

Butte 4H Ski Swap Database Project

Developed for Joe Casagrande of the Butte 4H Club

Developed by Scott Fleener, Joy Reistad, and Luke Schuler

Butte 4H Ski Swap Database Description

Scott Fleener

Joy Reistad

Luke Schuler

Database Title: Ski Swap Database

This application is being developed for the Butte 4H club Ski Swap event. Scott Fleener, Joy Reistad, and Luke Schuler will be developing this application. Our mission statement is: "Replace the prototype Ski Swap database with one that is more efficient and effective." The list of tasks for our project are as follows:

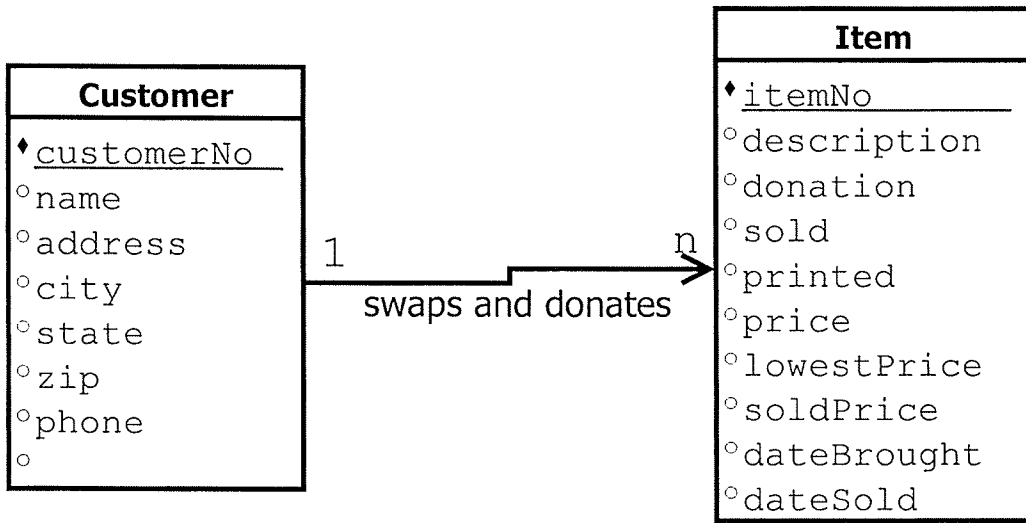
- Track of contact information of donating and swapping customers
- Track inventory of items along with the tags and sales of those items
- Be able to remove items from database (will be replaced with NULL values so there are no sequence breaks in the database)
- Keep archival data of previous years' events
- Generate a donation report and customer receipts

Objectives:

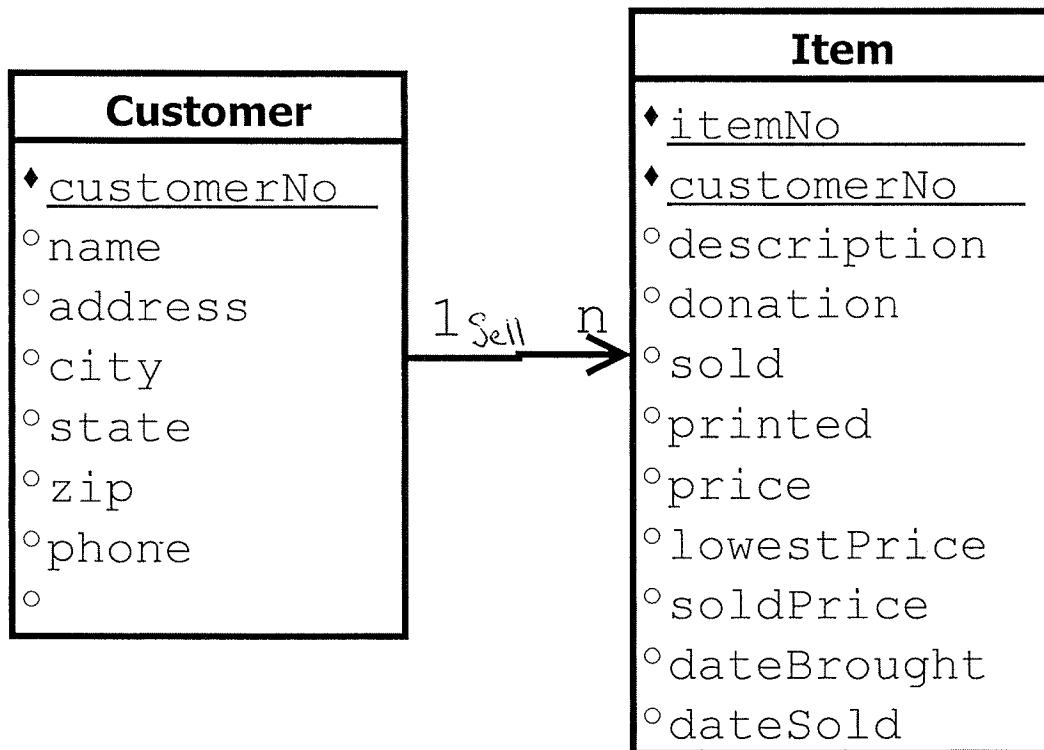
- Improve existing design
- Make database very simple to use and setup (usable by 20-year-old 4H members)
- Keep inventory of participating customers and their items being sold
- Database must be multi-user
- Allow tags to be printed only when they are needed, as opposed to current system
- Have item numbering be on a per-customer basis as opposed to a global basis
- Allow creation of donation report
- Allow integration of the previous year's records

Our system will have exactly one type of user. This user is the members of the Butte 4H club. While these users will vary in age and computer ability, they still require only one way to access our database. So, our system requires no effective boundaries between different types of users. Our primary user view will be simplistic and easy to navigate due to the varying levels of computer literacy between users. It should remain similar to what the current database looks like, albeit with the improvements outlined above implemented.

Conceptual Model



Logical Model



User Interface & Tables

Manage Items

Record: 1 of 1

Customer

Description:
 Asking price:
 Lowest price:
 Donation:

Record: 1 of 1

2014 customer

Name:
 Address:
 City:
 State:
 ZIP Code:
 Phone (no. area):

 Record: 1 of 1

Customer

Name:
 Address:
 City:
 State:
 ZIP Code:
 Phone (no. area):

Record: 1 of 1

Record: 1 of 1

Manage Records

Name: Description
 Address: Price
 City: Lowest price
 State: Sale price
 Zip: Donation
 Phone:

Record: 1 of 1

2014 Customer

customerNo	name	address	city	state	zip	phone
1	Laine Bellange	105 Crying Geo	Butte	Montana	59701	190-8763
2	David Berkheir	55 Milky Way	Butte	Montana	59701	299-2257
3	John Coulthard	805 W. Granite	Butte	Montana	59701	4393452

Record: 1 of 3

Items

itemNo	customerNo	description	donation	sold	printed	price	lowestPrice	soldPrice	dateBrought	dateSold
1	1	Solomon Skis	0	1	1	20	20	20	3/29/2014	3/29/2014
2	1	Scott Poles	0	1	1	5	5	5	6/1/2013	3/31/2014
3	1	Solomon Boots	1	1	1	5	5	5	6/1/2013	3/29/2014
4	1	Technica Boots	1	0	1	15	15	15	6/1/2013	
1	2	Fischer 147 CC Skis	0	1	1	5	5	5	3/29/2014	3/29/2014
2	2	Rossingol 160 XC Skis	1	0	1	30	30	30	3/29/2014	
3	2	Solomon Boots	1	1	1	5	5	5	6/1/2013	6/1/2013
1	3	Atomic Nomad Skis(red)	0	1	1	150	150	150	4/8/2014	4/8/2014
2	3	Phoenix Ski Pants	0	1	1	30	30	30	4/8/2014	4/8/2014
3	3	Discovery Vest (black)	1	0	1	5	5	0	4/8/2014	4/8/2014

