SAFETY CRITICAL SYSTEM

# INSULIN GLUCAGON PUMP SIMULATOR FINAL REPORT

January 29, 2018


**Group F**:

Prakrithi Gupta: 1205312

Kalaichelvi Rajendran: 1206625

Tran Anh Khoa: 1067505

Pham Nguyen Quang: 1104840

**Supervisor**: Prof. Dr. Matthias F. Wagner

# Contents

**Abstract**

Diabetes is an illness in which the body cannot deliver insulin to control blood glucose levels. When the body cannot produce sufficient insulin to maintain the blood glucose level, it will result in the increase of the BGL (blood glucose level) rate. This causes diabetes to occur in patients which may even lead to death. Prior to the disclosure of insulin in 1921, patients who were analysed with diabetes had a future not more than one year. The insulin pump makes it workable for individuals with diabetes to live effectively.

The Insulin/Glucagon Pump is a device that gives the diabetics consistent doses of insulin/glucagon to keep blood glucose levels in control and within the normal range (between 70mg/dl and 120 mg/dl). This device is user friendly and can be inserted into the patient's body which makes it convenient and not noticeable. In the near future, we can see an artificial pancreas which will fix the problems in diabetics.

# 1 Introduction

Insulin is a hormone produced by the beta cells of the pancreas. It allows your body to utilize sugar (glucose) from carbohydrates in the food that you consume for energy. Insulin helps in keeping your glucose level from getting too high (hyperglycemia) or too low (hypoglycemia). Hyperglycemia causes Type 1 and Type 2 diabetes. In Type 1 diabetes the pancreas produces very less insulin, where the sugar starts to build up in the blood which will lead to life threatening issues. Type 2 diabetes affect people regardless of their age and the symptoms are unnoticeable, where the body becomes slow in metabolizing carbohydrates leading to high blood sugar level. Hypoglycemia is caused due to low blood sugar level, that is often caused due to the side effects of diabetes medicines.

The cells in your body require sugar for strength. Nonetheless, sugar can't go into the majority of your cells directly. After you eat food, your glucose level rises, cells in your pancreas (known as beta cells) are activated to discharge insulin into your circulation system. When the insulin is discharged, the glucose level of the body goes down in order to maintain the blood sugar level.

# 2 Problem Statement

A person whose body is unable to produce Insulin/Glucagon in the pancreas to maintain the normal blood glucose level requires a safe and secure device which can balance the amount of insulin/glucagon in the blood.

# 3 TASK DISTRIBUTION

| Task | Responsibility |
|---|---|
| Requirement Analysis | Kalaichelvi Rajendran, Pham Nguyen Quang |
| Cost Estimation | Prakrithi Gupta,Tran Anh Khoa |
| Process Model | Prakrithi Gupta,Tran Anh Khoa |
| Implementation | Kalaichelvi Rajendran, Pham Nguyen Quang |
| Mathematical Model | Tran Anh Khoa, Prakrithi Gupta |
| Hazard Analysis | Tran Anh Khoa, Prakrithi Gupta |
| Insulin Pump Design | Kalaichelvi Rajendran, Pham Nguyen Quang |
| Test Cases Design | Tran Anh Khoa, Prakrithi Gupta |
| Testing | Kalaichelvi Rajendran, Pham Nguyen Quang |
| Documentation | Kalaichelvi Rajendran, Prakrithi Gupta, Pham Nguyen Quang, Tran Anh Khoa |

**Table 1:** Task Distribution

# 4 REQUIREMENT ANALYSIS

The following are the list of conditions on which the project is based upon:

## 4.1 Functional Requirements

- Using the mathematical model, the amount of Insulin/Glucagon to be injected is analysed.
- Inject insulin/glucagon when the blood sugar reaches critical levels.
- Real time tracking of blood sugar levels, variation in blood sugar level is shown in the message box.
- Both manual and automatic operations work depending upon the selection.
- The system sends alert messages in case of emergency.

## 4.2 Safety Requirements

- Warning message pops up in the below 3 cases,

- When the insulin level in the reservoir reaches the threshold.
- When the glucagon level in the reservoir reaches the threshold.
- When the battery goes below a certain limit.
- Only authorized person can access the system considering the safety of the patient.

The range of the Blood Sugar Level (BGL):

- **Unsafe**: BGL < 70 mg/dL, Glucagon is injected.
- **Safe**: BGL between 70-120 mg/dL.
- **Unsafe**: BGL > 120 mg/dL , Insulin is injected.

## 4.3 Software Requirements

- Platform: Windows10(64 Bit).
- Programming Language: Java
- Tools: Eclipse Neon, JavaFX [9].
- GUI: Swing library.

