SYSTEM DESIGN

**Introduction:**

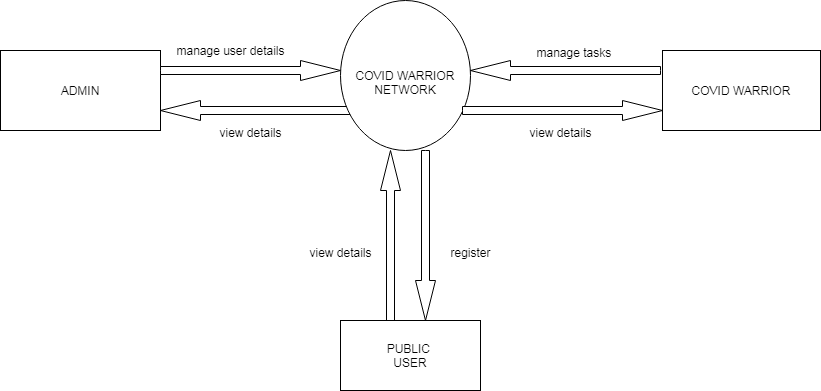
The goal of design process is to produce a model or representation of a system, which can be used later to build the system. Systems design is the process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements. The focus of system design is on deciding which modules are needed for system, the specifications of these modules and how the modules should be interconnected.

Systems design could be seen as the application of systems theory to product development. Systems design implies a systematic approach to the design of a system. It may take a bottom-up or top-down approach, but either way the process is systematic wherein it takes into account all related variables of the system that needs to be created. The system design controls the major structural characteristics of the system. It has a major impact on the testability and modifiability of system. The output is the architectural design for the software system to be built.

**Context Flow Diagram**

Context flow diagram is a top level data flow diagram. It only contains one process node that generalizes the function of the entire system in relationship to external entities. In context diagram the entire system is treated as a single process and all its inputs, outputs, sinks and sources are identified and shown.

**CFD:**

****

**Data flow diagram:**

Data Flow Diagram is a graphical representation of a system or a portion of the system. It consists of data flows, process, sources and sink and stores all the description through the use of easily understandable symbols.

DFD is one of the most important modelling tools. It is used to model the system, components that interact with the system, uses the data and information flows in the system.

DFD shows the information moves through the and how it is modified by a series of transformations. It is a graphical technique that depicts information moves from input or output.

DFD is also knows as bubble chart or Data Flow Graphs. DFD may be used to represent the system at any level of abstraction. DFD’s may partition into a level that represents increasing information flows and functional details.

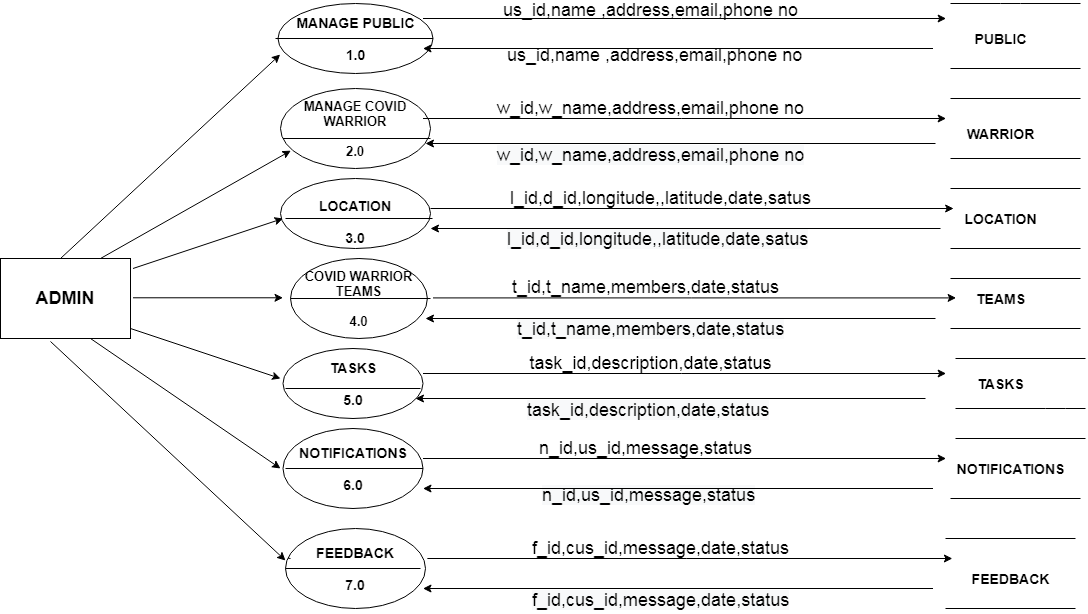
Rules Regarding DFD Construction:

* A process cannot have only outputs.
* A process cannot have only inputs.
* The inputs to a process must be sufficient to produce the outputs from the process.
* All data stores must be connected to at least one process.
* All data stores must be connected to a source or sink.
* A data flow can have only one direction of flow. Multiple data flows to and/or from the same process and data store must be shown by separate arrows.
* If the exact same data flows to two separate arrows, it should be represented by a forked arrow.
* Data cannot flow directly back into the process it has just left. All data flows must be named using a noun phrase.

