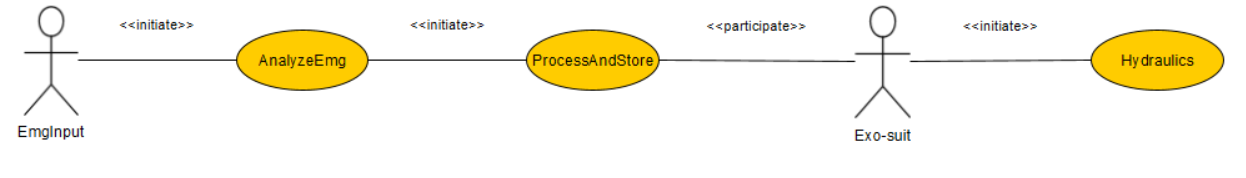
**B. Requirements Analysis**

**3.6 System Models**

**3.6.1 Use case model**



*Fig: 3.6.1 Use case model*

**3.6.2 Object model**

**3.6.2.1.I Identifying Boundary Objects**

|  |  |
| --- | --- |
| **Boundary Object** | **Definition** |
| **AuthenicationTouch** | Mechanism used by the **User** to gain inside the system. |
| **AuthenicationRequest** | Notice received by the **DataController** for requesting access. |
| **AuthenicationReply** | Notice received by the **User** indicating weather the user has been provided access to the system or not. |
| **AppltyFilter** | Filters out the unnecessary and noisy data from the received serial data. |
| **PerfomFFT** | Perfom fast fourier transform on the filtered data to find out the required frequency. |
| **InvalidData** | Sends back data back to **DataController** if data is not valid |
| **ProcessData** | Process the data to Gcode which motor could understand. |

*Table: 3.6.1.I Boundary Object*

3.6.2.1.II Identifying Control Objects

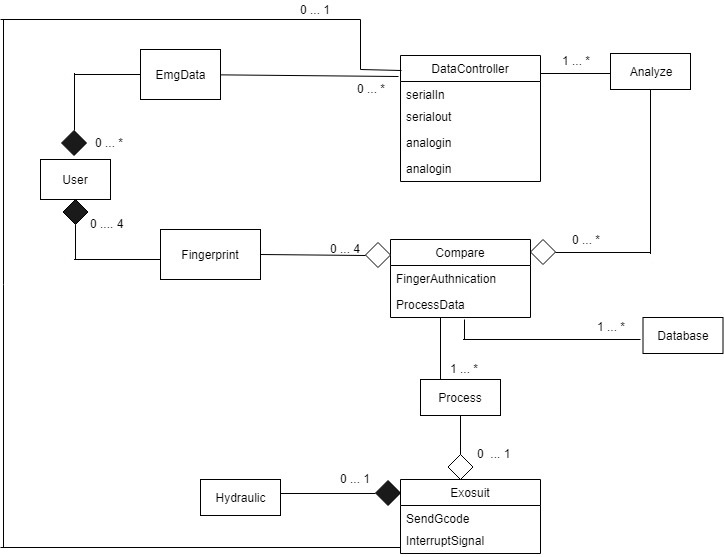
|  |  |
| --- | --- |
| **Control Object** | **Definition** |
| **SerialDataControl** | Indicates the flow of serial data from the U**ser** to the **DataController**. |
| **CompareAuthControl** | Compare authenticated data of **user** with store data on database. |
| **PowerSpectrumControl** | Perform the power spectrum analysis to calcite the power factor the **user** at the required frequency. |
| **CompareDataControl** | Compare the data between the **user** info and serial input with the data stored on the data base |
| **ServoControl** | Controls the servo motor for hydraulic fluid moment to control the **Exosuit.** |

*Table: 3.6.1.II Control Object*

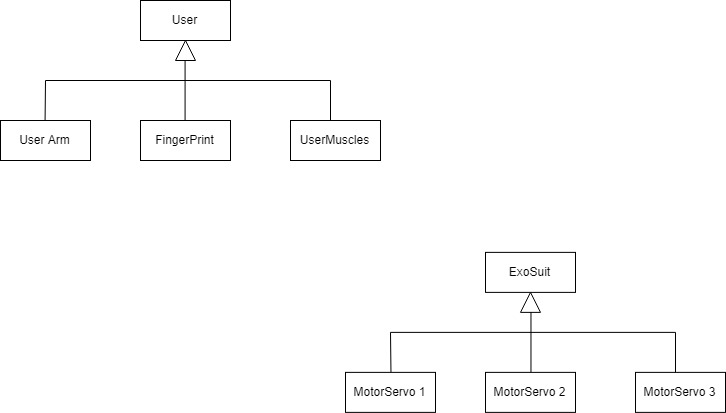
3.6.2.1.III Identifying Entity Objects

|  |  |  |
| --- | --- | --- |
| **Entity Object** | **Attributes and associations** | **Definition** |
| **User** | * Id * Name | An user represent the person/ part of muscle from whom the data is being taken and wear the suit. |
| **ExoSuit** | * Model * Name * Motor | Actor that represent the suit which used the processed data to control the suit based on various scenarios |
| **DataController** | * SerialIn * SerialOut * AnalogRead * AnalogWrite | Represent the data flow from **user** to the analysis program and program to the **suit** |