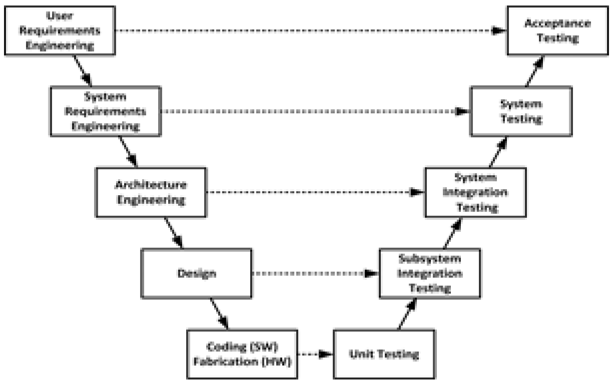
## 4.4 Graphical User Interface Requirements

- A power button will be displayed.
- Display of battery level and Time.
- Patient screen, Doctor screen.
- Insulin and Glucagon level indicator.
- Message screen for alerts.

# 5 PROCESS MODEL

The V-Model is an SDLC model where execution of procedures occurs in a consecutive way in a V-shape. It is also called Verification and Validation Model.

The V-Model is an expansion of the waterfall model and depends on the relationship of a testing stage for each corresponding development stage. This implies for each and every stage in the development cycle, there is a specifically related testing stage. This is an exceptionally restrained model and the following stage begins simply after completion of the previous stage.




# 6 COST ESTIMATION

Cost Estimation is an essential part of project management. Accurate estimation can provide good support for the decision-making process like the accurate assessment of costs can help the organization to better analyze the project and effectively manage the software development process, thus significantly reducing the risk. The comparative aggregate project cost, is utilized to approve a project's financial plan and deal with its costs. A precise cost estimate is fundamental for choosing whether to go up against a project, for deciding a project's possible degree, and for guaranteeing that tasks remains scally attainable and keep away from cost invades. Cost evaluations are commonly modified and refreshed as the project's degree turns out to be more exact and as project risks are acknowledged.

The Cost Constructive Model (COCOMO) developed by Barry Boehm which is used for the algorithmic estimation of costs involved in the development of our project. The main parameter of cost driver for project development is the effort, where effort is translated in to cost. The Primary element which affects the effort estimation is the developed kilo line of code(KLOC). Many software cost estimation models where proposed to help in providing a high-quality estimate which would be helpful in making accurate decision about the projects. In this estimation we are using a regression formula, including parameters that are derived from past project data and the characteristics of the project in hand.

The basic COCOMO equations takes the form:

- E = $a_b(KLOC)b_b$
- D = $c_b(E)d_b$
- P = $\frac{E}{D}$

where:

- E: Effort applied (in person-months)
- D: Development time (in months).
- KLOC: estimated number of delivered lines of code (in thousands).
- P: number of people required.

Calculation:

- Effort Estimation: $E$=2.4(2.8) $e^{1.05}$ Person-months
- Duration Estimation: $D$=2.5(5.38) $e^{0.38}$ months
- Person Estimation: $\frac{N}{d} = \frac{5.88}{3.89}$

# 7 Mathematical Model

Since the main aspect of the project is not to scientifically identify the metabolism of a single human body, the mathematical model implemented can be simple, yet describes considerably the change in blood glucose levels. Therefore, a model suggested by Carlos Estella [8]is taken into account:

$$G(t) = A_0 \frac{k_1}{k_2 - k_1}(e^{-k_1 t} - e^{-k_2 t}) + G_0 \quad (1)$$

with the parameters defined as below:

- $A_0$: amount of carbohydrates in the meal

- $G_0$=90 (mg/dL): the initial and target blood glucose levels.
- $k1$ = 0.0453: glycemic index, varies between different kinds of foods.
- $k2$ = 0.0224: the released rate of insulin.
- G: current blood glucose level.

In realistic cases, these parameters vary between different human bodies and different points of time. However, for simplicity, the model will be implemented with fix parameters as above to demonstrate its behavior when insulin bolus is injected. This behavior is discussed in the later part of this chapter.

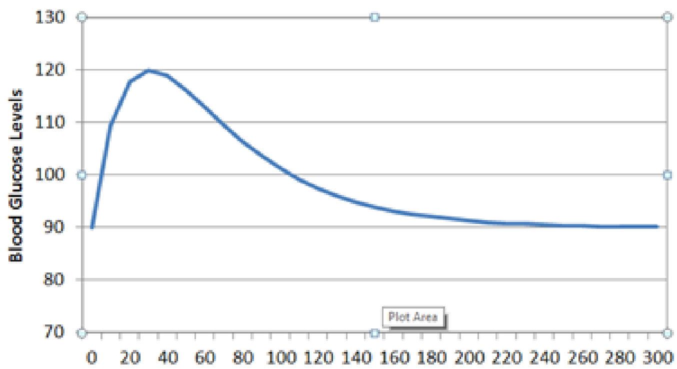
Examine the model: The peak value of the model, namely the maximum value of the blood glucose level, can be obtained within a period of time. Taking the derivation of equation (1), the time tmax can be calculated as:

$$t_{max} = \frac{lnk_1 - lnk_2}{k_1 - k_2} = 30.75 \text{ (mins)} \quad (2)$$

An acceptable range of a normal person blood glucose level is between 90mg/dL and 120mg/dL. Taking 120(mg/dL) as the peak value for the model, the value of A0 can be calculated to be:

$$A_0 = 59.67 \quad (3)$$

The graph of the model in this particular case is shown below:




Insulin Bolus and Glucagon Bolus Calculation:

The amount of insulin dose can be calculated as follow [2]:

$$B = \frac{CHO}{ICR} + \frac{G - G_0}{ISF} - IOB \quad (4)$$

where:

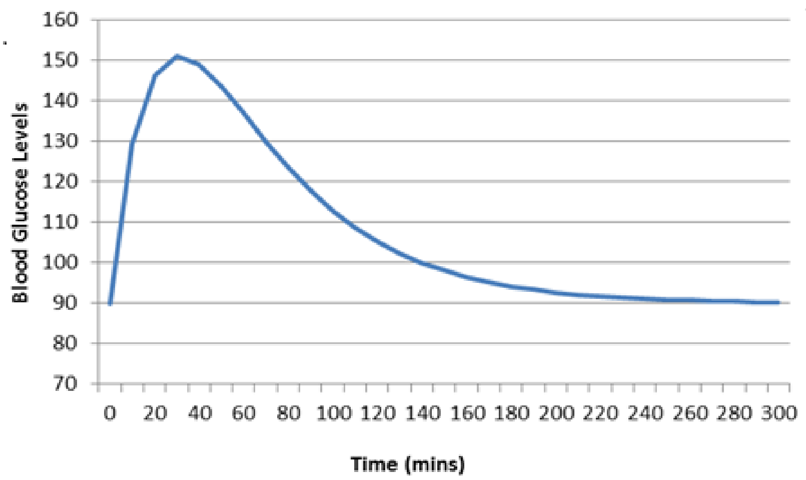
- B(IU): recommended dose of insulin.
- CHO (gram): the total amount of carbohydrate in the meal
- ICR (g/IU)= 10: insulin-to-carbohydrate ratio
- G(mg/dL): current blood glucose level
- $G_0$(mg/dL)=90: target blood glucose level.
- ISF(mg/dL/IU)= 50: insulin sensitivity factor
- IOB(IU): insulin dose left from previous injection.

Again for simplicity, we assume IOB to be equal to zero, which means all insulin doses are fully consumed in each injection. The amount of glucagon can be calculated equivalently as:

$$D = \frac{G_0 - G}{GS} \qquad (5)$$

Interactions between the two models:

Due to the complexity of human body metabolism, the fluctuation of blood glucose levels after insulin injection is manipulated. However, the interaction must consist of two crucial properties. Firstly, there should be no sudden drop or rise in the behavior of equation (1). Secondly, the blood glucose level should return to the target value, which is 90 mg/dL, approximately one hour after the injection. The manipulation is done as in the following example: Assume a person taking a meal and the amount of carbohydrates is A1=121.7. According to equation (1), his blood glucose levels will rise up to approximately 151 mg/dL in 30 minutes as shown in the graph below:



The insulin doses in this case is:

$$B = \frac{121.7-59.67}{10} + \frac{151-90}{50} = 7.423$$

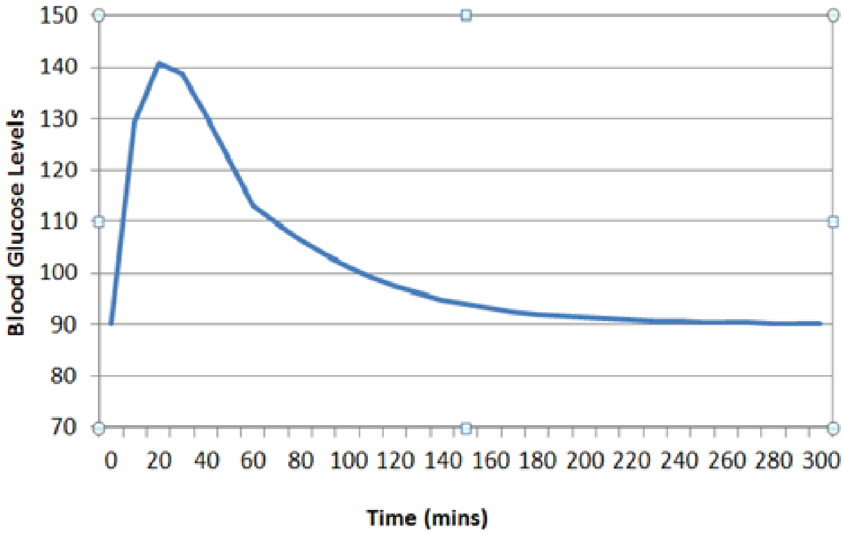
The interaction between the insulin bolus and the blood glucose model can be manipulated depending on the difference between the amounts of carbohydrates, namely A1-A0. The new model takes into consideration the new calculated amount of carbohydrates A as:

t ≤ 10: A = $A_1$

- 10 < t ≤ 20: A = $A_0 + 0.8(A_1 - A_0)$
- 20 < t ≤ 30: A = $A_0 + 0.6(A_1 - A_0)$
- 30 < t ≤ 40: A = $A_0 + 0.4(A_1 - A_0)$
- 40 < t ≤ 50: A = $A_0 + 0.2(A_1 - A_0)$

t ≥ 50: A = $A_0$

The graph of the new model is shown below:

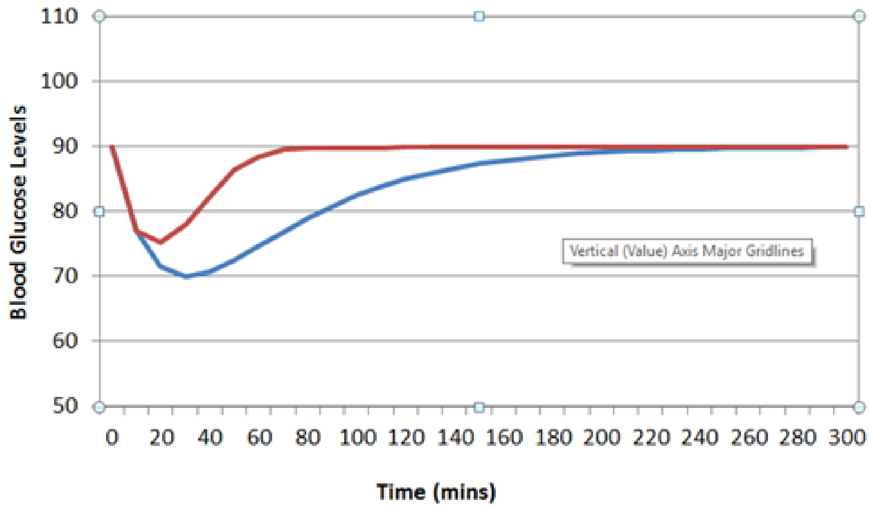



The result satisfies the two properties stated above.

In case of glucagon simulation, the model remains an exponential function of time:

$G(t) = -A_0 \frac{k_1}{k_2-k_1}(e^{-k_1 t} - e^{-k_2 t})G_0$

with the parameters defined as equation(1). The interaction algorithm resembles that of the insulin injection simulation. The result is shown in case of $A_0$=1 and $A_1$=39.83:

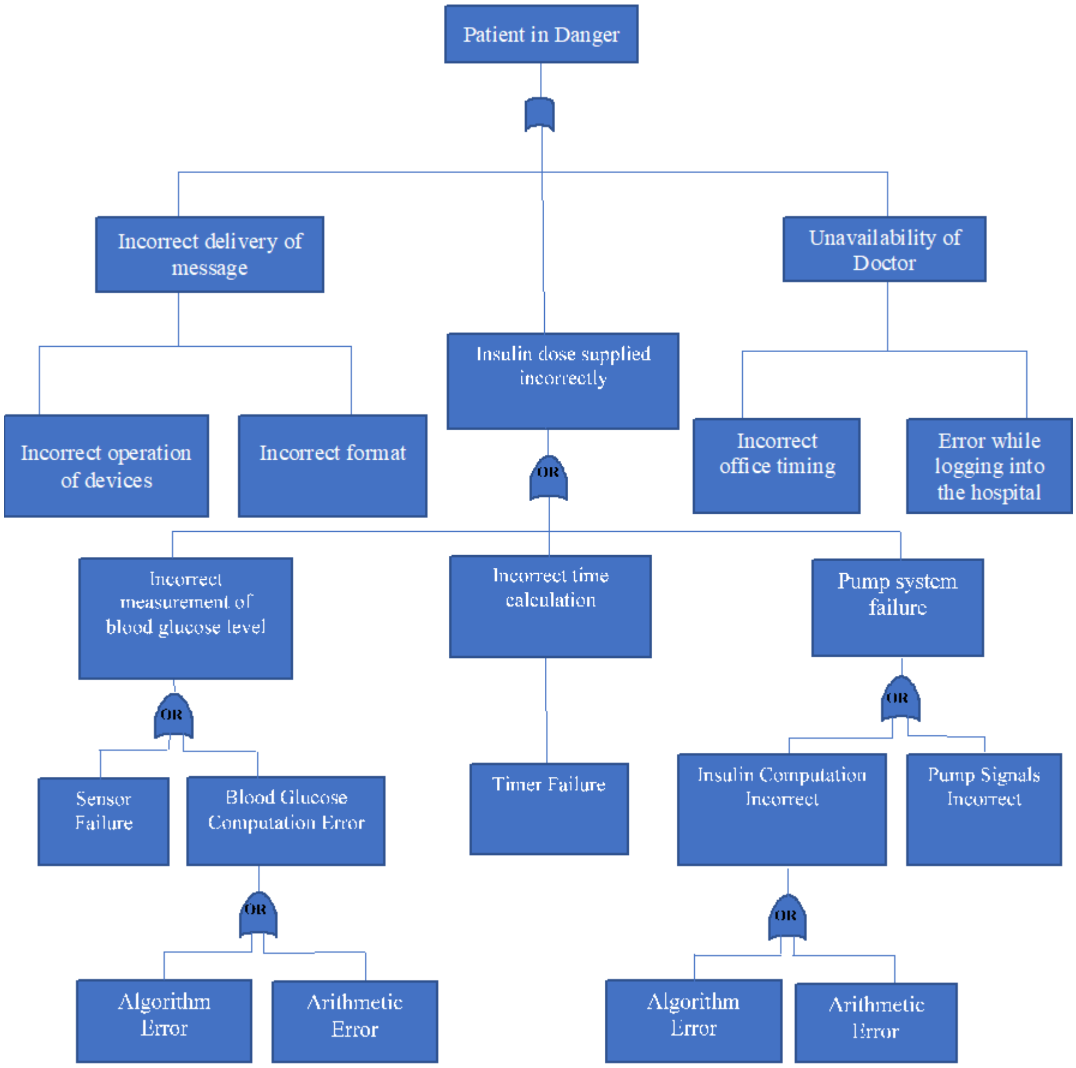



# 8 Hazard Analysis

## 8.1 Fault Tree Analysis

Fault Tree Analysis is a useful method of Hazard Analysis which helps determining the cause of failure or error. It follows a series of events logically which is used to test the reliability of a system for analyzing the cause of error.

Boolean logics are used to describe the combination of individual faults.




## 8.2 FMEA

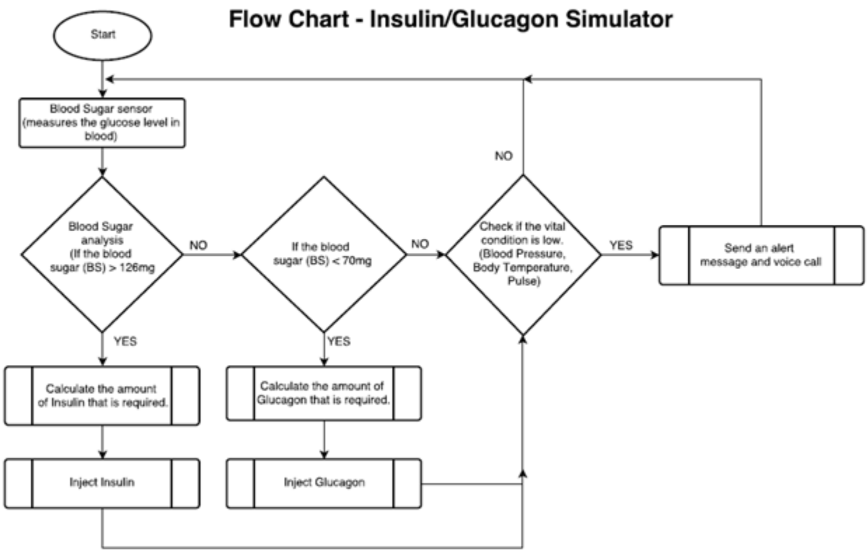
Failure Modes and Effects Analysis (FMEA) is another useful method of Hazard Analysis which identifies all possible failures in a design, a manufacturing process or a product through a step-by-step process.

