

利用予約システムと総合水槽実験棟の紹介

神戸大学大学院海事科学研究科技術部 杉原 直樹

1. 背景

2017 年 4 月に私が海事科学研究科技術部に採用されたとき、総合水槽実験棟の各水槽を利用しようと思うと、各水槽の担当教員に連絡し、予定の調整を行う必要があった。それをもう少し手軽にできないかと思い、途中グループウェアのクラウドサービスの利用を経て、自身で開発した利用予約システムを利用するに至り、2019 年 4 月より運用を開始して約 5 年が経過した。

学生時代に簡単な登録サービスを作成したことがあったので、その時に利用した Python と、Web アプリケーションフレームワークの Django を用いて開発を行った。その際、知識のアップデートも兼ねることができた。

ここでは、その利用予約システムの構成や、運用中のトラブルを紹介する。また、このシステムで利用予約が可能な総合水槽実験棟の各実験装置の紹介もおこなう。

2. システムの構成

- Python 3.8.10
- Django 3.2.7
- Dell 社製 OptiPlex 745(Intel Core2Duo E6300@1.86GHz、メモリ 4GB)
- Windows10 Pro 64bit
- VirtualBox 6.0.4
- Ubuntu20.04 64bit
- Nginx 1.18.0
- Bootstrap4
- 学内からのみアクセス可能

2.1. Django

Django は Python で Web アプリケーションを開発するためのフレームワークである。データベース操作のためのライブラリや、テストサーバ機能、管理サイト生成機能など、Web アプリケーションを開発するのに便利で、手間を減らせるような機能が多数提供されている。簡単なアプリケーションを開発するだけなら、Python やデータベースの知識があまりなくとも作ることができる。

3. 開発

3.1. 必要な機能

以下に、実装した機能をいくつか示す。*印がついているものは Django のライブラリに機能として入っており、ほとんど手間なく導入できるものである。

- ・ログイン機能*
- ・ユーザ名と表示名を分ける*
- ・予約重複の排除
- ・メール通知(登録時、利用前日)

- ・浅水槽の予定入力時にのみ水位等の入力欄を出現させる
- ・管理サイト＊
- ・予約をカレンダーに表示
- ・自分の登録した予約を一覧表示
- ・利用予約時の入力内容の確認画面
- ・日付入力時にカレンダーから入力
- ・予定カレンダーから予定入力画面に移る際に日付が既に入っているようにする
- ・ユーザによる閲覧制限

3.2. 苦労した点

開発中、特に苦労した点を示す。

- ・難しいことをしようとすると、Python の理解が必要

2.1 節に「簡単なアプリケーションを開発するだけなら、Python やデータベースの知識があまりなくとも作ることができる」と書いたが、Python の知識が浅いままだと、複雑な機能の実装で詰まってしまう、ということである。例えばクラスについてだが、Django の機能の多くはクラスで提供されており、それゆえクラスに関する知識は重要である。Django で提供されているクラスをそのまま使うのならよいが、ちょっと機能を追加した上で使いたい、などと考え始めると、クラスの継承とは何か、引数はどういう流れで処理されていくのか、と知識が要求され、当然不足していると正しくコードが書けない。「引数が多すぎる」というエラーメッセージは私が開発中にぶつかった大きな壁の一つである。

対処としては、エラーメッセージで検索をかけて、どういう対策をしなければいけなかったのか調べ、出てきた解説ページを読んで不足している情報を Python や Django のリファレンスで調べ、適切だと思われる形でコードを修正し実行、をエラーが出なくなるまで繰り返すことで知識を身につけた。

3.3. 開発期間

2018 年 4 月から 2019 年 4 月上旬までである。業務の傍らに作業を進めており、長期間触ることのできない時期もあったため、実際に作業を行ったのは 6 ヶ月程度である。

