

**Team: 1010 W. Springfield Posse and We all lived at ISR at some point**

Anand Sajnani – asajnani, Eric Jenkins – ejenkins  
Brian Gregg – bcgregg, Jeff Griffin- jagriff1

**WIC Database Final Report**

**What the Project Accomplishes**

For our project, we chose to design a more user friendly version of the WIC database. WIC is a supplemental nutrition program for women, infants, and children under 5 years of age. It is funded by the US department of agriculture. This database keeps track of who is enrolled into the WIC program and whether or not they fulfill all of the requirements to enroll into the program. It stores information starting with basic demographic information (name, date of birth, address, and race) of each participant and also takes WIC program specific information such as Race, Employment status, marital status, years of school, disabilities, and languages. This allows the people who work at the WIC program to evaluate applicant's information for program eligibility. Information about participant's children will also be stored into the database. In order to enroll into the WIC program, the participant must get a health assessment, and that information is stored into the database. If the participant receives a service, the entry will be recorded into the database. We categorize the service history into a table which shows multiple services that a participant might receive.

**Why We Chose to do our data Project on WIC**

We wanted to create the WIC database because Brian came up with the idea after talking to his grandmother about the class' assignment. She works with the current WIC

database system, and so we decided that this would be an effective use of our database knowledge. We created a GUI for this program using visual C++ as well as MySQL to implement the database. We wanted use the CS 411 course on databases to work on something that would have a real world application. This would help solidify what we learned in class and put the course concept into practice. Since a version of the database was already established, we could work on our project with specific goals in mind. This way we had something to measure our progress against, and there would be no temptation to change our database schema on a whim simply to make things easier to do.

### **Our Plan and Design Specifications**

Most of what we were able to accomplish went according to the design specifications. We did have to change the way the health assessment was accessed because it ended up needing its primary key in order to uniquely identify each health assessment for a person. We also did not originally account for the fact that a person could receive service entries. When we realized this problem, we modified our ER diagram to be able to have multiple service histories for an individual person as well as a way to look up multiple histories for a single participant (see attached and revised ER diagram). Our approach to this was to make a new form in our program to show the history of services that a single participant receives. Another thing we had to do was eliminate the participant eligibility of the program. We made this decision because we realized that the database would only be used to store WIC participants, in other words, they must already be eligible to enter into the program, so this would not be very useful in our program. Something we also chose to modify was the method in which a

participant's information can be modified. Originally we decided that we could modify the users data directly on the data grid, but later decided that this might lead to unwanted modification by mistake, so we decided to make an "edit participant" button on the participant's information. When you double click on the participant from the data grid a participant options menu comes up, and from there the information can be modified.

This is similar to the insert a new participant where a form would open up displaying all of the current information on the participant pending modification and allow the user to change the information according in a more organized fashion.

We also used PHP in order to aid us throughout the database creation process. We linked our database to a PHP server where we were able to test database query commands and from our browsers as well as from our program. Having an additional database manipulation model at hand helped us a lot because it gave us an idea of some of the functions our C++ database should support. The versatility it provided also allowed us to test our database program by giving us a way to manipulate the database information separately from our program. This was very helpful in the initial design stages (when, say, search might work but insert would be broken), and also at the end for testing purposes. Using this server, we were also able to link it with our program and when we modified the database with our program, the change would instantly appear online. There were some string issues we ran into when coding the C++ part of the program but these issues were eventually resolved when we were able to find out how to copy the strings into the box for modification. We also used an enumerated data value to query our data on the options with the drop down boxes. This allowed our database to

link the information with the value and pass back the original string. The reason we chose to do this is because it was an easier and more concise way to query our database. We also had to figure out how to resize the windows when we ran the program so they would automatically be resized to an appropriate size reflecting the main window of our program. We were also able to get all of the forms constrained within the main window of our program. This way, a user cannot make a form disappear from the viewable dimensions of our window. We were also able to implement the default search of the database. When the user queries the database without entering anything, all of the participants in the database will show up. Also, if part of the string you are searching for, name address, date of birth, or assessment ID is entered and the appropriate search method is used, the corresponding participants matching that part of the string will show up.

### **How we divided up the work**

The division of labor was pretty straightforward from the beginning of the project. We tried to stick as close as possible to the original plan in order for each of us to do about an equal share of work. Brian was able to implement most of the code in order to structure the GUI for our database. Jeff and Anand designed the insert and query commands as well as populated the database and made it diverse to show a realistic use for this type of application. Eric did most the back end of the GUI. We met as a group many times throughout the course of the project in order to budget our time and effectively get each stage of the project completed. There were times where we would end up helping each other when one of us was stuck at a certain point during the project.

For instance, there was a time that Brian got stuck while coding the GUI and would e-mail us all to look up a solution online in order to figure it out. Additionally, we had trouble implementing icons for the commands and Jeff was able to find a solution to this online. Eric also pulled through when Anand needed help writing the insert command. We all contributed ideas for making the GUI very user friendly so that even a beginner at computer would have enough knowledge and understanding to work the database. Brian constantly kept in contact with his Grandmother (who works with WIC), and she was able to answer any functionality questions we had. She gave us information on how the program works as well as what type of information they need to input for the people enrolled in the program. Brian would constantly give the group feedback on how we could do these things. He was able to keep all of us on schedule and make sure we would complete this project in a timely fashion. Sometimes when we were pressed for time, we would have a special group meeting and try to help each other get back on track. We constantly referred to online sources, as well as the Database Systems book for implementing the database. The C++ source library helped us with coding compilation errors.

