## CHAPTER FOUR

**System Design**

**4.1 Overview**

This is the system design document for The Advanced Metering Reader. It includes the design goals, the proposed system design and the object design.

**4.1.1 Purpose of the System Design**

This system describes the design issues of the overall system. It provides the complete architectural overview of the proposed system. It is intended to capture and express the significant architectural to capture and express the significant architectural decisions which have been made by the system.

**4.1.2 Design Goal**

The goal of the design goal is to model the system with high quality. This will be used by the developers to describe the qualities of the system that another developer should consider. The design goal represents the best quality that the system should have and provide a consist set of criteria that would be taken into consideration when making design decisions. They are grouped into four categories. These are:

* Performance
* Dependability
* Maintenance
* End User
* **Performance**

The Advanced Metering system as well as the software should respond fast and perform the task quickly as possible. Establishing communication, data sending and receiving, execution of received data and also communication of serial data, communication sensors is very fast. The AMI software performs its task such as receiving information and displaying the results.

* **Response Time:**  depending on the available network and speed of the information sent, the system will respond in s short period of time.
* **Memory:** to do the work efficiently the AMI software has more than 50% RAM and the software should install in the processor at least more than 2GB RAM.
* **Dependability**

Our system includes the following dependability criteria’s:

* **Reliability:** The system is reliable whenever the server fails or any hardware issues may occur.
* **Availability:** AMR can operate with any weather condition as well as day and night, also as long as there is local area network and establish communication the AMR software can operate and it will be available 24 hours a day.
* **Security:** AMR software should be secured, unauthorized user cannot manipulate AMR through AMR software. The system provides security since the information is sensitive.
* **Robustness:** AMR and AMR software should be fault tolerant due to termination of communication.
* **Maintainability**

To be maintainable the system should meet the following maintenance criteria:

* **Modifiability:** AMR is modifiable for further modification just preparing new  
  embedded code as well as hardware change. Also, AMR software is modifiable like version control.
* **Portability:** the system will be developed to be viewed and retrieved from any web browser regardless of their version and platform it resides in it.
* **Extensibility:** if it is needed to add new functionality to AMR, new circuit diagram and embedded code needed for AMR. New functionality to AMR software can be achieved by only making a separate page and integrate this page with the existing system.
* **Readability: -** the system code will be written by following universal standards and comments will be including for further understanding.
* **End User**

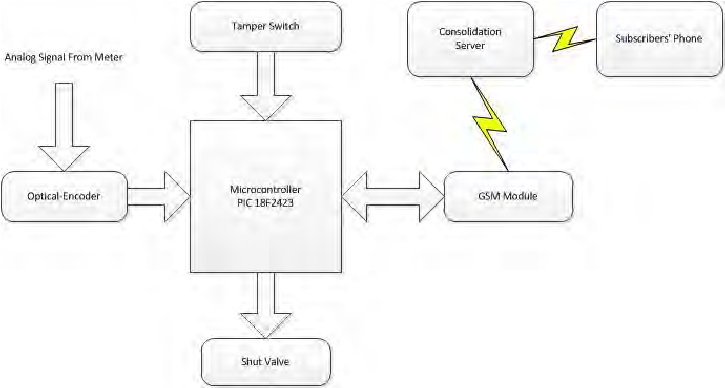
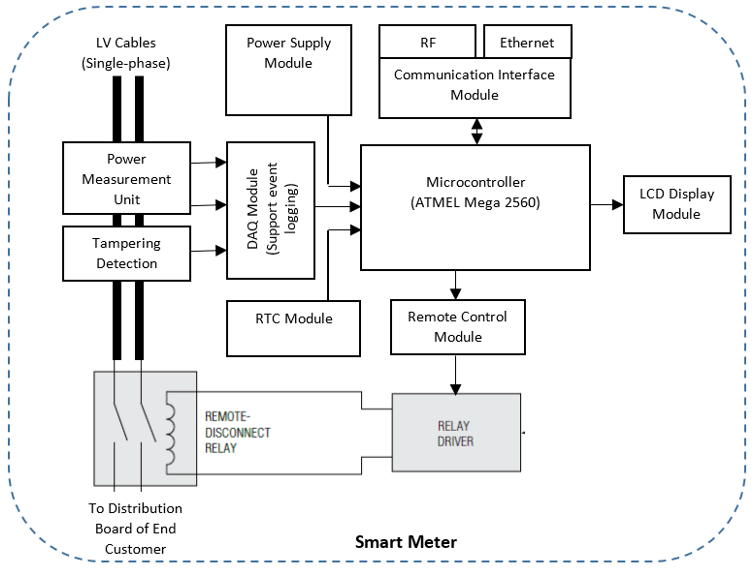
AMR software should be simple and user friendly. It has understandable graphical  
user interface such as forms and buttons, which have descriptive names. All the  
interfaces, forms and buttons are written or designed in a simple language or common  
language (English, Afaan Oromoo, and Amharic) so that the user can access it without  
any difficulty. Also, AMR may also have descriptive pushbutton.

