DBMS Mini Project
Community Living Database Management System in Apartment Societies

Submitted By:

Name: Hita Juneja

SRN: PES1UG20CS645

V Semester

Section K

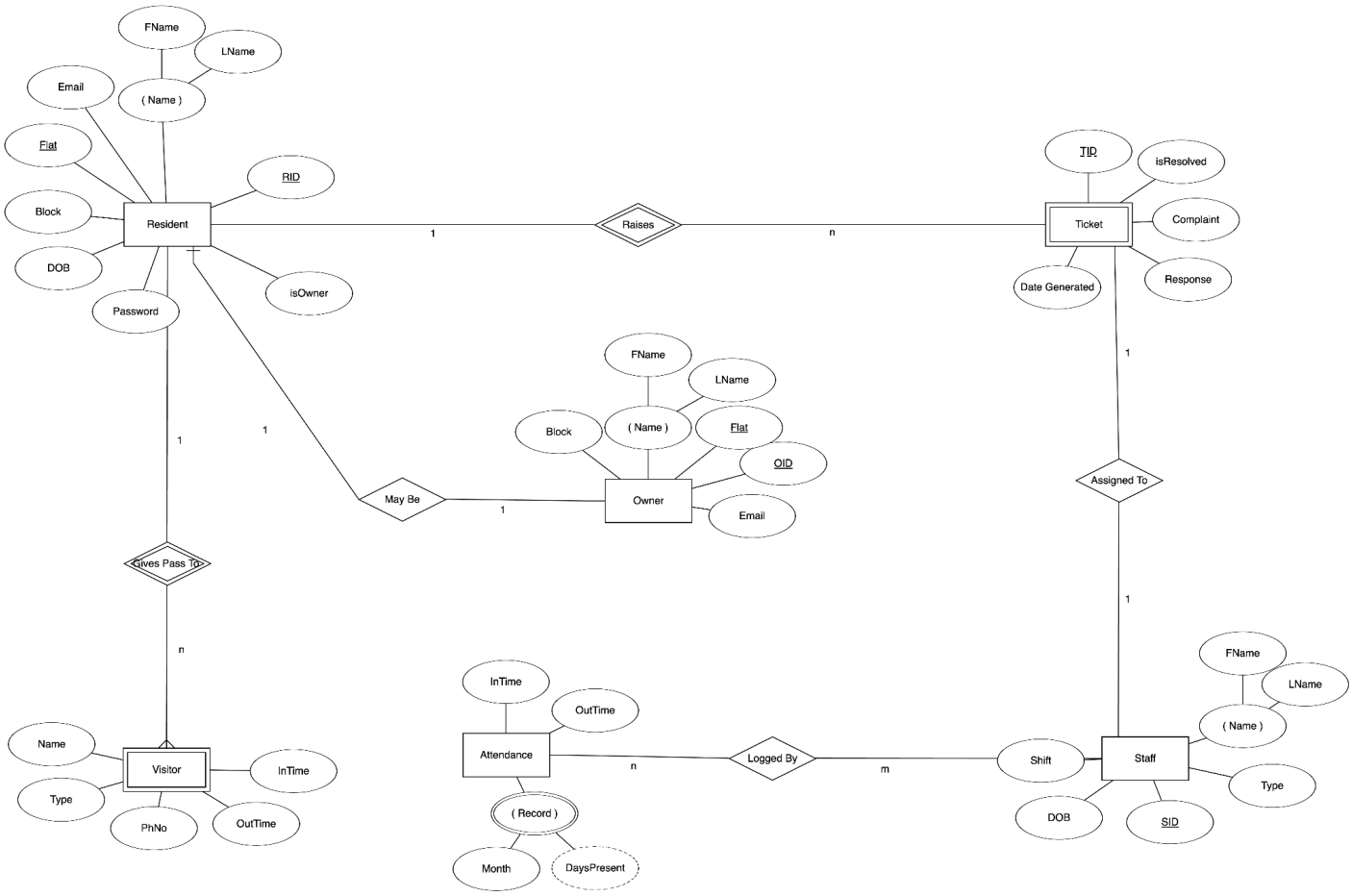
Description and Scope of Project

This project is an application that ensures the safety and convenience of gated societies. The gated community software offers numerous innovative features to create a wholesome community living experience and improve security standards around gated communities through its security, community modules.

Scope
Enhanced Security
1. Every visitor's (domestic help, cab drivers, guests) details and visit times are logged automatically.
2. No. of staff and details, attendance times and shifts are logged in the software.
3. Manage daily help (cooks, maid, drivers):
4. Use digital communication: Get updates on the on-platform notice board, raise complaints, find contact details of committee members & your neighbours.
5. Tickets once raised can be marked resolved, response updates and deleted only by help desk.

Interface allows for CRUD operations on table ticket and a text input to enter any SQL query and produces output accordingly.

ER Diagram



The given ER Diagram represents the relationship between entities in an apartment society. The apartment has many blocks – i.e. buildings. (Example Block A, B,C) and multiple flats in each block (Example 403 – 3rd flat on 4th floor).

- An owner owns a flat in the apartment. The owner has a unique ID denoted by OID and other personal information, along with the block and flat they own.

- Flats are occupied by residents. A resident has a block and flat associated with them, personal information and a unique ID RID. If the resident also owns the flat in which they are living, their RID is the same as their OID. Otherwise, the resident is a tenant.

