Frankfurt University of Applied Sciences

Faculty 2 – Computer Science and Engineering
M.Sc. Program – High Integrity Systems

Safety Critical Systems

Medical Pump Simulator Project

Authors
Group E
Lam Phuoc Huy 1104785
Pham Nhat Nam 1105331
Mohamed Omar Zayan 1248447
Tomusange Brian 1291025
Thomas Tena Nigussie 1291133

Supervisor
Prof. Dr. Matthias F. Wagner

July 19, 2019

Contents

1 Introduction

In this project, we created a simulator for medical insulin/glucagon pump to analyze the pumping process to implement for real application design. As the complexity of infusion pump devices increases, so does the likelihood of their failure. A significant part of these failures is due to a lack of understanding of the safety issues involved. This is particularly true in the case of software, for which it is often difficult to predict erroneous behavior. To address this issue, we apply Safety-Critical System concepts into the development process to ensure the safe execution of infusion pump software.

2 Project Management

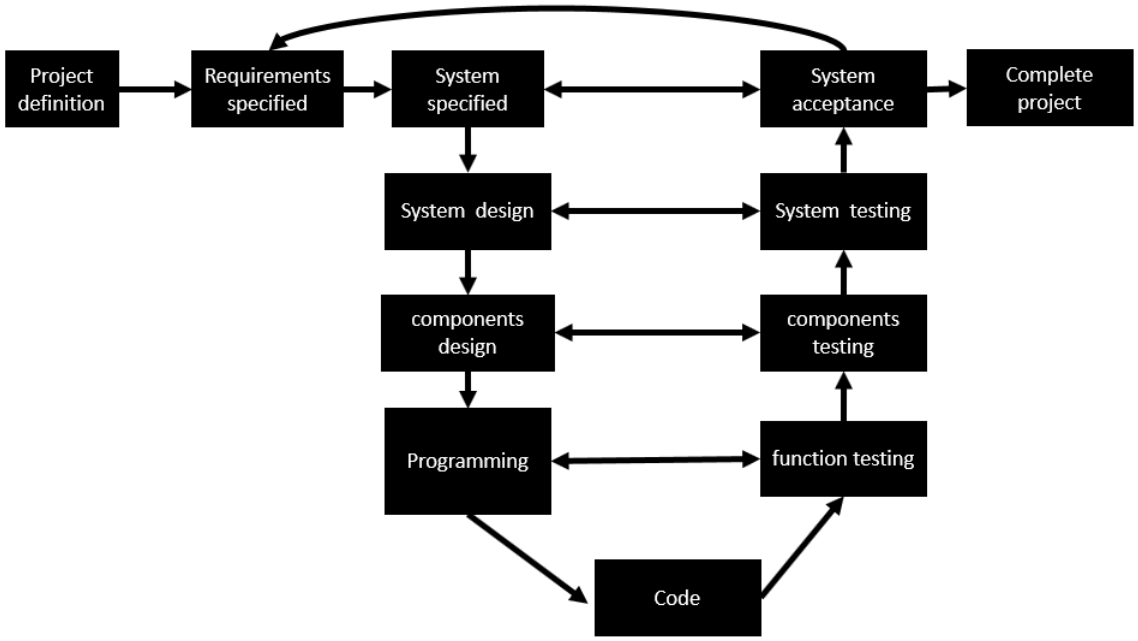

Figure 1: V-model XT for the Insulin/Glucagon pump simulator development

Our team decides to use V-model XT as the software development model because it is a worldwide software development model, especially in Germany. Moreover, the testing activities happen in each step so that the product quality can be guaranteed. For a project with unambiguous requirements like this one, V-model XT is applicable.

The responsibility for the project can be distributed as:

- Progress management, document responsibility, theoretical problem: Huy, Nam.
- Simulator Java version developer & mathematical model: Zayan, Thomas (tester).
- Problem related to backup web version simulator: Huy, Nam.

- Hazards Analysis: Brian, Nam.

List of tools and software team E used during the development: **Programming language**: Java SE 11 (main simulator), Javascript 3 (web-based backup simulator); **IDE**: Eclipse; **Communication**: Whatsapp; **Code management**: Github; **Diagram creation**: draw.io.

3 Use Cases In The Simulator

1. Doctor: can check the pump log and glucose level data.
2. Patient: can observe his/her own glucose level, the remaining dosage of insulin, battery and the log of glucose level when the pump works on the application interface.
3. Emergency: If the glucose level overcomes extreme level or the battery/insulin dosage are exhausted, the user interfaces change and alarm rings to notice patient and nearby people.
4. Tester: use a keyboard to input event such as eat, exercise to change the glucose blood level or increase time speed.