* **Priority of design goals**

The design goals of Advanced Metering System are prioritized as follows:

|  |  |
| --- | --- |
| **Priority** | **Design goal** |
| **1** | **End User** |
| **2** | **Dependability** |
| **3** | **Maintenance** |
| **4** | **Performance** |

**4.2 Proposed system architecture**

****fig:Hardware system architecture

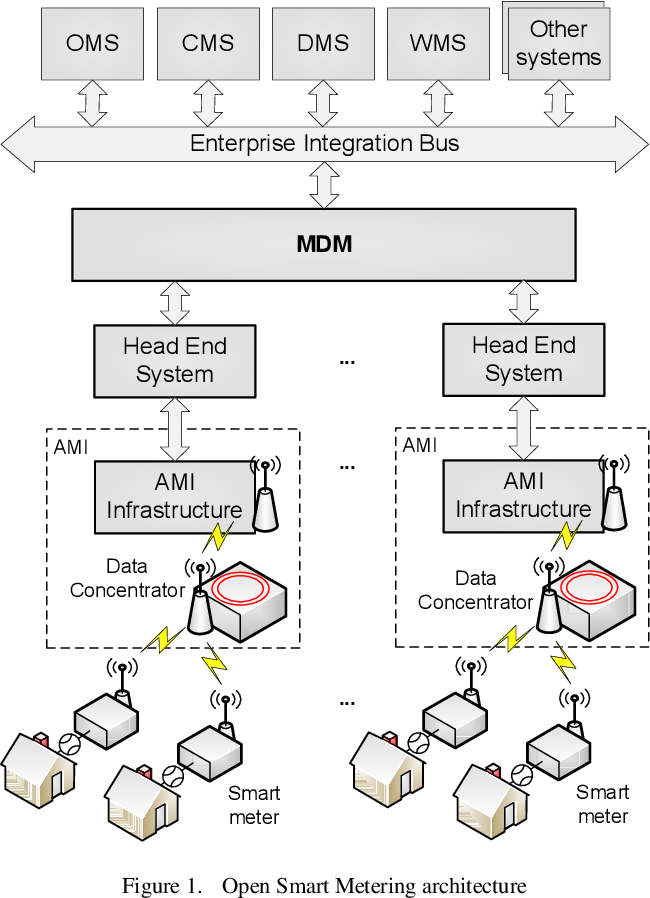
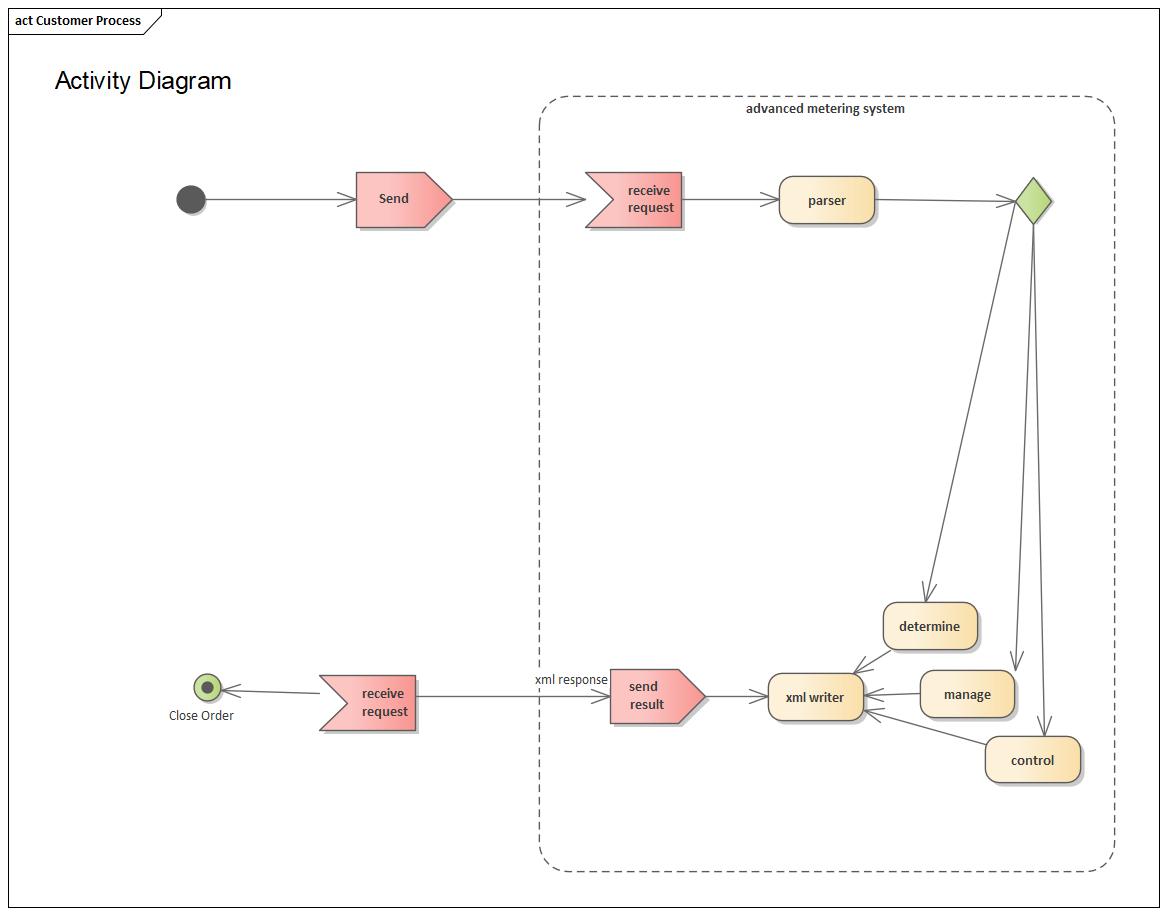
****

