**iv. DATABASE DESIGN**

**a) INTRODUCTION**

Database is an organized collection of data, generally stored and accessed electronically through computer systems. Database design is a collection of tasks or processes that enhance the designing, development, implementation, and maintenance data management system. The main objective of database design is to produce physical and logical design models of the proposed data-based system. The logical model is primarily concentrated on the requirement of the data. Physical data base design models include a translation of the logical design model of the database by keep control of physical media using hardware resources and software systems such as DBMS.

**b) PURPOSE AND SCOPE**

The ultimate purpose of a database management system is to store and transform data into information to support making decisions. The physical database: the collection of files that contain the data. The database engine: the software that makes it possible to access and modify the contents of the database

**4.2.1 PURPOSE**

This database requirement specification describes the function and performance requirements by the WATCHWORLD.

This database stores

* Admin login records
* The shopowner and customer details
* Blog details
* Booking details
* Cart details
* Feedback details.
* Product details.
* Rent details.
* Requested service details
* Service details

**iii. DATABASE IDENTIFICATION**

* A Primary key is a special relational database table column designed to uniquely identify each table record

A FORIEGN KEY is a column that is used to establish and enforce a link between the two tables to control the data that can be stored in a FORIEGN KEY table.

* The database consists of tables, each of which has columns and rows. Each row is a dataset that applies to a single item, and each column contains characteristics that describe the rows. In the database, these columns are called as attribute.
* Primary key and FORIEGN KEY are defined with same name.
* Make a separate table for each set of related attributes, and give each table a primary key.
* If an attribute depends on only part of a multi-valued key, remove it to a separate table.
* If attributes do not contribute to a description of the key, remove them to a separate table

## iv. SCHEMA INFORMATION

A database schema represents the logical configuration of all or part of a relational database. it can exist both as a visual representation and as asset of formulas known as integrity constraints that govern a database. These formulas are expressed in a data definition language such as SQL.

A database schema is the skeleton structure that represents the logical view of the entire database. It defines how the data is organized and how the relations among them are associated. It formulates all the constraints that are to be applied on the data.

A database schema defines its entities and the relationship among them. It contains a descriptive detail of the database, which can be depicted by means of schema Diagrams. It’s the database designers who design the schema to help programmers understand the database and make it useful.

**v. TABLE DEFINITION**

**TABLE NAME: admin**

|  |  |  |
| --- | --- | --- |
| username | password | a\_id |

**TABLE NAME: blog**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| bl\_id | title | des | image | date |

**TABLE NAME: booking**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| b\_id | c\_id | p\_id | quantity | amount |

**TABLE NAME: cart**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| cart\_id | c\_id | p\_id | quantity | price | date |

**TABLE NAME: customer**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| c\_id | username | password | c\_name | email | address | contact | image | status |

**TABLE NAME: feedback**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| f\_id | c\_id | message | name | email | subject | date |

**TABLE NAME:products**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| p\_id | p\_name | p\_name | p\_price | p\_description | p\_image | stock |

**TABLE NAME:rent**

|  |  |  |  |
| --- | --- | --- | --- |
| r\_id | c\_id | r\_name | r\_price |

**TABLE NAME:requested\_services**

|  |  |  |  |
| --- | --- | --- | --- |
| c\_id | s\_id | sr\_name | email |

**TABLE NAME:service**

|  |  |  |  |
| --- | --- | --- | --- |
| s\_id | s\_name | s\_price | s\_description |

**TABLE NAME:shopowner**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| s\_id | username | password | so\_name | email | address | contact | image | status |

**vii. DATA DICTIONARY**

**TABLE STRUCTURE**

The database watch\_world is organized into following tables:

* admin
* blog
* booking
* cart
* customer
* feedback
* products
* rent
* requested\_services
* shopowner

**TABLE NAME: admin**

This table is used to store the login details of admin

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ADMIN TABLE** | | | | | |
| SL NO | FIELD NAME | DATA TYPE | SIZE | CONSTRAINTS | DESCRIPTION |
| 1 | username | varchar | 50 | PRIMARY KEY | Admin’s user id |
| 2 | password | varchar | 10 | NOT NULL | Admin’s password |
| **3** | a\_id | int | 10 | NOT NULL | Admin id |

**TABLE NAME: blog**

This table is used to store blog details.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **BLOG TABLE** | | | | | |
| SL NO | FIELD NAME | DATA TYPE | SIZE | CONSTRAINTS | DESCRIPTION |
| 1 | bl\_id | int | 10 | PRIMARY KEY | Blog id |
| 2 | title | varchar | 5000 | NOT NULL | Title of blog |
| 3 | des | varchar | 100 | NOT NULL | Description |
| 4 | image | varchar | 100 | NOT NULL | Image |
| 5 | date | date | date | NOT NULL | Date |

**TABLE NAME: booking**

This table is used to store booking details

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **BOOKING TABLE** | | | | | |
| SL NO | FIELD NAME | DATA TYPE | SIZE | CONSTRAINTS | DESCRIPTION |
| 1 | b\_id | int | 10 | PRIMARY KEY | Booking id |
| 2 | c\_Id | int | 10 | FOREIGN KEY | Customer id |
| 3 | p\_id | int | 10 | FOREIGN KEY | Product id |
| 4 | quantity | int | 20 | NOT NULL | Quantity |
| 5 | amount | int | 20 | NOT NULL | Amount |