4 Design Model

4.1 Requirements

- When glucose level gets higher or lower than normal level, the alarm must ring and the message must show on interface continuously. Then the pump must work to make it return to the normal level and the alarm stop.
- When the insulin dosage and battery running low, the alarm must ring and the message must show on interface continuously until users refill/recharge.
- The critical detail on the interface must be transparent.
- The insulin dosage, battery amount must decrease while the simulator runs.
- Testers can increase the speed of the simulator. After simulator runs for a while, testers can look at the glucose level history.
- The time record on the clock must be correct

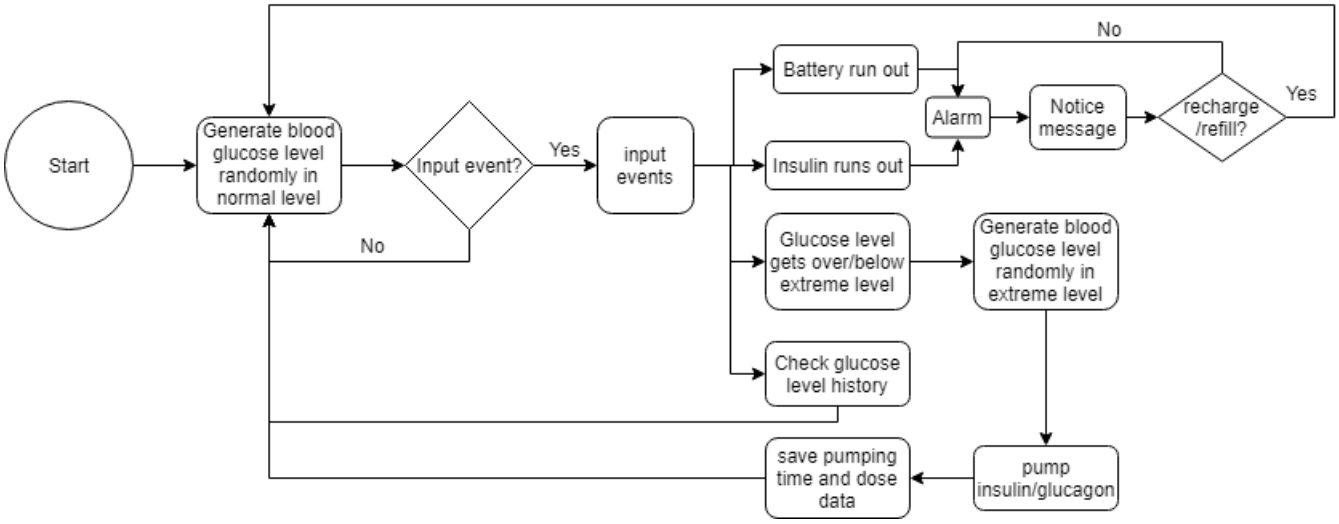

Figure 2: Work Flow Diagram

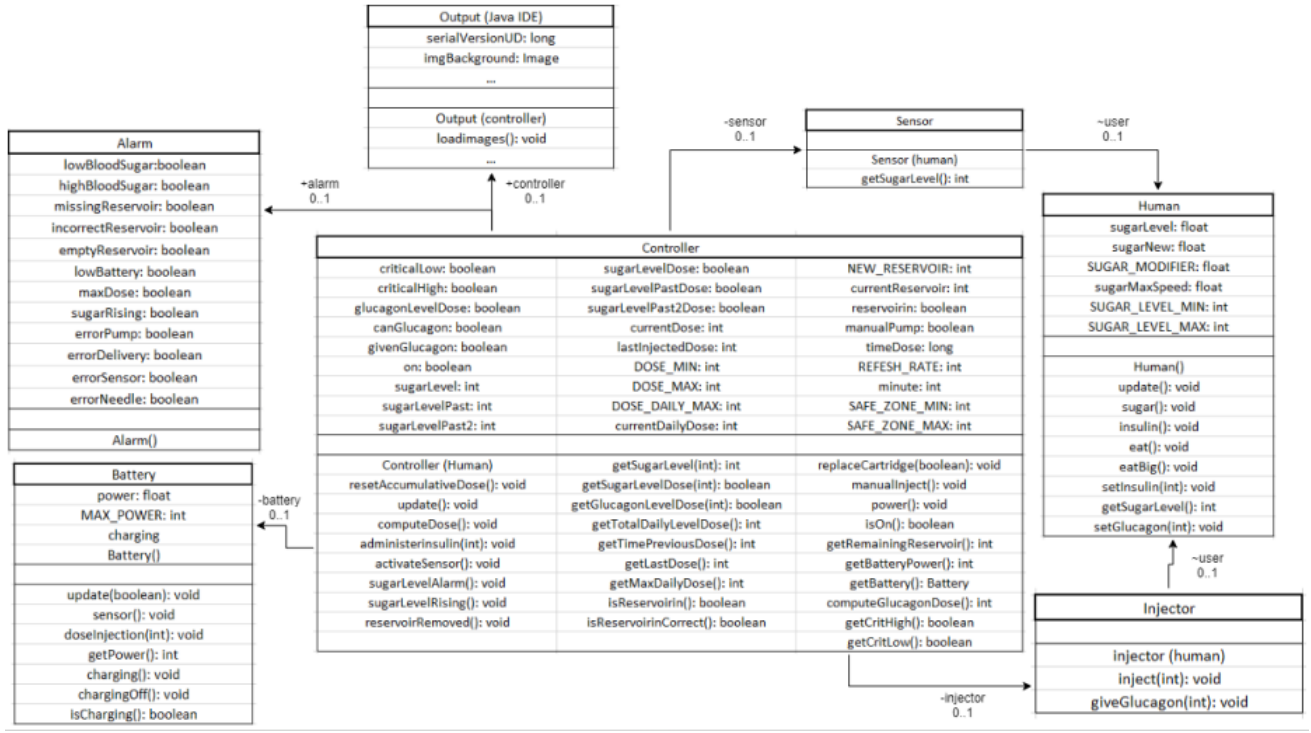

Figure 3: Class Diagram

4.2 Diagrams and Models

In order to fulfill the requirement for the simulator, the workflow diagram (Figure 2) and the class diagram (Figure 3 are created to give team members an overview of the structure of the program.