Fig: Software architecture

**4.2.1 System process**

****

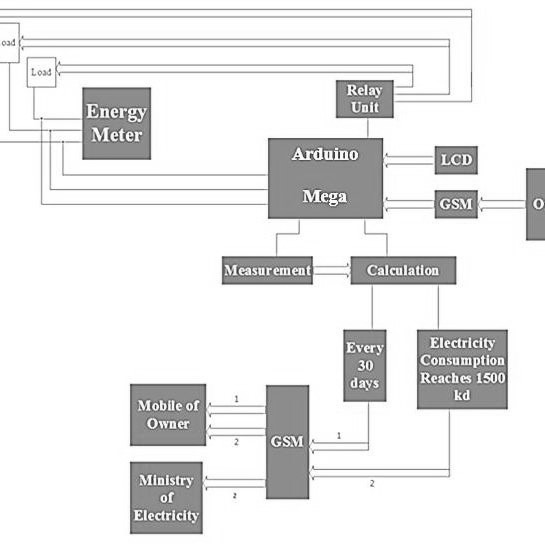


fig: AMR system processes

**4.2.3 Subsystem decomposition**

Subsystem decompositions will help reduce the complexity of the system. The subsystem can be considered as packages holding a collection of classes, associations, operations, and constraints that are closely interrelated with each other.