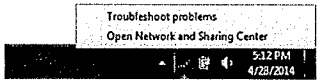
Record: 1 of 10

Tab - Set-up Tutorial

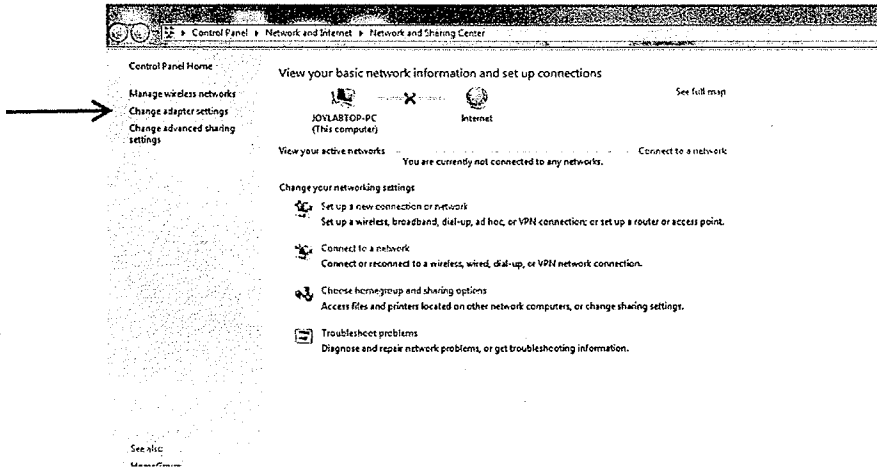
4H Ski Swap database setup tutorial

Super Cool!
Very helpful

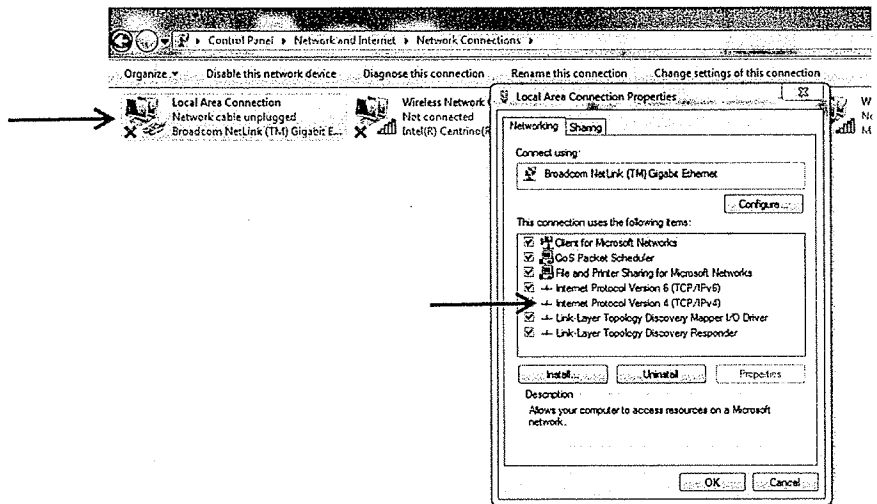
- 1) Connect each one of the computers to a network switch using an Ethernet cable.
- 2) Put the server package onto the computer that will be running the server, and the client package onto the other computers that will need to use the database.
- 3) On all of the computers:
 - a. The LAN IP addresses must be set to static in order for the computers to be able to connect to each other. Right click on the internet icon in the tray, in the lower right of the taskbar and click "Open sharing and network center".



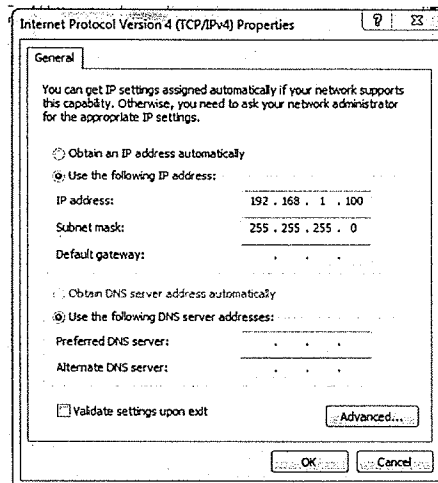
- b. On the top-left, click on "Change adapter settings". Right click on the "Local Area



Connection" and click "Properties". Double-click on "Internet Protocol Version 4 (TCP/IPv4)".

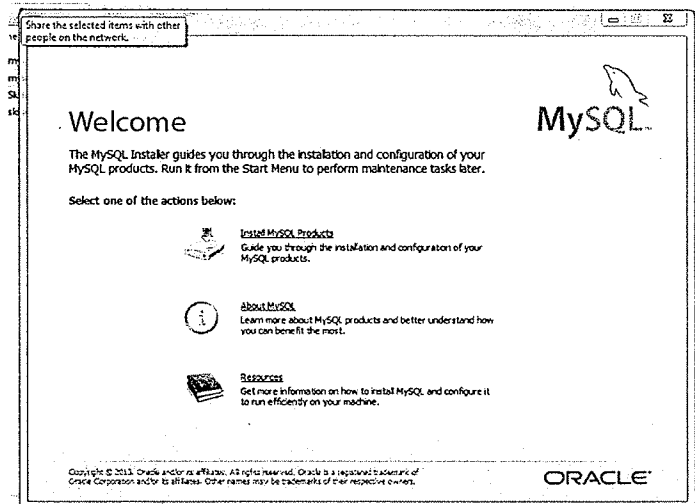


- c. Click on the "User the following IP address:" button and in the "IP address" field, enter in an IP address. For the server computer, use "192.168.1.100". For the client computers, use "192.168.1.101", "192.168.1.102" and so on if more computers are being used. Click once on the "Subnet mask" field and it should auto-fill with default values. Click the OK button on the windows, and close the "Network and sharing center" window.



4) On the server computer:

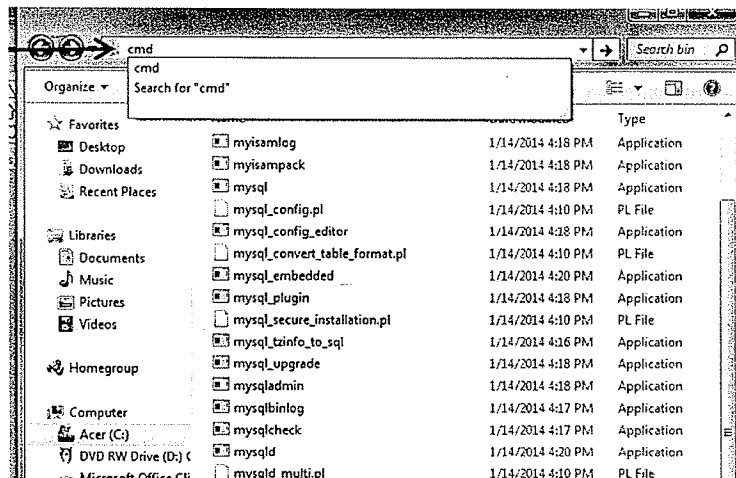
- a. Double-click on the "mysql-installer-community-5.6.13.0.msi" installer and click run and then yes on the windows that pop up. After a bit, you should see the following screen appear:



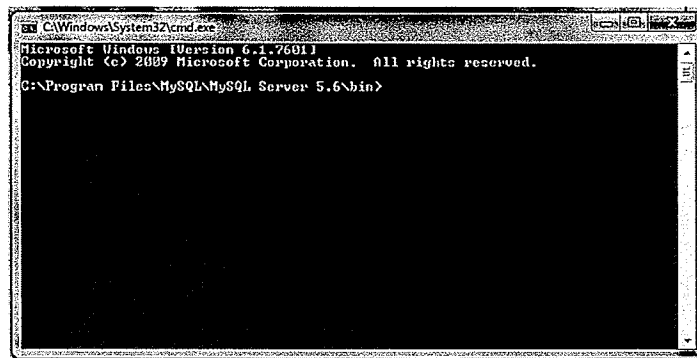
- b. Click the install button, then on the "I accept the license terms" checkbox. Then click Next. Check the skip the check for updates checkbox on the next screen and click next.
- c. Click the "custom" selection and click the next button. Unselect the "Documentation", "Connectors", and "Applications" checkboxes.
- d. Click the next button twice, then the execute button. Once the installation is complete, click the next button again three times. On the next screen, leave the "Current root password" field blank, and in the next two fields, type in a suitable root password for

the server. REMEMBER THIS PASSWORD. THIS PASSWORD WILL ALLOW ALL OF THE COMPUTERS TO CONNECT TO THE MySQL SERVER. Click next twice once a password is typed in.