After researching some relevant paper regarding the subject like [1], [2] and [3], we have decided to use the model suggested in [3] which is simple to understand. Throughout the implementation of the mathematical model, it proves that the model's result is acceptable and suitable for the project.

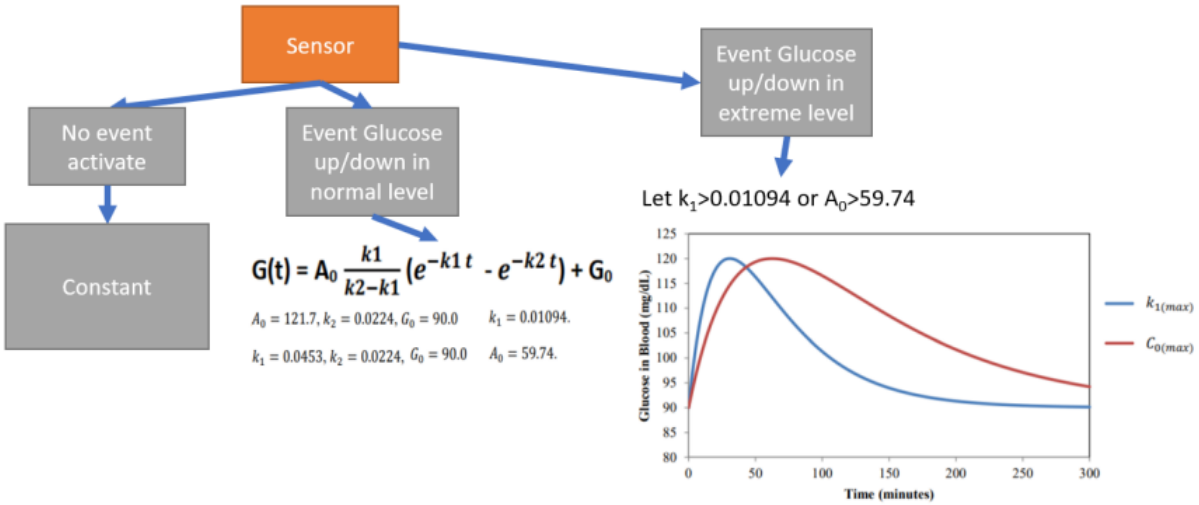

Figure 4: Mathematical Model

5 Hazard Analysis

The STAMP/STPA method is used to identify or predict erroneous behavior [4]. To address this issue, we investigate the use of formal methods based techniques to assure the safe execution of infusion pump software. The basis of the proposed approach is a safety analysis process wherein potential risk is identified. These risks are used to identify a set of core safety requirements that can be formally verified against infusion pump software. Initially, we applied the FTA method on a particular risk H1: Incorrect Insulin dose administration. Using a causal effect diagram, were able to identify 7 causes of risk after as shown below in the Fault Tree.

Next, STAMP analysis was used on H1 to compare with the FTA method. There are seven steps that we need to follow to do the hazard analysis for the insulin infusion pump project.