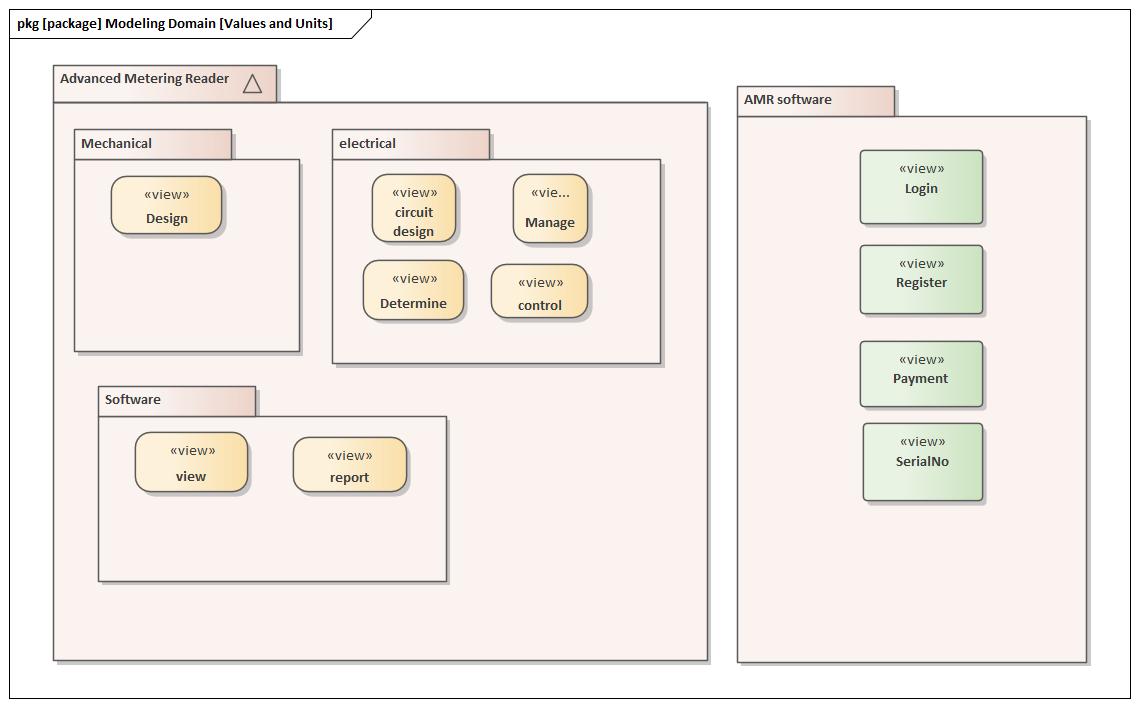
****

Fig: subsystem decomposition

**Subsystem decomposition description**

|  |  |  |
| --- | --- | --- |
| Subsystem | Purpose | Class |
| Software | Is responsible for viewing details, managing and generating report. |  |
| Electrical | Is responsible for controlling speed, movement, communication between the components, and the distance information is sent. |  |
| Mechanical | Is responsible for designing of the mechanical parts of the AMR system. |  |

**4.2.4 Hardware/software mapping**

When we say hardware/software mapping for the system, it describes how subsystems are  
assigned to hardware and off-the-shelf components. It also lists the issues introduced by multiple  
nodes and software reuse. In this system design mainly, there are three hardware components. The client side (user, AMR), server side and database side.