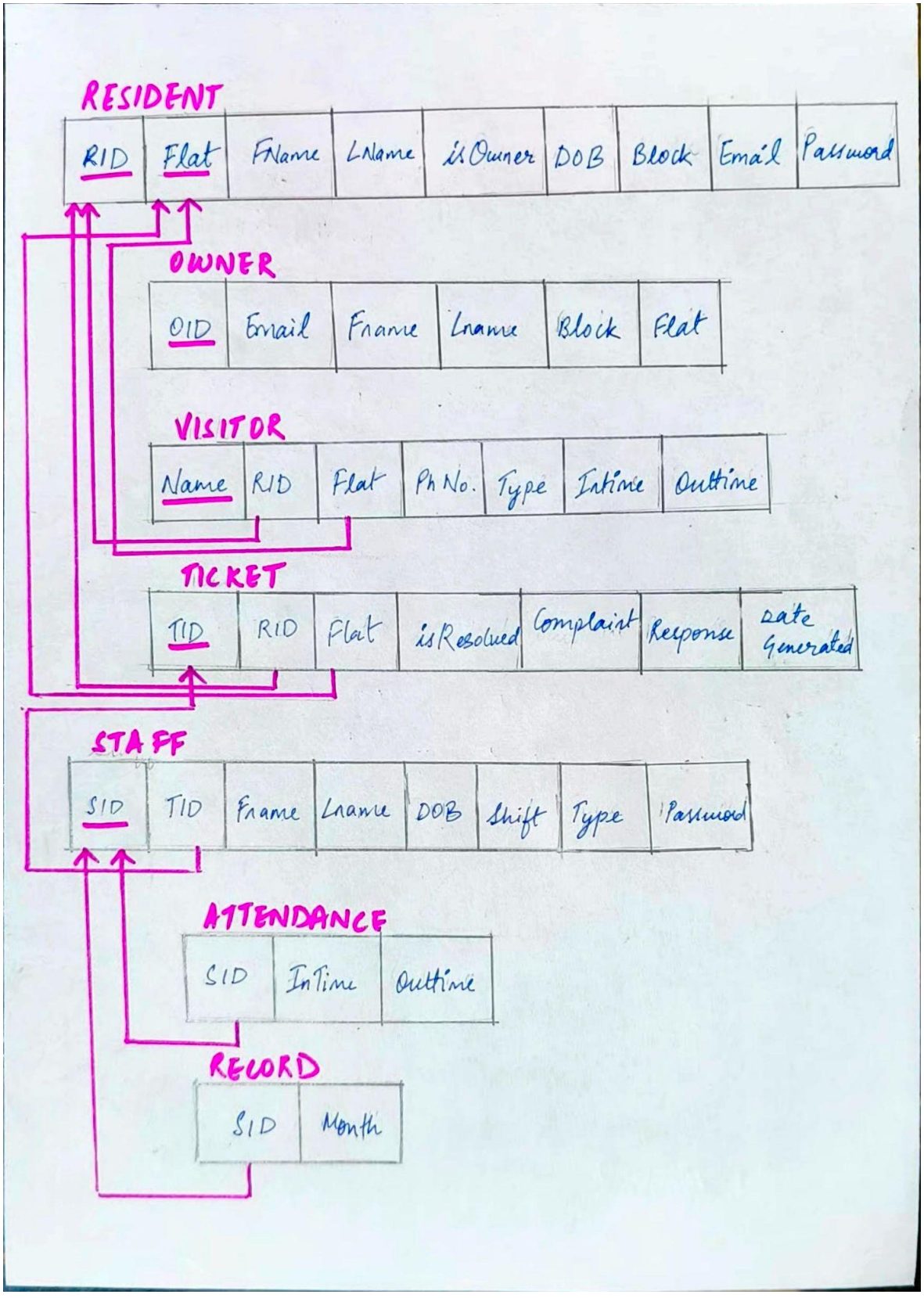
- Visitor who visits a flat(the resident) is recorded. Every visitor has a type(Cab, Guest, Daily Help) and an associated RID. From RID, we know the flat they are visiting and how long they were inside the premises from their Intime and Outtime.

- Residents can raise tickets, which is highlighting a relevant issue to the apartment association workers. The ticket has a unique ticket ID, TID, and is assigned to the appropriate staff at the apartment (Plumber, Electrician, Security, Housekeeping). Date it is raised is recorded, and a response as well as boolean value isResolved changes to 1 (default is 0) when it has been resolved by staff.

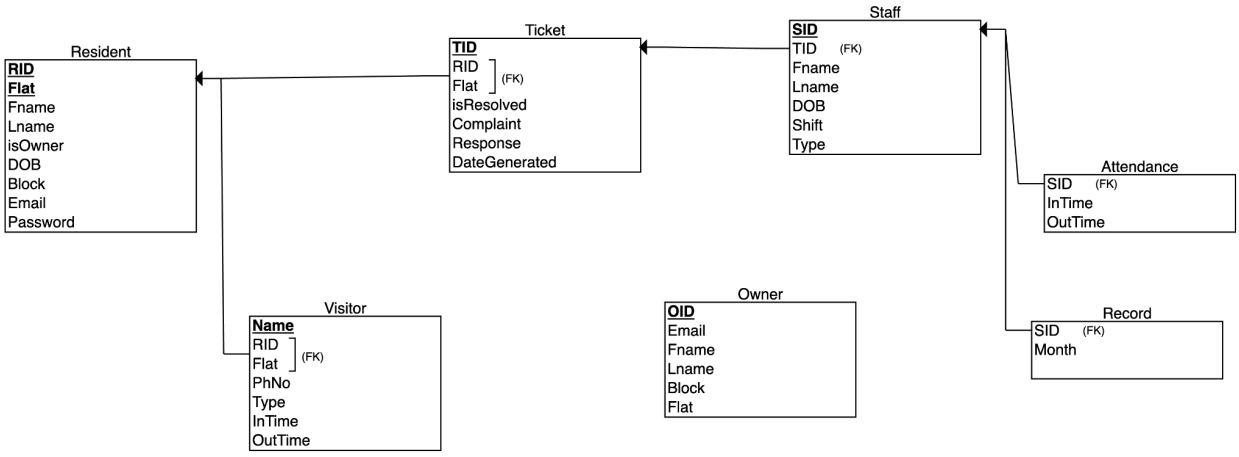
- Information about the staff is stored with unique id - SID, name, number, date of birth, shift(1- Day shift or 2 - Night Shift), type (Plumber, Electrician, Security, Housekeeping, Manager/Supervisor).

- Each staff member logs their attendance by storing Intimes and Outtimes everyday. From this information, we can derive the number of days they were present in a particular month(date timestamp).

Relational Schema



Same relational schema in vertical format.



Building Database (DDL Statements)

Create table **Resident**

```
CREATE TABLE `resident` (
 `rid` char(4) NOT NULL,
 `fname` varchar(10) NOT NULL,
 `lname` varchar(10) NOT NULL,
 `is_owner` tinyint(1) DEFAULT 0,
 `dob` date,
 `block` char(1) NOT NULL,
 `flat` int(3) NOT NULL,
 `email` varchar(30) DEFAULT NULL,
 `password` varchar(30) DEFAULT 'mygate',
 PRIMARY KEY(`rid`),
 KEY(`block`,`flat`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Create table **Owner**

```
CREATE TABLE `owner` (
 `oid` char(4) NOT NULL,
 `email` varchar(30) DEFAULT NULL,
 `fname` varchar(10) NOT NULL,
 `lname` varchar(10) NOT NULL,
 `block` char(1) NOT NULL,
 `flat` int(3) NOT NULL,
 PRIMARY KEY(`oid`),
 KEY(`block`,`flat`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Create table **Visitor**

```
CREATE TABLE `visitor` (
 `rid` char(4) NOT NULL,
 `name` varchar(20) NOT NULL,
 `phno` int(10) DEFAULT NULL,
 `type` char(1) DEFAULT NULL,
 `block` char(1) NOT NULL,
 `flat` int(3) NOT NULL,
 `intime` timestamp DEFAULT NOW(),
 `outtime` timestamp DEFAULT '2000-01-01 00:00:00'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Creating table **Staff**