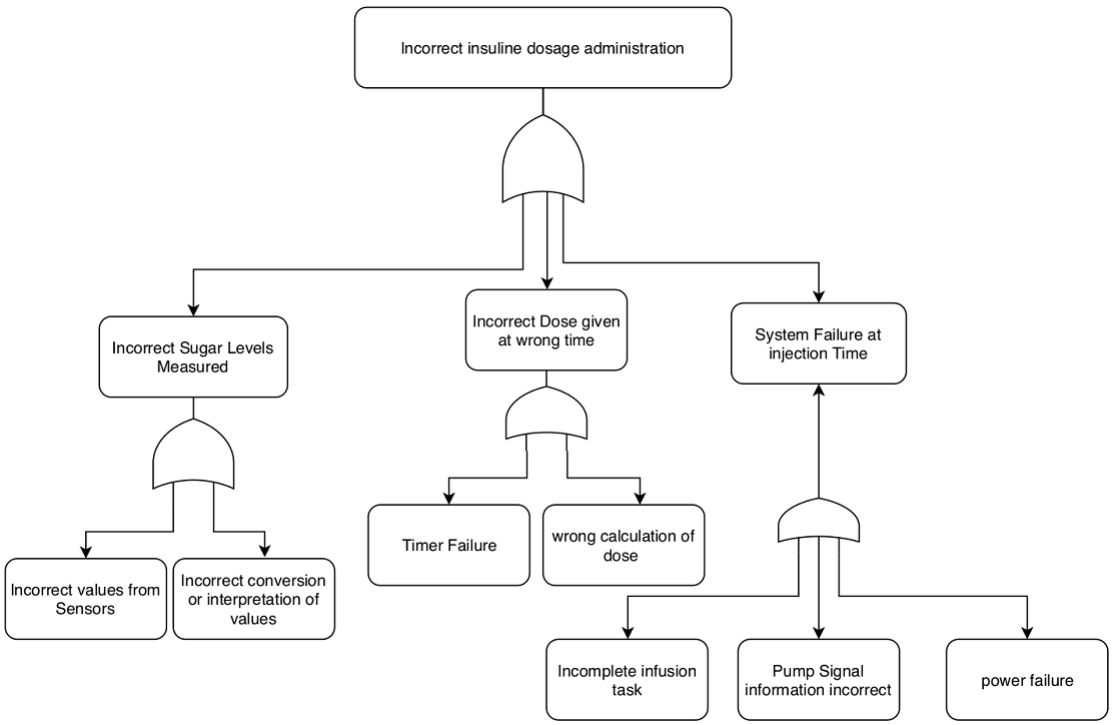

Figure 5: Fault Tree Analysis

5.1 Identified system objectives, system hazard, and safety constraint

The system is a simulation of a medical infusion pump. The objective and system scope has been clearly and explicitly explained in the previous section. There are very many hazards, we choose to analyze H1 which is important to the system performance. The system hazard H1: Incorrect Insulin dose administration using the casual effect analysis is a result of incorrect measurements of sugar levels, incorrect interpretation of readings to mention a few. The following Safety constraints and requirements were identified as well.

Risk	Safety Constraints (SC)	Safety Requirements (SR)
R1	SC1: Accurate dosage of insulin should be administered to the patient.	SR1: The system shall administer timely, accurately and correct amounts of insulin dose to a patient within an acceptable total allowable error.

5.2 Creating the Hierarchical Control Structure.

The system control structure was designed to investigate the hierarchical relationship of control throughout the system. The case study control structure was complex, contained

many elements within multiple layers because of its criticality as shown in the diagram below. Some of the control structures were eliminated for example the organization control, the different controllers not related to the insulin dose amount controller.

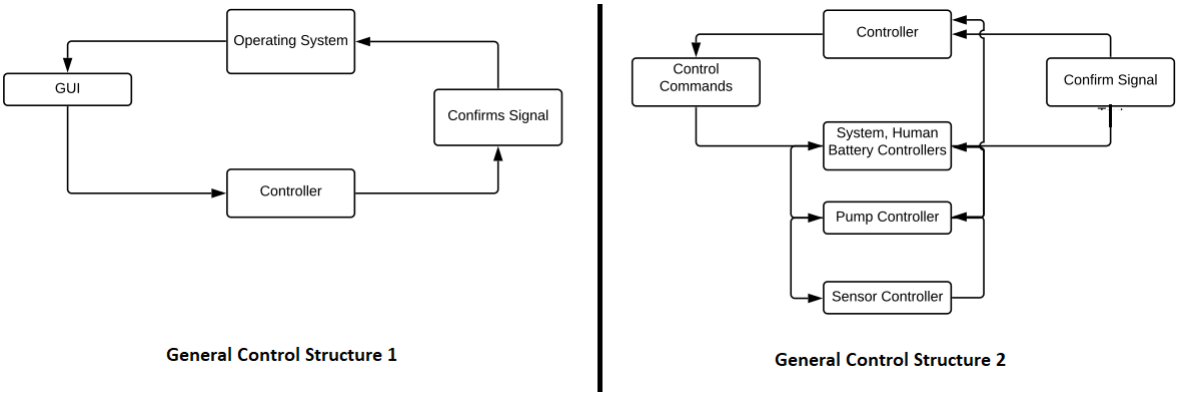

Figure 6: General Control Structures

For this hazard, we considered the pump controller, the sensor controller and system/battery controller as internal controllers which are all controlled by the master controller. Figure 6 shows the general structure of the program.

5.3 Defining control actions.

CAST analysis is done to identify which component violates the safety constraints and thus more details are added to or reduced from the controllers. After the changes are made, a more detailed control diagram for the technical system was generated to capture the Intra-Controller system. In this view, the direct controls of some software components were exhibited. The emergent function of these interactions was the system end value, the safe analysis, and reporting component.

5.4 Control structure under normal condition

Some of the loops identified work as independent, parallel or in series with others. The normal working condition was divided into phases namely time evaluation, sensor measurements, evaluations of results (including dosage), induce pump operation and pumping process after infusion. For a normal patient with the infusion pump, the user initiates the time evaluation process using the GUI. The sensor measurement controller will then control measurements, the system/human controller will evaluate the result and control the dosage, the pump controller will control the pump operation and the master controller will control all the process including the post infusion.