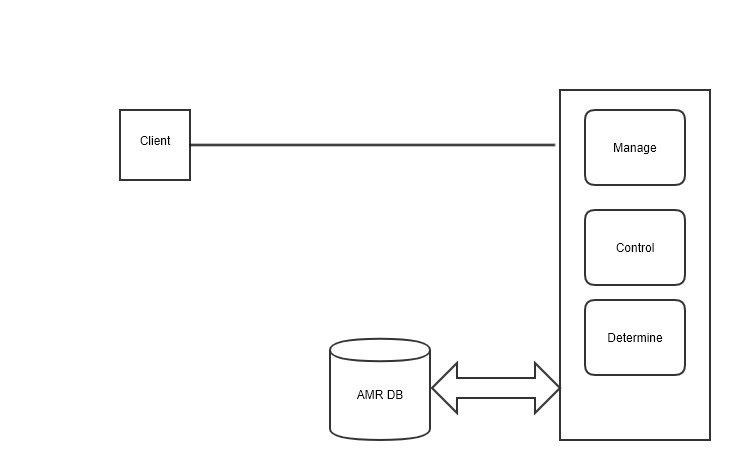


Fig: hardware/software mapping for software side

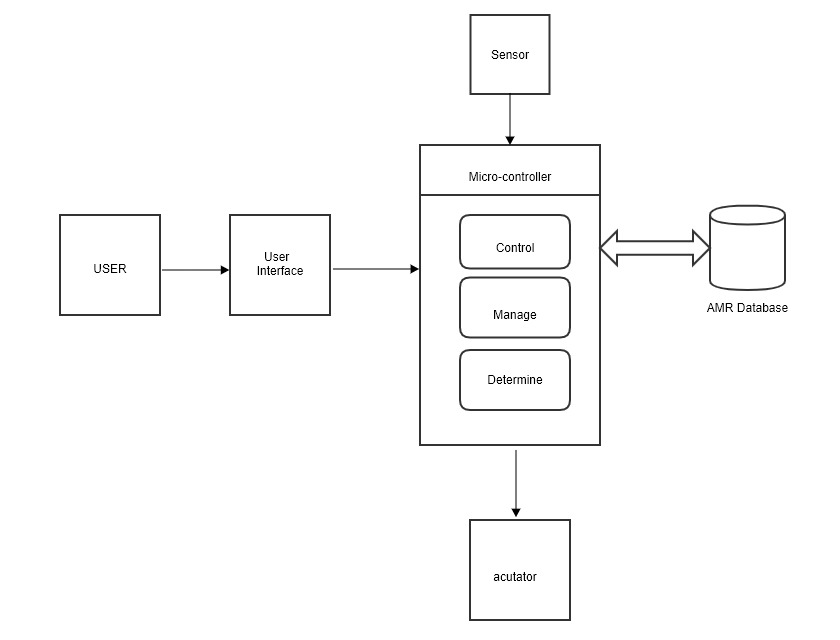
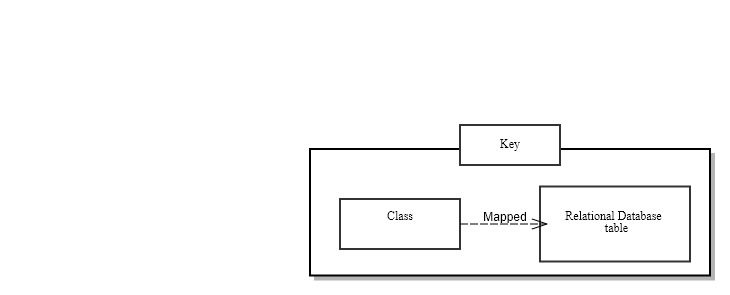
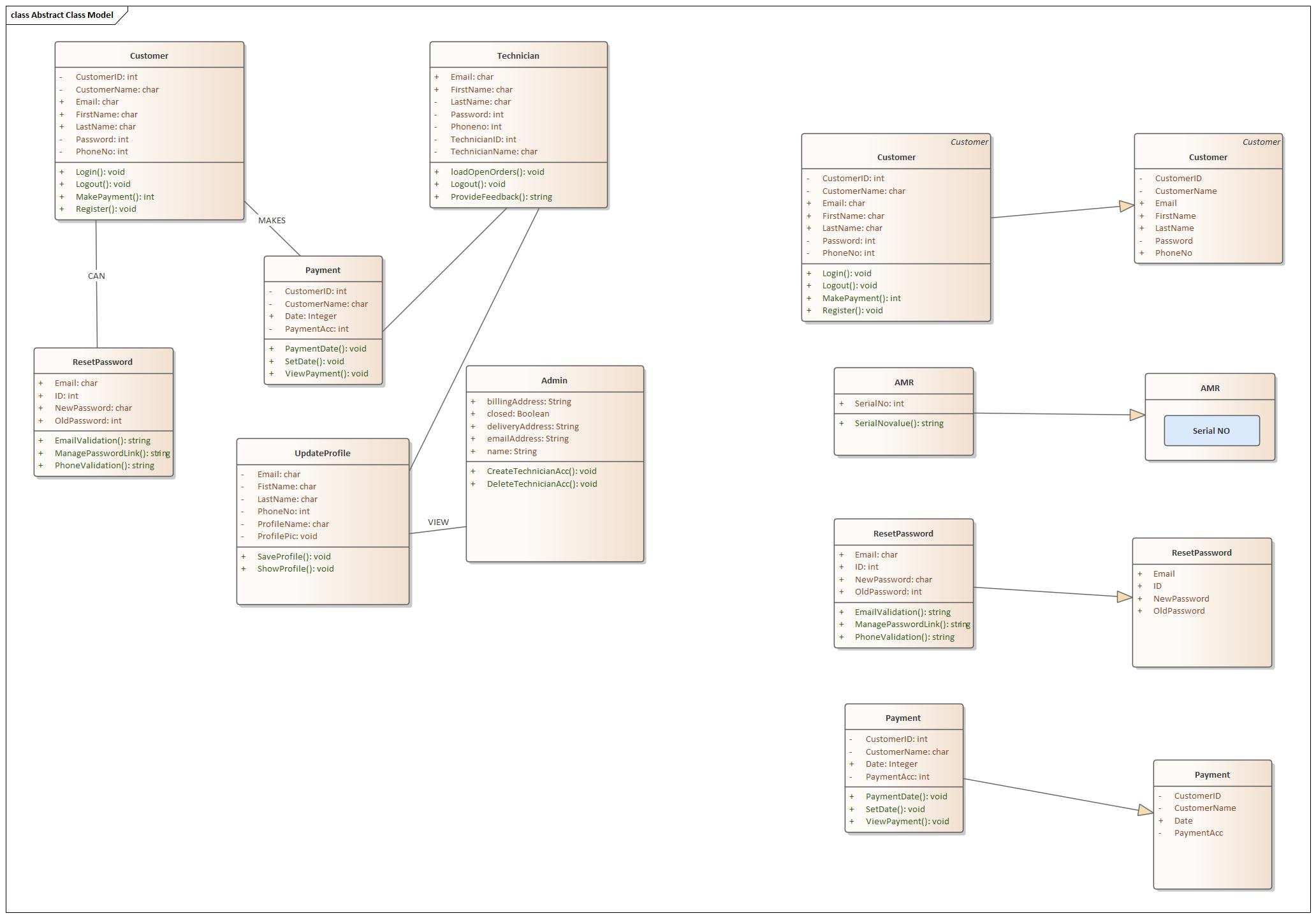
****