```
CREATE TABLE `staff` (
 `sid` char(4) NOT NULL,
 `tid` char(4) DEFAULT NULL,
 `fname` varchar(20) NOT NULL,
 `lname` varchar(20) DEFAULT NULL,
 `dob` date DEFAULT NULL,
 `shift` int(1) DEFAULT 1,
 `type` varchar(15) NOT NULL,
 `password` varchar(6) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
-- housekeeping(H), security(S), manager(M), maintainence()- plumber
and electrician
```

Creating table **Ticket**

```
CREATE TABLE `ticket` (
 `tid` char(4) NOT NULL,
 `rid` char(4) NOT NULL,
 `flat` int(3) NOT NULL,
```

```
 `block` char(1) NOT NULL,
 `is_resolved` tinyint(1) DEFAULT 0,
 `complaint` varchar(50) NOT NULL,
 `response` varchar(50) DEFAULT NULL,
 `date_generated` date
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
-- housekeeping(H), security(S), manager(M), maintainence()- plumber and electrician
```

Creating table **Attendance**

```
CREATE TABLE `attendance` (
 `sid` char(4) NOT NULL,
 `intime` timestamp NOT NULL,
 `outtime` timestamp DEFAULT '2000-01-01 00:00:00'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Creating table **Record**

```
CREATE TABLE `record` (
 `sid` char(4) NOT NULL,
 `month` char(3) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Populating the Database

Populate table **Resident**

```
INSERT INTO `resident` (`rid`, `email`,
`fname`,`lname`,`is_owner`,`block`,`flat`,`password`,`dob`) VALUES
('R333', 'dsouza@gmail.com','Christon','DSouza',0, 'A',
204,'abcd','2001-08-08'),
('R444', 'bsubr@gmail.com','Bala','Subramanyan',0, 'B',
304,'1234','1999-01-12'),
('R555', 'namgyal@gmail.com','Namgyal','G', 0,'C', 604,'abcd','1999-01-12'),
('R757', 'jimmg@gmail.com','Jimmy','G', 0,'A', 704,'abcd','1985-06-12'),
('R666', 'ashraf@gmail.com','Ashraf','Muhammed',0, 'B',
804,'1234','1985-06-12'),
('R888', 'simmychahal@gmail.com','Simmy','Chahal',0, 'C',
904,'abcd','1989-01-09'),
('R999', 'dimplek@gmail.com','Dimple','Kapadia',0, 'A',
101,'1234','1989-01-09'),
('R901', 'sprakash@gmail.com','Sajan','Prakash',0, 'B',
102,'abcd','1999-01-12'),
('R244', 'pradeep@gmail.com','Pradeep','K',0, 'C', 403,'abcd','1999-01-12'),
('R267', 'zintapreity@gmail.com','Preity','Zinta',0, 'A',
701,'abcd','1999-01-12'),
('R273', 'chahatthakur@gmail.com','Chahat','Thakur', 0,'B',
401,'abcd','1992-01-12'),
('R297', 'kalrakamal@gmail.com','Kamalpreet','Kaur',0, 'C',
202,'abcd','1992-01-12'),
('O777', 'manisha@gmail.com','Manisha','Solanki',1, 'A',
505,'abcd','1992-01-12'),
('O985', 'bhartidev@gmail.com','Bharti','Devgan',1, 'B',
601,'1234','1980-01-01'),
('O487', 'pallavi@gmail.com','Pallavi','Ram',1, 'C', 602,'1234','1980-01-01'),
('O911', 'vk@gmail.com','Virat','Kohli',1, 'A', 603,'abcd','1980-01-01'),
('O142', 'ali@gmail.com','Muhammed','Ali',1, 'B', 402,'abcd','1991-09-12'),
('O307', 'msingh@gmail.com','Manmohan','Singh',1, 'C',
201,'abcd','1991-09-12'),
('O327', 'nani@gmail.com','Urmil','Kalra',1, 'A', 203,'abcd','1986-01-03'),
('O907', 'maamu@gmail.com','Kamal','Kalra',1, 'B', 702,'1234','1986-01-03'),
```

```
('O407', 'mam@gmail.com','Seema','Kalra',1, 'C', 801,'abcd','1986-01-03'),
('O387', 'manju@gmail.com','Manju','Munjal',1, 'A', 802,'abcd','1986-01-03'),
('O222', 'veer@gmail.com','Veer','Singh',1, 'B', 803,'abcd','1996-01-12'),
('O892', 'manpreet@gmail.com','Manpreet','Kaur',1, 'C',
103,'abcd','1996-01-12'),
('O841', 'preetkaur@gmail.com','Preet','Sinha',1, 'A',
101,'1234','1996-01-12'),
('O648', 'nsingh@gmail.com','Navneet','Singh',1, 'B',
202,'abcd','1990-02-10'),
('O645', 'apdhil@gmail.com','AP','Dhillion',1, 'C', 703,'abcd','1990-02-10'),
('O765', 'alka@gmail.com','Alka','Yagnik',1, 'C', 101,'1234','1990-02-10');
```

Populate table **Owner**

```
INSERT INTO `owner` (`oid`, `email`, `fname`, `lname`, `block`,`flat`) VALUES
('O101', 'sh@gmail.com','Sheela','Reddy', 'A', 101),
('O102', 'udupa@gmail.com','Srihari','Udupa', 'B', 102),
('O603', 'dixit@gmail.com','Sheela','Dixit', 'C', 403),
('O104', 'ab@gmail.com','Anshu','Bharadwaj', 'A', 104),
('O105', 'arun@gmail.com','Arun','Kumar', 'B', 701),
('O106', 'snayar@gmail.com','Shradha','Nayar', 'C', 401),
('O107', 'ramk@gmail.com','Ram','Kapoor', 'A', 502),
('O777', 'manisha@gmail.com','Manisha','Solanki', 'A', 504),
('O985', 'bhartidev@gmail.com','Bharti','Devgan', 'B', 601),
('O487', 'pallavi@gmail.com','Pallavi','Ram', 'C', 602),
('O911', 'vk@gmail.com','Virat','Kohli', 'A', 603),
('O142', 'ali@gmail.com','Muhammed','Ali', 'B', 402),
('O307', 'msingh@gmail.com','Manmohan','Singh', 'C', 201),
('O327', 'nani@gmail.com','Urmil','Kalra', 'A', 203),
('O907', 'maamu@gmail.com','Kamal','Kalra', 'B', 702),
('O407', 'mam@gmail.com','Seema','Kalra', 'C', 801),
('O387', 'manju@gmail.com','Manju','Munjal', 'A', 802),
('O222', 'veer@gmail.com','Veer','Singh', 'B', 803),
('O892', 'manpreet@gmail.com','Manpreet','Kaur', 'C', 103),
('O841', 'preetkaur@gmail.com','Preet','Sinha', 'A', 101),
('O648', 'nsingh@gmail.com','Navneet','Singh', 'B', 202),
('O645', 'apdhil@gmail.com','AP','Dhillion', 'C', 703),
('O765', 'alka@gmail.com','Alka','Yagnik', 'C', 101);
```

Populate table **Visitor**