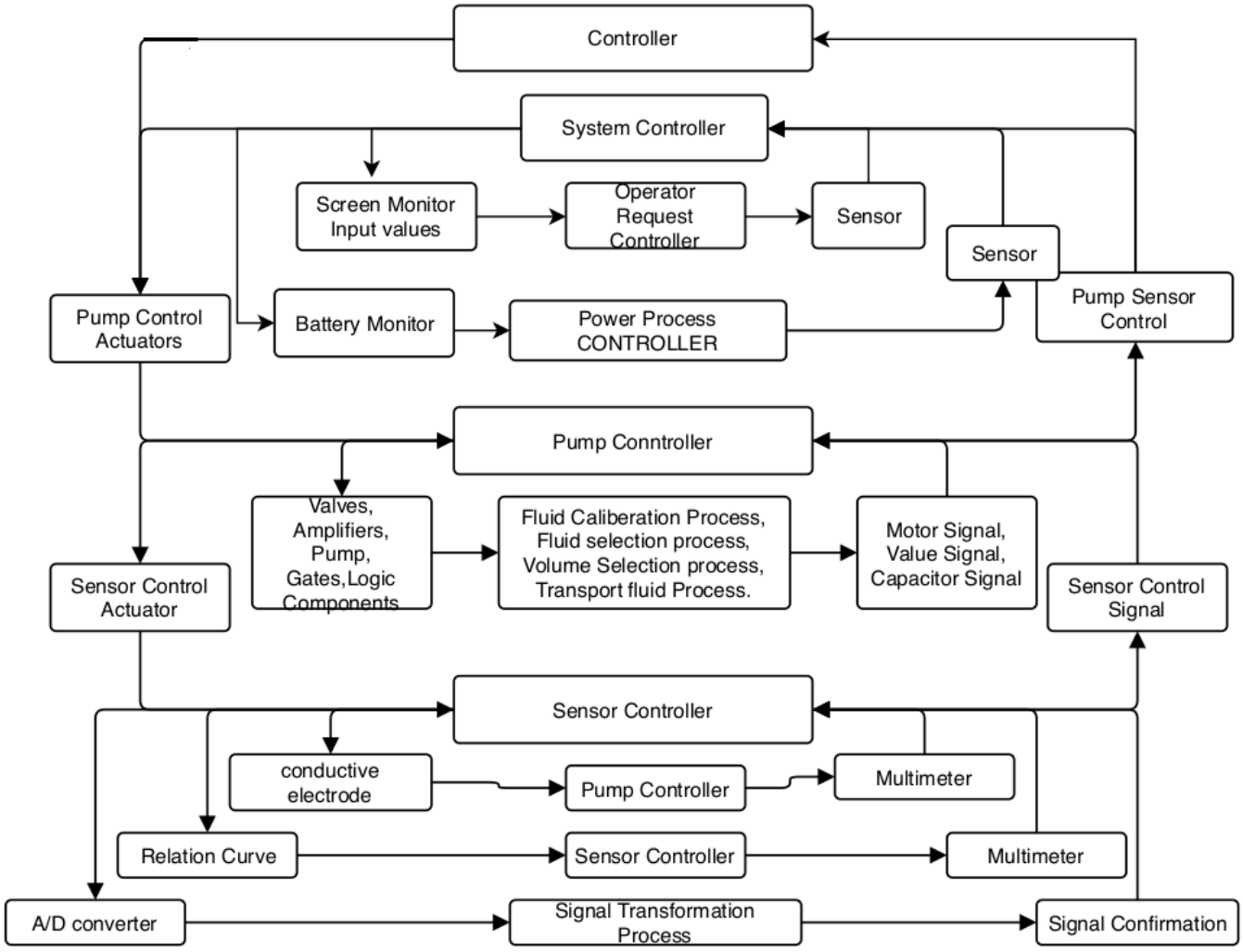

Figure 7: Full STAMP Analysis

Module	Controller	Control functions
Sensor measurement	Sensor Controller	• Recording sugar levels • Transfer readings to measured values on system scale • Analyzing of readings in relation to other parameters • Display of readings
Time and result evaluation (including dosage)	System/ human and Battery Controller	• Synchronize system clock to infusion timing • Record time of infusion • Amount of insulin in pump • Amount of insulin injected in body • Amount of insulin added in pump
Induce pump operation	Pump controller	• Rate of infusion • Method and mechanism of infusion
Evaluate body status and pumping process after infusion.	Master Controller	• Combination of all interactions if safe • External factors in infusion system • The successful execution of all process • Failure of uncontrolled components • Inconsistent system values

Figure 8: Control structure under normal condition

5.5 Identifying potential unsafe actions

We identified control loops of interest which we analyzed for factors that could contribute to a hazardous state. Leveson established several classifications of control loop deficiencies that could lead to hazards. Four scenarios were considered and represented in Figure 9 below:

1. Control action is not provided and causes a risk.
2. Control action is provided and causes a risk.
3. Control action is provided at a wrong time or order and causes a risk.
4. Control action is stopped too early or applied too long and causes a risk.

5.6 Using unidentified safe control to create safety requirements

In addition to those categories, there are several other considerations that can also elucidate potential causes of hazards. The framework for the CAST analysis, the intent is to identify the hazards that led to hazard. At this level, we have identified 51 hazards. If control tables are drawn, by going through each of the control loops, and their elements, and analyzing areas of deficiencies, many potential hazards could be further identified. In simple algorithmic terms at a ratio of 1:1, the hazards could multiply to 102. This typically shows the importance of this approach in hazard analysis.