Fig: hardware/software mapping for hardware side

The above figure indicates that the user give input to microcontroller using keypad or button  
or other input mechanisms and the microcontroller take sensing data from different sensors and the microcontroller control the mechanical parts by sending data to the actuators then the necessary information will save to database.

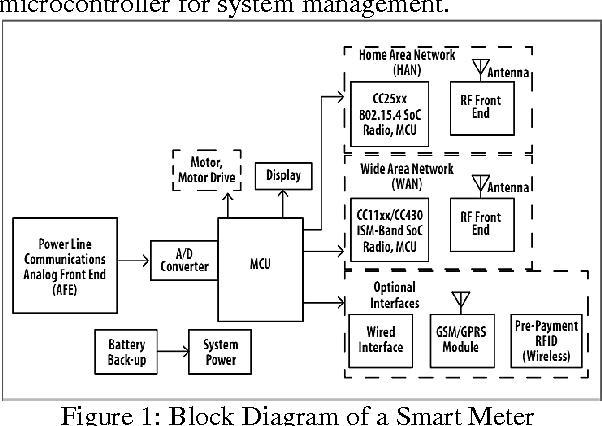
**4.2.5 Persistent data management**

The purpose of this section is to show the mapping of the objects/classes of the system, identified during the analysis stage, in to the corresponding relational database. ****

