Free Strata specs

\* = PK

# Data Dictionary

## strata

|  |  |  |  |
| --- | --- | --- | --- |
| KEY | Field | Data-Type | Constraints |
| PK | strata\_plan\_id | TEXT | NOT NULL, UNIQUE, PRIMARY, CHECK(SPnnnnn) |
|  | name | TEXT |  |
|  | office\_address | TEXT |  |
|  | office\_phone | TEXT | CHECK(0-9) |
|  | office\_email | TEXT | CHECK(*address*@*domain*) |
|  | strata\_address | TEXT |  |

## Assumptions

* + - 1. Council issued strata plan id can be used as a unique id. If not, replace with a SERIAL id.

## users

|  |  |  |  |
| --- | --- | --- | --- |
| KEY | Field | Data-Type | Constraints |
| PK | user\_id | SERIAL | NOT NULL, UNIQUE, PRIMARY |
|  | email | TEXT | NOT NULL, CHECK(*address*@*domain)* |
|  | password | TEXT | NOT NULL, cryptographed |
|  | name | TEXT |  |
| FK | strata | TEXT |  |
|  | unit | INTEGER | NOT NULL |
| FK | role | TEXT | NOT NULL |
|  | phone | TEXT | CHECK(0-9) |
|  | user\_status | TEXT | NOT NULL, CHECK[new, active, deleted], DEFAULT = new |
|  | creation\_time | TIMESTAMPTZ | DEFAULT = current time |

* No idea what I’m doing with password security

## roles

|  |  |  |  |
| --- | --- | --- | --- |
| KEY | Field | Data-Type | Constraints |
| PK | name | TEXT | NOT NULL, UNIQUE, PRIMARY, CHECK[owner, strata manager, building manager, tenant] |
|  | description | TEXT | NOT NULL |

## issues

|  |  |  |  |
| --- | --- | --- | --- |
| KEY | Field | Data-Type | Constraints |
| PK | issue\_number | SERIAL | NOT NULL, UNIQUE, PRIMARY |
|  | title | TEXT | NOT NULL |
|  | description | TEXT | NOT NULL |
| FK | user\_id | INTEGER | NOT NULL |
|  | status | TEXT | NOT NULL, CHECK[submitted, accepted, addressing, closed, rejected],  DEFAULT = submitted |
|  | creation\_time | TIMESTAMPTZ | DEFAULT = current time |
|  | last\_update | TIMESTAMPTZ | DEFAULT = current time |

## issue\_comments

|  |  |  |  |
| --- | --- | --- | --- |
| KEY | Field | Data-Type | Constraints |
| PK | comment\_number | SERIAL | NOT NULL, UNIQUE, PRIMARY |
|  | comment | TEXT | NOT NULL |
| FK | user\_id | INTEGER | NOT NULL |
| FK | Issue\_number | INTEGER | NOT NULL |
|  | comment\_time | TIMESTAMPTZ | DEFAULT = current time |

## Assumptions

* + - 1. Users will be able to leave back and forth comments, displayed by order of TIMESTAMPTZ.

## issue\_files

|  |  |  |  |
| --- | --- | --- | --- |
| KEY | Field | Data-Type | Constraints |
| PK | file\_number | SERIAL | NOT NULL, UNIQUE, PRIMARY |
|  | description | TEXT |  |
|  | file | BYTEA | NOT NULL |
| FK | user\_id | INTEGER | NOT NULL |
| FK | Issue\_number | INTEGER |  |
|  | upload\_time | TIMESTAMPTZ | DEFAULT = current time |

## Assumptions

* + - 1. Users will be able to post files after the issue is initially lodged.
      2. Strata representatives will be able to submit files. If not, the ‘user’ field is unnecessary.
* file – BYTEA is one of two PostgreSQL data-types. BYTEA will use large amounts of memory for larger files and is limited by filesize. The other data-type, Large Object, can handle much larger files but has serious security floors. We will likely want to consider limiting file-sizes, downsampling or only storing links to media hosted elsewhere.

# API

User

* createUser (login, password, name, role, phone, email)
* editUser(login, name, role, phone, email)
* editPassword(login, password)
* getUserInfo(login) -> login, name, role, phone, email, creation

Role

* getRoles() -> name, description

Issue

* createIssue(title, description, user, array of files) -> issue\_number
* getIssue(issue\_number) -> title, description, user, status, array of files, array of comments, creation
* addFile(description, user, issue\_number, file) -> number
* addComment(comment, user, issue\_number) -> number
* changeStatus(issue\_number, new\_status)
* deleteFile(number)
* deleteComment(number)