Project Identity
Shampoo-Conditioner Identification Device/ Medicine Reminder
Week 2: 1/28/07 - 2/3/07
Karla Sittnick

Work Completed

The last week has been spent learning and applying knowledge of C++ to the Medication Reminder Program. Sample tables of objects and unique identifiers were written out and stored within the lab notebook. Further development of algorithm subsections was also done within the Medication Reminder Notebook. To begin transferring some of the handwritten ideas for the Medication Reminder Program to actual C++ computer programming, a sample section of the Medication Reminder Input Code was developed before and typed during senior design lab this past Friday. After correcting a few initial errors, the program was able to run smoothly to the end and output the correct results.

The sample Medication Schedule Subsection of the program was created to represent the portion of the program which would allow the user, or caretaker through password protection, to select specific objects to input into the Medication Reminder Program. The subsection made use of nested “if” statements and was quite long. When the section is worked on further, the change to “switch” or if/else” statements will most likely be made. Transferring the subsection from paper to computer language allowed for analysis of possible sources of error and problems to be solved for the future.

The first problem that occurred was the cause of incorrect labeling of specific objects. For example, the user was asked to input a time for the medication session they were creating. Upon doing this, two types of entries could be made. One type of entry contained simply integers and the other a mix of numbers and characters, such as “9 AM.” If the user entered numbers and characters, the object could not support such an entry and caused the program to terminate. When simply entering an integer, the program followed through properly. Predicting possible sources of error that may be caused by the user’s choice of input is a very important concern of this
project. After a draft of all subsections has been created, a test run will be prepared and include a great deal of problem solving for user-input caused errors.

The next problem with the subsection written within the senior design lab is that although the program ran properly with a standard input device, or keyboard, if the program was transferred to the Palm, or device without a full keyboard, the user would not be able to input the proper data. This problem led to the need to continue reading more on formatting, especially for devices without keyboards. Further research must be done to avoid this problem.

While creating the Medication Schedule Subsection in lab, great concern was taken to write a subsection for the Medication Schedule portion that would work hand-in-hand with the Dose Count Subsection. The order of inputs and selection of objects and unique identifiers were created to work simultaneously with the Dose Count portion of the program. Problem solving on how to store specific objects was done and sections of the Medication Schedule Input subsection was modified to allow for simple creation of the Dose Count subsection.

The Palm Suite® was also downloaded to a personal computer. The program installed properly, but an error message popped up when attempting to execute one of the programs. It may have been caused by an unknown installation error. If the suite can be successfully loaded to the Design Lab computer, the original contents downloaded to the personal computer will be deleted and a new installation will be performed to allow progress to be made in lab, as well as, at home.

Future Work

The problem of the need to write the computer program for a device without a keyboard will be addressed during the following week. The program currently displays menus that are labeled by number and prompts the user to select an appropriate number. The number corresponds to that portion of the program and leads the user to the next step within the program upon selection. Ideally, in the actual Medication Reminder Program the user will be able to toggle up and down to select the appropriate menu option. Further reading in the C++ Programming book
or, possibly, other sources of C++ programming will need to be studied to solve this problem.

Within the following week, the Palm Suite® will be loaded onto the computers in the Senior Design Lab at school and sample subsections of the Medication Reminder Program will be able to be downloaded onto the Palm. At that point in time, the program will hopefully be able to work on a device without a standard keyboard and programming for a “toggle-able” menu will have been input into a sample subsection for testing on the Palm.

Finally, more in-depth study of C++ programming will continue for the following weeks and the rest of the semester until all of the necessary knowledge to write the program efficiently has been acquired. Alterations to the Medication Schedule subsection will be made within the following weeks and test runs on the subsection will be performed. Development of the Dose Count subsection will also be made during the following weeks. The two subsections will be written independently, but once written properly, be cumulated to ensure they will work together and not cause errors when working simultaneously.

**Project Review**

In review, the project is coming along as planned. The greatest challenge of this project is learning the actual C++ Programming Language itself and being able to apply it to write the Medication Reminder Program in its entirety. The first subsection has been drafted and created. Through its creation, many sources for improvement have already been noted and forward progress will be made in the following weeks. The sample subsection is included on the following page and was the original rough draft of the Medication Schedule Subsection.

**Hours Worked:** 12
Sample Medication Schedule Subsection*

```cpp
#include <iostream>
using namespace std;

int main()
{
    //Enter Password to Edit Inputs***MUST ADD

    //Menu of 3 Options; Medication Schedule, Dose Count, Log
    MainMenu:
    cout << "1) Medication Schedule" << endl
        << "2) Dose Count" << endl
        << "3) Log" << endl
        << "Please Enter Selection: ";
    int choice;
    cin >> choice;

    //Declaration of Variables
    char ses1, ses1name1;
    int ses1name1doseamt;
    char ses1name1doseunit;

    //Medication Schedule
    if (choice == 1)
    {
        MedicationSession:
        cout << "1) New Session" << endl
             << "Make selection or enter 0 to return to previous screen. ";
        int sessionchoice;
        cin >> sessionchoice;

        //New Session Option
        if (sessionchoice == 1)
        {
            cout << "Enter time for Medication Session: ";
            cin >> ses1;
            cout << "1) Add New" << endl
                 << "Make selection or enter 0 to return to previous screen. ";
            int medicationchoice;
            cin >> medicationchoice;

            //Add New Medication to Session
            if (medicationchoice == 1)
            {
                cout << "Enter medication name: ";
                cin >> ses1name1;
                cout << "Enter " << ses1name1 << " dosage units: ";
                cin >> ses1name1doseunit;
                cout << "Enter " << ses1name1 << " dosage amount: ";
                cin >> ses1name1doseamt;
                goto MedicationSession;
            }

            //Return to Add New Session Screen
            if (medicationchoice == 0)
            {
                goto MedicationSession;
            }
        }

        //Return to Main Menu
        if (sessionchoice == 0)
        {
            goto MainMenu;
        }
    }
```
cout << "Sample Medication Schedule" << endl
    << "Medication Session Time: " << ses1 << "," << endl
    << "Medication Name: " << ses1name1 << endl
    << "Medication Dosage: " << ses1name1doseamt << " " << ses1name1doseunit << endl;

//Dose Count
if (choice == 2)
{
    cout << "Programming not ready." << endl;
    return 0;
}

//Log
if (choice == 3)
{
    cout << "Programming not ready." << endl;
    return 0;
}

//Exit Program
return 0;

*Note: The program was greatly condensed for easier viewing of the sequence of the program. The actual code is spaced out for easier readability and troubleshooting.