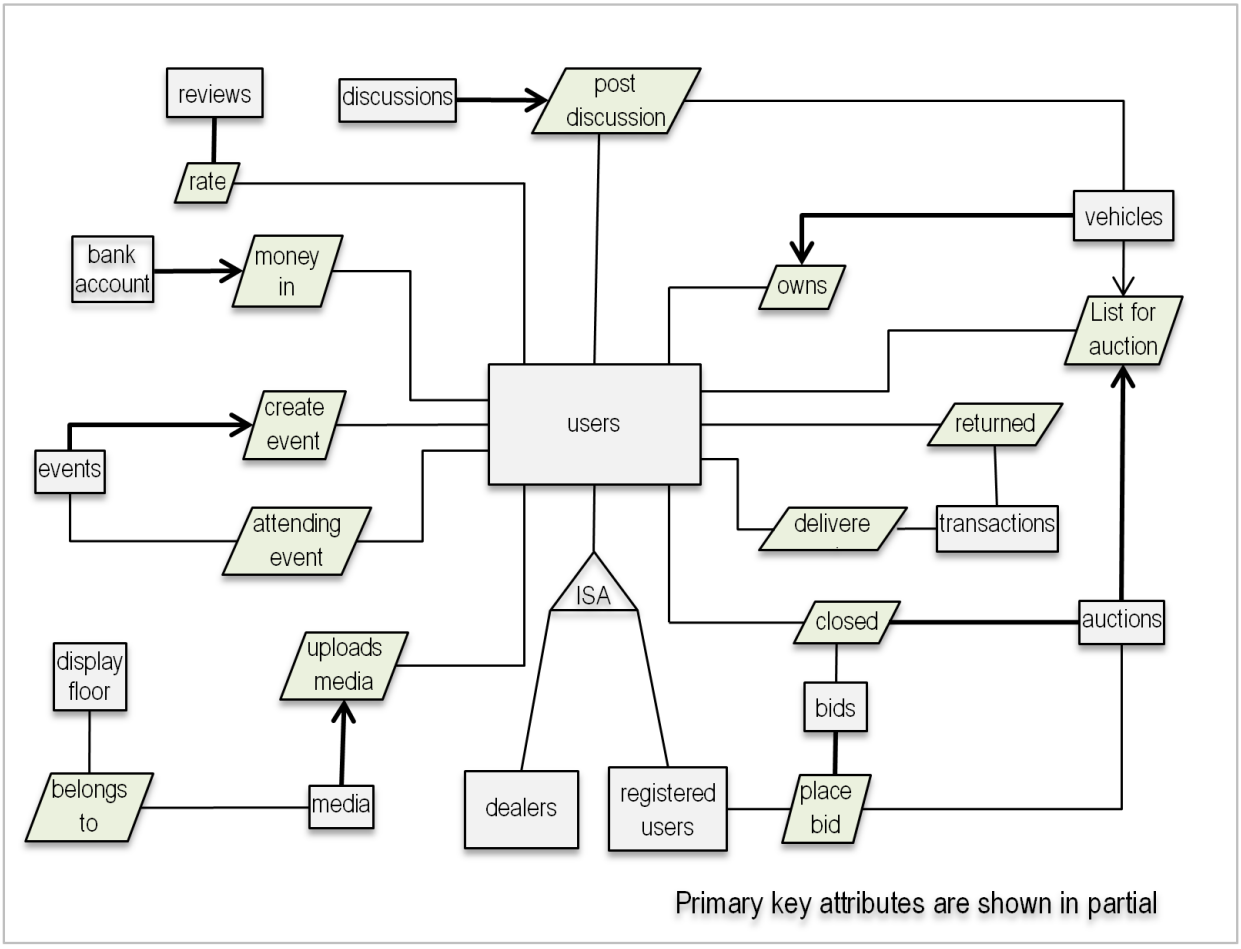


Figure 3.9: Full ER-Diagram

4. *Schema Definition and Normalization*

4.1 Schema Definitions

4.1.1 Initial Schema Produced By Translating the ER Diagram

The Following subsections provide the initial database schema definitions obtained from the ER diagram as shown in Figure 3.3. This initial database schema is the result of the straightforward translation from ER diagram to database schema. There are many redundancy and all integrity constraints are not enforced. This initial database schema serves as the starting point for the team to tackle the database schema design. It will be refined during constraints enforcement in section 4.1.2 and during normalization in section 4.2.

4.1.1.1 Users

e-Auction classifies all users into two categories: Dealers and Registered Users. As discussed in Section 3 and Figure 3.1, the separations between users are captured within an ISA hierarchy. Because registered users can bid on auctions to purchase vehicles, to protect the vehicle under transaction, the auction winner's auto insurance policy information need to be verified before delivering the vehicle.

Users(userid, email, password, address, phonenumber)
Dealers(userid, dealershipname, revenue, delershipcategorey)
RegisteredUsers(userid, age, gender, income, insurancepolicyno, insurancecompanycode, insurancecompanyname)

4.1.1.2 Banking

It is assumed that most users need to use money in multiple bank accounts to purchase a vehicle. Thus bank accounts are captured as an entity sets. The MoneyIn relationship set establish the ownership between users and bank accounts..

BankAccount(accountno, routingno, bankname)
MoneyIn(userid, accountno)

4.1.1.3 Vehicles

Users are encouraged to show vehicles they own for the community to see. Thus logically, users may list many different vehicles. Each vehicles must be owned by exactly one user. The Owns relationship set will establish the relation for who owns which vehicles, or which vehicle is owned by whom.

Vehicles(vin, category, make, model, year, mileage, description)
Owns(vin, userid)

4.1.1.4 Auctioning

The following definitions portray the all of the aspects associated with the auctioning processes. An user including dealers can list his vehicles for auction. Only registered user can bid on an auction. Each auction will be closed either at the auction end date or at when a user offers a buy-now price. After the vehicle is delivered, the transaction will be complete.

However, a user still can return the vehicle is the vehicle actual condition does not match the seller claimed or certified condition.

