GROUP PROJECT REQUIREMENTS AND SPECIFICATION

I PROJECT BACKGROUND

People with diabetes cannot make their own insulin, a hormone that is normally secreted by the pancreas. Insulin is essential to metabolize sugar and hence generate energy. Currently most diabetics inject insulin two or more times per day, with the dose injected based on readings of their blood sugar level. However, this results in artificial blood sugar fluctuations as it does not reflect the on-demand insulin production of the pancreas.

A personal insulin pump is an external device that mimics the function of the pancreas. It uses an embedded sensor to measure the blood sugar level at periodic intervals and then injects insulin to maintain the blood sugar at a 'normal' level. Using readings from the embedded sensor, the system automatically measures the level of glucose in the sufferer's body. Consecutive readings are compared. If they indicate that the level of glucose is rising then insulin is injected to counteract this rise. The ideal situation is a consistent level of sugar that is within some 'safe' band.

II MISSION (GOAL) STATEMENT

The project is required to construct an insulin pump simulator that can be used for diabetes patients, insulin pump developers and physicians.

III GLOSSARY

1) Diabetes: A medical disorder that causes the body to produce excessive amounts of urine.
2) Basal: Insulin infusion given as low regular flow throughout the day. This is done in AUTORUN mode of the insulin pump.
3) Bolus: Insulin infusion given as a one time delivery in addition to any basal delivery. A bolus is given, for example, when eating a high carbohydrate meal. This is done in MANUAL mode of the insulin pump.
4) Unsafe: A very low level of sugar (less than 3 units) is dangerous and can result in hypoglaecemia which can result in a diabetic coma and ultimately death.
5) Safe: Between 3 units and 7 units, the levels of sugar are 'safe' and are comparable to those in people without diabetes. This is the ideal band.
6) Undesirable: Above 7 units to 35 units is undesirable but high levels are not dangerous in the short-term. Continuous high-levels however can result in long-term side-effects.
7) Insulin: A hormone produced in the pancreas that regulates the level of glucose in the blood.
8) Pancreases: A large elongated glandular organ lying near the stomach. It secrets juices into the small intestine and the hormones insulin, glucagons and somatostatin into the bloodstream.
9) Sugar Level: Level of glucose in the blood. Levels rise after meals and are usually lowest in the morning, before the first meal of the day.

IV THE SCOPE OF THE WORK (IN/OUT OF SCOPE LIST)

1) IN: The project shall include the design and implementation of an insulin pump simulator, to be used to educate the diabetes patients.
2) IN: The deliverables shall include a requirements document, design document and source and executable code.
3) IN: The project shall include the design and implementation of a Graphical User Interface (GUI) to start, run (automatically and manually), and stop the insulin simulator. CASE tools can be used to generate GUI. *If GUI is generated by CASE tools, it is not necessary to include the details of GUI in the design documents.*
4) OUT: This project does not require to develop hardware.
5) OUT: The project doe not develop a patient simulator.

V FUNCTIONAL REQUIREMENTS

By and large, the functional requirements for the simulator are classified into six groups: safety requirements, insulin dose computation requirements, after-insulin-injection-blood-sugar-level requirements, insulin pump status requirements, and graphical user interface requirements.

V.1 Safety Constraints

1) The initial capacity of the insulin reservoir is set to 100. Maximum daily dose, maximum single dose and minimum dose are set to 35, 5, and 1, respectively. Maximum daily dose is defined the maximum dose that can be delivered in 12 minutes that is scaled down from a day; Maximum single dose is the maximum dose of insulin that can be delivered in a single injection; Minimum dose is the dose that may be delivered to maintain an existing trend in blood sugar levels.
2) Once the dose has been computed the simulator then must determine whether the dose calculated is safe. It is essential due to the critical nature of the simulator. There are two scenarios about safety requirements depending on simulator mode -- AUTORUN Mode and MANUAL Mode.
3) **AUTORUN Mode**: If maximum daily dose will be exceeded by the sum of the dose calculated and the cumulative doze, the dose to be injected should be calculated by subtracting the sum of the calculated doze and cumulative dose of insulin from maximum daily dose. If the sum of the dose calculated and cumulative doze is less than or equal to maximum daily doze, there are two possibilities: 1) If the dose calculated is less than or equal to maximum single dose, deliver the insulin calculated; 2) If the single dose computed is too high, deliver the maximum single dose. Alert the user and set the status to warning, if the safe constraints (maximum single doze and/or maximum daily doze) broken.
4) **MANUAL Mode**: If maximum daily dose is exceeded and/or maximum single dose is exceeded, alert the user and set the status to warning. The maximum daily and maximum single dose safety constraints will be overridden. In other words, amount of insulin units calculated will be injected as it is.
5) All safety requirements are reset every 12 minutes.

