



Portland State  
UNIVERSITY

## **Homework 5**

Fall 2016

Forrest Kaminski

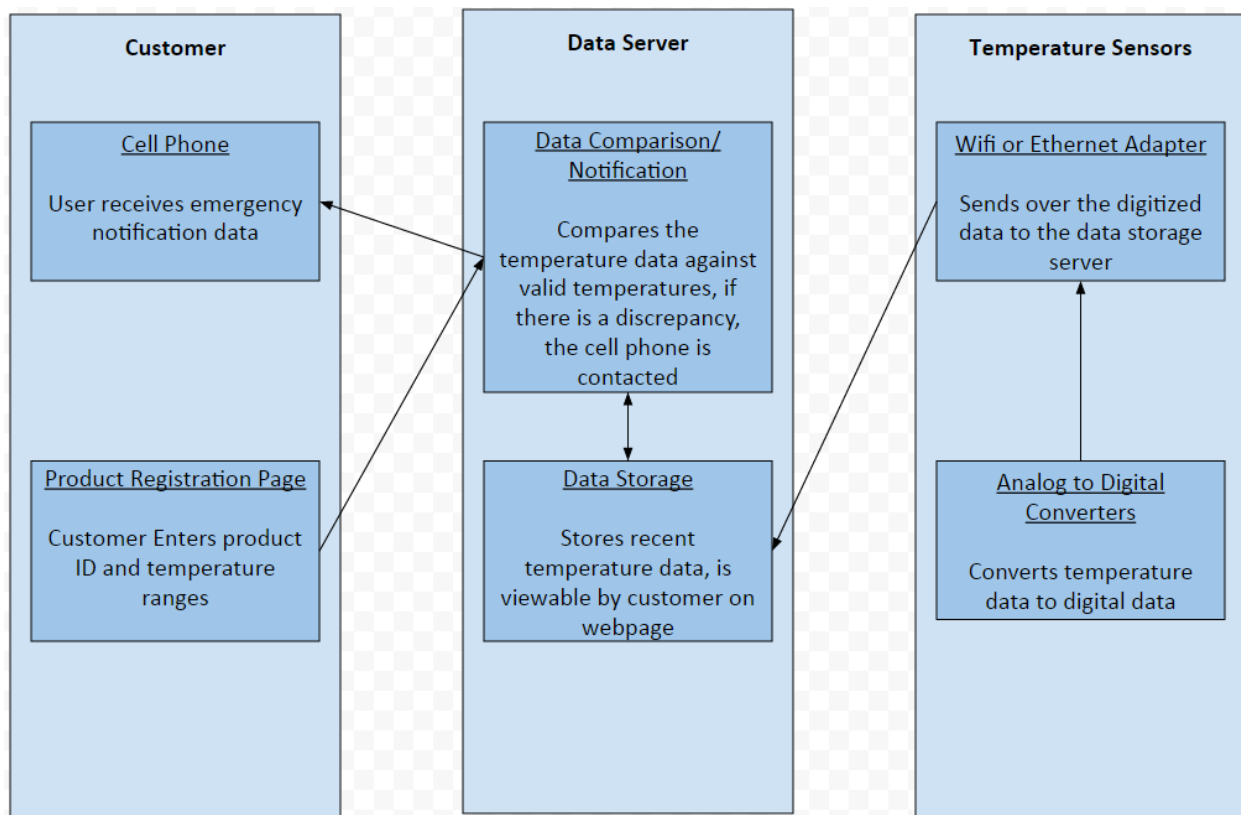
Thuan Pham

Justin Morgan

ECE 411

**Group # 12**

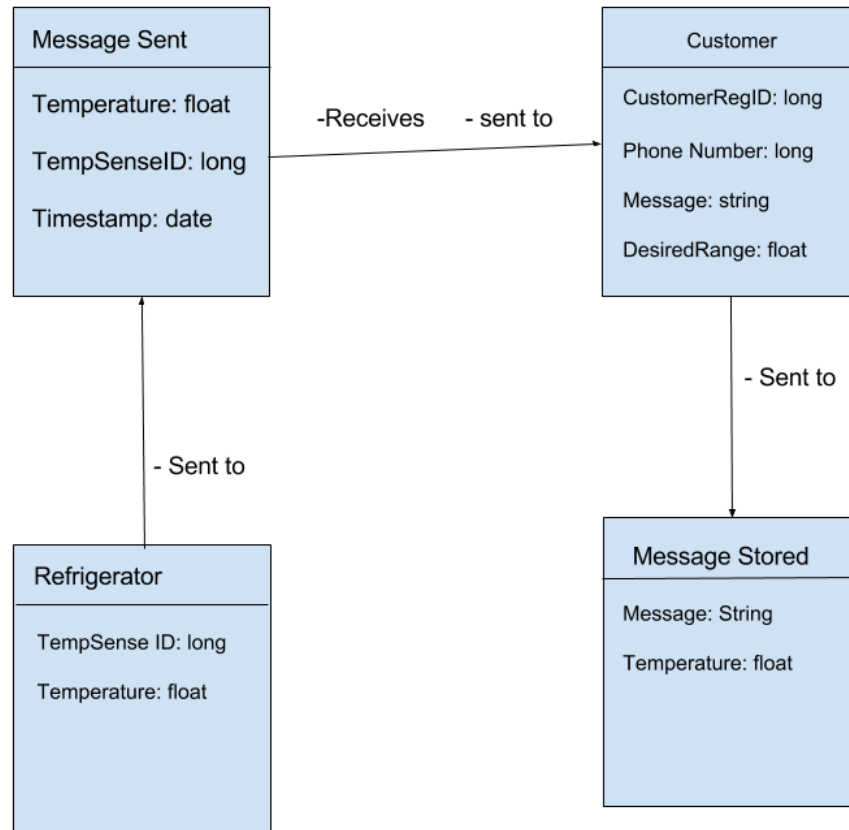
## UML Physical View



*Fig 1: UML Physical View*

In this system, the customer first has to register a product, which will define a product's ID and acceptable ranges, as well as a cell phone contact number. Once the product is registered and installed, the temperature sensors begin sending data to the data server, where information is compared against user entered ranges, if