**Design Document**

**for**

**Urban Service System,**

**a web portal of service provider system.**

Prepared by

Debopriyo Ghosh Roll- 33

Saikat Jana Roll-50

Mousumi Mondal Roll-40

Saradindu Rana Roll-28

B.Tech., Sem - V, Group- 1

Dated: 07.03.2021

Contents

**1. ER Diagram**

**2. Relational Schema**

**3. Data Dictionary**

**4. Context Diagram**

**5. DFD**

**6. UI Description**

**Entity-Relationship Diagram**

An **entity–relationship model** (or **ER model**) describes interrelated things of interest in a specific domain of knowledge. A basic ER model is composed of entity types (which classify the things of interest) and specifies relationships that can exist between [entities](https://en.wiktionary.org/wiki/entity) (instances of those entity types).

In [software engineering](https://en.wikipedia.org/wiki/Software_engineering), an ER model is commonly formed to represent things a business needs to remember in order to perform [business processes](https://en.wikipedia.org/wiki/Business_process). Consequently, the ER model becomes an abstract [data model](https://en.wikipedia.org/wiki/Data_modeling), that defines a data or information structure which can be implemented in a [database](https://en.wikipedia.org/wiki/Database), typically a [relational database](https://en.wikipedia.org/wiki/Relational_database).

**Relational Schema**

Relation schema defines the design and structure of the relation like it consists of the relation name, set of attributes/field names/column names. every attribute would have an associated domain.

A **relational schema** is a set of **relational** tables and associated items that are related to one another. All of the base tables, views, indexes, domains, user roles, stored modules, and other items that a user creates to fulfill the data needs of a particular enterprise or set of applications belong to one **schema**.

**Data Dictionary**

A **data dictionary** contains metadata i.e **data** about the database. The **data dictionary** is very important as it contains information such as what is in the database, who is allowed to access it, where is the database physically stored etc. . Physical information about the tables such as where they are stored and how.

**Context Diagram**

A **context diagram**, sometimes called a level 0 data-flow **diagram**, is drawn in order to define and clarify the boundaries of the software system. It identifies the flows of information between the system and external entities. The entire software system is shown as a single process.

**Data Flow Diagram**

A **data**-**flow diagram** is a way of representing a **flow** of **data** through a **process** or a system (usually an information system). The **DFD** also provides information about the outputs and inputs of each entity and the **process** itself. Specific operations based on the **data** can be represented by a **flowchart**.