Auctions(auctionid, vin, sellerid, startdate, enddate, reserveprice, buynow-price)

ListForAuction(auctionid,vin,userid)
Bids(bidid,bidderid, amount, timestamp)
PlaceBid(userid, auctionid, bidid)
Closed(auctionid,userid, bidid, closedate)
Transactions(transcationid,bidid, auctionid, transactionstate, transactiondate)

Delivered(userid, transactionid,delivereddate)
Returned(userid, transactionid,returneddate)

4.1.1.5 Review

An user can write reviews about an auction. A review includes a numbered ratings and a short text comments. A numbered rating is between 1 to 5, 1 for the worst auction and 5 for the best auction.

Reviews(reviewid, revewerid, sellerid, auctionid, reviewdate, review, rating)
Rate(userid, reviewid)

4.1.1.6 Community Aspect

This vehicle e-Auction System not only prive a platform for users/dealers to sell and buy vehicles. It also provides a social community for users to share vehicle purchasing experiences, exchange opinions about different vehicle performances, as well as let car-lovers to show off their antique cars or luxury cars, and let dealers to create car demonstration events.

DisplayFloor(displayid, userid, vin)
Media(mediaid, displayid, filetype, filename)
UploadsMedia(mediaid, userid)

BelongsTo(mediaid,displayid)
Events(eventid, eventdate, location, description)
CreateEvent(eventid, organizerid,)
AttendingEvent(userid, eventid)
Discussions(discussionid, discussion, timestamp)
PostDiscussion(discussionid, userid)

4.1.2 Schema Refinement via Key & Total participation Constraints Analysis

Based on the initial schema obtained in section 4.1.1, this subsection will analyze each entity set within each relationship set to identify possible key constraints and possible total participation constraints. The initial schema will be refined during constraints enforcement in following subsections.

4.1.2.1 Users

Since ownership of vehicles is a crucial aspect within this community and it is also important to identify which users can place bids and which can not, it is the most logical to use three different entity sets to capture the ISA structure for two different categories of users(dealership and individual registered users) while avoiding redundancy. Dealership and individual registered users inherit common attributes from Users. No constraints are for the users entity sets:

Users(userid, email, password, address, phonenumber)
Dealers(userid, dealershipname, revenue, delershipcategorey)
RegisteredUsers(userid, age, gender, income, insurancepolicyno,
insurancecompanycode, insurancecompanyname)

4.1.2.2 Bank Accounts

As mentioned in section 4.1.1.2, it is logical to assume that users need to use money from multiple bank accounts to buy expensive vehicles. Each bank account must be owned by exactly one user. Therefore, the bank accounts entity set has both key constraint and total participation constraint in the MoneyIn relationship. To enforce this key constraint and total participation constraint, the MoneyIn relationship is merged into the BankAccount entity set as:

BankAccount(<u>userid</u>, <u>accountno</u>, routingno, bankname)

4.1.2.3 Vehicles and Owns

Aforementioned, there is a third relation, owns, which is portrayed as a relationship in the ER-model. Logically, a vehicle must be owned by a user to be able to be sold. Thus there is an owns relation. However, since all vehicles must be owned by exactly one user, there is a key constraint as well as a total participation constraint on vehicles within the **Owns** relationship. The relation Owns(<u>vin</u>, userid)and the relation **Vehicles** separately will not capture these constraints. After identifying this flaw in the schema, a new relation is created by include the Owns relation into the Vehicles relation as:

Vehicles(<u>vin</u>, ownerid(userid),category, make, model, year, mileage, description)

4.1.2.4 Auctions, ListForAuction, and Closed

Each auction must be listed by an user, there is a total participation constraint for Auction in the ListForAuction relationship. To enforce this total participation constraint, the ListForActuion relation is included into the Auctions relation.

Furthermore, each auction must be closed either at the auction end date or at when a buy-now price is offered. Therefore, there is a total participation constraint for the Closed relation in the Closed-Auction relationship. To enforce this total participation constraint, the Closed relation is included into the Auctions relation.

The Auctions, ListForAuction, and Closed are merged together as:

Auctions(auctionid, vin, sellerid, startdate, enddate, reserveprice, buynowprice, closedate)

4.1.2.5 Bids and PlaceBid

As shown in Figure 3.3, each bid must be placed by one registered user. Therefore, Bids has total participation constraint in the PlaceBid relationship. To enforce this total participation constraint, PlaceBid is included into Bids relation as:

Bids(bidid, auctionid, bidderid, amount, timestamp)

4.1.2.6 Transactions, Delivered, and Returned

For when a vehicle is successfully delivered for completed auction or a vehicle is returned because of flaw during vehicle certification, a transaction record will be created in the database.

Transactions, Delivered, and Returned are separate relations as defined below:

Transactions(transcationid, bidid, auctionid, transactionstate, transactiondate)
Delivered(userid, transactionid, delivereddate)
Returned(userid, transactionid, returneddate)

4.1.2.7 Review

As shown in Figure 3.3, each review must be created/rated by an user. Therefore, there is a total participation constraint for Reviews in the Rate relationship. To enforce this total participation constraint, the Rate relation is merged into the Reviews as:

Reviews(reviewid, revewerid, sellerid, auctionid, reviewdate, review, rating)

4.1.2.8 Media, UploadsMedia, BelongsTo, and DisplayFloor

Each media file must be uploaded by an user. There is a total participation constraint for Media in the UploadsMedia relationship. To enforce this total participation constraint, UploadsMedia is merged into the Media relation as:

Media(mediaid,userid, displayid, filetype, filename)

Each media can be used for any number (zero or more than one) of DisplayFloor. There is no constraint for the BelongsTo relationship. The BelongsTo and DisplayFloor are defined as:

BelongsTo(mediaid,displayid)
DisplayFloor(displayid, userid, vin)

4.1.2.8 Events, CreateEvent, and AttendingEvent

As shown in Figure 3.3, each event must be organized/created by an user. There is a total participation constraint for Events in the CreateEvent relationship. To enforce this total participation constraint, the CreateEvent is merged into Events relation as:

Events(eventid, organizerid, eventdate, location, description)

An user can attend any number of (zero or more than one) events. There is no constraint for the AttendingEvent relationship.