4. 運用中のトラブル

かなり自業自得な話ではあるが、利用予約システムが稼働している Ubuntu をうっかり破壊してしまうというトラブルを起こした。

2021 年 9 月のことであった。その時はベースの Windows の方でトラブルが生じており、本体の再起動をする必要があった。そのついでに仮想マシンの Ubuntu をアップデートしようとした。「apt update」を実行するとエラーが生じたため、調べてみると、python3 のシンボリックリンクの宛先をデフォルトの Python3.6 から予約システム用に導入した Python3.8 に以前書き換えたのが原因であることが分かり、それを戻そうとしたところ、コマンドを間違え Python3.6 の実行ファイルそのものを削除してしまった。この段階でまずい状況に気づき、予約システム自体は Python3.8 で動いているので当面は問題ないものの、念のためデータベースと予約システムのプログラムのバックアップを実施した。現状のバックアップ状況を確認すると、物理 PC 自体のイメージバックアップと、仮想マシンのスナップショットがあるようだったが、前者は物理 PC ごと電源を落とさないと復旧することができず、現状予約システムの仮想マシンの電源を落とすと再度起動したときに正常に起動するかどうか定かではなかったため、後者はスナップショットを確認するボタンが見当たらず、仮想マシンを終了しないと確認できないと思い込んだため、バックアップからの復旧を断念した。現状の仮想マシンを修復するか、新しく仮想マシンを

一から構築するかの 2 択で動くことにした。しかし、うまい修復方法を見つけないことができなかったため、以前の Ubuntu18.04 から Ubuntu20.04 に変更した上で新しく仮想マシンを立ち上げ、バックアップを復元して利用予約システムを復旧させた。

バックアップ体制の改善を決意するには十分な体験であったため、まずスナップショット機能の使い方を確認した。また利用予約のプログラムの変更や、仮想マシンそのもののアップデートの前と後にはスナップショットの作成をすることとした。さらに、メインの仮想マシンの他にテストサーバとしてメインの仮想マシンのクローンを作成した。仮想マシンに変更を加える場合でも、テストサーバで試してからメインの方に適用することができるようになる。むしろ今までなぜそうしてなかったのか、という話ではあるのだが、少しはましな状況になった。

5. 今後の課題

5.1. セキュリティ

Django にはさまざまな Web アプリケーション攻撃に対する対策が実装されている。クロス・サイト・スクリプティング、クロス・サイト・リクエスト・フォージェリ、SQL インジェクションなどの対策が含まれているようだが、まず私自身に Web サイトのセキュリティに関する教養があまりない。公式のリファレンスや、非公式の解説サイトに「このように書けばこの攻撃に対する対策ができるから書いてくださいね」と書かれてはいるが、その攻撃がどのようなものなのか、その解説に書かれている程度の知識しかない。また、その対策もブラックボックスで、どのような仕組みで対策が行われているのか定かではない。

公開範囲が学内のみとはいえ、それに安心せず知識を仕入れる必要性を感じている。

5.2. サーバが古い

前述の通り、利用予約システムを稼働させているサーバは私の採用前から水槽棟に保管されていた PC であり、スペックを見るに明らかに古い PC である。2025 年に Windows10 のサポート終了が控えているが、Windows11 へのアップグレードは当然非対応である。ハードを新しいものに切り替えるか、ベースの OS を Windows にするのを諦める必要がある。



図 1 利用予約システムのトップページ



図 2 予約画面

総合水槽実験棟利用予約システム

実験計画の編集

既にある予約と利用時間が重複しています。

実験タイトル

利用開始日時

利用終了日時

利用する装置

浅水槽の水位の初期設定 [cm]

