Pilot Logbook Manager

Jennifer Sebek Kemp

March 17, 2023

Project Overview

Pilot Logbook Manager is a digital pilot logbook application. This application is designed to be a user-friendly tool for all kinds of pilots. Pilot Logbook Manager will assist pilots in keeping an accurate and complete logbook, allow pilots to keep track of their progress toward earning pilot certifications, store current pilot certification information, and aid pilots in complying with FAA medical requirements.

Purpose

Maintaining a logbook is a chore that many pilots do not enjoy, and using a classic paper logbook is not always the best solution. Paper logbooks must be safeguarded so that critical flight information is not lost, and for this reason many pilots do not want to carry their logbook with them when traveling. Digital logbooks allow pilots to enter flight information using mobile devices that pilots are carrying already. Pilot Logbook Manager aims to offer a secure and persistent record of a pilot's flight time. The application will take care of totaling a user's flight time in all the necessary categories to save the user from needing to do that work themselves and also to prevent possible mistakes. It will also be designed to walk the user through the steps of entering their information easily so that users will not be frustrated when beginning to use the application. Finally, it will display time and progress reports in a simple and easy to understand way, as well as display information to let the pilot know if they are meeting current requirements with their medical certificate.

User Personas

There are three main types of users that the application plans to support initially:

- Student pilots who are just beginning their training and are working towards their initial pilot certificates and type ratings. These users will be the most unfamiliar with the requirements they need to meet for each goal and will find progress reports helpful to visualize where they are in their training. They will be able to share their progress with flight instructors who need to verify flight time before signing off on their certifications.
- Recreational pilots who need to maintain a log of their hours and have different medical requirements from professional pilots.
- Professional pilots who are very busy and need to be able to enter their flight logs easily and accurately. They need to maintain records for inspection and for employment reasons. These pilots also have stricter medical requirements that they need to meet in order to legally do their jobs.

Value and Benefits

The main purpose of this application is to provide an improvement upon the options currently available to pilots. Many digital logbook applications do not have all the features pilots desire or are so difficult to use that pilots abandon them. The goal of Pilot Logbook Manager is to give pilots a tool they will enjoy using and will stick with for long-term use. Pilot referrals for the product will then increase the number of users.

Use and Interaction

To maintain logs most accurately, the best practice is for a pilot to enter their flight information as they complete flights. Many pilots do not follow through with this because adding one more task to their day is a hassle. Pilot Logbook Manager will give clear instructions to the user to set up their profile. Then, this application will offer a quick solution by allowing pilots to complete each log by filling in just a few pieces of information for each flight. As much information as possible will be automatically entered for the pilot, and the pilot can simply adjust those fields as needed. Additionally, pilots will use the summary and reporting features to view their current total hours in important categories. Finally, pilots will view their current medical information so that they know when it is time to schedule an appointment to renew their medical certificate.

Minimum Viable Product

Overview

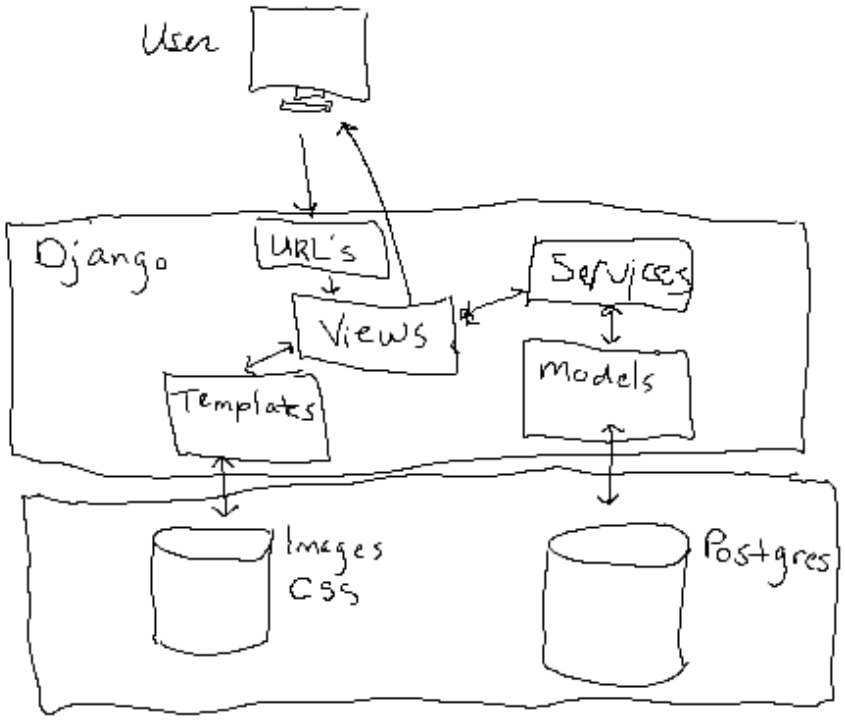
To create a useful application, at a minimum the application will need to collect some data from the user, including their age, the date and class of their last FAA medical examination, any current certifications held, and current aircraft type. This information will be used to populate fields in the flight log entry form to make entering flight logs faster for the user. A user will still be able to enter flight logs without this information, but they will need to complete more fields in the form. Users will be able to navigate to pages that show their FAA medical status, current flight hour totals and a list of the most recent flight log entries, and progress toward meeting minimum hours for certifications. Users will be able to update or make corrections to data that has been entered. They will also be able to search for flight log entries and display these as a list for review.

Features

- Users will be able to log in to the app.
- Users will be able to create, read, update, and delete data related to their current flight certifications.
- Users will be able to create, read, update, and delete data related to their current medical certification.
- Users will be able to create, read, update, and delete data related to individual flights.
- Users will be able to view reports of their progress toward certifications.
- Users will be able to view statistics of their current flight hour totals.
- Users will be able to view a list of flight data in logbook form.
- Users will be able to view the date that their next medical certification is due.

Architecture

This application will use the Django framework with PostgreSQL. The front end will use Django views to render HTML pages. The backend will use Django URL routes with models and templates to generate views. The Django models will interact with PostgreSQL to work with data, and the Django templates will retrieve necessary static files. A layer for the business logic, services, will work between models and views.



Services needed for the application include a flight log service, a medical certificate service, an account details service, a logbook summary service, and a reports service.

Data

The application will store the following data:

User Object:

User
user_id: integer, primary key email: string password: string

Account Detail Object:

Account Detail
account_detail_id: integer, primary key user_id: integer, foreign key first_name: string last_name: string age: integer pilot_in_command: boolean current_aircraft_type: string

Pilot Certificate object:

Certificate
certificate_id: integer certificate_name: string

Rating object:

Rating
rating_id: integer rating_name: string

Medical exam object:

Medical
medical_id: integer, primary key
user_id: integer, foreign key
exam_date: date
class: integer
examiner_name: string
examiner_phone: string
notes: string

Flight data object:

Flight
flight_id: integer, primary key
user_id: integer, foreign key
flight_date: date
origin: string
destination: string
tail_number: string
total_time: float
landings: integer
multi_engine_time: float
single_engine_time: float
vfr_time: float
ifr_time: float
night_time: float
notes: string

The account detail, medical, and flight data objects will be stored in tables and related to the user with the user id as a foreign key. There will be a table of pilot certificate types and rating types, and tables that relate users with pilot certificate types and rating types.