AttendingEvent(userid, eventid)

4.1.2.9 Discussions and PostDiscussions

Each discussion about a vehicle is posted by a user. There is a total participation constraint for Discussions in the PostDiscussion relationship. To enforce this total participation constraint, the PostDiscussion is merged into Discussions as:

Discussions(discussionid, userid, discussion, timestamp)

4.2 Schema Normalization

Most redundancies have already been eliminated in section 4.1.2 "Schema Refinement via Key & Total participation Constraints Analysis" . This section will perform schema normalization to normalize all relations into 3NF to further eliminated functional dependencies if there is any.

4.2.1 Functional Dependency Analysis

This section will depict the schema in a more visual sense. In essence, this section describes the functional dependencies within each relation in a visual and/or accompanying verbal description. The goal for this round schema refinement is to achieve Boyce Codd Normal Form (BCNF) for all relations.

4.2.1.1 Users, Dealer, and RegisteredUsers

Users: There is no FD in Users relation. It is already in BCNF.

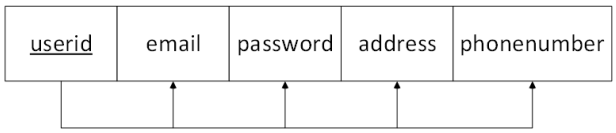


Figure 4.1: Users

Dealers: There is no FD in Dealers Relation. It is already in BCNF.

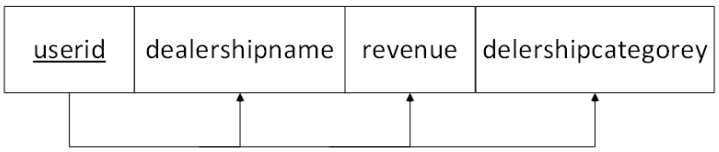


Figure 4.2: Dealers

RegisteredUsers: The RegisteredUser Relation is not in BCNF. It is only in 2NF.

Registered users can bid on auctions. When a registered user wins an auction, to protect the vehicle under transaction, the user's auto insurance policy must be verified.

Each auto insurance company is registered with an unique DMV code. The insurance company code can uniquely identify the insurance company name. Many registered users may

use the same insurance company. The insurance company code and company name may be repeated in the RegisteredUsers relation. Therefore, there is a functional dependency in ReqisteredUsers relation as FD: **insurancecompanycode → insurancecompanyname.**

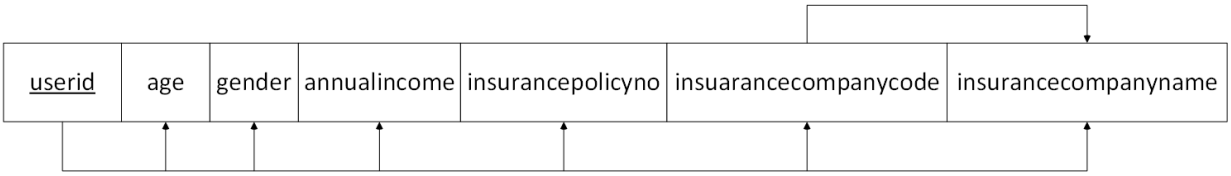


Figure 4.3: Registered Users

4.2.1.2 Bank Accounts

Bank Account: The Bank Account Relation is not in BCNF. It is only in 2NF.

Because a bank's routing number can uniquely identify the bank name, therefore in Bank Accounts relation, there is a functional dependency FD: **routingno → bankname**

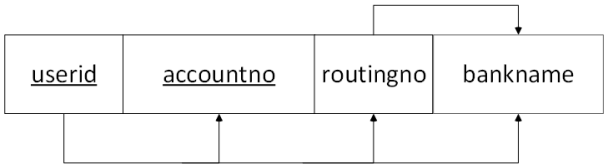


Figure 4.4: Bank Account

4.2.1.3 Vehicles

Vehicles: There is no FD in Vehicles Relation. It is already in BCNF.

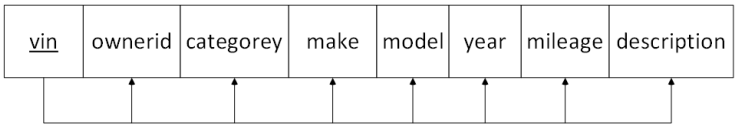


Figure 4.5: Vehicles

4.2.1.4 Auctioning

Auctions: There is no FD in Auctions Relation. It is already in BCNF.

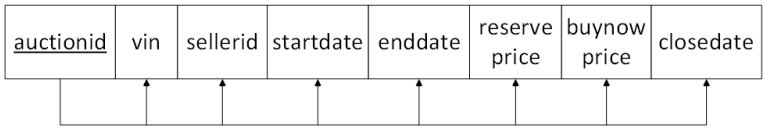


Figure 4.6: Auctions

Bids: There is no FD in Bids Relation. It is already in BCNF.

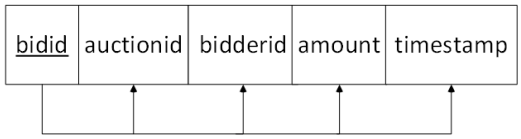


Figure 4.7: Bids

Transactions: There is no FD in Transactions Relation. It is already in BCNF.

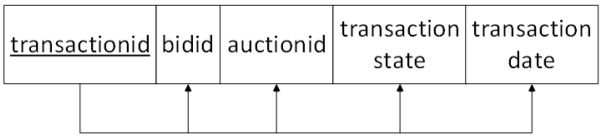


Figure 4.8: Transactions

Delivered: There is no FD in Delivered Relation. It is already in BCNF.

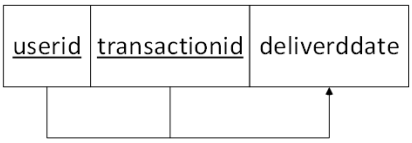


Figure 4.9: Delivered

Returned: There is no FD in Returned Relation. It is already in BCNF.

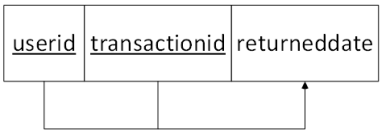


Figure 4.10: Returned

4.2.1.5 Reviews

Reviews: There is no FD in Reviews Relation. It is already in BCNF.

