**Analysis Workflow**

**Step 1: Functional Modeling**

**Our team has one simple use case:**

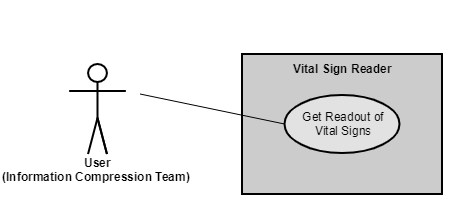


Figure : Use-case diagram for the Vital Signs program.

**There is one scenario for this use case:**

Figure : Normal scenario of the use case.

1. User requests readout of temperature, heart rate, systolic blood pressure, diastolic blood pressure, and blood oxygen level.
2. The temperature is generated and evaluated.
3. The heart rate is generated and evaluated.
4. The systolic blood pressure is generated and evaluated.
5. The diastolic blood pressure is generated and evaluated.
6. The blood oxygen level is generated and evaluated.
7. Evaluations of all 5 vital signs are combined into one result, which is returned to the user.

**Step 2: Entity Class Modeling**

**Noun-Extraction Method**

**Stage 1. Describe the Software Product in a Single Paragraph**

A vital sign reader will generate readings for each vital sign. These readings will be evaluated to determine if they fall in the normal, low, or high range for each vital sign. The evaluations will be combined into one result.

**Stage 2. Identify the Nouns**

1. Vital sign reader (**We select this as an entity class)**
2. Readings, range, evaluations, result **(These abstract nouns might be used as attributes)**

**Initial Class Diagrams**

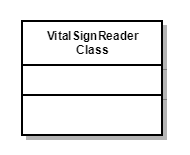


Figure : Initial class diagram.

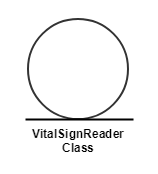


Figure : Initial class diagram showing UML stereotype for entity class.

**Step 3: Dynamic Modeling**

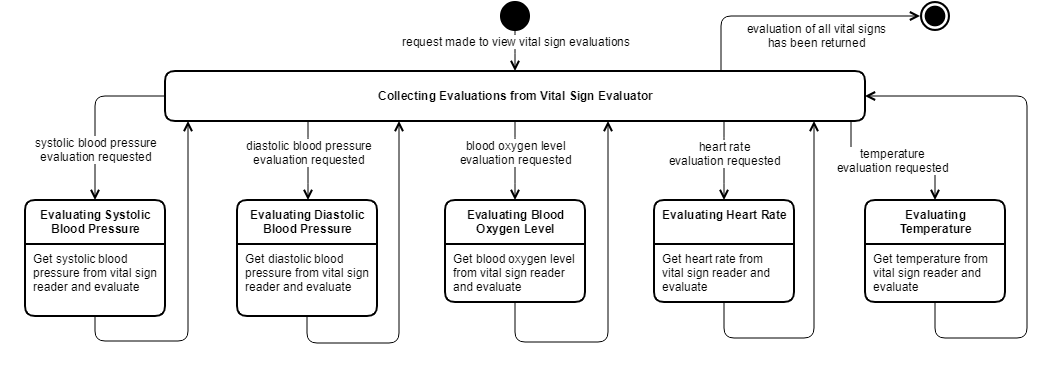
**Statechart:**

Figure : Statechart for Vital Signs program.

**Step 4: Extract the Boundary and Control Classes**

**From the statechart, we see that in addition to the VitalSignReader entity class, we need:**

1. **VitalSignViewer boundary class**
2. **VitalSignEvaluator control class**

**Revised Class Diagrams**

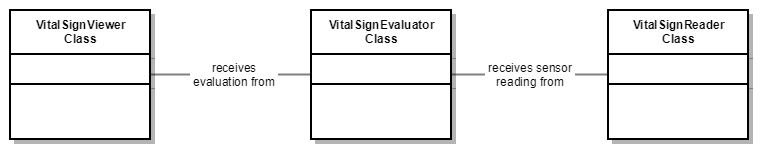


Figure : Revised class diagram.

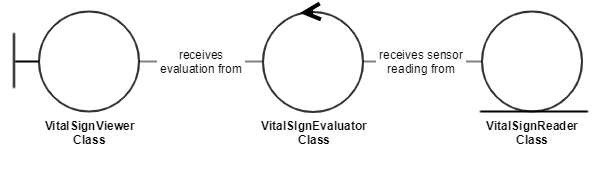
****

Figure : Revised class diagram showing UML stereotypes for boundary, control, and entity classes (left to right).

**Step 5: Use-Case Realization**

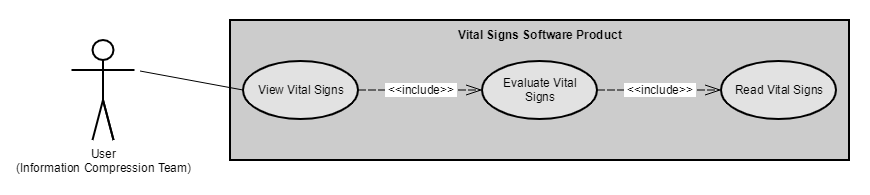
**Refine Use Case:**

Figure : Revised use-case diagram for the Vital Signs program.

Figure :The description of the View Vital Signs use case.

**Brief Description**

The View Vital Signs use case enables the user to view a combined evaluation of 5 different vital signs for the patient.

**Step-by-Step Description**

1. For each of the 5 vital signs:
   1. Evaluate the vital sign.
      1. Get a numerical reading for the vital sign.
      2. Based on the reading, determine if the vital sign is in the normal, high, or low range.
2. Combine the evaluations into one summary result and return to the user.

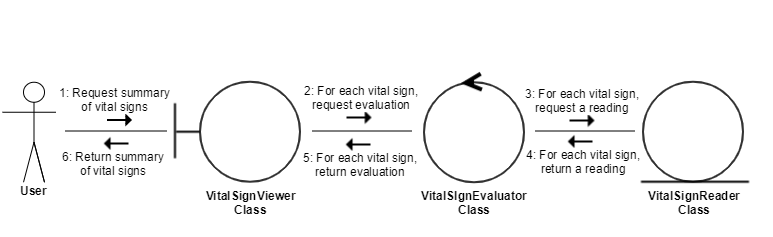
**Communication Diagram:**

Figure : A communication diagram of the realization of the use case View Vital Signs for the Vital Signs software product.

**Flow of Events:**

Figure : The flow of events of the communication diagram in Figure 10.

The information compression team requests a summary of the patient’s vital signs (1). For each vital sign, an evaluation is requested (2). To complete each evaluation, a reading of the vital sign must be taken (3) and returned (4), and then the evaluation for that vital sign can be returned (5). Finally, a summary of all five vital signs is returned to the compression team (6).