Controller	Control functions	Control action that could lead to hazard							
		Not provided	Danger Level	Provided incorrectly	Danger Level	Too early or too late	Danger Level	Early stopping or too long	Danger Level
Sensor Controller	Recording sugar levels	Sugar level not recorded	2	Sugar level not recorded properly	2				
	Transfer readings to measured values on system scale	Readings transferred not provided	7	Incorrect values	8				
	Analyzing of readings in relation to other parameters	Readings not wholesomely provided	7	Incorrect and miss leading analysis	8	Analysis appearing late	5		
	Display of readings	Not display provided	6	Incorrect display provided	6				
System / Human and Battery controller	Amount of insulin in pump	No insulin in pump	9			Late insulin provided in pump	8		
	Amount of insulin injected in body	No insulin injected in body	10	Incorrect insulin provided	10	Insulin provided late	9	Over dose	10
	Record time of infusion	No previous time of infusion is provided	5	Incorrect times recorded	6				
	Synchronize system clock to infusion timing	No synchronization of clock to infusion timing	6	Incorrect synchronization of clock with infusion timing	6				
Pump controller	Rate of infusion	Infusion rate not provided	7	Infusion rate is terrible	7			Early stopping of infusion	8
	Method and mechanism of infusion	Method of infusion not provided	6	Incorrect and method of infusion	6				
Master Controller	Combination of all interactions if safe	All combinations of interaction not provided for safety	7	All combinations incorrectly interact	6	Combinations interact at different wrong durations	6	Effect of on interaction on another either performance stopped early or late	6
	External factors in infusion system	External factors in infusion system not provided for design	7	External factors in infusion system wrongly provided for.	7				
	Failure of uncontrolled components	Failure of uncontrolled components considered	6						
	Inconsistent system values	Inconsistent system values provided for.	6	Inconsistent system values incorrectly considered	6				

Figure 9: STAMP Analysis Table

5.7 Determining how each potentially hazardous control action could occur to enable mitigation action

These hazards are a result of the violation of the safety constraints which have been identified by the control loops. The potential occurrence of the hazards is beyond the scope of this research, as a measure has to be generated to ascertain the gravity of the hazard. It can be recommended for further research in the field of using STAMP and CAST to do hazard analysis. Many hazards were generated through this analysis, and some are directly related to the hazard. A contributing factor for safety-critical design changes.

6 Human-Machine interface design

6.1 Graphical User Interface

The graphical user interface displays the results of the simulation over a certain period of time. The program interface is designed to provide user-friendly experience [5] [6]. Current glucose measurement, past glucose measurements, previous dose amount with a timestamp, remaining insulin supply, and the current battery. Also, indicators show other information, such as whether, an insulin/glucagon dose was given, whether the glucose level is on the rise, along with fatal errors associated with sensors, and the whole injection system.

6.2 Safety Requirement

Concerning the safety aspects of the user interface, several issues were considered.
The glucose level measurement color is changed to Red, whenever the glucose level is outside

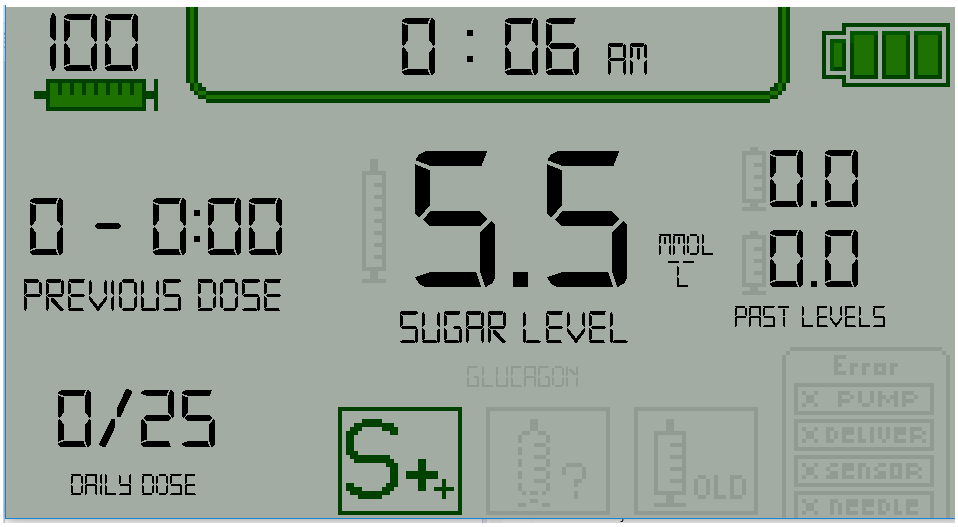

Figure 10: GUI

the safety range. For color-blind patients, this is a safety hazard, since the patient might not be able to see that there is a problem with their blood glucose. Taking this into account, when the blood sugar level becomes too high or too low. The screen changes and a new error is displayed mentioning that the blood sugar level needs to be adjusted. The issue of the battery running out of charge is treated in the exact same manner. The system also indicates the number of insulin doses administered until a certain point in time of the day. Knowing this, the user can adjust his upcoming meals accordingly, in order not to stay below the limit of maximum doses per day.