| Function | Failure Mode | Failure Effect | Severity | Potential Failure Cause | Occurence | Preventive Action | RPN | Action(s) Recommended | Action(s) Taken | Detection Action |
|---|---|---|---|---|---|---|---|---|---|---|
| Hypoglycaemia | When blood sugar level goes below 70 mg/dl., Patient does not recognise alarm | Coma or Possible Death. | 10 | No balance between food, medication and activity. | 8 | Required amount of Glucagon is injected. | 160 | Proper diet, medication and activity. | Required amount of glucagon injected and continue to monitor the readings until it is back to normal. | Check Glucose Level with a Glucose Sensor. |
| Battery failure | Battery falls below threshold and causes System to shut down. | Patient Death. | 10 | Low battery quality | 5 | Frequent condition checks and Warning alerts are configured. | 100 | 1.Prior knowledge on battery life and capacity. 2.Battery backup for emergency use. | System will not proceed until the battery is back to normal. | Check Battery with a Battery Sensor. |
| Hyperglycemia | Increased thirst, Headaches,,Blurred vision,And Fatigue. | Coma, Damage to your eyes, blood vessels, or kidneys | 10 | Incorrect calculation. | 2 | Alarm Notification. | 24 | Set dose limit and change calculation. | Required amount of glucagon is injected and continue to monitor the readings until it is back to normal. | Check Glucose Level with a Sensor. |