V.2 Insulin Dose Computation Requirements

1) The dose of insulin to be delivered shall be computed by measuring the current level of blood sugar, comparing this to previous measured levels and computing the required dose as described below.

2) The possible blood sugar level is between 1 and 35 and the simulator shall measure the blood sugar level and deliver insulin if required every 10 seconds.

3) The amount of insulin to be delivered shall be computed according to the current sugar reading as read by the simulator:

   a. If the reading is below the safe minimum, no insulin shall be delivered. Alarm is on and a warning message must be displayed.

   b. If the reading is within the safe zone, then insulin is only delivered if the level of sugar is rising and the rate of increase of sugar level is increasing. The amount of insulin required must be defined based on the following algorithm:

```
//CompDose is the insulin dose required according to the insulin dosage computation.

//This may be overridden for safety reasons; cumulative_dose is the total dose that has already //been delivered over the last 12 minutes. r0, r1, and r2 maintain //information about

//the last three readings from the sugar sensor. r2 holds the current reading, r1 the previous //reading and r0 the reading before that.

if (r2 >= safemin && r2 <= safemax) {

        //sugar level stable or falling
        if (r2 <= r1) then CompDose = 0
        // sugar level increasing but rate of increasing falling
        else if ((r2 > r1) && ((r2 – r1) < (r1 – r0))) then CompDose = 0
        // sugar level increasing and rate of increase increasing, compute dose
        // a minimum dose must be delivered if rounded to zero
        else if ((r2 > r1) && ((r2 – r1) >= (r1 – r0)) && (round ((r2 – r1) / 4) > 0))
          then CompDose = round((r2 – r1) / 4)
        else if ((r2 > r1) && ((r2 – r1) >= (r1 – r0)) && ((round (r2 – r1) / 4) = 0))
          then CompDose = minimum_doze
}
```

   c. If the reading is above the safe zone, insulin is delivered unless the level of blood sugar is falling and the rate of decrease of the blood sugar level is increasing. The amount of insulin required must be defined based on the following algorithm.

```
if (r2 > safemax) {

        //sugar level increasing.
        if ((r2 > r1) && (round((r2 – r1) / 4) = 0)) then CompDose = minimum_dose
        else if ((r2 > r1) && (round((r2 – r1) / 4) > 0)) then CompDose = round((r2 – r1) / 4)
        // sugar level stable
        else if (r2 == r1) then CompDose = minimum_dose
        // sugar level falling and rate of decrease increasing
        else if ((r2 < r1) && ((r2 – r1) <= (r1 – r0))) then CompDose = 0
        // sugar level falling and rate of decrease decreasing
        else if ((r2 < r1) && ((r2 – r1) > (r1 – r0))) then CompDose = minimum_doze
}
```

d. The amount of insulin actually delivered may be different from the computed dose as various safety constraints specified at V.1. There is a limit on the maximum dose to be delivered in a single injection and a limit on the total cumulative dose in 12 minutes. If maximum daily dose will be exceeded by the dose calculated, the dose to be injected should be calculated by subtracting cumulative dose of insulin from maximum daily dose. In the mean time the warning message shall be issued. This message can be reset by the user within the 12 minute time window.

V.3 Blood Sugar Behavior Requirement

When a certain amount of insulin is injected based on the algorithms as shown in V.2, V.5.(4), and V.5.(5) the blood sugar level needs to be updated at next 10-second reading (we assume that next 10-second blood sugar level reading is "updated blood sugar level" and next 10-second time is "current time" in the formula below) unless the user updates the blood sugar level between current time and latest injection time. A possible formula is as follows:

updated blood sugar level =
previous blood sugar level – 3 * e$^{(-(\text{current time} - \text{latest injection time})/\text{latest injected insulin dose} * 0.04)}$

The constant numbers such as 3 can be adjusted, resulting different blood-level drops. Note that BSL is updated every 10 seconds after first dosage of insulin injection.