- e. After the installer finishes configuring the server, click the next button, then the finish button on the next screen.
- f. Now that the MySQL server is installed, we have to put the database onto the server. To do this, open the start menu and click "Computer". Then, double click on the files in this order: C:, Program Files, MySQL, MySQL Server 5.6, bin. Then, copy the "skiswapsave.sql" file from the server package into this folder.
- g. In the same explorer window, click on the address bar and type in "cmd" and hit enter. This is where you should be typing:



This is the window that should pop up:



In this window, type this series of commands exactly as they appear here without the quotes (hit enter after each one):

"mysql -u root -p"

(enter your password and hit enter)

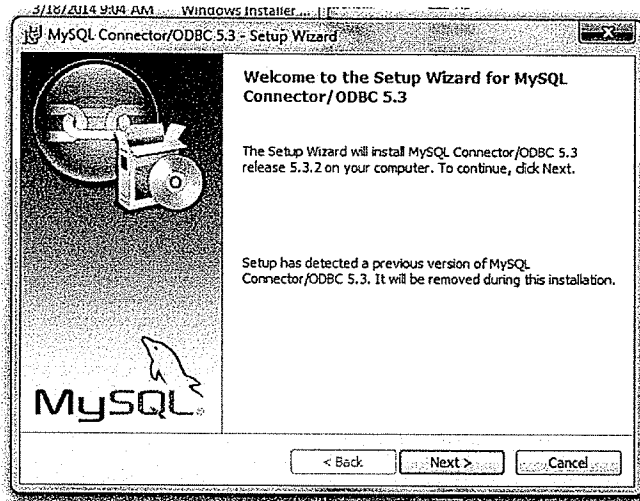
"create database skiswap;"

"quit"

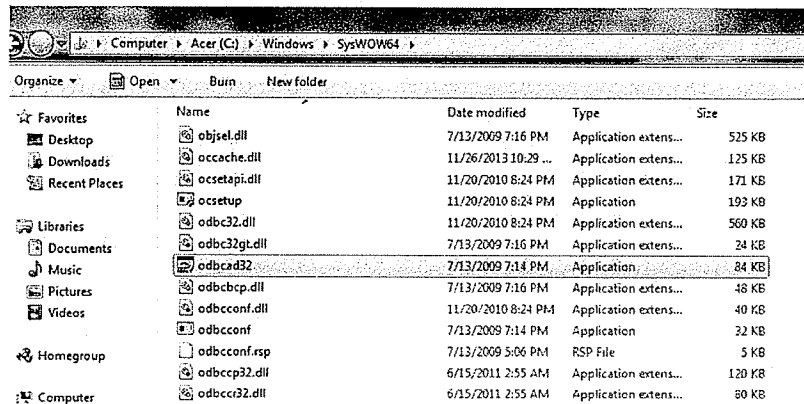
"mysql skiswap -u root -p < skiswapsave.sql"

(enter your password and hit enter)

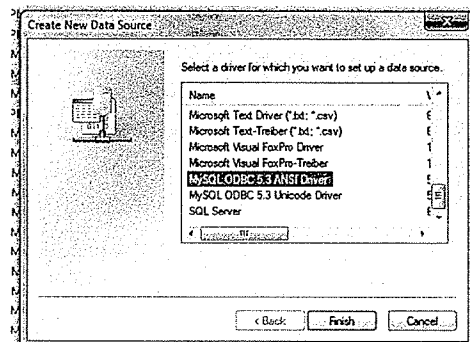
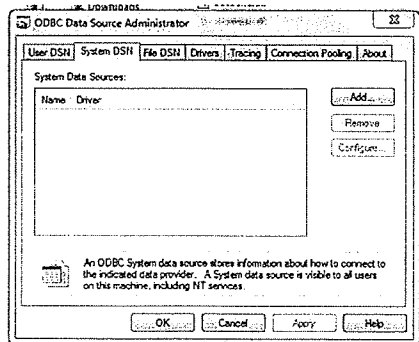
- h. Now that we have the data for the database set up, we need to set up the connector so Microsoft Access can access this data. In the server package, double-click on the “mysql-connector-odbc-5.3.2-win32” installer. This window should appear:



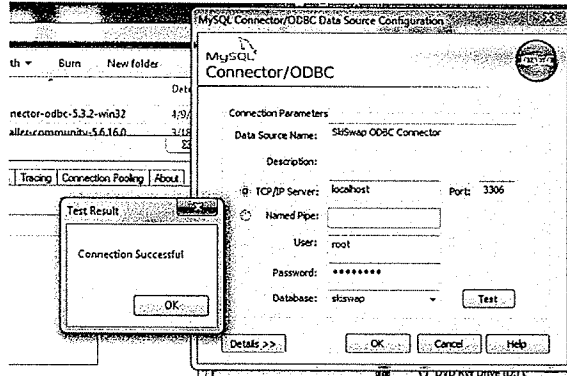
- i. Click the next button, then “I accept the terms in the license agreement” and then next twice. Click the install button, wait for the installation to complete (click yes on the window if it pops up), then click finish.
- j. In your explorer window, navigate to the “C:\Windows\SysWOW64” directory and open the program called “odbcad32.exe” (it may appear as “odbcad32” in the explorer window).



- k. On the windows that opens, click on the “System DSN” tab, then click the “Add...” button. On the windows that pops up, scroll down to the “MySQL ODBC 5.3 ANSI Driver”. Click on that, and then the finish button.



- I. On the window that pops up, enter this information into the following fields:
Data Source Name – SkiSwap ODBC Connector
TCP/IP Server – localhost
User – root
Password – (your password)
Database – skiswap



- m. Click the test button to ensure that the database can be connected to, then click OK on all windows. Now you can open the Microsoft Access file (SkiSwap.accdb) that is included in the sever package.
- 5) On the client computers:
 - a. Repeat steps h-m for the server computer instructions, with one small change to step I. On the "TCP/IP Server" field, instead of "localhost", type in "192.168.1.100"
- 6) Now, all of the computers should be able to access the database on the server computer. Open up the SkiSwap.accdb file included with the respective packages for each one of the computers. Double click on the "Main Menu" button under forms to begin using the database.

Tab - Ski Swap
Info

Butte 4H Ski Swap Database Strategy Sheet

Ver 1
1/2/14

Goal:

Replace the "prototype" Ski Swap database with a more efficient and effective database. The prototype was merely developed to show proof of concept but ended up being used in 2013.

Background:

Each year, the Butte 4H sponsors a ski swap where the community brings items to sell (or donate to 4H to sell). The event only last a few hours each year:

• **7:30am-8:30am -- setup swap area**

The area is usually a small space rented for the few hours the swap occurs (it was recently held at the Butte KC but has also been held at a vacant shop in the mall). The setup includes arranging racks/tables for displaying items and setting up the "sales" area where people:

- drop-off and pick-up items for the swap (i.e. participants)
- pay for an item (i.e. customers)

Take-away: Any setup (including computer, router, etc.) must:

- be simple/easy
- be quick
- allow for two "work stations" to simultaneously process participants and customers

• **8:30am-10:00am -- log swap items**

Swap participants drop off items. This entails 4H members:

- asking each participant if they are donating (items to 4H) or swapping (donating 20% of the sale of an item to 4H)
 - If swapping:
 - logging the participant's personal information (name, address, number, etc.)
 - inventorying the various items the participant is supplying so at the end of the swap the participant is either:
 - reimbursed for sold items (80%)
 - returned all un-sold items
 - logging the participant's asking price for each inventoried item
 - if donating:
 - inventorying the item as newly acquired 4H item
- displaying all items which entails:
 - labeling the item so that it ties back to the inventory
 - tagging the item with the asking price

Take-away: Any logging must be:

- quick – because participant's can come in bunches with many items

- accurate – because the participant must be properly reimbursed or their items returned
- **10:00am-4:00pm -- process swap items**

Swap customers make purchases. This entails 4H members:

- taking the customer's cash/check for the item(s) they wish to buy
- referencing the label to tie back to the inventory from which:
 - if a 4H item (i.e. donated), 4H receives 100% of the proceeds
 - if a swap item, 4H receives 20% percent

Take-away: Inventory references must be quick because a single customer may buy various items spanning various participants.

- **by 4:00pm -- process swap items**

During the swap and no later than 4pm, participants of the swap are:

- paid 80% for their items that sold
- returned their items that did not sell

Take-away: Inventory must be accurate because at the end of the day each participant must be paid money (one check covering 80% of all sold items) and/or returned their unsold items.

Strategy:

Strategy for 2013

In 2013, the attached database was "prototyped" to assist with the process and was used exclusively (so the data in the databases reflects all 2013 swap activity):



4HSkiSwap_b.mdb



4HSkiSwap_b_2.mdb

NOTE: For 2013, two databases were used because 1) two "workstations" were required and 2) to avoid potential issues with a multi-user database and networking.