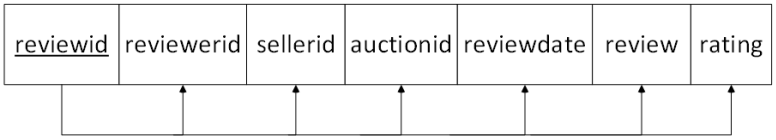


Figure 4.11: Reviews

4.2.1.6 Community

Media: There is no FD in Media Relation. It is already in BCNF.

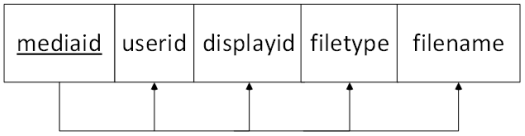


Figure 4.12: Media

BelongsTo: There is no FD in BelongsTo Relation. It is already in BCNF.

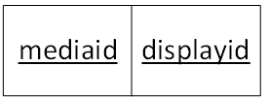


Figure 4.13: Belongs To

DisplayFloor: There is no FD in DisplyFloor Relation. It is already in BCNF.

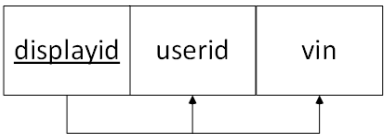


Figure 4.14: Display Floor

Events: There is no FD in Events Relation. It is already in BCNF.

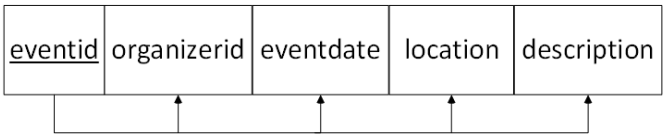


Figure 4.15: Events

AttendingEvents: There is no FD in AttnedingEvents Relation. It is already in BCNF.

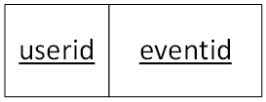


Figure 4.16: Attending Events

Discussions:There is no FD in Discussions Relation. It is already in BCNF.

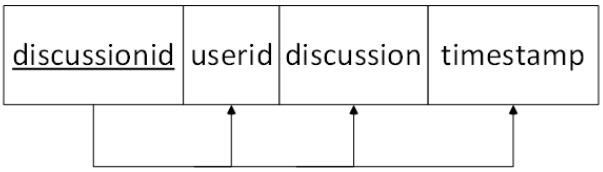


Figure 4.17: Discussions

4.2.2 Forms Normalization

This section will focus on normalizing all relations into BCNF. Normalization will be achieved through lossless-join decomposition. Lossless-join decomposition is the concept of removing redundancy in data while preserving the integrity and originality of the data safely. More specifically: let R be a relational schema and R_1 and R_2 be its decomposition. R is decomposed losslessly iff $R_{12} = R$.

Within the Vehicle e-Auction relational schema as analyzed in section 4.2.1, only two relations, ***RegisteredUsers*** and ***BankAccounts***, need to be normalized. The following section will illustrate the normalization process to transform the two 2NF relations into BCNF.

4.2.2.1 RegisteredUsers

Currently the only violation for RegisteredUsers to be in BCNF is the functional dependency between non-key attributes **inssurancecompanycode→ inssurancecompanyname**. In order to achieve BCNF, lossless-join decomposition must be applied to this relation. Let RegisteredUsers relation be denoted as R and its attributes as U (userid), G (gender), I (annualincome),N (insurancepolicyno), C (insurancecompanycode), and E (insurancecompanyname). As mentioned, the only functional dependency for R is $C \rightarrow E$. Decomposing R into R_1 and R_2 ,where $R_1 = (U, G, I, N, C)$ and $R_2 = (C, E)$, lossless-join decomposition is achieved. This decomposition is achieved with an introduction of a new relation InsCompany and revised R to become:

RegisteredUsers(userid, age, gender, income, insurancepolicyno, insurancecompanycode)

InsCompany(insurancecompanycode, insurancecompanyname)

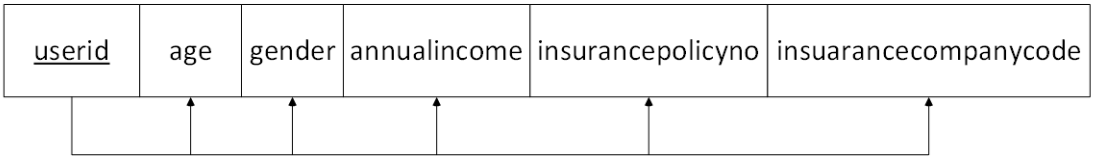


Figure 4.18: InsCompany Relation

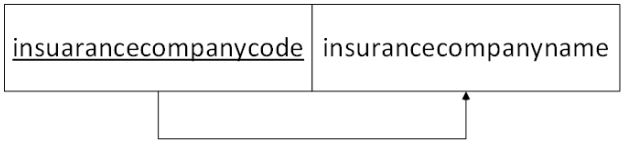


Figure 4.19: InsCompany Relation

This transformation makes relation RegisterdUsers into BCNF. Since there is no FD in relation InsCompany, it is also in BCNF.

4.2.2.1 Bank Accounts

Currently the only violation for BankAccounts to be in BCNF is the functional dependency between non-key attributes **routingno**→ **bankname**. In order to achieve BCNF lossless join decomposition must be applied to this relation. Let BankAccounts relation be denoted as B and its attributes as U(userid), A(accountno), $R($routingno), and N(bankname). As mentioned the only functional dependency for B is $R\rightarrow N$. Decomposing B into B_1 and B_2 ,where $B_1 = (U, A, R)$ and $B_2 = (R, N)$, lossless join decomposition is achieved. This decomposition is achieved with an introduction of a new relation Banks and revising B to become:

BankAccount(userid, accountno, routingno)
Banks(routingno, bankname)

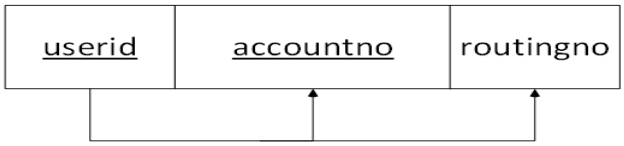


Figure 4.20: Bank Account in BCNF

Figure 4.21: Banks Relation

This transformation makes relation BankAccount into BCNF. Since there is no FD in relation Banks, it is also in BCNF.