*Table: 3.6.1.III Entity Object*



*Fig: 3.6.2.a Object Diagram with link and aggregation*

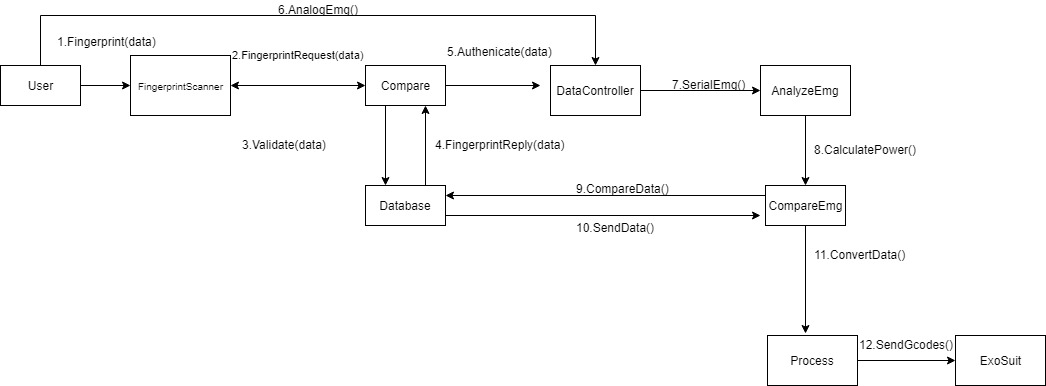


*Fig: 3.6.2.b Inheritance DIagram*

3.6.2.2 Class diagrams

(REMAING)

3.6.2.3 Interaction among objects

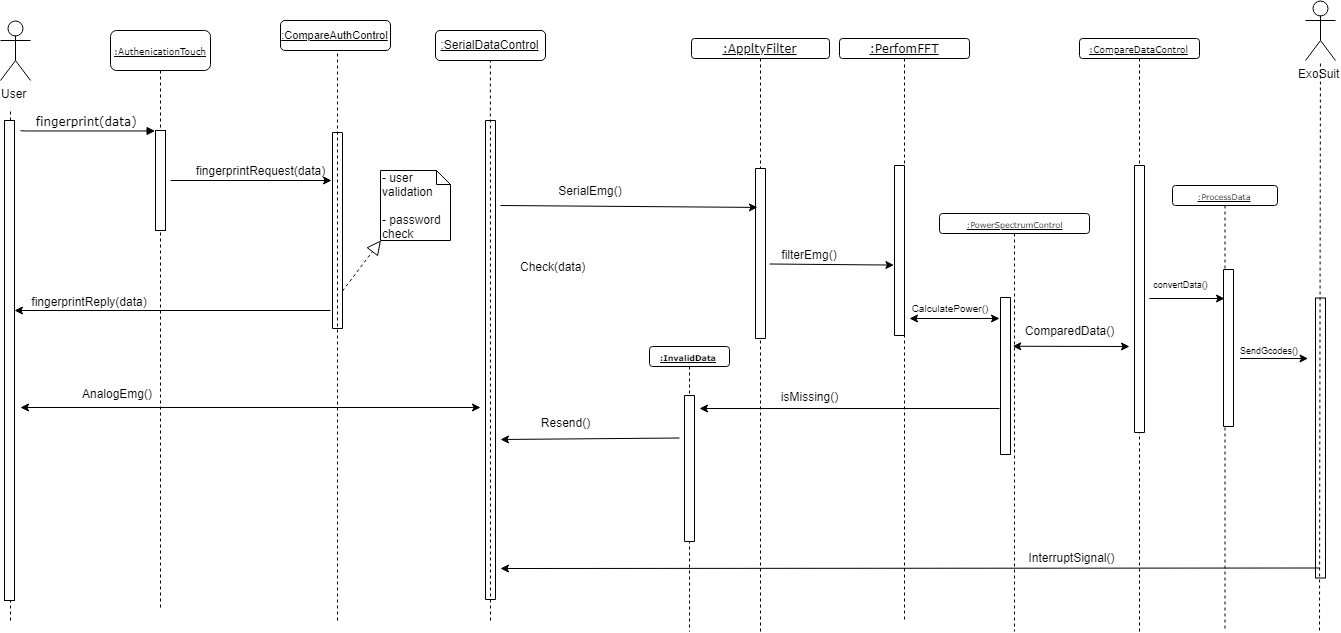


*Fig3.6.2.3.1: Sequence Diagram*

*(CRC PENDING)*

**3.6.3 Dynamic models**

**3.6.3.1 Sequence diagram**



*Fig3.6.3.1: Sequence Diagram*

**3.6.3.2 State machine diagram**

**3.6.4 User interface with navigational paths and screen mock-ups**

**3.6.5 Completeness, correctness, consistency, reality of the analysis**