An overriding goal of the prototype was to keep it simple because:

- the laptops used during the swap are "borrowed" from the county (the requirement was that they simply had Access loaded)
- there was little (to no) training opportunity
- kids (12years to 20 years) are the users
- setup of any computer equipment must be minimal (for 2013, two laptops and a printer were networked through a spare, wireless router)

The database replaced the paper inventory system documented in the appendix.

Strategy for 2014

Since the prototype worked well, the strategy for 2014 is to replace the prototype database with a new database that minimally additionally includes:

- **migrating to a multi-user database**

This entails consolidating the existing data into one database and ensuring the remaining single database can be accessed by concurrent users.

- **eliminating redundancy**

The prototype was developed to avoid any macro or module development (simply because of time and complexity issues). In the process, queries/forms/reports were duplicated to provide ability to manage customers by name or by item number. The 4H members did not care about the “by name/by number” screens because they always used the search capability – so half of what was developed for the prototype can simply be scrapped.

Adding code is not a problem either, as long as it is documented so others can build upon it in the future.

- **including a “participant” void function**

Having a record of all activity is beneficial (i.e. no breaks in the automated sequence number) so that mistakes can be reviewed yet invalid records voided.

- **including an “item” print flag**

A participant may not bring in all items at one time; therefore all tags for one participant may not be printed at one time. The current process causes existing tags to be “re-printed” with new tags wasting paper and causing confusion with “existing” tags being re-produced.

ItemEntry_ByName

Consignment Items for Number: 21
Bellanger, Laine

	Item	Price	Sell Price	printed
▶	21-142 Solomon Skis	\$20.00	\$20.00	
	21-143 Scott Poles	\$5.00	\$0.00	
	21-144 Solomon Boots	\$5.00	\$5.00	
	21-145 Technica Boots	\$15.00	\$15.00	
*		\$0.00	\$0.00	
		\$45.00	\$40.00	

Record: 14 1 of 4 ▶▶▶▶ No Filter Search

- **redoing the item numbering**

To minimize complexity, a simple auto-indexing function was used to identify unique identifiers for the items across all participants. The desire is to have an item numbering system that re-sequences off of each participant number. The "tag" report should then print tag numbers in "item order" rather than alphabetical order.

Example: Per the sample record below, the desire is to uniquely number items for participant 23 (Jim Dolan) from "1 to n" rather than simply picking up the next (and system generated) item number (in this case 151-155, and then 161-162):

ItemEntry_ByName

Consignment Items for Number: 23
Dolan, Jim

	Item	Price	Sell Price
▶	23-151 Waltonen CC Skis	\$5.00	\$5.00
	23-152 Heierling Boots CC	\$10.00	\$10.00
	23-153 Marathon Step Skis	\$5.00	\$0.00
	23-154 Marathon Jarvinen	\$5.00	\$0.00
	23-155 K2 ECS Alpine Skis	\$20.00	\$0.00
	23-161 Poles	\$3.00	\$3.00
	23-162 Poles	\$3.00	\$3.00
		\$60.00	\$27.00

Record: 1 of 10 No Filter Search

- **separating 4H "donations" from "participants"**

Currently, items donated to 4H are simply recorded under a "4H participant" record. Since this participant entry is really showing 4H donations, these 4H donation records should be managed in a separate table.

ItemEntry_ByName

Consignment Items for Number: 2
4-H, Junior Leaders

	Item	Price	Sell Price
▶	2-16 Prizm JXC Skis (white)	\$5.00	\$5.00
	2-17 Karhi Skis (grey)	\$5.00	\$0.00
	2-18 Lake Placid Skis (blue)	\$5.00	\$0.00
	2-19 Rossingol 735 Skis (white)	\$5.00	\$0.00
	2-20 Crystal Skis (white)	\$5.00	\$5.00
	2-21 Kastle Skis (black)	\$15.00	\$0.00
	2-22 Rossingol Skis (grey and red)	\$15.00	\$0.00
		\$590.00	\$190.00

Record: 1 of 78 No Filter Search

- ***Develop a 4H “donations” report***

With the 4H donated items distinguishable from the “participants”, a separate 4H donations report is desired.

Issues:

Issues to consider when developing the database include:

- Users of the database are randomly picked 4H members who may or may not have familiarity with a database type application; therefore the application must be simple and straight-forward to use.
- The application will only be used once a year, therefore even those familiar with the database must be able to “re-acquaint” themselves quickly and easily.
- Since “participant” records from 2013 are now retained, a new process to the 4H members for 2014 will be checking if a participant is already logged in the database (rather than simply “recreating” another record for them). In addition, 2013 “items” must be considered because without changes to the report and entry screen past items will appear; so should items be purged, labeled as 2013 and filtered so they do not appear, etc?
- On-going maintenance of the database will be the responsibility of 4H; therefore development of the database must be simple, straight-forward, and with adequate code documentation (if code is used).

Appendix


Inventory Sheet

Prior to the 2013 prototype database, inventory sheet(s) were used to record the following key information:

- inventory number which was assigned to each participant and displayed with the items
- participant information such as name and address
- list of items provided
- at the end of the day, the check number 4H wrote reflecting 80% of the total of the items sold and the check amount

NOTE: A participant may have had multiple "inventory" sheets if they had more items than what would fit on a single sheet

Number **103**



SKI SWAP

Consignment Information:

Name: Edie Nekl

Address: 3172 Krause Way Helena

Telephone: 406 431 8550

Items:

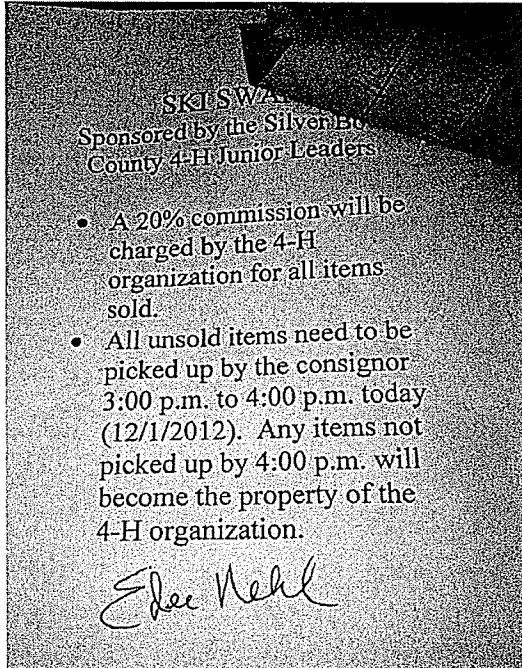
A. Two Signal Horned Skis	Price: <u>10⁰⁰</u>	<u>Donated</u>
B. Yellow Snow Machine	Price: <u>50⁰⁰</u>	<u>pd</u>
C. Yellow Snow Machine	Price: <u>50⁰⁰</u>	<u>pd</u>
D. Blue Bergin 1/2 w/ metal tips	Price: <u>25⁰⁰</u>	<u>30⁰⁰ not sold</u>
E. Wack Crystal Snow Candy Skis	Price: <u>15⁰⁰</u>	<u>not sold 80⁰⁰</u>
F. Snowshoes w/ Helmet - silver	Price: <u>25⁰⁰</u>	<u>not sold 40⁰⁰</u>
G. Brown Knie Ski	Price: <u>79⁰⁰</u>	<u>not sold 36⁰⁰</u>
H. Short Poles - Teal - Purple	Price: <u>15⁰⁰</u>	<u>not sold 144⁰⁰</u>

Check # 671 Amount \$144.00

Edie Nekl

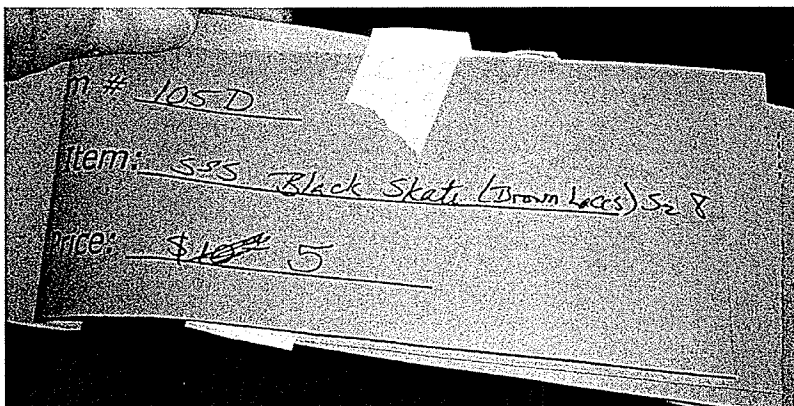
Agreement

Agreements are simply a sheet of paper noting the following and signed by the participant. Agreements are not currently managed or printed from the prototype and are simply printed ahead of the event:



Tags

Prior to the 2013 prototype database, tags were completed for each item each participant provided, and tied back to the inventory sheet via the number (in this example, item number 105) and sub-item number (in this example, sub-item "D").