the received data exceeds the range, the customer is contacted via text on their cell phone.

## UML Server Class Diagram



*Fig 2: Server Class Diagram*

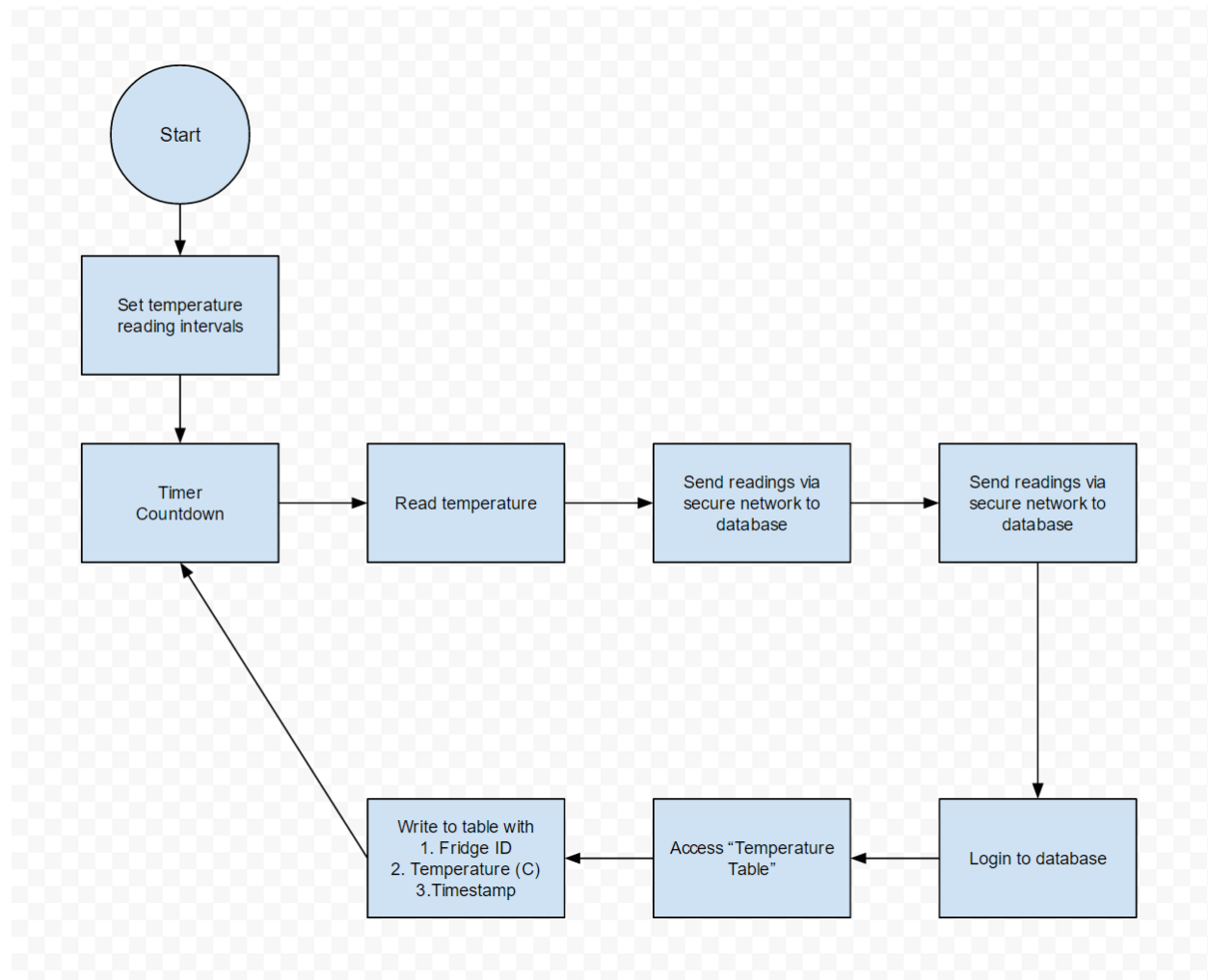
There are four distinct classes in this system. We start with the Refrigerator itself. The ID contains is the temperature sensor, and the value of the temperature itself. The Temp and ID is sent to the "Message Sent" class. This is a delegate that will determine whether or not to notify customer. If it does, it will sent to customer, and also to message stored class, for archival purposes.