```
INSERT INTO `visitor` (`rid`,`name`, `phno`, `block`, `flat`,
`type`,`intime`,`outtime`) VALUES
('R267', 'Rakesh', '9986573000', 'A', 701,'C','2022-25-10
08:30:01','2022-25-10 08:45:01'),
('O307', 'Sarita', '9916573000', 'C', 201,'D', '2022-12-08
10:45:01','2022-12-08 11:45:00'),
('O841', 'Divyansh Ghosh', '8310860218','A' ,101, 'G', '2022-03-04
12:15:01','2022-03-04 13:30:01'),
('R666', 'Naresh Gadi', '9935473000', 'B', 804,'G','2022-03-04
12:30:01','2022-03-04 18:30:00');
```

Populate table **Staff**

```
INSERT INTO `staff` (`sid`, `tid`, `fname`, `lname`, `dob`, `shift`,
`type`,`password`) VALUES
('S001', 'T111', 'Gopal', 'Murthy', '1985-06-12', 1,'plumber','1234'),
('S002', 'T222', 'Imtiaz', 'Ali', '1989-01-09', 1,'electrician','1234'),
('S003', 'T333', 'Abhishekh','Srinivasa', '1999-01-12', 1,
'housekeeping','1234'),
('S004', NULL, 'Sampoorna ', 'K', '1992-01-12', 1,'housekeeping','1234'),
('S005', NULL, 'Manjumma ', 'Kumari', '1990-02-10',1, 'housekeeping','1234'),
('S006', NULL, 'Madhu', 'Kishore', '1980-01-01',1, 'housekeeping','1234'),
('S007', NULL, 'Radha', 'Gopalchand', '1991-09-12',1, 'housekeeping','1234'),
('S008', NULL, 'Abdul', 'Mukit', '1986-01-03',1, 'security','1234'),
('S009', NULL, 'Ranjit', 'Kumar', '1996-01-12',2, 'security','1234'),
('S010', NULL, 'Sampath', 'Kumar', '1996-01-12',2, 'supervisor','1234');
```

Populate table **Ticket**

```
INSERT INTO `ticket` (`tid`, `rid`, `block`, `flat`, `is_resolved`,
`complaint`, `response`, `date_generated`) VALUES
('T111', 'R333','A',204, 0, 'MCB switch - light tripped. Check with
electrician', NULL, '2022-10-27'),
('T222', 'R444','B',304, 1,'No water in kitchen taps', NULL, '2022-10-28'),
('T333', 'R555', 'C',604, 0,'Dry waste not collected today', NULL,
'2022-10-26');
```

Populate table **Attendance**

```
INSERT INTO `attendance` (`sid`, `intime`, `outtime`) VALUES
('S010', '2022-11-01 08:08:48', NULL),
('S009', '2022-11-01 08:08:48', '2022-11-01 12:08:48'),
('S004', '2022-10-01 08:08:48', NULL),
('S008', '2022-11-01 08:08:48', NULL),
('S004', '2022-11-01 08:08:48', '2022-11-01 16:08:48'),
('S004', '2022-11-02 08:08:48', '2022-11-02 16:08:48'),
('S004', '2022-11-07 08:08:48', '2022-11-07 16:08:48'),
('S004', '2022-11-10 08:08:48', '2022-11-10 16:08:48'),
('S004', '2022-11-12 08:08:48', '2022-11-12 16:08:48'),
('S004', '2022-11-14 08:08:48', '2022-11-14 16:08:48'),
('S004', '2022-11-15 08:08:48', '2022-11-15 16:08:48'),
('S004', '2022-11-23 08:08:48', '2022-11-23 16:08:48'),
('S001', '2022-11-15 08:08:48', '2022-11-15 16:08:48'),
('S002', '2022-11-15 08:08:48', '2022-11-15 16:08:48'),
('S005', '2022-11-15 08:08:48', '2022-11-15 16:08:48'),
('S006', '2022-11-15 08:08:48', '2022-11-15 16:08:48');
```

Populate table **Record**

```
INSERT INTO `record` (`sid`, `month`) VALUES
('S001', 'NOV'),
('S002', 'NOV'),
('S003', 'NOV'),
('S004', 'NOV'),
('S005', 'NOV'),
('S006', 'NOV'),
('S007', 'NOV'),
('S008', 'NOV'),
('S009', 'NOV'),
('S010', 'NOV');
```

Join Queries

1. **Get the names, designation of staff who have been assigned a ticket at the apartment.**

```
SELECT  t.tid, s.sid, s.fname, s.lname, s.type
FROM ticket t
INNER JOIN staff s
ON t.tid = s.tid;
```

```
mysql> SELECT  t.tid, s.sid, s.fname, s.lname, s.type
    -> FROM ticket t
    -> INNER JOIN staff s
    -> ON t.tid = s.tid;
+------+------+----------+----------+--------------+
| tid  | sid  | fname    | lname    | type         |
+------+------+----------+----------+--------------+
| T111 | S001 | Gopal    | Murthy   | plumber      |
| T222 | S002 | Imtiaz   | Ali      | electrician  |
| T333 | S003 | Abhishekh | Srinivasa | housekeeping |
+------+------+----------+----------+--------------+
3 rows in set (0.03 sec)
```

2. Display the staff that hasn't come for work at the apartment.

```
SELECT s.sid, s.fname, s.lname, s.type
FROM staff s
```

```
WHERE s.sid NOT IN (SELECT DISTINCT sid FROM attendance);
```

```
mysql> SELECT s.sid, s.fname, s.lname, s.type
    -> FROM staff s
    -> WHERE s.sid NOT IN (SELECT DISTINCT sid FROM attendance);
+------+-----------+------------+--------------+
| sid  | fname     | lname      | type         |
+------+-----------+------------+--------------+
| S003 | Abhishekh | Srinivasa  | housekeeping |
| S007 | Radha     | Gopalchand | housekeeping |
+------+-----------+------------+--------------+
2 rows in set (0.01 sec)
```

3. Get the names, designations of staff that have worked for less than 8 hours on any day in month of November.

```
FROM staff s
WHERE s.sid IN (SELECT DISTINCT sid FROM attendance WHERE
timestampdiff(hour,intime, outtime) < 8);
```

```
mysql> SELECT s.sid, s.fname, s.lname, s.type
    -> FROM staff s
    -> WHERE s.sid IN (SELECT DISTINCT sid FROM attendance WHERE timestampdiff(hour,
intime, outtime) < 8);
+------+--------+-------+----------+
| sid  | fname  | lname | type     |
+------+--------+-------+----------+
| S009 | Ranjit | Kumar | security |
+------+--------+-------+----------+
1 row in set (0.00 sec)
```

4. Get details of the residents who have raised a ticket.