The tags are what are displayed with each item and what the customer presented at the "workstations" when checking-out.

Tab - Minutes

Database Design, CSCI 340, Spring 2014
Meeting Minute template

Purpose of meeting: meet with client & present Prototype

Date/time: 4/11/12 6:00PM

Location: mTech Library

Recorder: Jay Reistad

Attendees: Jay Reistad, Luke Schuler, Scott Fleener, Joe Casagrande

Include everyone who is there at the beginning of the meeting. If someone arrives late, make a note of this at the appropriate spot in the minutes. Similarly make a note if someone leaves early.

Minutes do not need to be highly detailed but all decisions and the rationale for that decision should be recorded.

End time: 7:00PM

Date, time, location and purpose of next meeting if another meeting is planned. -Not presently

Met with Joe (client) to present the current Ski Swap database prototype. Joe like the new features & easier user interface. 😊 We talked to Joe about the multiuser set up & decide to create a Guide to setting it up.

Joe suggested a few changes for if we had any extra time

- Refresh Button

- Ability to correct sale → unsell if sold wrong item

Talked about final presentation of project at end of month. Joe said he would send a 4-H leader if he was not able to attend

Database Design, CSCI 340, Spring 2014
Meeting Minute template

Purpose of meeting: Make corrections + talk about Project

Date/time: 4/8/14 7:00

Location: Our House

Recorder: Joy Reisted

Attendees: Joy Reisted, Scott Fleener, Luke Schuler

Include everyone who is there at the beginning of the meeting. If someone arrives late, make a note of this at the appropriate spot in the minutes. Similarly make a note if someone leaves early.

Minutes do not need to be highly detailed but all decisions and the rationale for that decision should be recorded.

End time: 9:00

Date, time, location and purpose of next meeting if another meeting is planned.

Next Meeting: 4/10/14 6:00 Mtech Library ~ Meet with Joe to discuss database; Joe was away last week & had to reschedule meeting planned on 4/8

Redid Conceptual Model ~ Rethought adding a donation/swap table, only needed a boolean value stating if the item was donated

Added the objectives to List of tasks

System Boundaries are in the List of tasks, last paragraph

Simplified our user interface

* See Screenshots

Database Design, CSCI 340, Spring 2014
Meeting Minute template

Purpose of meeting: Meet with Joe & Discuss Database

Date/time: 3/6/14 6:00

Location: Montana Tech Library

Recorder: Joy Reistad

Attendees: Joy Reistad, Luke Schuler, Scott Fleener, Joe Casagrande
Include everyone who is there at the beginning of the meeting. If someone arrives late, make a note of this at the appropriate spot in the minutes. Similarly make a note if someone leaves early.

Minutes do not need to be highly detailed but all decisions and the rationale for that decision should be recorded.

End time: 6:30

Date, time, location and purpose of next meeting if another meeting is planned. No Meeting Planned Presently.

Joe answered questions we had about the database. He clarified that anyone can buy items at the 4th swap and he loved the idea of keeping track of buyers and wanted us to add this to the database.

We explained the conceptual model & new interface to be and he explained how he wanted the reports generated. Added a section to the interface specifically for generating reports.

Decided to meet again when more has been done on database. Next meeting date and time to be decided upon at a later time

Database Design, CSCI 340, Spring 2014
Meeting Minute template

Purpose of meeting: Create Conceptual Model, User Interface, & fix List of Tasks

Date/time: 3/6/14 3:45

Location: Our House

Recorder: Joy Reistad

Attendees: Joy Reistad, Scott Fleener, Luke Schuler

Include everyone who is there at the beginning of the meeting. If someone arrives late, make a note of this at the appropriate spot in the minutes. Similarly make a note if someone leaves early.

Minutes do not need to be highly detailed but all decisions and the rationale for that decision should be recorded.

End time: 4:30

Date, time, location and purpose of next meeting if another meeting is planned. 3/6/14 6:00

Fixed Previous List of Tasks. Rame up
With a few questions to ask Joe
at meeting later tonight.

Meet w/ Joe
for Questions &
to review new
Details

- Can only customers swapping/donating buy items?
- Would you like data base to keep track of information for buying customers?

Developed a conceptual Model with two tables, Customer & Item. Logical Model will probably have joint tables for Buying & Swapping/Donating

Decided to use the old database as a model for the user interface, but to redo it & make it more user friendly, eliminate redundancy, and add a few other features. Also decided to have buttons for navigating, ok and back to make interface easier to use.

Will meet with Joe later this evening to discuss questions and design of database.

Database Design, CSCI 340, Spring 2014
Meeting Minute template

Purpose of meeting: Develop a task list for Ski Swap Database

Date/time: 2/20/14 5:00 P.M.

Location: Our house

Recorder: Joy Reistad

Attendees: Joy Reistad, Scott Fleener, Luke Schuler

Include everyone who is there at the beginning of the meeting. If someone arrives late, make a note of this at the appropriate spot in the minutes. Similarly make a note if someone leaves early.

Minutes do not need to be highly detailed but all decisions and the rationale for that decision should be recorded.

End time: 5:30pm

Date, time, location and purpose of next meeting if another meeting is planned.

Meeting planned for next Thursday at 6:00pm to meet with Joe and explain conceptual model & develop user interface.

Meeting to be held at Montana Tech Library. 2/27

Developed the task list for our database project. The name decided on for the application was Ski-Swap Database. We decided on it because it was descriptive & self-explanatory. We added details Joe gave us to our objectives and project details. Had Joe approve final task list. *Good.*

According to Joe's specifications & the previous database, database will have only one user view. User view will be simplistic & easy to navigate. View will be similar to previous databases, but will have improvements specified by Joe such as:

- eliminating redundancy
- different ways for printing
- improved numbering
- separate donations from participants

Nice

Finally the system will be able to generate reports for donations, null items, and inventory.

Database Design, CSCI 340, Spring 2014
Meeting Minute template

Purpose of meeting: Meet w/ Client, Clarify Objectives & Mission

Date/time: 2/13/14 6:00pm

Location: Montana Tech Library

Recorder: Joy Reistad

Attendees: Joy Reistad, Scott Fleener, Luke Schuler, Joe Casagrande
Include everyone who is there at the beginning of the meeting. If someone arrives late, make a note of this at the appropriate spot in the minutes. Similarly make a note if someone leaves early.

Minutes do not need to be highly detailed but all decisions and the rationale for that decision should be recorded.

End time: 6:20

Date, time, location and purpose of next meeting if another meeting is planned.

It was decided that we would get 4-4 members to test database product, once we have something concrete. Great!

Reviewed mission & objectives with be, changes to be made:

- Remove redundancy from database
- Set up void entry for items removed
- For Multiusers, set up database for more than 2 users at time
- Setup must be very simple

Joe also stated that all coding done must be documented. 😊

Need to get a copy of previous database for skiisway from Celia, if she has it.

For subsequent meeting, time & location will be the same.

Tab - Previous
Material

Butte 4H Ski Swap Database Description

Scott Fleener

Joy Reistad

Luke Schuler

Mission Statement:

Replace the prototype Ski Swap database with one that is more efficient and effective.