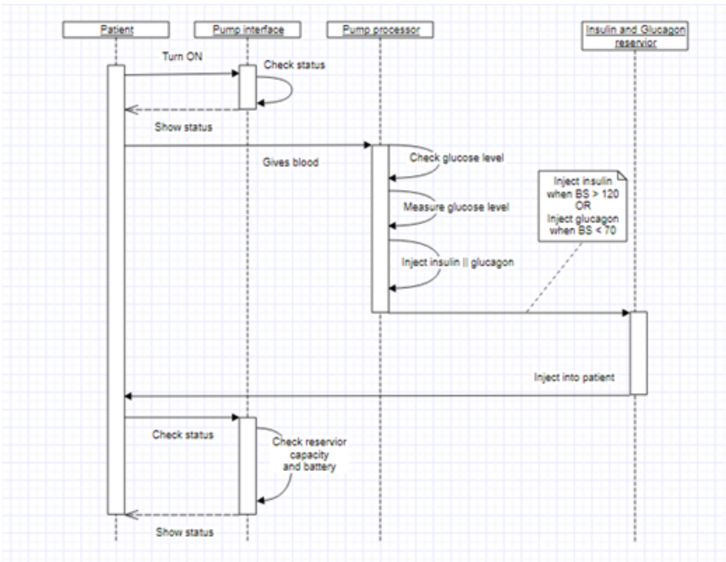
**Table 2:** FMEA

# 9 DESIGNING

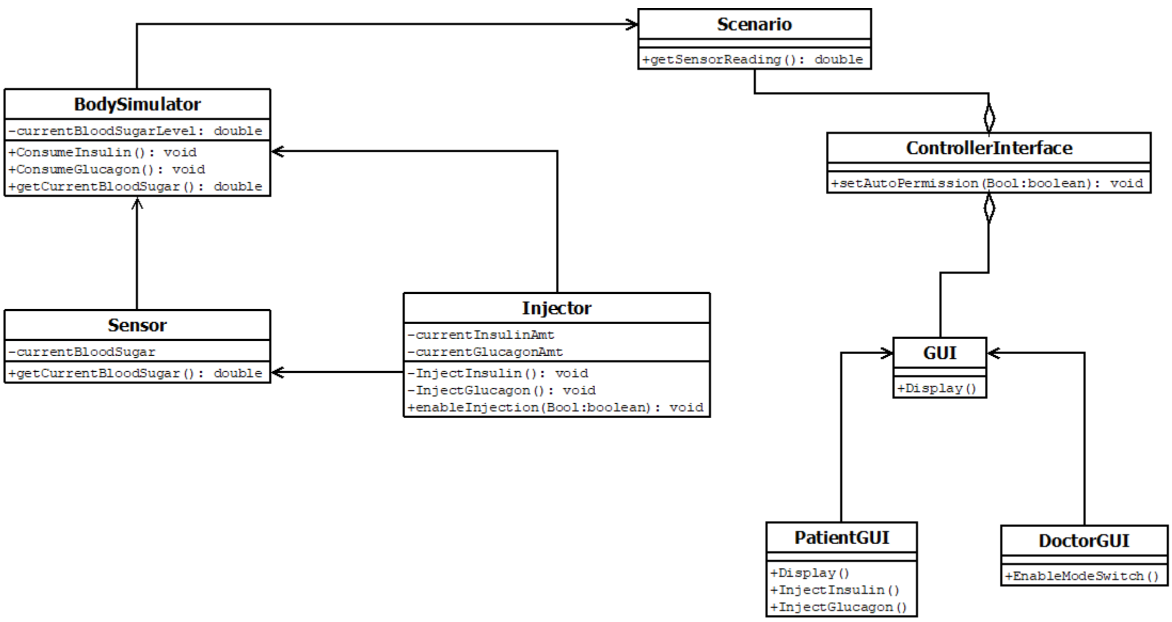
## 9.1 Flow Diagram



## 9.2 Sequence Diagram



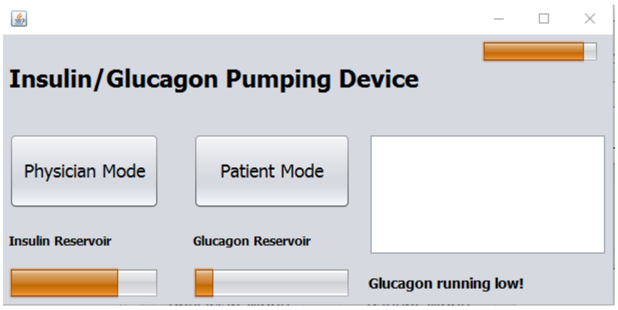

## 9.3 UML Diagram




## 9.4 GUI Design

GUI Design: The system interface is designed so that it can fit into a wrist-wearable device as shown in the figure below:

**Figure 1:** The Designed Interface

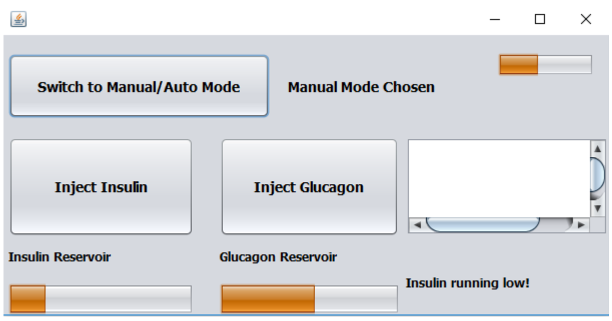



In the upper right corner, a progress bar indicates the battery level, whereas two at the bottom left corner show the amount of insulin and glucagon left in their reservoirs. The message is displayed at the bottom right corner to inform users about these parameters. For example, in this case, the amount of glucagon is running low and a warning is presented.

User can also choose between "Physician Mode" and "Patient Mode" with different functionalities by clicking the two buttons with the respect names. A text area is also included in order to inform users about actions need to be done or already done. For example, a message "need to inject insulin" or "insulin injected" may appear depending on the mode chosen inside the "Patient Mode".

Patient Mode:

**Figure 2:** Patient Mode Interface



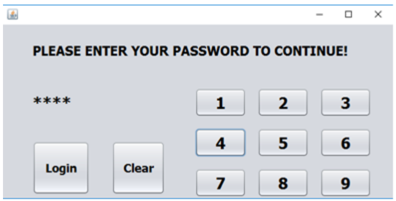

In "Patient Mode", user can switch between manual and automatic mode by pressing the button indicated. It is compulsory to design this feature, since the blood glucose levels can vary outside the acceptable range in the time the patient is sleeping.

If "Manual Mode" chosen as in the above figure, both "Inject Insulin" and "Inject Glucagon" buttons are disabled. On the other hand, if "Auto Mode" is chosen, the user can choose to do appropriate actions based on different scenarios.

Another text area is used to display the patient blood glucose levels over time. The warning message and progress bars resemble those of the starting interface.

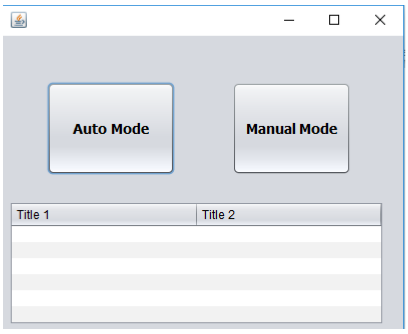
Physician Mode:

**Figure 3:** Log in Interface




In order to enter "Physician Mode", the physician must log in using his password. Passwords in this case can only be sequence of numbers from 1 to 9. A "Clear" button is designed in case a wrong number is mistakenly entered.

**Figure 4:** Physician Mode Interface




After logged in, a table of critical values of blood glucose level with respect to time is displayed.

The two buttons here are for a physician to override the mode chosen in "Patient Mode". For example, a patient with dementia should never use "Manual Mode". Therefore, his respective physician can override his selection by clicking "Auto Mode" button in this interface.