```
SELECT * FROM resident
WHERE rid IN(SELECT DISTINCT rid from ticket)
```

```
mysql> SELECT * FROM resident
    -> WHERE rid IN(SELECT DISTINCT rid from ticket);
+------+----------+------------+----------+------------+-------+------+-------------------+----------+
| rid  | fname    | lname      | is_owner | dob        | block | flat | email             | password |
+------+----------+------------+----------+------------+-------+------+-------------------+----------+
| R333 | Christon | DSouza     |        0 | 2001-08-08 | A     |  204 | dsouza@gmail.com  | abcd     |
| R444 | Bala     | Subramanya |        0 | 1999-01-12 | B     |  304 | bsubr@gmail.com   | 1234     |
| R555 | Namgyal  | G          |        0 | 1999-01-12 | C     |  604 | namgyal@gmail.com | abcd     |
+------+----------+------------+----------+------------+-------+------+-------------------+----------+
3 rows in set (0.04 sec)
```

5. Get the tickets that have not been resolved and number of days since.

```
SELECT tid, complaint, is_resolved, date_generated,
DATEDIFF(curdate(), date_generated ) as no_of_days
FROM ticket
WHERE is_resolved=0;
```

```
mysql> SELECT tid, complaint, is_resolved, date_generated,
    -> DATEDIFF(curdate(), date_generated ) as no_of_days
    -> FROM ticket
    -> WHERE is_resolved=0;
+------+-----------------------------------------------------+-------------+----------------+------------+
| tid  | complaint                                           | is_resolved | date_generated | no_of_days |
+------+-----------------------------------------------------+-------------+----------------+------------+
| T111 | MCB switch - light tripped. Check with electrician  |           0 | 2022-10-27     |         17 |
| T333 | Dry waste not collected today                       |           0 | 2022-10-26     |         18 |
+------+-----------------------------------------------------+-------------+----------------+------------+
2 rows in set (0.00 sec)
```

6. Fetch the staff details that have a record of attendance.
Correspondingly fetch the staff details that don't have a record of attendance yet.

```
mysql> SELECT sid, fname, lname, type
    -> FROM staff s
    -> WHERE EXISTS(SELECT sid
    -> FROM attendance
    -> WHERE sid =
    -> s.sid);
+------+------------+---------+-------------+
| sid  | fname      | lname   | type        |
+------+------------+---------+-------------+
| S001 | Gopal      | Murthy  | plumber     |
| S002 | Imtiaz     | Ali     | electrician |
| S004 | Sampoorna  | K       | housekeeping |
| S005 | Manjumma   | Kumari  | housekeeping |
| S006 | Madhu      | Kishore | housekeeping |
| S008 | Abdul      | Mukit   | security    |
| S009 | Ranjit     | Kumar   | security    |
| S010 | Sampath    | Kumar   | supervisor  |
+------+------------+---------+-------------+
8 rows in set (0.00 sec)

mysql> SELECT sid, fname, lname, type
    -> FROM staff s
    -> WHERE NOT EXISTS(SELECT sid
    -> FROM attendance
    -> WHERE sid =
    -> s.sid);
+------+----------+-----------+-------------+
| sid  | fname    | lname     | type        |
+------+----------+-----------+-------------+
| S003 | Abhishekh | Srinivasa | housekeeping |
| S007 | Radha    | Gopalchand | housekeeping |
+------+----------+-----------+-------------+
2 rows in set (0.00 sec)
```

7. Join with View

Create a view that stores resident details whom a visitor visits.
Every visitor has an associated resident rid.
Join the 2 tables resident and visitor as a view.

Query to get the flat and block of all cab drivers who has come to pick up a resident.
(visitor_type='C' -> Cab driver)

```
CREATE VIEW view3 AS SELECT r.rid, r.fname as resident_fname, r.lname
as resident_lname, r.block, r.flat, v.name as visitor_name, v.type as
visitor_type
FROM resident r
```

```
INNER JOIN visitor v
WHERE r.rid=v.rid;


SELECT * from view3;


SELECT flat, block from view3
WHERE visitor_type='C';
```

```
mysql> CREATE VIEW view3 AS SELECT r.rid, r.fname as resident_fname, r.lname as resident_lname, r.block, r.flat, v.name
 as visitor_name, v.type as visitor_type FROM resident r INNER JOIN visitor v where r.rid=v.rid;
Query OK, 0 rows affected (0.01 sec)

mysql> select * from view3;
+------+---------------+---------------+-------+------+---------------+--------------+
| rid  | resident_fname | resident_lname | block | flat | visitor_name   | visitor_type |
+------+---------------+---------------+-------+------+---------------+--------------+
| 0307 | Manmohan       | Singh          | C     |  201 | Sarita         | D            |
| 0841 | Preet          | Sinha          | A     |  101 | Divyansh Ghosh | G            |
| R267 | Preity         | Zinta          | A     |  701 | Rakesh         | C            |
| R666 | Ashraf         | Muhammed       | B     |  804 | Naresh Gadi    | G            |
+------+---------------+---------------+-------+------+---------------+--------------+
4 rows in set (0.01 sec)

mysql> SELECT flat, block from view3
    -> WHERE visitor_type='C';
```

```
mysql> SELECT flat, block from view3
    -> WHERE visitor_type='C';
+------+-------+
| flat | block |
+------+-------+
|  701 | A     |
+------+-------+
1 row in set (0.00 sec)
```

Aggregate Functions

1. Get number of residents that are tenants at the apartment.

```
SELECT count(rid) as no_of_tenants
FROM resident
WHERE is_owner=0;
```

```
mysql> SELECT count(rid) as no_of_tenants
    -> FROM resident
    -> WHERE is_owner=0;
+---------------+
| no_of_tenants |
+---------------+
|            12 |
+---------------+
1 row in set (0.00 sec)
```

2. List the occupancy of each block. (How many residents occupy each block)

```
SELECT count(rid) as occupancy, block
FROM resident
GROUP BY(block);
```

```
mysql> SELECT count(rid) as occupancy, block
    -> FROM resident
    -> GROUP BY(block);
+-----------+-------+
| occupancy | block |
+-----------+-------+
|         9 | A     |
|         9 | B     |
|        10 | C     |
+-----------+-------+
3 rows in set (0.00 sec)
```

3. List the maximum occupancy in all blocks. (Which block has the maximum occupants)

```
SELECT count(rid) as occupancy, BLOCK from resident
GROUP BY(block)
ORDER BY occupancy DESC LIMIT 1;
```

```
mysql> SELECT count(rid) as occupancy, BLOCK from resident
    -> GROUP BY(block)
    -> ORDER BY occupancy DESC LIMIT 1;
+-----------+-------+
| occupancy | BLOCK |
+-----------+-------+
|        10 | C     |
+-----------+-------+
1 row in set (0.03 sec)
```

4. Display the count of staff on duty at night.