The graphical user interface consists of another display, which is relevant to medical professionals. The system outputs a graph, recording the glucose level at every hour for a certain time period. The medical professional can determine how the patient's glucose level was fluctuating, and can accordingly tell if the system was indeed providing insulin and glucagon doses when it was required.

7 Testing

The objective of the test is to verify that the functionality of Medical Insulin Pump works according to the specifications. The test will execute and verify the test scripts, identify, fix and retest all high, medium and low severity defects per the entrance criteria, prioritize higher severity defects for future fixing. The final product of the test is two-fold:

- Functional Testing: Testing team will use preloaded data which is available at the time of execution.
- Test Principles: Testing will be focused on safety, reliability and quality of the sim-

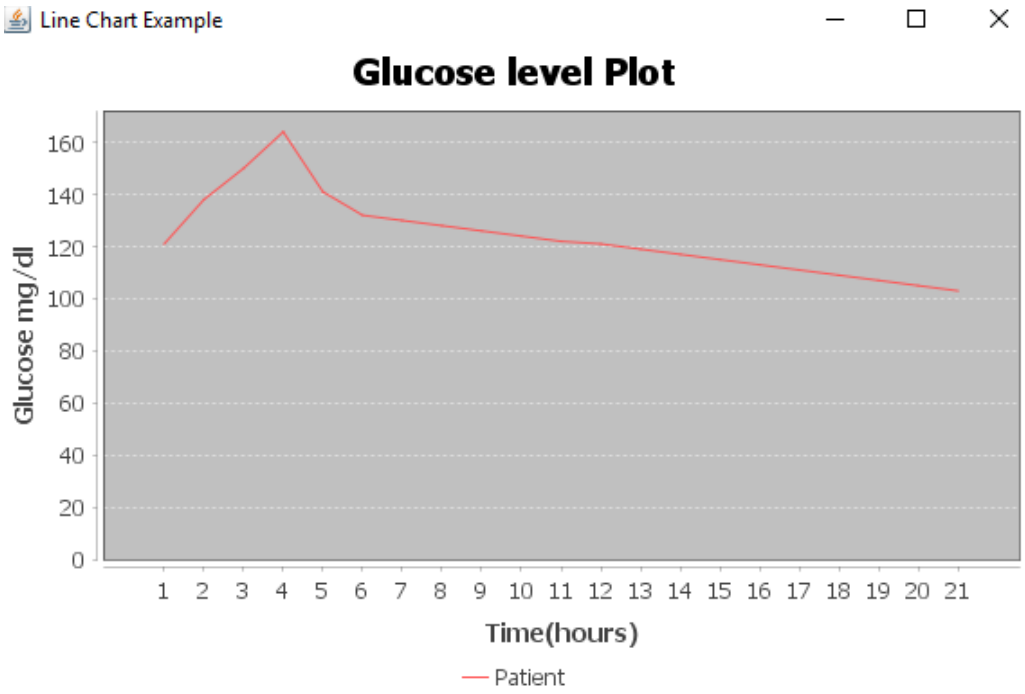

Figure 11: Glucose Level Graph

Test Case	Input	Expected Output	Actual Output
1	Needle ON	No error Message	No error Message Displayed
2	Needle OFF	Warning Message	Warning Message Displayed
3	Sensor ON	No error Message	No error Message Displayed
4	Senson OFF	Warning Message	Warning Message Displayed
5	Reservior is ON	No error Message	No error Message Displayed
6	Reservior is OFF	Warning Message	Warning Message Displayed
7	Pump is Working ON	No error Message	No error Message Displayed
8	Pump is Working OFF	Warning Message	Warning Message Displayed
9	Drain Battery	Battery Empty	Battery is Drained
10	Drain Insulin	Insulin Level Empty	Insulin is Drained
11	Recharge Battery	Battery Must Start Recharging	Battery is being Recharged
12	Refill Insulin	Insulin level must increase	Insulin level Raised
13	AutoMode	Sugar level is calculated, accordingly Insulin is injected automatically	Sugar level is calculated, accordingly Insulin is injected automatically
14	Manual Mode - Add Sugar	Sugar Level is raised in the monitor	Sugar level increases in the monitor
15	Manual Mode - Add Excess Sugar	Maximum Insulin Dosage must be reached	Maximum Insulin Dosage Reached
16	Manual Mode - Set Insulin Dosage	Insulin Dosage level is set Manually	Insulin Dosage level is adjusted by patient
17	Clock Reading	Current Time	Real Time Displayed
18	Auto Mode OFF	Switch to Manual Mode	Switched to Manual Mode
19	Turn OFF the System	Data is reset	Reset the data
20	Auto Mode - Red Zone	Goes into Red Zone	Goes into Red Zone
21	Auto Mode - Yellow Zone	Goes into Yellow Zone	Goes into Yellow Zone
22	Auto Mode - Green Zone	Goes into Green Zone	Goes into Green Zone

Figure 12: Blackbox test cases

ulator. The processes will be well defined and build upon previous stages to avoid redundancy or duplication of effort. Testing will be a repeatable, quantifiable, and measurable activity.