Objectives:

- Improve existing design
- Make database very simple to use and setup (usable by 20-year-old 4H members)
- Keep inventory of participating customers and their items being sold
- Database must be multi-user
- Allow tags to be printed only when they are needed, as opposed to current system
- Have item numbering be on a per-customer basis as opposed to a global basis
- Allow creation of donation report
- Allow integration of the previous year's records

These are the expected objectives of this project, objectives will be finalized after meeting with our client

Nice

Overview of Tasks

Scott Fleener

Joy Reistad

Luke Schuler

Database Title: Ski Swap Database

This application is being developed for the Butte 4H club Ski Swap event. Scott Fleener, Joy Reistad, and Luke Schuler will be developing this application. Our mission statement is: "Replace the prototype Ski Swap database with one that is more efficient and effective." The list of tasks for our project are as follows:

- Track of contact information of donating, swapping, and buying customers
- Track inventory of items along with the tags and sales of those items
- Be able to remove items from database (will be replaced with NULL values so there are no sequence breaks in the database)
- Keep archival data of previous years' events
- Generate a donation report and customer receipts

Objectives:

- Improve existing design
- Make database very simple to use and setup (usable by 20-year-old 4H members)
- Keep inventory of participating customers and their items being sold
- Database must be multi-user
- Allow tags to be printed only when they are needed, as opposed to current system
- Have item numbering be on a per-customer basis as opposed to a global basis
- Allow creation of donation report
- Allow integration of the previous year's records

Our system will have exactly one type of user. This user is the members of the Butte 4H club. While these users will vary in age and computer ability, they still require only one way to access our database. So, our system requires no effective boundaries between different types of users. Our primary user view will be simplistic and easy to navigate due to the varying levels of computer literacy between users. It should remain similar to what the current database looks like, albeit with the improvements outlined above implemented.

Item
Item No
Customer No *
Description
donation
Sold
printed
Date Bought
Date Sold
price Asked For
price Sold For
Lowest Asking Price

Customer
Customer No
fname
lname
address
city
state
zip
phone

0..m

Selling

0..m

Buying

print

1:1

What do
relationships
represent?
Selling/Buying
Donation/Print

Assumptions:

donation, sold, & printed ~ boolean values

Overview of Tasks

Scott Fleener

Joy Reistad

Luke Schuler

Database Title: Ski Swap Database

This application is being developed for the Butte 4H club Ski Swap event. Scott Fleener, Joy Reistad, and Luke Schuler will be developing this application. Our mission statement is: "Replace the prototype Ski Swap database with one that is more efficient and effective." The list of tasks for our project are as follows:

- Track of contact information of donating, swapping, and buying customers
- Track inventory of items along with the tags and sales of those items
- Be able to remove items from database (will be replaced with NULL values so there are no sequence breaks in the database)
- Keep archival data of previous years' events
- Differentiate current and archive data ← a characteristic more than a task -1

Our system will have exactly one type of user. This user is the members of the Butte 4H club. While these users will vary in age and computer ability, they still require only one way to access our database. So, our system requires no effective boundaries between different types of users. Our primary user view will be simplistic and easy to navigate due to the varying levels of computer literacy between users. It should remain similar to what the current database looks like, albeit with the improvements outlined above implemented.

System boundary?

Generate a donation report -1

Overview of Tasks

Scott Fleener

Joy Reistad

Luke Schuler

Database Title: Ski Swap Database

This application is being developed for the Butte 4H club Ski Swap event. Scott Fleener, Joy Reistad, and Luke Schuler will be developing this application. Our mission statement is: "Replace the prototype Ski Swap database with one that is more efficient and effective." Our objectives for this

~~project are as follows:~~

application:

- Improve existing design *multi-user and*
- Make database very simple to use and setup (usable by 20-year-old 4H members)
- Keep inventory of participating customers and their items being sold
- Database must be multi-user
- Allow tags to be printed only when they are needed, as opposed to current system
- Have item numbering be on a per-customer basis as opposed to a global basis
- Allow creation of donation report
- Allow integration of the previous year's records *but differentiate current + archive data*
- Keep good documentation of any code used in the database

Really, keep people but purge items sold?

Our system will have exactly one type of user, ~~this user is~~ the members of the Butte 4H club. While ~~these~~ users will vary in age and computer ability, they still require only one way to access our database. So, our system requires no effective boundaries between different types of users. Our primary user view will be simplistic and easy to navigate due to the varying levels of computer literacy between users. It should remain similar to what the current database looks like, albeit with the improvements outlined above implemented.

List of tasks the system will perform is still needed. -5

Butte 4H Ski Swap Database Description

Scott Fleener
Joy Reistad
Luke Schuler

Mission Statement:

Replace the prototype Ski Swap database with one that is more efficient and effective.

Excellent

Objectives:

- Improve existing design
- Make database very simple to use and setup (usable by 20-year-old 4H members)
- Keep inventory of participating customers and their items being sold
- Will be used on laptops borrowed from county, so cannot be machine-specific
- Database must be multi-user
- Allow tags to be printed only when they are needed
- Re-do item numbering to simplify UI - ?
- Allow creation of donation report
- Allow integration of the previous year's records

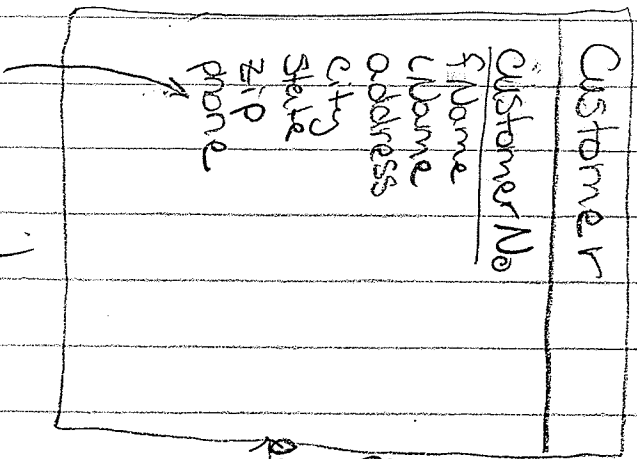
*When else
will they
be printed*

Important

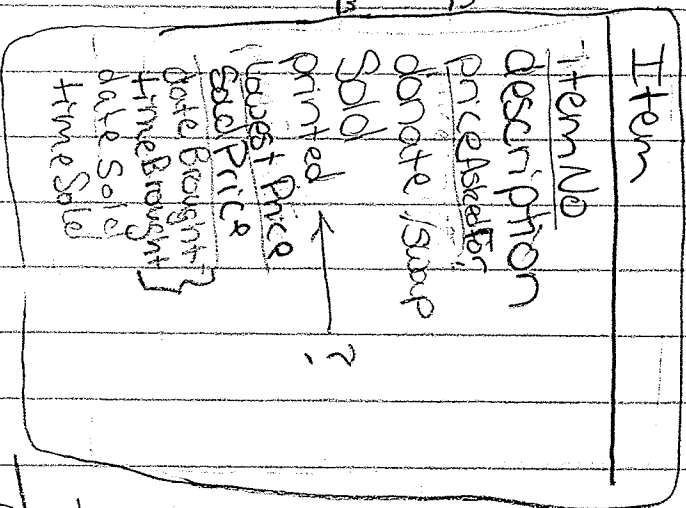
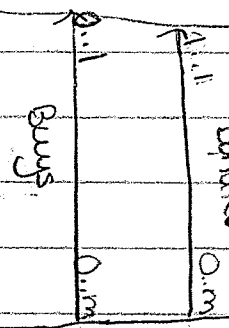
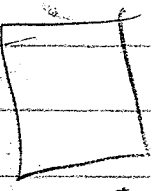
*- Not an
objective, but
is useful
info.*

These are the expected objectives of this project, objectives will be finalized after meeting with our client

*Separate objectives
from characteristics
of the system.*



I number
from own numbers? No
store two



Assumptions: Item will have a price being asked for, a lowest price, and a price that the item sold for
 printed
 sold or donate / swap : boolean values