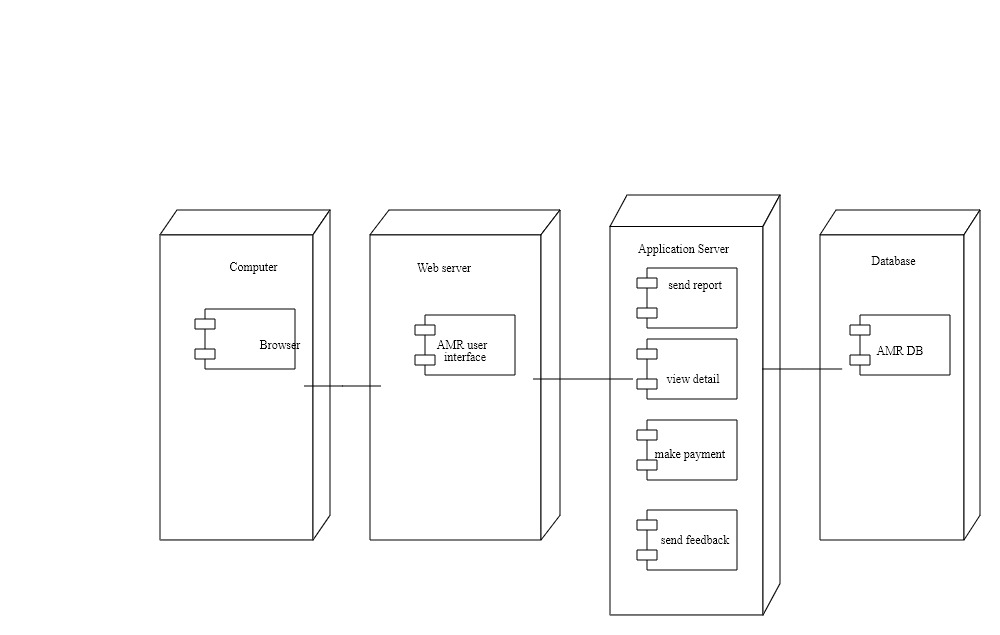


**4.2.6 Component diagram**

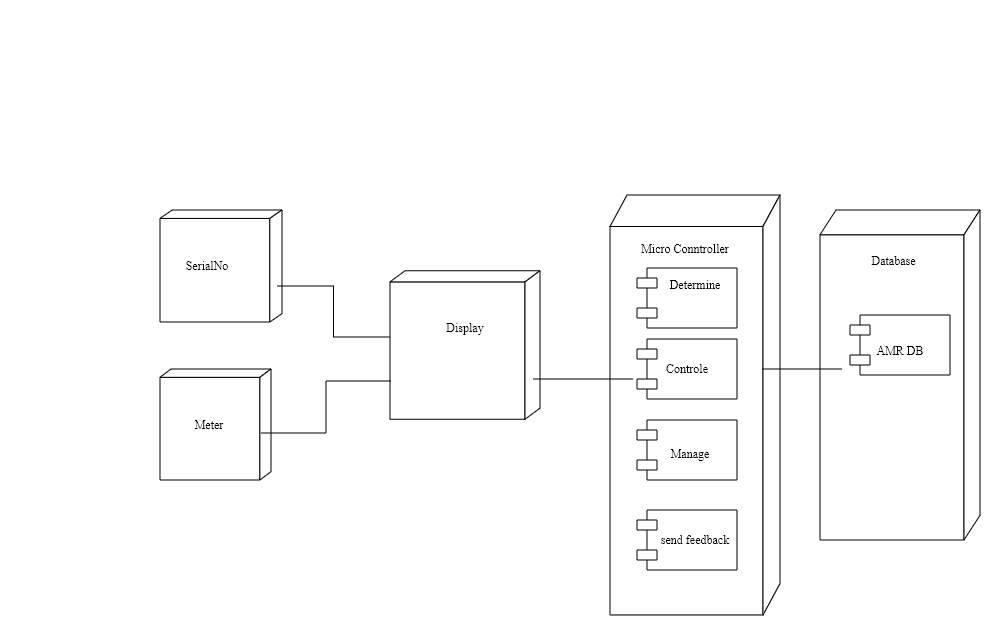
The following component diagram represents a group of graphs of components connected by  
dependency relationships and dependencies are shown as dashed arrows from the client  
component to the supplier component.