For example, after the simulator starts, let's assume that the current BSL is 5 (note that an initial BSL is any number between 3 and 7). If the user does not change BSL, the BSL remains same, which is 5 in this case. Suppose that at the 00:14 simulation time the user increases BSL to 14 (which means the use eats some food). Then, at the 00:20 simulation time (note that every 10 second the simulator reads BSL), the simulator reads a new BSL and decides whether or not it needs to inject some amount of insulin based on the algorithms. In this case, 2 insulin units are expected to be injected. At the 00:30 simulation time the simulator should update BSL by using the formula above. In other words, at this time, the update BSL = $13 - 3 * e^{(-(30-20)/2*0.04)} = 12.18$. Another 10 seconds later, the updated BSL = $12.18 - 3 * e^{(-(40-20)/2*0.04)} = 11.49$ unless the user changes BSL.

V.4 Error and Warning Requirement

Whenever errors and/or warnings occur, the simulator displays the corresponding messages immediately. Since there is no insulin pump hardware for the simulator, the error messages such as "No Needle Unit," "Sensor Failure," "Insulin Reservoir Removed," and "Pump Failure" are activated and deactivated by the user. The error message, "No Insulin," and the warning messages such as "Maximum Daily Dose," "Maximum Single Dose," "Low Insulin Level," "Low Battery," and "Low Blood Sugar Level" shall be activated and deactivated by the simulator. These messages can be also reset every 12 minute. Note that these messages reflect the status of the insulin simulator. For example, "Low Battery" shall be issued if the status of the battery has fallen to one-third of the full charge after the simulator starts. Note that the level of battery shall decrease by 4 units every 30 second. Note that when the battery is completely empty, the simulator shall stop. The table below specifies the error and warning messages and their descriptions. When at least one error occurs, AUTORUN and MANUAL modes are suspended until the errors are removed; simulation timer and battery keep going.

	Messages to be displayed	Message description
Error	No Needle Unit	The delivery needle has been removed.
	Sensor Failure	The self test of the sugar sensor has resulted in an error.
	Insulin Reservoir Removed	Insulin reservoir has been removed.
	Insulin Pump Failure	The pump is not working.
	No Insulin	There isn't enough remaining insulin to suffice the next dose.
Warning	Maximum Daily Dose	Maximum daily dose will be exceeded by the dose calculated
	Maximum Single Dose	Maximum single dose will be exceeded by the dose calculated
	Low Insulin Level	The level of insulin in the reservoir is low (i.e., less than 20 units)
	Low Battery	The status of the battery has fallen to one-third of the full charge.
	Low Blood Sugar Level	The blood sugar level is below the safe minimum value.

Table 1. Error and warning conditions of the insulin pump simulator.

V.5 Insulin Pump Status Requirements

1) The possible status of the insulin pump simulator includes SWITCHON, AUTORUN, MANUAL, and SWITCHOFF.

2) When the simulator is loaded, it goes to SWITCHOFF mode.

3) The SWITCHON mode simulates the behavior of the pump when the user pushes the switch-on button. When the user switches on the simulator, it goes to AUTORUN mode. It is assumed that the user's blood sugar is at SAFE stage (for example, the blood sugar level is five.)

4) Under normal operating conditions, the simulator is defined by the AUTORUN. If the user increases the blood sugar level, the simulator should respond properly based on V.2. Unless the user increases the blood sugar level, the simulator does not take any insulin injections and the blood sugar level remains SAFE. Note that cumulative dose is reset to zero at the beginning of the each 12 minutes period. The AUTORUN can be further defined in three statuses: normal, warning, and error. In normal operation, the status is running; if a state exists which is potentially hazardous such as low battery and low insulin level, then the status is warning and The simulator keep functioning with displaying warning messages; if a state exists such as pump fail, sensor fail, delivery fail and insulin empty, the status is error. In the error state, normal operation (AUTORUN mode and MANUAL mode) are suspended until the errors are removed.