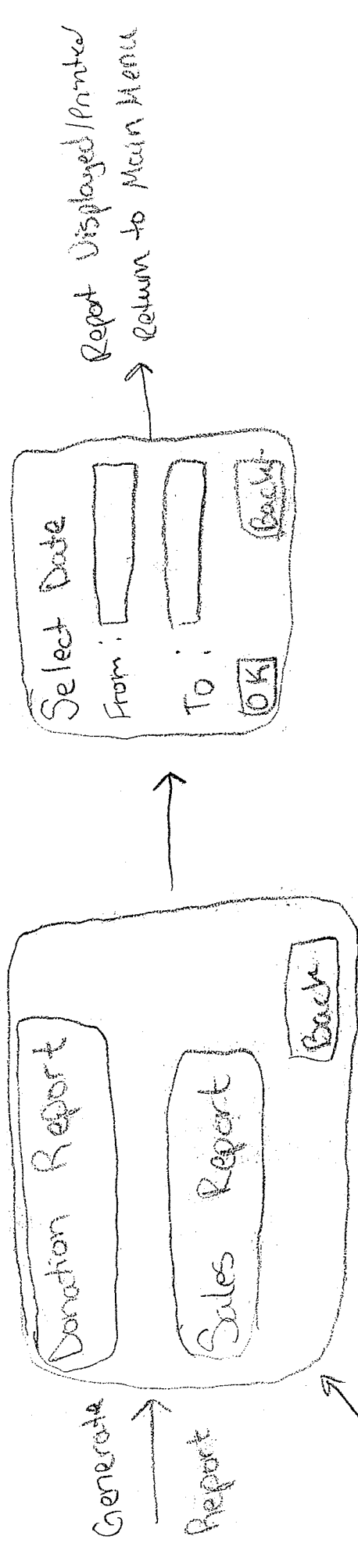
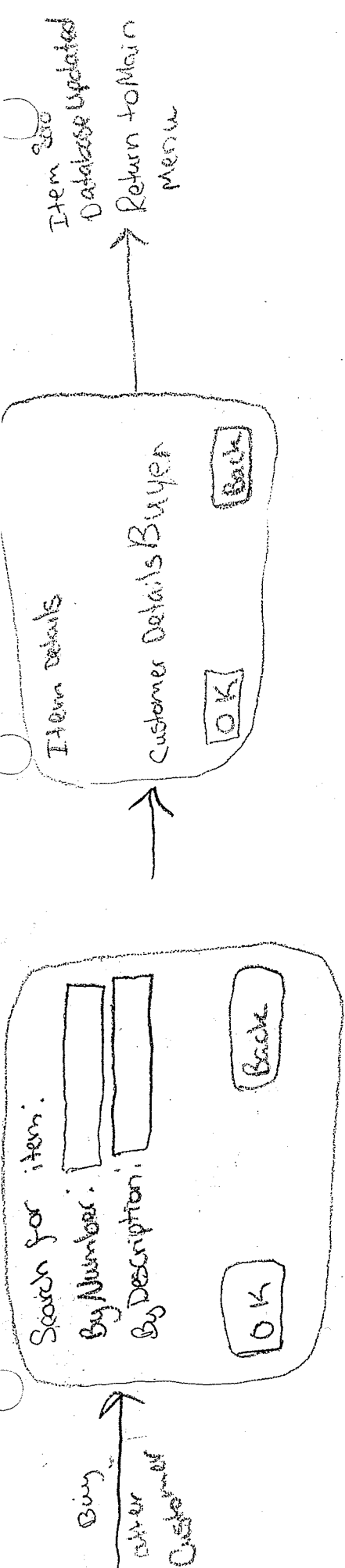
logical model will have joining table with optional refs

outputs sale & Donating / swap

Question: Earlier you said have item

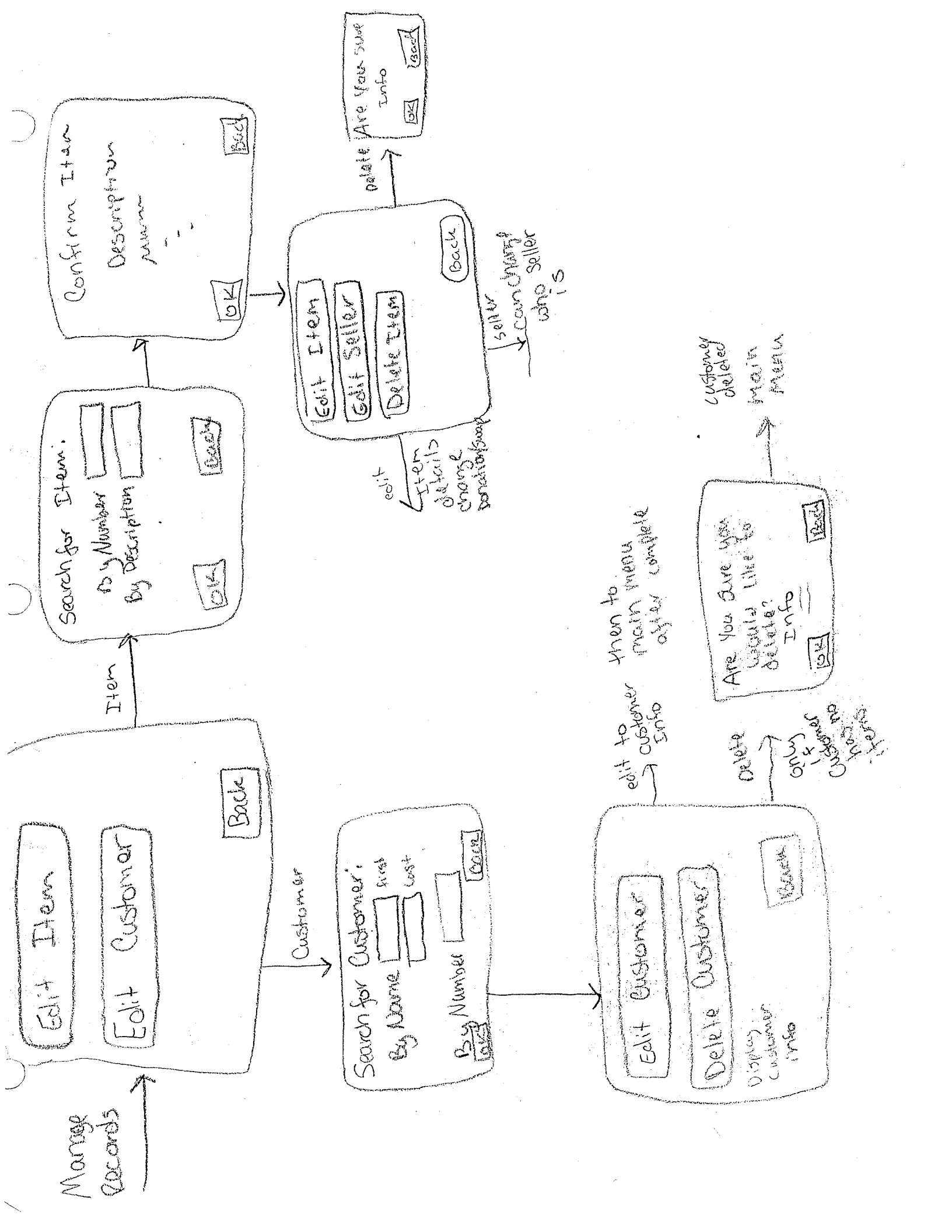
numbering on a per customer rather than a global level. Therefore the itemNo is only a partial primary key. -2

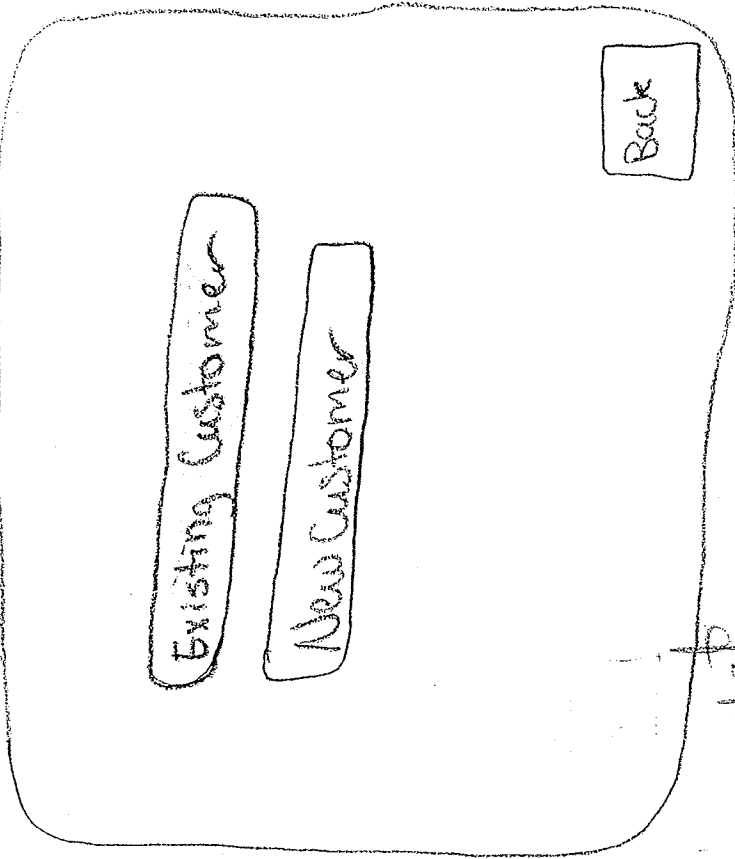
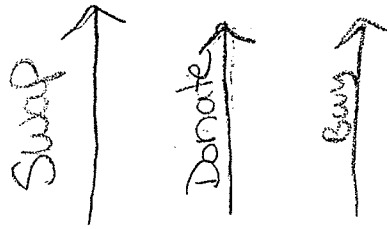
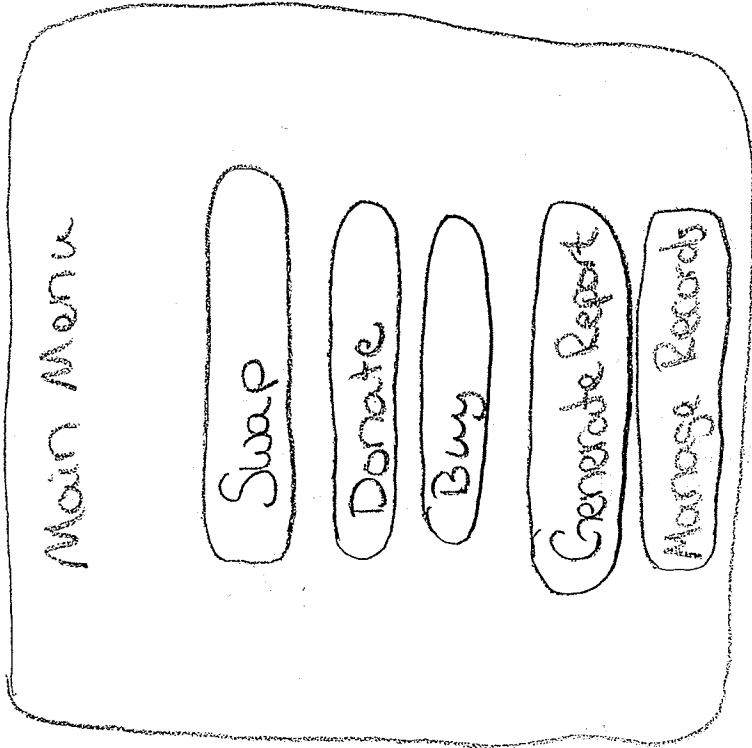
This belongs in the conceptual model. Didn't actually need attributes with alternative numbers