8 Safety Plan

Because the time for this project is short, there are many shortcomings in our simulator which we have not solved yet. First, we need to deal with functions relates to glucagon's dosage since we only assumed it is unlimited now. A database to store patient's information is also necessary since the memory of the device can be lost due to many reasons. We also consider implementing the network function and the emergency alarm to connect patients to nearby health facilities in an emergency case. Some implementations related to patient's condition recognition will be researched also because it can help predict and prevent dangerous situations when small abnormalities are found out before it becomes worse. Unlike testers that have some knowledge about the system, many patients will have trouble when dealing with modern devices so a manual should be integrated into the application when it is made.

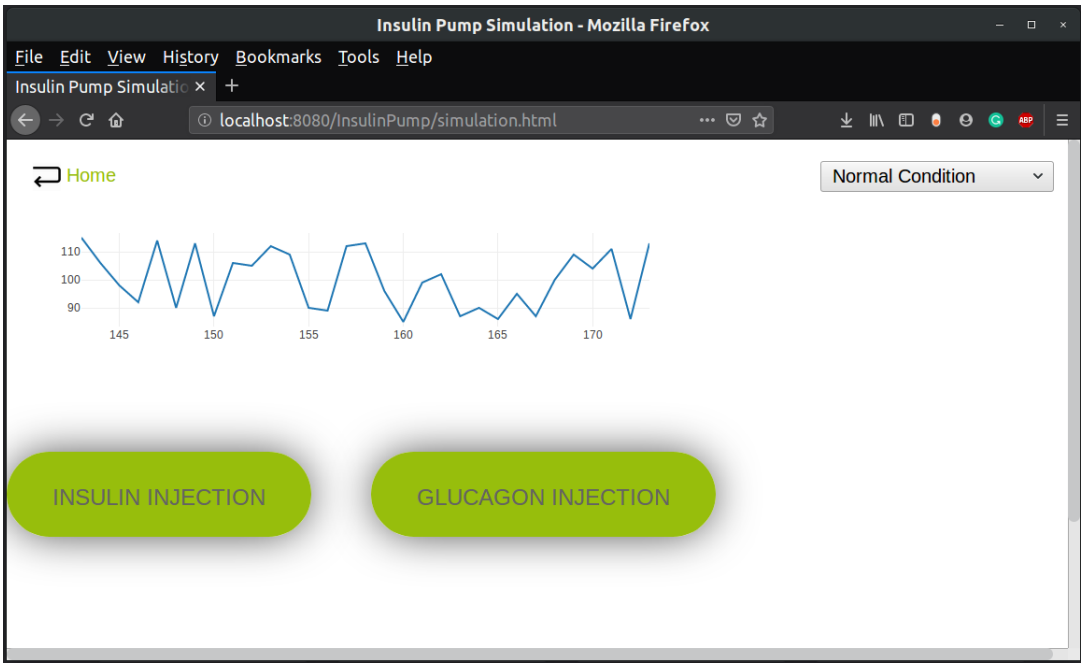

Figure 13: Web-based Backup GUI

Besides the plan mentioned above, we also created a web-based simulator (Figure 13) to simulate a web-based application to control the pump through a web interface in case there are problems with the Java application (buttons do not react, screen malfunctions...).

With this web version, user can connect to the website with their personal devices like smartphone or computer and use the network connection to control their devices. This web-based simulator has almost all the important functions as the main one such as choose events to increase or decrease glucose level and the injection buttons for insulin and glucagon function. However, the user interface is not yet completed and we have some issues with the integration process so we will not use it in the final product.

9 Conclusion

This report shows that Safety-Critical System concepts can help us streamline the application development process and increase the reliability of our program. Even though there are still many things that need to be improved like the HMI design, we can say that our final product satisfies all the use cases initially planned. Through this project, our team members learn a lot of different methods and procedures that are necessary for Safety-Critical System development.

A Source code

Main Java application: https://github.com/mohamedzayan19/Insulinpump
Web-based backup version: https://github.com/NaginataAI/InsulinPump
Last accessed date: 18/07/2019

References

[1] N. Bazaev, K. Pozhar, and P. Rudenko, "Mathematical modeling of blood glucose concentration dynamics," *Biomedical Engineering*, vol. 48, no. 6, pp. 292–296, 2015.

[2] P. Palumbo, S. Ditlevsen, A. Bertuzzi, and A. De Gaetano, "Mathematical modeling of the glucose–insulin system: A review," *Mathematical biosciences*, vol. 244, no. 2, pp. 69–81, 2013.

[3] C. Estela, "Blood glucose levels," *Undergraduate Journal of Mathematical Modeling: One+ Two*, vol. 3, no. 2, p. 12, 2011.

[4] N. Leveson and J. Thomas, "Stpa handbook," *NANCY LEVESON AND JOHN THOMAS*, vol. 3, 2018.

[5] C. Gong, "Human-machine interface: Design principles of visual information in human-machine interface design," in *2009 International Conference on Intelligent Human-Machine Systems and Cybernetics*, vol. 2, pp. 262–265, IEEE, 2009.

[6] J.-M. Hoc, "From human–machine interaction to human–machine cooperation," *Ergonomics*, vol. 43, no. 7, pp. 833–843, 2000.