未定の場合は「-1」と入力してください。

☒ 曳航台車の使用

☒ 造波機の使用

図 3 登録時に出る重複エラー

Django 管理サイト

変更する schedule を選択

番号	実験タイトル	装置名	利用開始日時	利用終了日時
57	レーン半端試験	hachimoto	2019年10月7日11:00	浅水槽
58	水漏れ試験	hachimoto	2019年10月7日11:00	浅水槽
59	曳航機試験	nohwa	2019年10月7日11:00	浅水槽
60	曳航機試験	nohwa	2019年10月7日11:00	浅水槽
61	曳航機試験	nohwa	2019年10月7日11:00	浅水槽
62	曳航機試験	nohwa	2019年10月7日11:00	浅水槽
63	曳航機試験	nohwa	2019年10月7日11:00	浅水槽
64	曳航機試験	nohwa	2019年10月7日11:00	浅水槽
65	曳航機試験	nohwa	2019年10月7日11:00	浅水槽
66	曳航機試験	nohwa	2019年10月7日11:00	浅水槽
67	曳航機試験	nohwa	2019年10月7日11:00	浅水槽
68	曳航機試験	nohwa	2019年10月7日11:00	浅水槽
69	曳航機試験	nohwa	2019年10月7日11:00	浅水槽
70	曳航機試験	nohwa	2019年10月7日11:00	浅水槽
71	曳航機試験	nohwa	2019年10月7日11:00	浅水槽
72	曳航機試験	nohwa	2019年10月7日11:00	浅水槽
73	曳航機試験	nohwa	2019年10月7日11:00	浅水槽
74	曳航機試験	nohwa	2019年10月7日11:00	浅水槽
75	曳航機試験	nohwa	2019年10月7日11:00	浅水槽
76	曳航機試験	nohwa	2019年10月7日11:00	浅水槽
77	曳航機試験	nohwa	2019年10月7日11:00	浅水槽
78	曳航機試験	nohwa	2019年10月7日11:00	浅水槽
79	曳航機試験	nohwa	2019年10月7日11:00	浅水槽
80	曳航機試験	nohwa	2019年10月7日11:00	浅水槽
81	曳航機試験	nohwa	2019年10月7日11:00	浅水槽
82	曳航機試験	nohwa	2019年10月7日11:00	浅水槽
83	曳航機試験	nohwa	2019年10月7日11:00	浅水槽
84	曳航機試験	nohwa	2019年10月7日11:00	浅水槽
85	曳航機試験	nohwa	2019年10月7日11:00	浅水槽
86	曳航機試験	nohwa	2019年10月7日11:00	浅水槽
87	曳航機試験	nohwa	2019年10月7日11:00	浅水槽
88	曳航機試験	nohwa	2019年10月7日11:00	浅水槽
89	曳航機試験	nohwa	2019年10月7日11:00	浅水槽
90	曳航機試験	nohwa	2019年10月7日11:00	浅水槽
91	曳航機試験	nohwa	2019年10月7日11:00	浅水槽
92	曳航機試験	nohwa	2019年10月7日11:00	浅水槽
93	曳航機試験	nohwa	2019年10月7日11:00	浅水槽
94	曳航機試験	nohwa	2019年10月7日11:00	浅水槽
95	曳航機試験	nohwa	2019年10月7日11:00	浅水槽
96	曳航機試験	nohwa	2019年10月7日11:00	浅水槽
97	曳航機試験	nohwa	2019年10月7日11:00	浅水槽
98	曳航機試験	nohwa	2019年10月7日11:00	浅水槽
99	曳航機試験	nohwa	2019年10月7日11:00	浅水槽
100	曳航機試験	nohwa	2019年10月7日11:00	浅水槽

図 4 Django 標準のものを使った管理サイト

6. 総合水槽実験棟の実験装置の紹介

6.1. 浅水槽

長さ 60m、幅 6m、水深最大 1.5m の曳航台車、造波装置が付属した水槽である。最大の特徴として水深を完全排水の 0 から 1.5m まで自由に変更できるという点である。それにより、水深の浅い港などを模擬した実験や、水底に置いたおもりに係留した浮体の実験などが可能である。

6.2. 回流水槽

閉水路中で水を循環させて、模型などの被験体と水との間で相対的な流れを生み出す水槽である。プロペラの性能計測など、没水体に対する実験や、物体に対して長時間流れを当てたい実験をするときに有用である。

6.3. アンカー水槽

砂が詰まった水槽である。水を入れて、泥とすることもできる。船を留めておくためのアンカーが、水底の砂や泥に引っかかる力である把駐力を調べる実験が可能である。



図 5 浅水槽



図 6 回流水槽



図 7 アンカー水槽

参考文献

- ・ Django ドキュメント (バージョン 3.2) <https://docs.djangoproject.com/ja/3.2/>
- ・ Python ドキュメント (バージョン 3.8) <https://docs.python.org/ja/3.8/>
- ・ Python Django 入門 https://qiita.com/kaki_k/items/511611cadac1d0c69c54