```
SELECT count(sid) as night_duty_staff
FROM staff

WHERE shift=2;
```

```
mysql> SELECT count(sid) as night_duty_staff
    -> FROM staff
    -> WHERE shift=2;
+------------------+
| night_duty_staff |
+------------------+
|                2 |
+------------------+
1 row in set (0.00 sec)
```

Set Operations

1. **Get the tickets that is not been assigned to staff.**

```
SELECT tid FROM ticket
EXCEPT
SELECT tid FROM staff;
```

```
mysql> SELECT tid FROM ticket
    -> EXCEPT
    -> SELECT tid FROM staff;
Empty set (0.01 sec)
```

This implies all tickets raised have been assigned to some staff.

2.Get owners in block A who are also residents.

```
SELECT * FROM resident where rid IN(
SELECT rid FROM  resident WHERE block='A'
INTERSECT
SELECT oid FROM owner);
```

```
mysql> SELECT * FROM resident where rid IN(
    -> SELECT rid FROM  resident WHERE block='A'
    -> INTERSECT
    -> SELECT oid FROM owner);
+------+---------+---------+----------+------------+-------+------+---------------------+----------+
| rid  | fname   | lname   | is_owner | dob        | block | flat | email               | password |
+------+---------+---------+----------+------------+-------+------+---------------------+----------+
| 0327 | Urmil   | Kalra   |        1 | 1986-01-03 | A     |  203 | nani@gmail.com      | abcd     |
| 0387 | Manju   | Munjal  |        1 | 1986-01-03 | A     |  802 | manju@gmail.com     | abcd     |
| 0777 | Manisha | Solanki |        1 | 1992-01-12 | A     |  505 | manisha@gmail.com   | abcd     |
| 0841 | Preet   | Sinha   |        1 | 1996-01-12 | A     |  101 | preetkaur@gmail.com | 1234     |
| 0911 | Virat   | Kohli   |        1 | 1980-01-01 | A     |  603 | vk@gmail.com        | abcd     |
+------+---------+---------+----------+------------+-------+------+---------------------+----------+
5 rows in set (0.06 sec)
```

3. List those staff members that don’t have a record of attendance in the month of November.

```
SELECT * from staff where sid IN(
SELECT sid FROM staff
EXCEPT
SELECT sid FROM record WHERE sid IN (SELECT sid FROM record WHERE
month='NOV' ));

INSERT INTO `staff` (`sid`, `tid`, `fname`, `lname`, `dob`, `shift`,
`type`,`password`) VALUES
('S011', NULL, 'Fiyaz', 'Shaqib', '1996-01-12',2,
'electrician','1234');

DELETE FROM staff WHERE sid='S011';
```

```
mysql> INSERT INTO `staff` (`sid`, `tid`, `fname`, `lname`, `dob`, `shift`, `type`,`password`) VALUES
    -> ('S011', NULL, 'Fiyaz', 'Shaqib', '1996-01-12',2, 'electrician','1234');
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * from staff where sid IN(
    -> SELECT sid FROM staff
    -> EXCEPT
    -> SELECT sid FROM record WHERE sid IN (SELECT sid FROM record WHERE month='NOV' ));
+------+------+-------+--------+------------+-------+-------------+----------+
| sid  | tid  | fname | lname  | dob        | shift | type        | password |
+------+------+-------+--------+------------+-------+-------------+----------+
| S011 | NULL | Fiyaz | Shaqib | 1996-01-12 |     2 | electrician | 1234     |
+------+------+-------+--------+------------+-------+-------------+----------+
1 row in set (0.01 sec)

mysql> DELETE FROM staff WHERE sid='S011';
Query OK, 1 row affected (0.01 sec)
```

Functions

Function func2() returns the number of guests that visited the apartment within 24 hours.

```
DELIMITER $$

CREATE FUNCTION func2()

RETURNS int

READS SQL DATA

DETERMINISTIC

BEGIN

DECLARE tot_count int;

SET tot_count = (SELECT count(name) AS guests FROM visitor WHERE
type='G' and timestampdiff(hour, outtime, curdate()) < 24;);

RETURN tot_count;

END;$$

DELIMITER ;
```

```
mysql> select func2();
+---------+
| func2() |
+---------+
|       0 |
+---------+
1 row in set (0.00 sec)
```

This is correct as the last visitor's outtime is not 16th November. (This command was run on 17th November).

rid	name	phno	type	block	flat	intime	outtime
O307	Sarita	2147483647	D	C	201	2022-12-08 16:15:01	2022-12-08 17:15:00
O841	Divyansh Ghosh	2147483647	G	A	101	2022-03-04 17:45:01	2022-03-04 19:00:01
R267	Rakesh	2147483647	C	A	701	0000-00-00 00:00:00	0000-00-00 00:00:00
R666	Naresh Gadi	2147483647	G	B	804	2022-03-04 18:00:01	2022-03-05 00:00:00
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

On updating a O841's outtime to 17th november(today), the following result is shown.

```
UPDATE TABLE visitor SET intime = '2022-11-17 12:15:01', outtime =
'2022-11-17 14:15:00' WHERE type='G' and OID='O841';
```

```
mysql> select func2();
+---------+
| func2() |
+---------+
|       1 |
+---------+
1 row in set (0.00 sec)
```

This implies visitor for resident with rid O841 had a guest in the last 24 hours.

Triggers

A trigger function that validates the type of staff when one is created.
plumber, electrician, housekeeping, security, supervisor are the different types of staff.
Any other type inserted is not valid and trigger should check for that before inserting a row on table staff.

```
DELIMITER $$
CREATE TRIGGER check_staff_type BEFORE INSERT
ON STAFF FOR EACH ROW
BEGIN
DECLARE error_msg VARCHAR(255);
IF new.type != 'plumber'  or  new.type = 'electrician'  or
new.type='security' or new.type='supervisor' or new.type='housekeeping'
THEN
SET error_msg = ('Invalid type. Accepts any one of (plumber,
electrician, housekeeping, security, supervisor) as valid inputs');
SIGNAL SQLSTATE '45000'
SET MESSAGE_TEXT = error_msg;
END IF;
END $$
DELIMITER ;