5. *SQL Statements*

Following are the SQL statements to create the tables as defined in section 4.2.

```
CREATE TABLE Users (
        userid INT(10) NOT NULL AUTO_INCREMENT,
        email VARCHAR(100)NOT NULL,
        password VARCHAR(32)NOT NULL,
        address VARCHAR(100),
        phonenumber VARCHAR(16),
        PRIMARY KEY( userid),
        UNIQUE(email) );

CREATE TABLE Dealers (
        userid INT(10) NOT NULL,
        dealershipname VARCHAR(64) NOT NULL,
        personofcontact VARCHAR(64)NOT NULL,
        revenue real NOT NULL,
        dealershipcategorey VARCHAR(64),
        PRIMARY KEY( userid ),
        FOREIGN KEY(userid) REFERENCES users ON DELETE CASCADE);
CREATE TABLE RegisteredUsers (
        userid INT(10) NOT NULL,
        name VARCHAR(64) NOT NULL,
        age real,
        gender VARCHAR(1),
        income real NOT NULL,
        insurancepolicyno VARCHAR(16),
        insurancecompanycode VARCHAR(10),
        PRIMARY KEY(userid),
        FOREIGN KEY(userid) REFERENCES users ON DELETE CASCADE,
        FOREIGN KEY(insurancecompanycode) REFERENCES InsCompany ON DELETE
                                        SET TO NULL );
```

```
CREATE TABLE InsCompany (
        insurancecompanycode VARCHAR(10),
        insurancecompanyname VARCHAR(64) NOT NULL,
        PRIMARY KEY(insurancecompanycode) );
CREATE TABLE BankAccount(
        userid INT(10),
        accountno INT(10),
        routingno INT(10) NOT NULL,
        PRIMARY KEY(userid,accountno),
        FOREIGN KEY(userid) REFERENCES Users ON DELETE CASCADE,
        FOREIGN KEY(routingno) REFERENCES Banks ON DELETE CASCADE );
        CREATE TABLE Banks (
        routingno INT(10),
        bankname VARCHAR(64),
        PRIMARY KEY(routingno) );

CREATE TABLE vehicles (
        vin CHAR(17),
        ownerid INT(10),
        category VARCHAR(64),
        make VARCHAR(64) NOT NULL,
        model VARCHAR(64) NOT NULL,
        year CHAR(4) NOT NULL,
        mileage INT DEFAULT VALUE 0,
        description VARCHAR(255),
        PRIMARY KEY( vin ),
        FOREIGN KEY(ownerid) REFERENCES Users(userid) ON DELETE CASCADE);
CREATE TABLE Auctions(
        auctionid INT(10) AUTO_INCREMENT,
        vin CHAR(17) NOT NULL,
        sellerid INT(10),
        startdate DATE,
        enddate DATE NOT NULL,
        reserveprice REAL NOT NULL,
        buynowprice REAL NOT NULL,
        closedate DATE,
        PRIMARY KEY(auctionid),
        FOREIGN KEY(vin) REFERENCES Vehicles ON DELETE CASCADE,
        FOREIGN KEY(sellerid) REFERENCES users(userid) ON DELETE NO CASCADE );

CREATE TABLE Bids(
       bidid INT(10) AUTO_INCREMENT,
       auctionid INT(10) NOT NULL,
       bidderid INT(10) NOT NULL,
       amount REAL NOT NULL,
```

```
    timestamp TIMESTAMP NOT NULL,
    PRIMARY KEY(bidid),
    FOREIGN KEY(bidderid) REFERENCES Users(userid) ON DELETE CASCADE,
    FOREIGN KEY(auctionid) REFERENCES Auctions ON DELETE CASCADE );
CREATE TABLE Transactions{
    transactionid INT(10) AUTO_INCREMENT,
    bidid INT(10) NOT NULL,
    auctionid INT(10) NOT NULL,
    transactionstate VARCHAR(255) NOT NULL,
    transactiondate DATE NOT NULL,
    PRIMARY KEY (transactionid),
    FOREIGN KEY (bidid) REFERENCES Bids ON DELETE CASCADE,
    FOREIGN KEY (auctionid) REFERENCES Auctions ON DELETE CASCADE );
CREATE TABLE Delivered(
    userid INT(10),
    transactionid INT(10),
    deliveredDate DATE NOT NULL,
    PRIMARY KEY (userid, transactionid),
    FOREIGN KEY (userid) REFERENCES Users ON DELETE CASCADE
    FOREIGN KEY (transactionid) REFERENCES Transactions ON DELETE CASCADE);


CREATE TABLE Returned(
    userid INT(10),
    transactionid INT(10),
    returnedDate date NOT NULL,
    PRIMARY KEY (userid, transactionid),
    FOREIGN KEY (userid) REFERENCES Users ON DELETE CASCADE
    FOREIGN KEY (transactionid) REFERENCES Transactions ON DELETE CASCADE);


CREATE TABLE Reviews (
    reviewid INT(10) NOT NULL AUTO_INCREMENT,
    reviewerid INT(10),
    sellerid INT(10),
    auctionid INT(10),
    reviewdate date,
    review VARCHAR(255),
    rating INT,
    PRIMARY KEY( reviewid ),
    FOREIGN KEY(reverwrid) REFERENCES Users(userid) ON DELETE NO ACTION,
    FOREIGN KEY(sellerid) REFERENCES Users(userid) ON DELETE CASCADE,
    FOREIGN KEY(auctionid) REFERENCES Auctions ON DELETE NO ACTION);
CREATE TABLE Media (
    mediaid INT(10) AUTO_INCREMENT,
```

```
    userid INT(10),
    displayid INT(10),
    filetype VARCHAR(16),
    filename VARCHAR(255)
    PRIMARY KEY mediaid,
    FOREIGN KEY userid REFERENCES Users ON DELETE CASCADE,
    FOREIGN KEY displayid REFERENCES DisplayFloor ON DELETE NO ACTION);

CREATE TABLE BelongsTo (
    mediaid INT(10),
    displayid INT(10),
    PRIMARY KEY (mediaid, displayid),
    FOREIGN KEY mediaid REFERENCES Media ON DELETE CASCADE,
    FOREIGN KEY displayid REFERENCES Displayfloor ON DELETE CASCADE);
CREATE TABLE DisplayFloor(
    displayid INT(10) AUTO_INCREMENT,
    userid INT(10) NOT NULL,
    vin CHAR(17) NOT NULL,
    PRIMARY KEY(displayid),
    UNIQUE(vin),
    FOREIGN KEY(vin) REFERENCES Vehicles ON DELETE CASCADE,
    FOREIGN KEY(userid) REFERENCES Users ON DELETE CASCADE );
CREATE TABLE Events(
    eventid INT(10) AUTO_INCREMENT,
    organizerid INT(10) NOT NULL,
    eventdate date NOT NULL,
    location VARCHAR(255) NOT NULL,
    description CHAR(255)NOT NULL,
    PRIMARY KEY(eventid),
    FOREIGN KEY organizerid REFERENCES Users(userid) ON DELETE CASCADE);
CREATE TABLE AttendingEvent(
    userid INT(10),
    eventid INT(10),
    PRIMARY KEY(userid, eventid),
    FOREIGN KEY(userid) REFERENCES Users ON DELETE CASCADE
    FOREIGN KEY(eventid) REFERENCES Events ON DELETE CASCADE );
CREATE TABLE Discussions {
    discussionid INT(10) AUTO_INCREMENT,
    vin VARCHAR(17) NOT NULL,
    userid INT(10) NOT NULL,
    discussion VARCHAR(255) NOT NULL,
    timestamp TIMESTAMP NOT NULL,
    PRIMARY KEY (discussionid),
    FOREIGN KEY (vin) REFERENCES Vehicles ON DELETE CASCADE,
    FOREIGN KEY (userid) REFERENCES Users ON DELETE CASCADE);
```