# 10 TESTING

| Module Name | Title | Test Scenario | Expected | Actual | Status | Comments |
|---|---|---|---|---|---|---|
| Patient/Physician Mode Screen | Action Button Patient and Physician Mode | Trigger the Patient Mode Button | New window showing the patient screen should open | New window showing the patient screen is opened | Passed | Working as expected |
| Patient/Physician Mode Screen | Action Button Patient and Physician Mode | Trigger the Physician Mode Button | New window showing the physician screen should open | New window showing the physician screen is opened | Passed | Working as expected |
| Patient Mode Screen | Manual insulin injection | The patient should be able to inject Insulin when blood glucose level is more than 120 mg/dL. | Insulin should be injected | Insulin is injected | Passed | Working as expected |
| Patient Mode Screen | Manual glucagon injection | The patient should be able to inject Glucagon when blood glucose level is less than 70 mg/dL. | Glucagon should be injected | Glucagon is injected | Passed | Working as expected |
| Physician Mode Screen | Login Page | Physicians should be able to log-in to handle physician mode | When correct passcode is entered, the physiscian should be logged in | The physician is logged in when correct passcode is entered | Passed | Working as expected |
| Patient Mode Screen | Automatic Insulin Injection | Insulin injection is done automatically when blood sugar level is more than 120 mg/dL | Insulin should be automatically injected | Insulin is injected automatically | Passed | Working as expected |
| Patient Mode Screen | Automatic Glucagon Injection | Glucagon injection is done automatically when blood sugar level is less than 70 mg/dL | Glucagon should be automatically injected | Glucagon is injected automatically | Passed | Working as expected |
| Physician Mode Screen | Log in Page | Unsuccessful log-in | The physician should not be logged in when incorrect passcode is entered | The physician is not logged in when incorrect passcode is entered | Passed | Working as expected |
| Physician Mode Screen | Disabe Automatic/ Manual mode switching | The physician should be able todisable automatic/ manual mode switching | Patient's permission to switch between automatic and manual mode should be disabled | Patient's permission to switch between automatic and manual mode is disabled | Passed | Working as expected |
| Physician Mode Screen | Enable Automatic/ Manual mode switching | The physician should be able to enable automatic/ manual mode switching | Patient's permission to switch between automatic and manual mode should be enabled | Patient's permission to switch between automatic and manual mode is enabled | Passed | Working as expected |
| Patient Mode Screen | Selecting Manual Mode | The patient selects manual mode | Manual mode should be enabled and automatic mode should be disabled | Manual mode is enabled, automatic mode is disabled | Passed | Working as expected |
| Patient Mode Screen | Selecting Automatic Mode | The patient selects automatic mode | Automatic mode should be enabled and manual mode should be disabled | Automatic mode is enabled, manual mode is disabled | Passed | Working as expected |



**Table 3:** Test Cases

The software was tested based on the test case document which was specially designed for this project. Most of the functional behavior of the software was tested using its GUI.

# 11 FUTURE IMPROVEMENT

Artificial Pancreas is the future for treating a diabetic patient and maintaining the normal blood glucose level. It can be described as a device used by patients which are having type 1 diabetes. It checks the blood glucose in a patient's body and then automatically adjust the amount of insulin which needs to be inserted into the patient's body.

Given the complexity of the human body metabolism mechanic, this software could be further improved to take in account into the simulation of several parameters such as the patient's heart condition, different types of insulin and glucagon for different treatment schematic.

Given the unpredictable behaviour of human beings, the software should taken account into scenarios where the patient will make an unpredicted action.

# 12 CONCLUSION

The project was successfully completed within the allocated time and satisfying the project requirements. The bugs detected based on the test cases designed for this project were solved successfully. However, considering the criticality of this software as it will be used in a life support mechanism, the functionality of this software can only be guaranteed if the possible further improvements mentioned in the previous section is made. Further testing needs to be conducted thoroughly on a hardware in order to improve reliability of this software.

# REFERENCES


- [1] Introduction:
  - https://www.medicinenet.com/hyperglycemia/article.htm last accessed: 26/01/2018
  - https://www.medicinenet.com/type_2_diabetes_pictures_slideshow/article.htm last accessed: 26/01/2018
  - https://www.medicinenet.com/type_1_diabetes_pictures_slideshow/article.htm last accessed: 26/01/2018
- [2] Cost Estimation: Liming Wu. The comparison of the software cost estimating methods. University of Calgary, pages 2âĂŞ12, 1997
- [3] Process Model (Image): https://insights.sei.cmu.edu/sei_blog/2013/11/using-v-models-for-testing.html last accessed: 26/01/2018
- [4] Hazard Analysis:http://www.conceptdraw.com/examples/hazard-analysis last accessed: 26/01/2018
- [5] Testing:http://softwaretestingfundamentals.com/test-case/ last accessed: 26/01/2018
- [7] Future improvement: https://futurism.com/2018-may-be-the-year-of-the-artificial- last accessed: 26/01/2018
- [8]F. Islam, A. H. A. Khan, N. Singh and A. A. Ali, "Advanced simulation of an insulin & glucagon pump,"2016 International Conference on Information Technology (InCITe) - The Next Generation IT Summit on the Theme - Internet of Things: Connect your Worlds, Noida, 2016, pp. 197-202.
- Git Repository: https://github.com/phamnguyenquang/SCS