5) As soon as the user clicks a manual mode button, the simulator goes to MANULA mode. The user is able to specify the amount of insulin to be delivered by pressing the button within 5 second period. The number of clicks specifies the number of units of insulin to be injected. Insulin injection occurs when the five second time window is expired. Note that safety checks, specified at Requirement V.1, are overruled. If maximum daily dose and maximum single dose are exceeded, the amount of insulin is injected while the simulator displays the corresponding warning messages. Cumulative dose and the last dose of insulin are still updated. After insulin injection, the simulator automatically goes to the AUTORUN mode. Note that the blood sugar level will be updated when the simulator goes back to the AUTORUN mode.

6) When switch off, the simulator goes to the SWITCHOFF mode that clear all information. When the user switches on, the simulator starts new.

Figure 1 shows the statechart diagram for the insulin pump simulator.

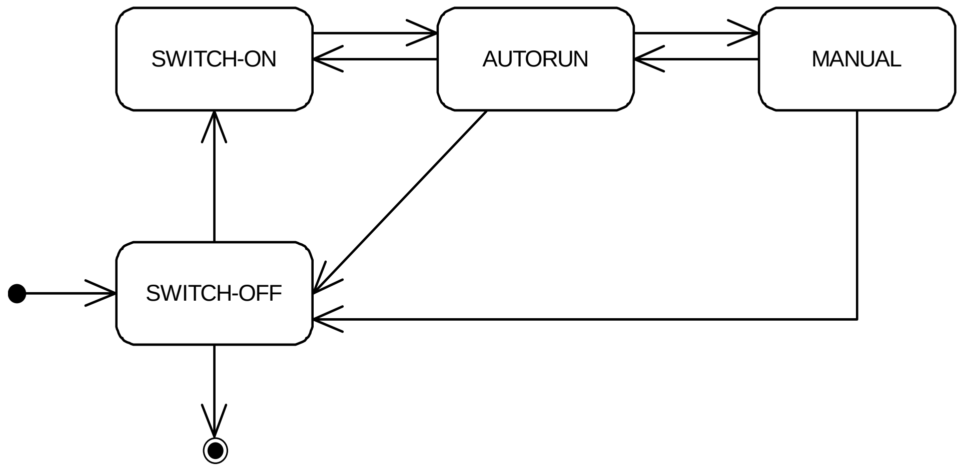

Figure 1. Statechart diagram for insulin pump simulator

V.6 Graphical User Interface Requirements

1) A power on/off switch shall be displayed.
2) A MANNUAL button is used to inject insulin manually. The number of clicks within five second is the amount of insulin injected.
3) Display the current mode (AUTORUN, MANNUAL).
4) All values are reset every 12 minutes.
5) A current blood sugar level and at least two previous levels should be displayed. Implementing a graphical representation (with numerical values) to a current sugar level and previous sugar levels is required.
6) The amount of the last dose of insulin to be administered shall be displayed.
7) The amount of the cumulative of insulin injected shall be displayed.
8) The amount of time the simulator run shall be displayed.
9) Current real-time shall be displayed.
10) The status of battery shall be displayed with a numeric value and progress bar.
11) The status of the insulin reservoir shall be displayed.
12) Warning and error messages shall be displayed in *the single line text box*. If there is one message, keep displaying that message. When there is more than one message, each message is displayed for 3 seconds until all messages have been displayed. The display sequence then restarts with the first message unless warnings and errors are clear. Also an audio alarm for each message shall be given.
13) Users are allowed to change the current blood sugar level in order to test the insulin pump simulator's reaction to varying blood sugar readings.
14) If users decide that they require insulin immediately they can manually override the automatically delivering system. Users specify how much insulin to be injected by pressing a button. The number of button presses within 5 second period specifies the number of units of

insulin to be injected. The system must be in manual mode for this to be operational and you may assume that after the first press is detected, the number of presses is counted automatically.

VI ANALYSIS, DESIGN AND IMPLEMENTATION REQUIREMENTS

Object-Oriented Analysis (OOA), Object-Oriented Design (OOD) and Object-Oriented Programming (OOP) are required. Use case diagrams, class diagrams, and sequence diagrams are required. An OOP language such as Java, C#, C++, and Visual Basic is required. CASE tools should be used for OOA, OOD and OOP: Microsoft Visio and UMLDiagrammer (or other UML drawing tools) can be used for OOA and OOD (However, word processors or presentation tools such as Microsoft Word and PowerPoint are allowed); Eclipse, NetBeans, and Microsoft Visual Studio can be used for OOP. Note that NetBeans provides an automatic GUI generation function.