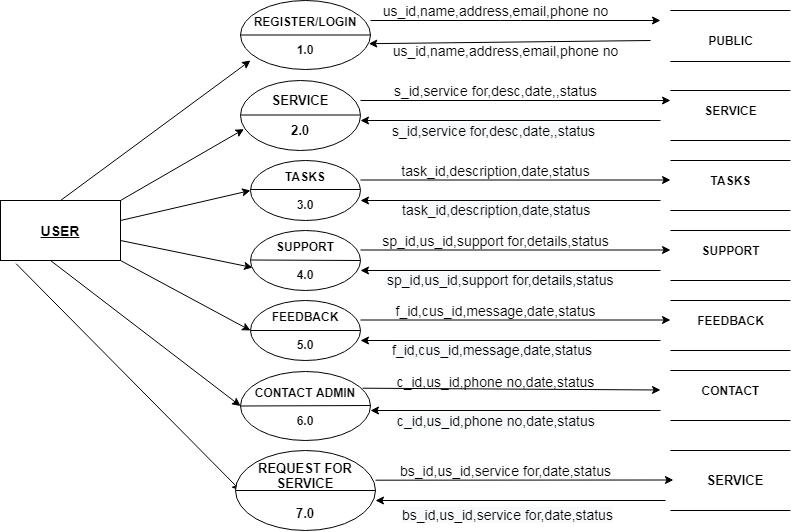
**DFD Symbols:**

|  |  |  |
| --- | --- | --- |
| **Name** | **Notation** | **Description** |
| **Process** |  | A process transforms incoming data flow into outgoing data flow. The processes are shown by named circles. |
| **Datastore** |  | Data stores are repositories of data in the system. They are sometimes also referred to as files. |
| **Dataflows** |  | Data flows are pipelines through which packets of information flow. Label the arrows with the name of the data that moves through it. |
| **External Entity** |  | External entities are objects outside the system with which the system communicates. External Entities are sources and destinations of the system’s inputs and outputs |

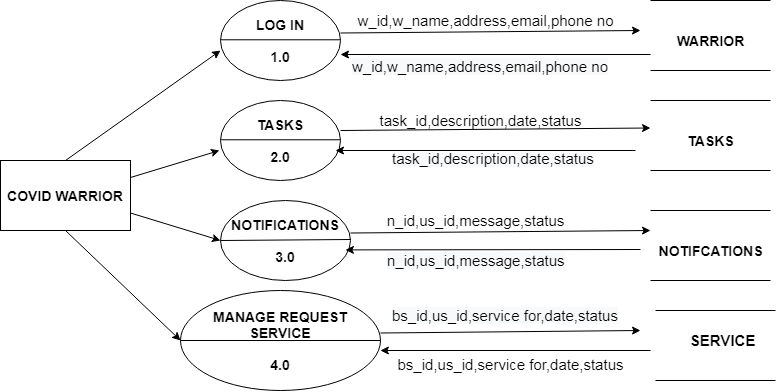
**DFD level 1(Admin)**

****

**DFD level 1 (Public User)**

****

**DFD level 1 (Covid warrior)**

****

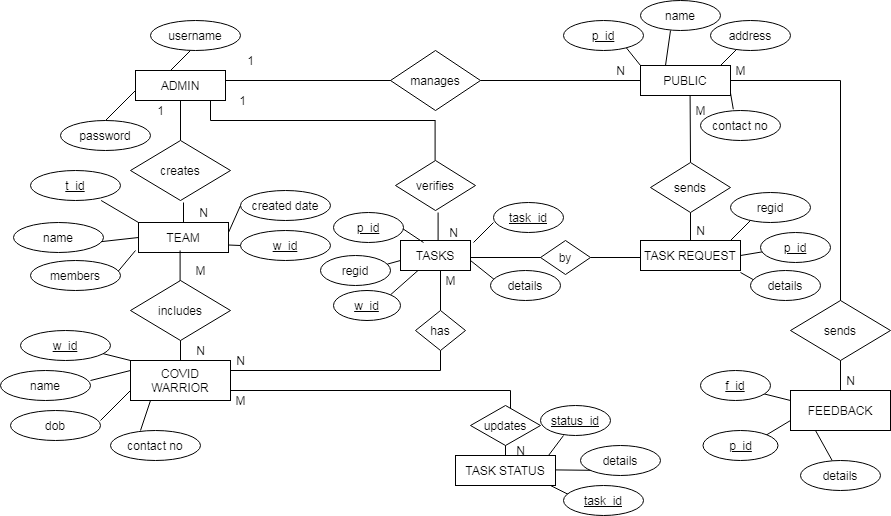
**Entity-Relationship Diagram:**

The basic objective of the ER model representation is an entity which is a “thing” in a real world with an independent existence. Entities are physical items or aggregations of data items that are important to the business we analyze or to the system; we intend to build. An entity represents an object defined within the information system about which you want to store information. Entities are named as singular nouns and are shown in rectangles in an ER-Diagram. Each entity is described by several attributes; individual instances of an entity will have different attribute values.

**ER-Diagram Symbols:**

|  |  |  |
| --- | --- | --- |
| **Name** | **Notation** | **Description** |
| **Entity** | Entity name | It may be an object with the physical existence or conceptual existence. It is represented by a Rectangle. |
| **Attribute** |  | The properties of the entity can be a attribute. It is represented by a Ellipse. |
| **Relationship** | Relation | Whenever an attribute of one entity refers to another entity, some relationship exists. It is represented by a Diamond. |
| **Link** |  | Lines link attributes to entity sets and entity sets to relation. |
| **Derived Attribute** |  | Dashed ellipse denotes derived attributes. |
| **Key Attribute** |  | An entity type usually has an attribute whose values are distinct for each individual entry in the entity set. It is represented by a Underlined word in ellipse. |
| **Multivalued Attribute** |  | Attributes that have different numbers of values for a particular attribute. It is represented by a Double ellipse represents multi-valued attributes. |
| **Cardinality Ratio** | 1. 1:1 2. 1:M 3. M:1 4. M:M | It specifies the maximum number of relationships instances that an entity can participate in. There are four cardinality ratios. |

**ER Diagram:**

****