INSERT INTO staff (sid, tid, fname, lname, dob, shift, type, password)
VALUES
('S111', NULL, 'Guru', 'Veer', '1985-06-12', 1,'delivery','1234');
```

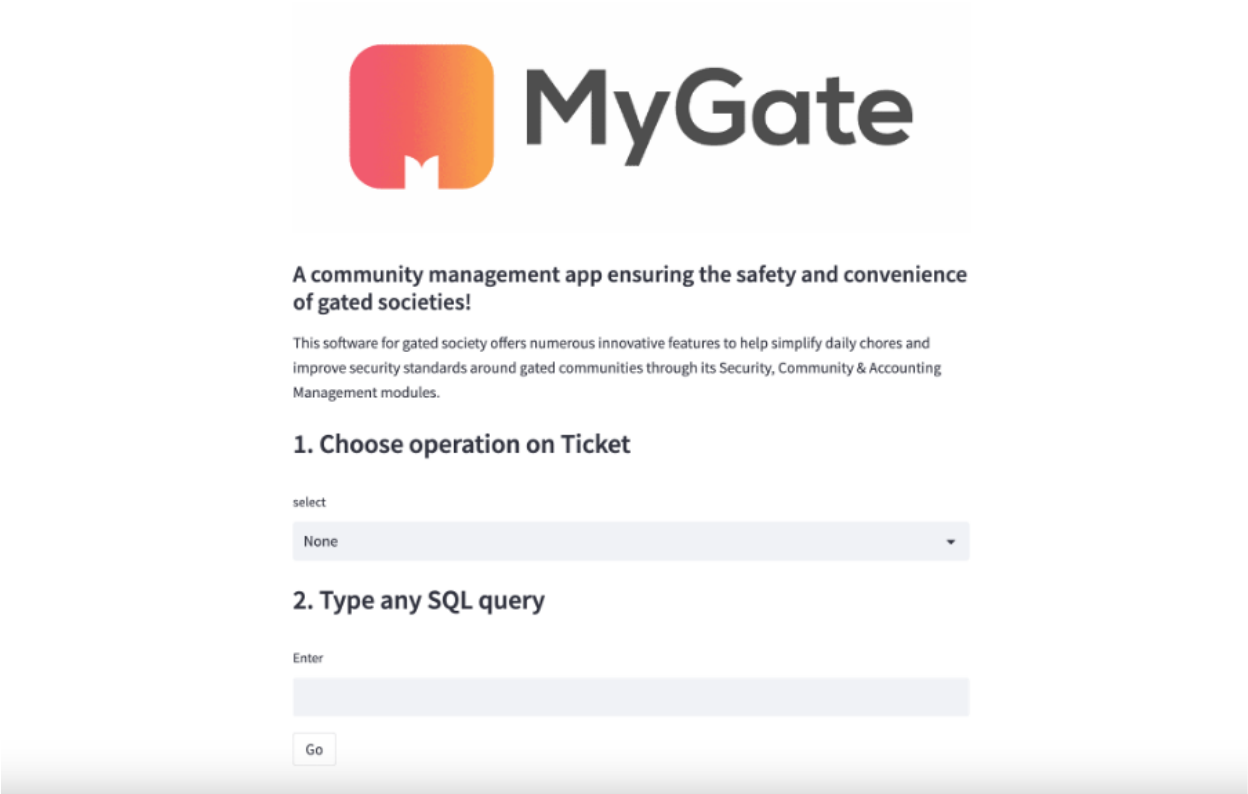
```
mysql> DELIMITER $$
mysql> CREATE TRIGGER check_staff_type BEFORE INSERT
    -> ON STAFF FOR EACH ROW
    -> BEGIN
    -> DECLARE error_msg VARCHAR(255);
    -> IF new.type != 'plumber'  or  new.type = 'electrician'  or new.type='security' or n
ew.type='supervisor' or new.type='housekeeping'
    -> THEN
    -> SET error_msg = ('Invalid type. Accepts any one of (plumber, electrician, housekeep
ing, security, supervisor) as valid inputs');
    -> SIGNAL SQLSTATE '45000'
    -> SET MESSAGE_TEXT = error_msg;
    -> END IF;
    -> END $$
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> INSERT INTO staff (sid, tid, fname, lname, dob, shift, type, password) VALUES
    -> ('S111', NULL, 'Guru', 'Veer', '1985-06-12', 1,'delivery','1234');
ERROR 1644 (45000): Invalid type. Accepts any one of (plumber, electrician, housekeeping,
security, supervisor) as valid inputs
```

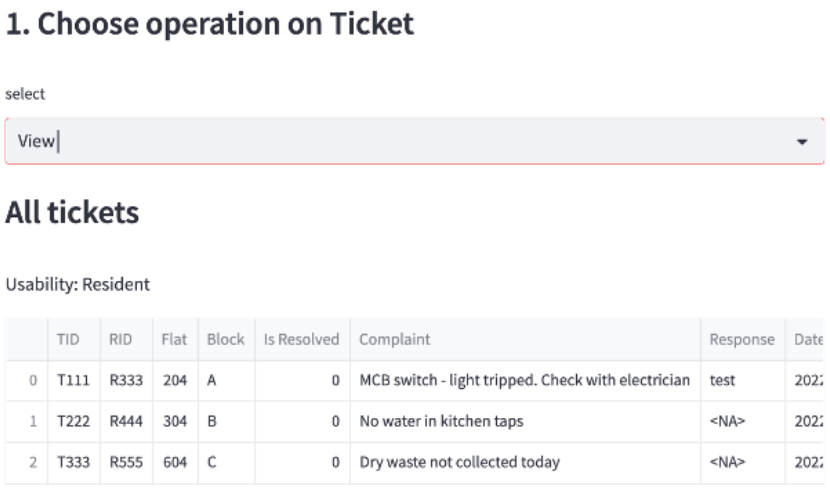
Developing a Frontend

Implemented using streamlit and mysql connector

Homepage

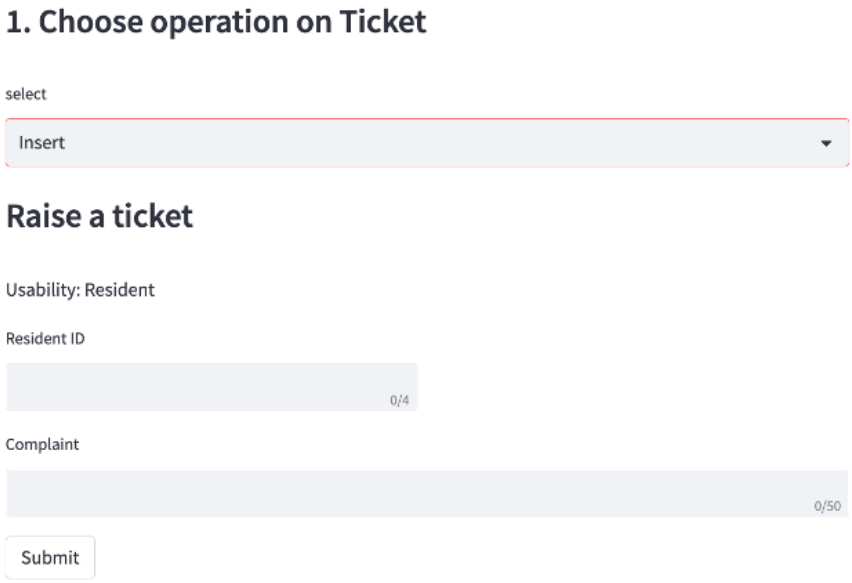


View/Read all tickets

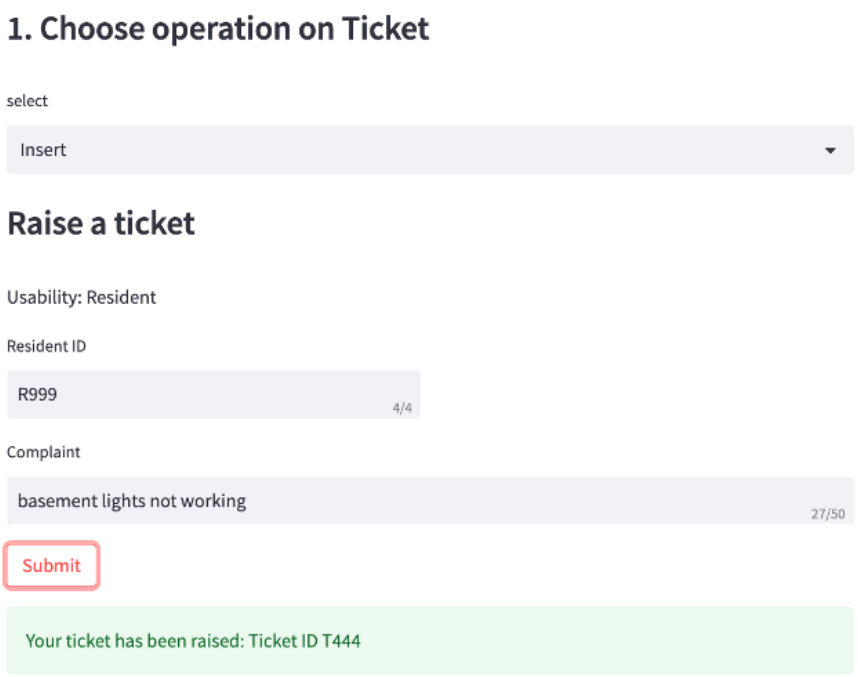


	TID	RID	Flat	Block	Is Resolved	Complaint	Response	Date
0	T111	R333	204	A	0	MCB switch - light tripped. Check with electrician	test	202
1	T222	R444	304	B	0	No water in kitchen taps	<NA>	202
2	T333	R555	604	C	0	Dry waste not collected today	<NA>	202

Insert ticket



Ticket created

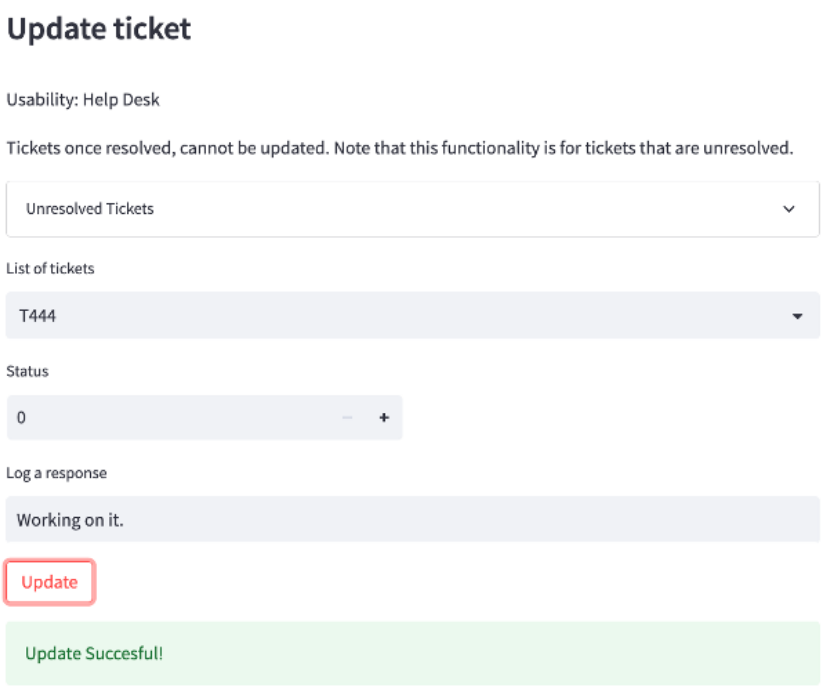


Ticket generated

	TID	RID	Flat	Block	Is Resolved	Complaint	Response	Date generated
0	T111	R333	204	A	0	MCB switch - light tripped. Check with electrician	test	2022-10-27