6. *Technologies selected for implementation*

6.1 Major implementation tools analysis

Firebase

- **Nosql**, realtime database
- Queries can instantly be sent to and from various devices without page refreshes
- Google have aggressively marked Firebase as being an entire ecosystem for building web and mobile apps

Pros

- Instant data updates without refreshing the webpage
- Easy to synchronize multiple computers with your database
- No need to worry about your server going into meltdown if you suddenly get tonnes of traffic
- Comes with other cool features such as firebase storage, google cloud functions, google auth and more

Cons

- Shortage of bookings and training resources
- Esoteric security protocols
- Pricing/bandwidth concerns
- Seriously limited queries

MongoBD

- Open source Nosql database
- Built with an emphasis on scalability and consistency

Pros

- Flexible database structure
- Can store lots of data inexpensively
- More secure than MySQL and firebase
- Exceptional performance on high traffic apps with lots of simultaneous write statements

Cons

- Query syntax looks more complicated than MySQL or even firebase
- Steep learning curve for developers who manage very high traffic websites

- Complex queries are very difficult
- Third party apps like mongoose make MongoDB more pointless

MySQL

1) Open source, relational sql database
2) Traditionally used with php but now used with lots of other web technologies
3) Free with most web hosting packages

Pros

1) Easy to learn
2) Can handle surprisingly large amounts of data
3) Good at handling complicated queries
4) Stable
5) A good choice for companies who have traditionally used spreadsheets to store their data
6) Very good third party software in available, such as phpmyadmin, navicat

Cons

1) Potential image/branding challenge
2) Constant threat of sql injection attacks

3) Potential question marks over how mysql performs on very high traffic websites

	MySql	MongoDB	NoSQL
Open source	Yes	Yes	Yes
ACID transactions	Yes	Yes	No
Flexible, Rich data model	No	Yes	Partial: Schema flexibility but support for only simple data structure
Schema governance	Yes	Yes	No
Expressive joins, faceted search, graphs queries, powerful aggressions	Yes	Yes	No
Enterprise grade security and mature management tolls	Yes	Yes	No
Database as a service on all major clouds	Yes	Yes	No

6.2 Major Tools Selected

Based on the analysis outlined in section 6.1, we decided to use DjangoDB as our Web framework and use MySQL as back-end database. The reason why we choose DjangoDB as our Web framework because all of our group members have in-depth programming expertise with Python, secondly because Django is a high-level Python framework which supports rapid development and clean, pragmatic design. Django have debugging tools with python packages and the best package management which allow developers easily use it in other application. And the reason that we choose MySQL as our back-end database is that MySQL fully support SQL so that all the SQL knowledge and relational database theory which we learned from CMPSC 431W can be intensively utilized and exercised during this Vehicle e-Auction project development.

7. *SQL for Populating Database*

The following MySQL script code is used to populate testing data into the Vehicle e-Auction database. This script creates:

