**LUNAMAR’S MANAGEMENT**

**Project Phase 1** Nandhimandalam, Lakshmi – 1001719397

Garikipati, Kavya Sri – 1001953373

Allakonda, Sai Teja – 100

**Abstract**

The purpose of this project is to design a web application for Lunamar’s Management. The three main categories here are Management of Buildings, Management of Garden and Surroundings and Management of Visitors

**Web Application Explanation**

Web application consists of

Home – It is the default page and is available for all users. Users can navigate to multiple pages from Home Page

About – About page consists of detailed explanation about Lunamar’s Management

Sign Up – New users can register themselves using this page providing their personal details and contact information

Sign In – Registered users login to application using the username and password details given at the time of registration

Service Requests – Based on the user type (owner) that is logged in, this page displays requests associated to them while Manager and Staff can view all requests raised by owners

Contact – Contact page displays the address, contact details of Lunamar’s Management and location details.

**Technology Requirements**

HTML, CSS, JavaScript, React, node, PHP, LARAVEL, MySQL

**Roles**

Manager – Manager is responsible for creating all the tables

Staff – Staff is responsible for all the DML operations on the table and also assigning service requests to maintenance person

Owner – Owner can view apartment details and can also create service request for apartment/building

Visitor – Visitor can create his personal details and also enter the apartment\_no which he is visiting

Maintenance Staff - Maintenance Staff can enter his personal details and register himself and can view the service requests assigned to him

**ER Diagram Explanation**

* Relation **Building** is createdby Manager and has attributes – building\_no (primary key), trash\_date, laundry\_date
* Relation **Apartment** is created by Manager and maintained by staff and has attributes – apartment\_no (primary key), no\_of\_bedrooms, no\_of\_bathrooms, total\_sqfeet, gas\_usage, electricity\_usage, water\_usage, home\_association, owner\_id (foreign key), parking\_id, building\_no (foreign key).
* Relation **Owner** is created when the owner enters his details during registration and has attributes – owner\_id (primary key), owner\_name, owner\_email, owner\_mobile, apartment\_no (foreign key), vehicle\_no
* Relation **Garden** is created by Manager and maintained by staff and has attributes – plant\_id (primary key), plant\_name, plant\_type, last\_maintained, next\_service, maintenance\_person\_id (foreign key)
* Relation **Pool** is created by Manager and maintained by staff and has attributes – pool\_no (primary key), open\_time, close\_time, last\_cleaned, next\_clean, maintenance\_person\_id (foreign key)
* Relation **Maintenance** is createdwhen the maintenance staff enters his details during registration and has attributes - maintenance\_person\_id (primary key), maintenance\_person\_name, maintenance\_person\_mobile, salary
* Relation **Visitor** is created when he enters his details in order to visit an owner and has attributes – visitor\_id (primary key), visitor\_name, visitor\_mobile, vehicle\_no, apartment\_no (foreign key)
* Relation **Complaints** is created by Manager or staff and has attributes – complaint\_id (primary key), visitor\_id (foreign key), complaint\_date, complaint\_description
* Relation **ServiceRequests** is created by owner/staff and has attributes – service\_request\_id (primary key), description, priority, apartment\_no (foreign key), building\_no (foreign key), maintenance\_person\_id (foreign key), status

**Cardinality**

* Cardinality relation between Building and Apartment is 1: N because 1 building can have many apartments but 1 apartment cannot be part of multiple buildings
* Cardinality relation between Building and ServiceRequests is 1: N because 1 building can have many service requests raised but 1 service request cannot be associated with multiple buildings
* Cardinality relation between Apartment and ServiceRequests is 1: N because 1 apartment can have many service requests raised but 1 service request cannot be associated with multiple apartments
* Cardinality relation between Owner and Apartment is N: N because 1 apartment can have only one or more owners and 1 owner can own multiple apartments
* Cardinality relation between Apartment and Complaints is 1: N because 1 apartment can have many complaints registered against it and 1 complaint cannot be associated with multiple apartments
* Cardinality relation between Maintenance and Garden is N: N because 1 maintenance person can be associated with 1 or more plants and 1 plant can be associated with 1 or more maintenance person
* Cardinality relation between Maintenance and Pool is N: N because 1 maintenance person can be associated with 1 or more pool and 1 pool can be associated with 1 or more maintenance person
* Cardinality relation between Maintenance and ServiceRequests is N: N because 1 maintenance person can have many service requests to fix and 1 service request can be assigned to 1 or more maintenance person
* Cardinality relation between Visitor and Complaints is 1: N because 1 visitor can have many complaints registered against him/her but 1 complaint cannot be associated with multiple visitors
* Cardinality relation between Visitor and Apartment is N: N because 1 visitor can be associated with one or more apartments and 1 apartment can be associated with 1 or more visitors