### **Project Organization and Commentary**

The way our project is organized is pretty simple. We spent a lot of time making this database accessible for beginners with minimal computer knowledge. When you open the executable file after it has been compiled, you see a standard toolbar on top with File, Edit, Tools, and Help. You also see the icons: New, Search, Connect, and Disconnect. Upon opening the program, you will notice that the disconnect icon is

grayed out and unusable. The reason for this is because you cannot disconnect until you have connected to the database server.

First, you must start by connecting to the database server. When you click connect, you are able to enter the IP address of the server which you wish to work with. There is also a 'user id' and 'password' box that we added for security issues. The IP box is not hard-coded, meaning you can work with several servers at different sessions in order to enable the user to work with different databases. This is effective because there may be many WIC programs at different locations that wish to work with separate databases. Once you have connected to the database you wish to use, you can click the search button to see a list of participants on a data grid of the database as well as all of the information about them. You will notice that if you want to search for a participant, the type box what enables you to exclusively see a Mother, Infant, Children that you want to search for. They 'search by' and 'search for' boxes are grayed out in order to make the user first select the 'type of person'. You can then select "search by" box to the right of that which will allow you to search by name, address, date of birth, or assessment id. The 'search for' box is still grayed out until you select what you wish to search by. You can then type what you would like to search for in the "search for" box which will allow the user to type in whatever name, address, date of birth, or assessment id they wish to search for and it will return the results if there are any results to return once you hit the search button. If you wish to delete one of the participants, you just hit the 'delete' key on the keyboard while you have selected the participant you wish to delete on the data grid from the database. When you delete a participant, a confirmation box comes up stating if you

are sure you want to delete the participant. If you select 'yes', the participant along with its child will be deleted. If you select 'no', that person will be kept in the database. If you delete a Mother and she had corresponding children, we used foreign keys to delete her children too. Accordingly, all of the person's information will be deleted along with them if you chose to do so.

If you decide you want to enroll a new person, you can click on "new participant" and a window will pop up that allows the user to enter the type of person, name, address, years of school, disabilities, date of birth, race, employment statuses, marital status, and languages spoken. If you enter a child, the employment status, marital status, languages, and years of school are grayed out because this information is not necessary for the entry of a child into the WIC program. The date of birth was implemented using the calendar function that allows a calendar to pop up once clicked on, making it easy for the user to just choose a desired date. You can also just manually type the date in the 'yyyy-mm-dd' format. Once you click the "save" button, the new participant is added to the database. You can add people with the same information, but they will have unique participant ID's assigned to them because that is a primary key in our database. In order to modify information, you can search for the participant, and click on the participant and you will notice many options you can select from.

You can edit the participant's health assessment and if they do not have a health assessment, you can create one. While you are editing the health assessment, the 'edit assessment' box is grayed out so you cannot edit the assessment twice at the same time. Once you hit the okay button, the participant's information is updated instantly as well as

reflected on the data table. You can also hit the 'edit participant' button which will enable you to edit the participant's basic information. You are also able to click the 'add/view children' button which enables you to view the list of children if the participant has any as well as add a child to a Mother. You can also create a service entry for a participant by clicking on the 'create service entry'. You could then click the 'service history' box which enables you to see a list of service histories that a participant received. If you click the 'close' button, you will be reverted back to the data grid.

Lastly, you can exit the program by click exit from the file toolbar which will exit you from the program and disconnect you from the database server. If you simply wish to only disconnect from the database server, you can click the disconnect button which will disconnect you from the server and you then click connect to connect to the same database server or another database server. We added windows that show the query commands when it is accessed but they can be easily taken out of the program if the user chooses not to see this. We did this mainly for our own convenience in order to see exactly which commands are being accessed from the database. This is how our program works.

### **What we would change about the project**

There is very little we would like to change for this program. We were happy with most of the progress we made throughout the duration of creating the database application. If we were to expand the functionality of the program, we would add a program eligibility entity. This would be a more accurate reflection how the database can be further utilized. If the participant is able to check their program eligibility, then they



are able to find out what aid they are able to enroll into the WIC program. The reason we did not implement this is because we wanted to spend more time on the project making a great GUI. The GUI eliminates any complication that could occur when the user is running our program. We also wanted to minimize the time it takes for the user to learn how to use this program so we could cater it to a myriad people.

### **What to consider when grading our project**

What you should consider while grading the project is the simple user interface that we were able to implement with our database application. We think this is something that could realistically be used for the WIC program. All of the bugs and quirks in the program that we found were addressed and fixed. This is a smooth running program and it is able to do all of the things we set forth to accomplish and then some. We put in a lot of work in this program and took in a lot of database knowledge from the course. This should be apparent upon our presentation of this project. We hope you enjoyed our final project.