- 10 users (2 dealers, 8 registered users)
- 10 vehicles to be auctioned.
- 3 vehicles are listed for auction.
- The first auction received buy-now price bid. The auction is closed/completed, but not delivered yet.
- The second auction received 1 bid.
- The third auction has not received any bid yet.
- 2 users have commented/reviewed the first auction.

```
INSERT INTO Users
VALUES ('a1@psu.edu', '431', '229 sparks', '123-457-7890'),
    ('a2@psu.edu', '432', 'penn state', '211-222-3333'),
    ('a3@psu.edu', '433', 'penn state', '311-222-3333'),
    ('a4@psu.edu', '434', 'penn state', '411-222-3333'),
    ('a5@psu.edu', '435', 'penn state', '511-222-3333'),
    ('a6@psu.edu', '436', 'penn state', '611-222-3333'),
    ('a7@psu.edu', '437', 'penn state', '711-222-3333'),
```

```
    ('a8@psu.edu', '438', 'penn state', '811-222-3333'),
    ('a9@psu.edu', '439', 'penn state', '911-222-3333'),
    ('a0@psu.edu', '430', 'penn state', '011-222-3333');

INSERT INTO Dealers
VALUES (1, 'My BMW', 'W WANG', 99999999, '4S'),
    (2, 'DCH',   'John', 10000000, '3c');


INSERT INTO RegisteredUsers
VALUES (3, 'William W', 20, 'M', 500000, 'MTL88182', '518'),
    (4, 'John K', 35, 'M', 110000, NULL, NULL),
    (5, 'Jane H', 26, 'F', 130000, 'AST31-23', '617'),
    (6, 'Ken L',  31, 'M', 90000, NULL, NULL),
    (7, 'Smith F', 43, 'M', 150000, '74HL64673', '518'),
    (8, 'Jenny Z', 18, 'F', 70000, NULL, NULL),
    (9, 'Tony X', 39, 'M', 150000, '316K4566-1','617');

INSERT INTO InsCompany
VALUES ('518', 'MetLife'),
    ('617', 'All State');

INSERT INTO BankAccount
VALUES (3, 368012385618, 200041201),
    (4, 468012385619, 200041201),
    (5, 568012385618, 200041201),
    (6, 668012385618, 300041201),
    (7, 768012385618, 200041201),
    (8, 868012385618, 300041201),
    (9, 968012385618, 300041201),
    (10, 168012385618,200041201);

INSERT INTO Banks
VALUES (200041201, 'Bank of America'),
    (300041201, 'PNC');

INSERT INTO Vehicles
VALUES ('BMW238I16856831', 3, 'Sedan', 'BMW', '328i','2016', 15000, 'Love it'),
    ('BMW168X36856831', 3, 'SUV', 'BMW', 'X3', '2017', 9800, 'Best'),
    ('H238I16856831', 5, 'Truck', 'Honda', 'Ti','2008', 75000, 'Fair'),
    ('K238I16856831',   6, 'Van', 'Kia', 'Rio', '2016', 15000, 'Love it'),
    ('F538I16856831', 1, 'Sport', 'Ford', 'Fc', '2018', 0, 'New'),
    ('D238I16856832', 2, 'Pickup', 'Hadai', 'Jk', '2017', 5000, 'Good'),
    ('K238I16856833', 2, 'Bus', 'Kia', 'Tr', '2018', 1000, 'Loaner'),
    ('J238I16856834', 1, 'Tractor', 'JD', 'F16', '2010', 85000, 'Ok'),
```

```
    ('F238I16856835', 7, 'Antique', 'Ford', 'MS', '1948', 125000, 'Fair'),
    ('G238I16856836', 2, 'MiniVan', 'GMC', 'G130', '2018', 41, 'New');

INSERT INTO Auctions
VALUES ('BMW238I16856831', 3, 2/20/18, 2/28/18, 51000, 55000, 2/21/18),
    ('K238I16856832', 2, 2/21/18, 3/20/18, 15000, 17000, NULL),
    ('F538I16856831', 1, 2/25/18, 3/28/18, 25000, 27000, NULL);

INSERT INTO Bids
VALUES (1, 5, 55000, 10:18:26-2/21/18),
    (2, 7, 11000, 09:28:16-2/23/18);

INSERT INTO Reviews
VALUES (5, 3, 1, 2/21/18, 'Best Auction', 5),
    (8, 3, 1, 2/22/18, 'Nice car', 5);
```

8. *Conclusion*

Following Phase II, the team will begin implementing the design. The team will implement full features as described in Section 2. The implementation will serve as a proof of concept. A full framework involving the front-end design as well as the back-end will be in place to demonstrate Generic Team Name's unique and innovative design. With the completion of the framework, the team will strive to present our concept to the committee for selection.

Please note that during the implementation phase, some refinements will still be made. It is inherently impossible to predict all outcomes which arises during the next rigorous refinement phase. Even though the schema may not condense, it may increase in size due to the need for data collection and analysis.

Appendix A: Progress Reports

Group Progress Reports

Group: Generic Team Name **Reporting Period:** 3/1/2018

Group Meetings	**Attendance**	**Date/Time/Location**	**Duration**
Discussed Phase II Status	**Present:** Yuchen William JingRui Rob **Missing** Mike	2/25 8:00pm Discord VOIP	.5 Hour
Began discussing Phase II	Present: Yuchen William JingRui Rob Mike	2/8 7:30pm 204W Westgate	2 Hour
Group Total Hours			2.5

Group: Generic Team Name **Reporting Period:** 2/4/2018

report

Group Meetings	**Attendance**	**Date/Time/Location**	**Duration**
Discussed the breakdown of the presentation and worked on the slides supporting it.	**Present:** William Yuchen Rob **Missing:** JingRui Mike	2/4, 12:30pm W204 Westgate Building	2 hrs
The writing of the project report. Yuchen came to help William write the project report. Yuchen was tasked to work on the Introduction.	**Present:** William Yuchen Missing: JingRui Rob Mike	2/3 , 10:00pm W204 Westage Building	2.5 hours
Determine the exactly what entities were needed.	**Present:** Yuchen William JingRui Rob Mike	2/1, 7:30pm W204 Westgate Building	2 hours
Group Total Hours			6.5

Group: Generic Team Name **Reporting Period:** 1/28/2018

Group Meetings	**Attendance**	**Date/Time/Location**	**Duration**
Began discussing the conceptual diagram	**Present:** Yuchen William **Missing:** JingRui Rob Mike	1/28, 3:30pm Pattee Library	2 hour
Group Total Hours			2

Group: Generic Team Name **Reporting Period:** 1/21/2018

Group Meetings	**Attendance**	**Date/Time/Location**	**Duration**
Initial team meeting. Discussed the requirements for the project including which additional features we plan to add. Decided on cars as a topic for our site, and a few of the technologies we plan to use (Bootstrap, MySQL, Python, PHP). Set up a Github for our project.	**Present:** Rob Mike Yuchen William **Missing:** JingRui	1/20, 3:30pm Discord	1 hour
Group Total Hours			1

Signature:

Michael Berezanich

William Wang

Robert Beck

Yuchen Zeng

Individual Progress Reports

William Wang

Date	Hours	Activities
3/1 4:00pm	6	Work on final formatting of the paper. Print out and submission.
2/28 3:00pm	8	Translated all of the normalized schema into SQL statements. Completed Section 5. Created SQL data population scripts for sample data. Completed Section 7.
2/27 3:30pm	10	Started Working on Section 4.2 of the Phase II report. During the Functional Dependency stage, realized that all of the relations in the schema were already in BCNF. Created new attributes to entities to create functional dependencies to meet requirements. Changed Section 3 to incorporate these changes. Finished all of Section 4.
2/26 6:00pm	5	Wrote Section 4.1 of the Phase II report by using newly revised ER Model to create the schema
2/25 5:00pm	2	Incorporated the tables for Section 3 as well as finalized the whole ER model and breakdowns for relationships
2/24 12:00pm	4	Finalized the revision of Section 3

continued on next page

continued from previous page

2/21 5:00pm	3	Started Editing Section 3
2/19 3:00pm	2	Started to reorganize the ER Model
2/4 9:00pm	3	Wrote Appendix C and finalized the report through formatting
2/4 2:30pm	5	Added more subsections to Section 2: Unique features. Refined Section 3 with needed entities, and relationships. Reworked the ER-model Diagram. Drew the Category tree.
2/4 12:30pm	2	Worked with Rob, and Yuchen on the presentation formatting and slide.
2/3 10:30pm	2	Worked with Yuchen on the Introduction. Helped him edit his introduction which properly expresses his thoughts in English.
2/3 4:30pm	6	Worked on Sections 2 and 3 of the Project Report. Drew the preliminary ER model as a first draft.
2/1 7:30pm	2	Determine exactly what entities were needed. With the whole group.
1/28, 3:30pm	2	Began discussing the conceptual diagram with Yuchen

Yuchen Zeng

Date	Hours	Activities
3/1, 4pm	1	Complete the conclusion of section 6
2/27, 8pm	2	Working on the technologies selected form implement in phase 2
2/4, 11pm	1	Format Phase 1 Report Final Draft
2/4, 12:30pm	3.5	Making powerpoint and prepared presentation
2/3, 10pm	2.5	Finish introduction part of Phrase 1 report
2/1, 7:30pm	2	Determine exactly what entities were needed. With the whole group.
1/28, 3pm	2	Discussed and drew ER diagram with William in Pattee

Robert Beck

Date	Hours	Activities
3/1 4:00pm	3	Looked over the report for errors. Formated the flow between each section.
2/26 8:00pm	2	Looked into potential software/web tools for implementation
2/23 5:00pm	2	Suggested better logical flow of the outline for the report. Made sure that Phase II tasks were met within the report.
2/11, 3:30pm	2	Researched and experimented with phpmyadmin for generating SQL statements
2/4, 8:30pm	1	Proofread progress report and reviewed presentation
2/4, 12:00pm	2	Created slides for section 2 of report for presentation

Michael Berezanich

Date	**Hours**	**Activities**
2/26, 6:00pm	2	Helped a bit with the sql commands by exporting the sql commands to generate the database set up in phpmyadmin. Generated the Relational Schema diagrams for easier optimization. Helped William with Optimization.
2/16, 3:00pm	6	Set up an initial mysql database covering the basic operations of our schema. Revised some of the entities in our model to support more in the future (multiple addresses, phone numbers). Hosted Server for the rest of the group to practice SQL and PHP.
2/11, 12:00pm	2	Set up a new database on my phpmyadmin server to demo the options available on phpmyadmin to the team in our subsequent meeting
2/4, 7:30pm	1	Wrote the conclusion and fixed some formatting for the progress report.
1/20, 2:30pm	1	Reviewed project details prior to meeting to plan discussion points and possible ideas. Set up a github repository for the project.

Jingrui Duan

Date	Hours	Activities
2/8, 9:30 pm	1	Conceptual Database Design
2/10 6:00 pm	1	Schematic diagram of relationships
2/17 3:00 pm	2	Label model (Event, discussions)
2/18 9:00 pm	3	Label model (display floor, media) Modifying the diagram
2/22 7:00 pm	2	Review PHP and SQL Hosted Server
2/25 10:00 pm	2	Looking for the Network tool that can be used to optimize project

Appendix B: Project Plan

Project Plan (Schedule, deliverables/milestone)

Week of	**Items**	**Deliverables / Milestones**
Week 2 01/15/18	Initial team meeting: discuss project requirements identify additional features to implement	Meeting minutes
Week 3 01/22/28	Requirement Analysis Conceptual Database Design	Drafted requirement analysis and conceptual design
Week 4 01/29/18	Finalize the conceptual database design Write Phase I report Prepare presentation slides	Final version of Phase I report Presentation slides
Week 5 02/05/18	Define database schema Perform normalization	Database schema
Week 6 02/12/18	Write SQL statements to create all the relations/tables/views	MySQL code

Week 7 02/19/18	Create MySQL scripts to populate test/demo data Write Phase II report	MySQL code and e-Auction database created in MySQL
Week 8 02/26/18	Finalize Phase II report Prepare Phase II Presentation slides	Final version of Phase II report Phase II presentation
Week 9 03/05/18	Develop web app GUI and Python code to implement: 1) Register User 2) Add Auction Item 3) Browse Items	Implemented Web Application including required back-end database operations
Week 10 03/12/18	Develop web app GUI and Python code to implement: • Search items • Bid Items • Post Reviews	Implemented Web Application including required back-end database operations

Week 11 03/19/18	Develop web app GUI and Python code to implement: 1) close/terminate action 2) Generate statistics and marketing report 3) Post Reviews	Implemented Web Application including required back-end database operations
Week 12 03/26/18	Develop web app GUI and Python code to implement: 1) Display floor 2) Event broadcast 3) Event RSVP	Implemented Web Application including required back-end database operations
Week 13 04/02/18	Prepare 4th progress report Debug and Integration test	4th Progress report Complete Vehicle e-Auction System
Week 14 04/09/18	Reserved week in case anything takes longer than estimated	TDB

Week 15 04/16/18	Write project final report	Draft of project final report
Week 16 04/23/18	Finalize project final report Prepare project final presentation and demonstration	Project final report Complete Vehicle e-Auction System