****

**4.2.7 Deployment diagram**

****

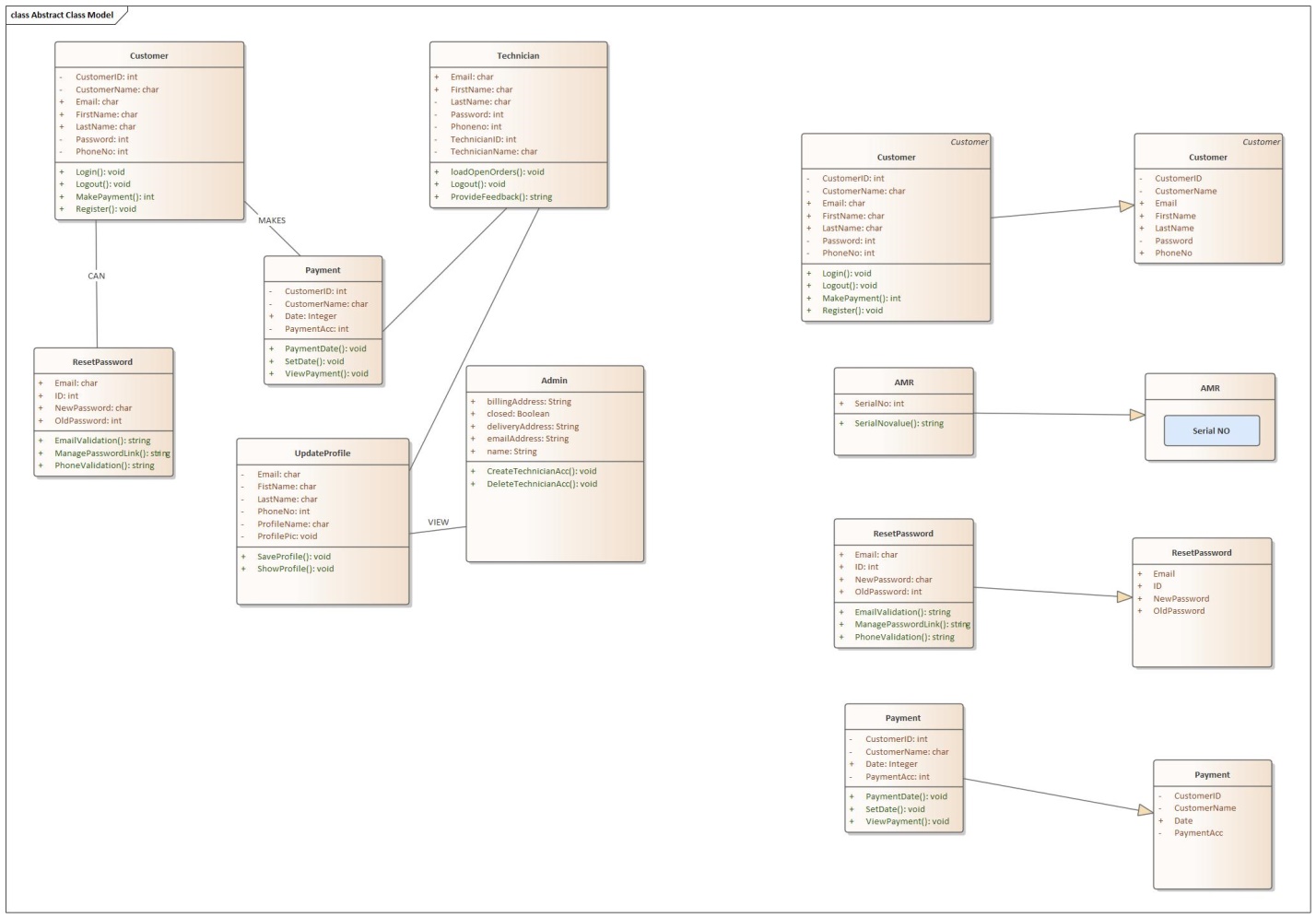
**fig: software side**

****

**Fig: hardware side**

**4.2.8 Database design**

The following figure shows the relationship between each table in a relational database.



\

**4.2.9 User interface design**