## UML Use Case

<b>Use Case</b>	<p><i>Registering a new temperature sensor to a new or existing customer with a phone number:</i></p> <ul style="list-style-type: none"><li>• The customer is a new customer and is registering their information to the data server, and connecting a single temperature sensor.</li><li>• The customer is registering another product to their existing customer information</li></ul>
<b>Description</b>	<p><i>Register a new product, store temperature sensor ID and link it to a customer:</i></p> <ul style="list-style-type: none"><li>• If the customer doesn't exist in a database, the end of the use case should result in the customer having a registered emergency textable phone number, temperature sensor that they can view recorded data, and relevant login information.</li><li>• If the customer exists, they should end the use case with having an additional temperature sensor linked to their account.</li><li>• Alternatively, view recorded data from your sensors.</li></ul>
<b>Actors</b>	<ul style="list-style-type: none"><li>• Hospital or vaccine caretaker (the customer), as well as the automated server.</li></ul>
<b>Assumptions</b>	<ul style="list-style-type: none"><li>• Customer owns at least one temperature sensor with a valid ID.</li><li>• Customer owns a phone capable of receiving text messages.</li><li>• Customer's clinic has internet access that can reach the fridge either by wifi or ethernet.</li></ul>
<b>Steps</b>	<p><i>For adding a new customer w/ sensor:</i></p> <ol style="list-style-type: none"><li>1. Navigate to server web page and click on the create new account.</li><li>2. Create login credentials.</li><li>3. Navigate to connect a temperature sensor and enter the temperature sensor's product ID, lower and upper bounds as well as a product nickname for easy identification.</li><li>4. Navigate to connect a phone number, enter the number and wait for a verification text stating product &lt;nickname&gt; has been registered.</li><li>5. Once received, end the use case.</li></ol>

	<p><i>For adding a new temperature sensor:</i></p> <ol style="list-style-type: none"><li>1. Login to server web page using your credentials.</li><li>2. Go to step 3 of the adding a new sensor and customer.</li></ol> <p><i>For viewing data:</i></p> <ol style="list-style-type: none"><li>1. Login using your credentials.</li><li>2. Navigate to view recorded data.</li></ol>
--	---

## UML Temperature Activity



*Fig 3: Temperature Activity*

This UML temperature activity diagram shows us the process in which the temperature sensor reads the temperature from the fridge and then sends the data over wifi to a database. Once at the database it will write into the database table labeled temperature which will record the readings down with the fridge id, temperature and timestamp of which the data was recorded. From there the database will have the data stored and the timer will restart and will loop until the user finishes recording.

## **HTTP Get Specification**

[www.methlab.gov/customerid=1234/?fridgeid=101&temp=30&time=20161117084011](http://www.methlab.gov/customerid=1234/?fridgeid=101&temp=30&time=20161117084011)

So this HTTP Get string will be the basis for our backend servers which will record all the data necessary from our temperature sensors. Using this link the fridge will access a database using the customerid and then it will write to the table with fridgeid, temp and timestamp. Each fridge will correspond to a different id. By having different fridge id's we can later group up each fridge to their temperature readings and sort out the data.