1	T222	R444	304	B	0	No water in kitchen taps	<NA>	2022-10-28
2	T333	R555	604	C	0	Dry waste not collected today	<NA>	2022-10-26
3	T444	R999	101	A	0	basement lights not working	None	2022-11-18

Updating newly created ticket

Response changed from ‘None’ to ‘Working on it’



	TID	RID	Flat	Block	Is Resolved	Complaint	Response	Date generated
0	T111	R333	204	A	0	MCB switch - light tripped. Check with electrician	test	2022-10-27
1	T222	R444	304	B	0	No water in kitchen taps	<NA>	2022-10-28
2	T333	R555	604	C	0	Dry waste not collected today	<NA>	2022-10-26
3	T444	R999	101	A	0	basement lights not working	Working on it.	2022-11-18

Delete operation

1. Choose operation on Ticket

select

Delete

Delete ticket

Usability: Help Desk

Current Tickets

Select ticket

T444

Do you want to delete T444?

Delete

Ticket has been deleted successfully!

1. Choose operation on Ticket

select

View

All tickets

Usability: Resident

	TID	RID	Flat	Block	Is Resolved	Complaint	Response	Date
0	T111	R333	204	A	0	MCB switch - light tripped. Check with electrician	test	2022
1	T222	R444	304	B	0	No water in kitchen taps	<NA>	2022
2	T333	R555	604	C	0	Dry waste not collected today	<NA>	2022

Any SQL query

Get number of residents that are tenants at the apartment

Select count(rid) as no_of_tenants
From resident
Where is_owner=0;

Select * from attendance;

2. Type any SQL query

Enter

```
Select * from attendance;
```

Go

Success!

	0	1	2
0	S001	2022-11-15T13:38	2022-11-15T2
1	S002	2022-11-15T13:38	2022-11-15T2
2	S004	2022-10-01T13:38	<NA>
3	S004	2022-11-01T13:38	2022-11-01T2
4	S004	2022-11-02T13:38	2022-11-02T2
5	S004	2022-11-07T13:38	2022-11-07T2
6	S004	2022-11-10T13:38	2022-11-10T2
7	S004	2022-11-12T13:38	2022-11-12T2
8	S004	2022-11-14T13:38	2022-11-14T2
9	S004	2022-11-15T13:38	2022-11-15T2