you might be able to have a button on the screen that says 'Generate Report' and then you can also have a 'Back' button.

Manage Records





I hope this part will help
 determine if the system
 exists or not.
 Can + the
 Auto-completion
 for user their records
 Customer records
 ask if
 them.

Existing Customer

Search By Name:

Search By ID:

with a search bar
 with a search bar
 with a search bar
 with a search bar

New Customer

First Name:

Last Name:

Address:

City:

State:

Zip Code:

Phone Number:

any other be
 any other be

swap
 or
 delete

Add

Description:

Price:

For Customer:
 customer do
 customer item
 with
 step

asking
 or price
 + forward
 accept
 price (if
 not a
 donation)

Would you like to add another item?

ADD
 press
 pass
 Main Menu

YES

edit
 info

Tab - General
info

Database Design, CSCI 340, Spring 2014 DB Application

Last updated: Feb.3

Submissions:

Feb. 12 – Project Description (mission and objectives)

Feb. 21– List of Tasks (scope of system, types of users, “what” the system will do)

March 7 – Conceptual Model and Draft of User Interface

April 9 – Draft Report

April 30 – Final Report

April 30 & May 2 – Project Presentation to Clients

Working in a group, you are to develop a database application, for an external client. Your group will meet with the client to develop requirements for the database application, develop a data model, create the tables, and a user interface (forms and reports) for the application. You will meet with the clients throughout the semester to ensure that the system that you are building is the system that the users want. Each group must establish a weekly meeting time and record minutes for all meetings - meeting with the clients, your group meetings, and times you meet with me. Use the 'Meeting Minute Template' under the Projects tab on the website.

Get a ringed binder for your project and keep all materials in it, old and new. Your binder should at least have a section for current materials, one for past materials, and one meeting minutes.

Project Description

Establish a weekly meeting time, meet with your client (this may or may not be during your established meeting time) and create a mission statement and objectives for the project.

Turn in: Binder containing minutes from your meeting (which states your established meeting time), the project mission statement, objectives and any other information which will be useful to you in creating the database application.

List of Tasks

Develop a high level overview of the application. Include the scope of the system, the types of users of the system and a list of the tasks that the application is to perform.

Overview document that has been accepted by the clients (client feedback and acceptance must be documented in meeting minutes). The Overview document should be in paragraph form, typed and should contain the following:

- A name for the database application
- Who the application is being developed for
- Who is developing the application

- Mission statement for the application (see Figure 10.8, page 327 of the text)
- List of objectives (see Figure 10.9, page 331) for the application
- Systems boundary (see Figure 10.10, page 332)
- If there will be different types of users of the system, describe each user view.

Turn in: Binder containing overview, task list, meeting minutes, updated versions of previous information, old versions of previous information and any feedback which I have given your group.

Conceptual Model and Draft of User Interface

Develop a conceptual model (attributes which are foreign keys are not included in the UML diagram), and a user interface for the application. Working with the clients, walk through the tasks that the database is to perform, using your suggested user interface and making certain that all needed data is present in your conceptual model.

Conceptual model that has been accepted by the clients. (Client feedback and acceptance must be documented in meeting minutes.) The data model must be typed. This diagram should contain entities, relationships, attributes, keys and cardinalities. If there are constraints on the data these should be noted on the data model. Turn in the entire database application folder along with all of the old materials.

Sample forms and reports for the database. These can simply be sketched on paper. Have your users look over these forms and reports and tell you if they look right. Turn in the entire database application folder along with all of the old materials, original and updated, and the old feedback sheets. Use tabs to organize your folder.

Turn in: Binder containing conceptual model, possible user interface, updates of all other information, along with the originals and any feedback which I have given your group.

Draft Report

Create the database application, turn in a copy of each table, populated with sample data, screen shots of your application, and the application itself.

Tables and primitive data entry forms. Referential integrity should be enforced for the relationships amongst the tables.

Once the tables are created, use the Form Wizard to create primitive data entry forms to facilitate placing data into the tables. These do not need to be the final forms that will be used in the database application. Use these forms to put realistic data into each table. Put

at least three records into each table. Look directly at the tables to determine if the data is stored as expected.

Once you have convinced yourself that the tables and forms are correct create paper copies of the following to turn in:

- relationships between the tables (use the relationship tool from the tool bar),
- the sample data in each table,
- one copy of each form (DO NOT turn in a copy of the form for every record in the table!).

One easy way to get these paper copies is to press the *printscreen* button when the information is being displayed on the screen. You can then paste the screen image into another document for printing.

Turn in the entire database application folder. Also give the database application to the clients so that they can enter realistic data into the tables for the next deliverable.

Tables containing realistic client data. The clients should have received the database application on March 24 so that they could use the forms to place data into the tables. Once the clients have entered data into the tables, create the same paper copies to turn in as you did for the previous deliverable. You do not need to make new copies of the information if it has not changed. Turn in the entire database application folder.

Forms and reports: Develop forms and reports for the system. This should be the first version of forms and reports that you show to the clients. Use *printscreen* to print your forms so that only one page of each form is shown. You do not need to show all of the data that is in your system. These forms and reports should have been shown to the clients and meeting minutes should document this. Turn in the entire database application folder.

Turn in: Binder containing all the screen shots updates of all other information, along with the originals and any feedback which I have given your group.

Final Report

Complete the database application.

Turn in: Binder containing all the screen shots updates of all other information, along with the originals and any feedback which I have given your group.

April 27 – Presentation to clients. Invite the clients to class to see the final project. You will be graded on this presentation.

May 4 – Final database application delivery. Turn in an electronic copy of your application, your binder with the updated and old materials, another folder that contains a fresh copy of the overview of the application, final data model, fresh copies of form and report (this is for the clients), and an addition fresh copy of your documents for me to keep. You will be graded on each of these 3 deliverables.

The final application will also receive a grade that will count three times as much as the previous grades. The final application will be graded as follows:

Appearance and usability - is it attractive, does the layout of the forms and reports make it easy to comprehend the information, can all portions of the system be gotten to easily from the switch board and with a small number of clicks, are tab orders correct, are forms that should be updated after an action updated? Are all labels on the forms and reports English words rather than field names? (Particularly, watch the display name for the forms and reports.) Auto-number fields are not displayed as they are meaningless.

Coding - are object names descriptive and do they begin with a three-letter code that indicates the type of object, are controls that are referred to in other places clearly named,