**TABLE NAME: cart**

This table is used to store booking details

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **CART TABLE** | | | | | |
| SL NO | FIELD NAME | DATA TYPE | SIZE | CONSTRAINTS | DESCRIPTION |
| 1 | cart\_id | int | 10 | PRIMARY KEY | Cart id |
| 2 | c\_id | int | 10 | FOREIGN KEY | Customer id |
| 3 | p\_id | int | 10 | FOREIGN KEY | Product id |
| 4 | quantity | int | 50 | NOT NULL | Quantity |
| 5 | price | double |  | NOT NULL | Price |
| 6 | date | date |  | NOT NULL | Date |

**TABLE NAME: customer**

This table is used to store customer details

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **CUSTOMER TABLE** | | | | | |
| SL NO | FIELD NAME | DATA TYPE | SIZE | CONSTRAINTS | DESCRIPTION |
| 1 | c\_id | int | 10 | PRIMARY KEY | Customer id |
| 2 | username | text |  | NOT NULL | Username |
| 3 | password | varchar | 20 | NOT NULL | Password |
| 4 | c\_name | text |  | NOT NULL | Customer name |
| 5 | email | varchar | 50 | NOT NULL | Email |
| 6 | address | varchar | 200 | NOT NULL | Address |
| 7 | contact | int | 15 | NOT NULL | Contact |
| 8 | image | varbinary | 500 | NOT NULL | Image |
| 9 | status | varchar | 30 | NOT NULL | Status |

**TABLE NAME: feedback**

This table is used to store feedback details

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **FEEDBACK TABLE** | | | | | |
| SL NO | FIELD NAME | DATA TYPE | SIZE | CONSTRAINTS | DESCRIPTION |
| 1 | f\_id | int | 10 | PRIMARY KEY | Feedback id |
| 2 | c\_id | int | 10 | FOREIGN KEY | Customer id |
| 3 | message | varchar | 500 | NOT NULL | Message |
| 4 | name | varchar | 20 | NOT NULL | Name |
| 5 | email | varchar | 50 | NOT NULL | Email |
| 6 | subject | varchar | 100 | NOT NULL | Subject |
| 7 | date | date |  | NOT NULL | Date |

**TABLE NAME: products**

This table is used to store products details

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **PRODUCTS TABLE** | | | | | |
| SL NO | FIELD NAME | DATA TYPE | SIZE | CONSTRAINTS | DESCRIPTION |
| 1 | p\_id | int | 10 | PRIMARY KEY | Product id |
| 2 | p\_name | varchar | 1000 | NOT NULL | Product name |
| 3 | p\_price | double |  | NOT NULL | Product price |
| 4 | p\_description | varchar | 1000 | NOT NULL | Description |
| 5 | p\_image | varchar | 500 | NOT NULL | Image |
| 6 | stock | varchar | 30 | NOT NULL | Stock |

**TABLE NAME: rent**

This table is used to store rent details

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **RENT TABLE** | | | | | |
| SL NO | FIELD NAME | DATA TYPE | SIZE | CONSTRAINTS | DESCRIPTION |
| 1 | r\_id | int | 10 | PRIMARY KEY | Rent Id |
| 2 | c\_id | int | 10 | FOREIGN KEY | Customer id |
| 3 | r\_name | text |  | NOT NULL | Rent name |
| 4 | r\_price | int | 10 | NOT NULL | Rent price |

**TABLE NAME: requested\_services**

This table is used to store requested services details

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **REQUESTED\_SERVICES TABLE** | | | | | |
| SL NO | FIELD NAME | DATA TYPE | SIZE | CONSTRAINTS | DESCRIPTION |
| 1 | c\_id | int | 10 | FOREIGN KEY | Customer id |
| 2 | s\_id | int | 10 | FOREIGN KEY | Service id |
| 3 | sr\_name | text |  | NOT NULL | Service name |
| 4 | email | varchar | 50 | NOT NULL | Email |

**TABLE NAME: service**

This table is used to store service details

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SERVICE TABLE** | | | | | |
| SL NO | FIELD NAME | DATA TYPE | SIZE | CONSTRAINTS | DESCRIPTION |
| 1 | s\_id | int | 10 | PRIMARY KEY | Service id |
| 2 | s\_name | text |  | NOT NULL | Service name |
| 3 | s\_price | int | 10 | NOT NULL | Service price |
| 4 | s\_description | varchar | 200 | NOT NULL | Description |

**TABLE NAME: shopowner**

This table is used to store shopowner details

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SHOPOWNER TABLE** | | | | | |
| SL NO | FIELD NAME | DATA TYPE | SIZE | CONSTRAINTS | DESCRIPTION |
| 1 | s\_id | int | 10 | PRIMARY KEY | Shopowner id |
| 2 | username | text |  | NOT NULL | Username |
| 3 | password | varchar | 20 | NOT NULL | Password |
| 4 | so\_name | varchar | 30 | NOT NULL | Full name |
| 5 | email | varchar | 50 | NOT NULL | Email |
| 6 | address | varchar | 100 | NOT NULL | Address |
| 7 | contact | int | 10 | NOT NULL | Contact |
| 8 | image | varbinary | 300 | NOT NULL | Image |
| 9 | status | varchar | 30 | NOT NULL | status |

**v. DETAIL DESIGN**

**a) INTRODUCTION**

Detailed design is the second level of the design process. During detailed design, we specify how the module in the system interacts with each other and the internal logic of each of the modules specified during system design is decided, hence it is also called as logic design.

Detailed design essentially expands the system design and database design to contain a more detailed description of the processing logic and data structures so that the design is sufficiently complete for coding. Detailed design is the phase where the design is refined and plans, specifications and estimates are created. Detailed design will include outputs such as 2D and 3D models, P & ID's, cost build up estimates, procurement plans etc. This phase is where the full cost of the project is identified.