Django Project

Before you start working on this project, I *strongly* recommend you go through the tutorial in the link below to familiarize yourself with Django.

<https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django>

# Models

The models define the fields that will be stored in the SQLite database. Create a new class for every function/mechanism that required data to be stored.

*Locallibrary > Catalog > models.py*

## RoomKey

This model represents the rooms/facilities that have keys available at WalterFedy. This could be expanded to include all rooms so that the RoomKey model can be related and accessed by other models.

* This model does not represent the physical key but what the key unlocks
* Room name, key symbol, and bitting number all serve as identifiers for rooms

### Fields

Room\_name: name of room/facility

Room\_des: description of the room/facility

Key\_symb: string of characters unique to roomkey (optional)

Bitting\_num: set of integers unique to roomkey (optional)

Office: Kitchener/Hamilton

## KeyInstance

This model represents each individual copy of a key related to a specific RoomKey. Each KeyInstance has a unique ID code that is stored in the database. Each KeyInstance also can have a user assigned to it which allows it to be signed in and out.

### Fields

RoomKey: relates each KeyInstance to a RoomKey in a many-to-one relationship (allows many KeyInstances to be assigned to a single RoomKey)

KeyRequest: relates a KeyRequest to a KeyInstance in a one-to-one relationship

Status: status of the particular KeyInstance (available/on loan/reserved)

Date\_out: date that the key is issued

Date\_in: date that the key is to be returned (optional)

Key\_notes: notes if applicable

Id: unique ID for specific KeyInstance

## KeyRequest

This model is a request model. It will store all request details made by users for **RoomKeys**.

### Fields

Roomkey: defines the RoomKey that the KeyRequest is made for

Requester: Django user that is currently logged in and making the request

Borrower: person that the key is for

(this field is required in the situation that managers request keys on behalf of an employee)

Request\_status: status of key request (pending/approved/denied)

Date\_requested: automatically added to database upon submission

Date\_completed: date that the request is updated/fulfilled

Request\_comments: comments about request

## MaintenanceRequest

A request model that holds the maintenance requests submitted. This is not related to any other models and is standalone.

### Fields

Requester: Django user that is currently logged in and making the request

Office: Kitchener/Hamilton

Urgency: Low/Medium/High

Status: Submitted/Work-in-Progress/Completed

Date\_submitted: Time request is made (automatically recorded)

Date\_completed: Date that the request is updated/fulfilled

Request\_comments: Any additional notes that is needed with request

# Work Flow

This section outlines how a user will interact with the system and how the system responds to their requests and interactions.

## Keys

### User is Requesting a New Key

1. User navigates to the specific RoomKey they want a KeyInstance for
2. If a KeyInstance for the RoomKey has a status of “available”, a request button will be available for the user to press
   1. No available KeyInstance will prompt the user to contact the service team
3. This will lead them to the form that will have two fields from the KeyRequest model that the user must manually fill in; the borrower field and the request\_comments field
4. User presses the submit button which triggers a pop up with the key holder agreement (see below)
   1. Submit button can be pressed once the checkbox that indicates that the user agrees to the statements is selected
5. Once the form is submitted, the first KeyInstance with an “available” status will be assigned to the user
   1. Fields from the KeyRequest model that will be automatically filled in
      1. roomkey field: RoomKey that was selected earlier is recorded
      2. requester field: user that is logged in is recorded
      3. date\_requested field: time of submission is recorded
   2. Fields from the KeyInstance model that will automatically filled in
      1. Keyrequest field: KeyRequest is assigned to the KeyInstance
      2. Status field: changed to “on-loan”
      3. Date\_out: date that the status field changed is recorded

### Key Holder Agreement

This was created by HR and requires the user acknowledge reading and agreeing to the terms before they can borrow a key.

1. The key herein remains property of WalterFedy/AEC Developments
2. The key is entrusted to me for my exclusive use. I will not duplicate, loan, transfer, exchange, misuse, modify, alter, or otherwise allow its use or possession by any other person
3. I will report its loss, theft, or destruction immediately to my department
4. If this key becomes lost, stolen, or not available for return, I will pay the key replacement charge
5. Upon employment termination or request from team leader, I will return the key promptly, in person, and only to the Operations team

## Maintenance Request

* On the homepage of the maintenance request function, a user can create a new request or view pending and completed requests

### Submitting a Request

1. User selects the “Create” button to create a new request
2. Leads to new maintenance request form
   1. Fields from the MaintenanceRequest model that need to be filled in by the user
      1. Office: default Kitchener
      2. Urgency: default low
      3. Request\_comments: must outline reason for submitting a request
   2. Fields from the MaintenanceRequest model that will be filled in upon submission
      1. Requester: user logged in
      2. Status: submitted
      3. Date\_submitted: date the form is created
3. Request will then be available in the “Pending Requests” section

### Viewing Requests (User)

* User can view their current request under the “Pending Requests” or “Completed Requests” sections
  + The location of the request depends on the status of the request
* User can expand the request to see a detailed overview of the request

### Viewing and Updating Requests (Admin)

* Admin can view requests the same way users do but once the request is expanded to the detailed overview, the admin will see an “Update” button
* Update button will lead to an update form with the following fields from the MaintenanceRequest model that the admin can update
  + Status: default work-in-progress
  + Date\_completed: default to current date (this field will show as date updated on the user side)
  + Request\_comments: allows the admin to add any details necessary to the request