

Swift project

-App using GPS

目的:

教育で学んでいるJava Script以外のSwift言語を自ら学び、
開発することで新しいことに取り組む際の調べる、問題解決する
スキルの醸成する。
Swiftとお友達になる。

MVP:

リストに情報を表示させ、選択すると詳細画面に遷移する。
位置情報を取得する。

アプリについて

消去ボタン
※機能未実装

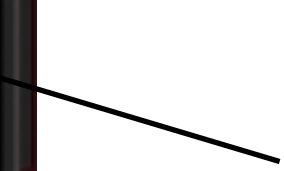


記事投稿



記事一覧に追加する新規投稿画面に遷移します

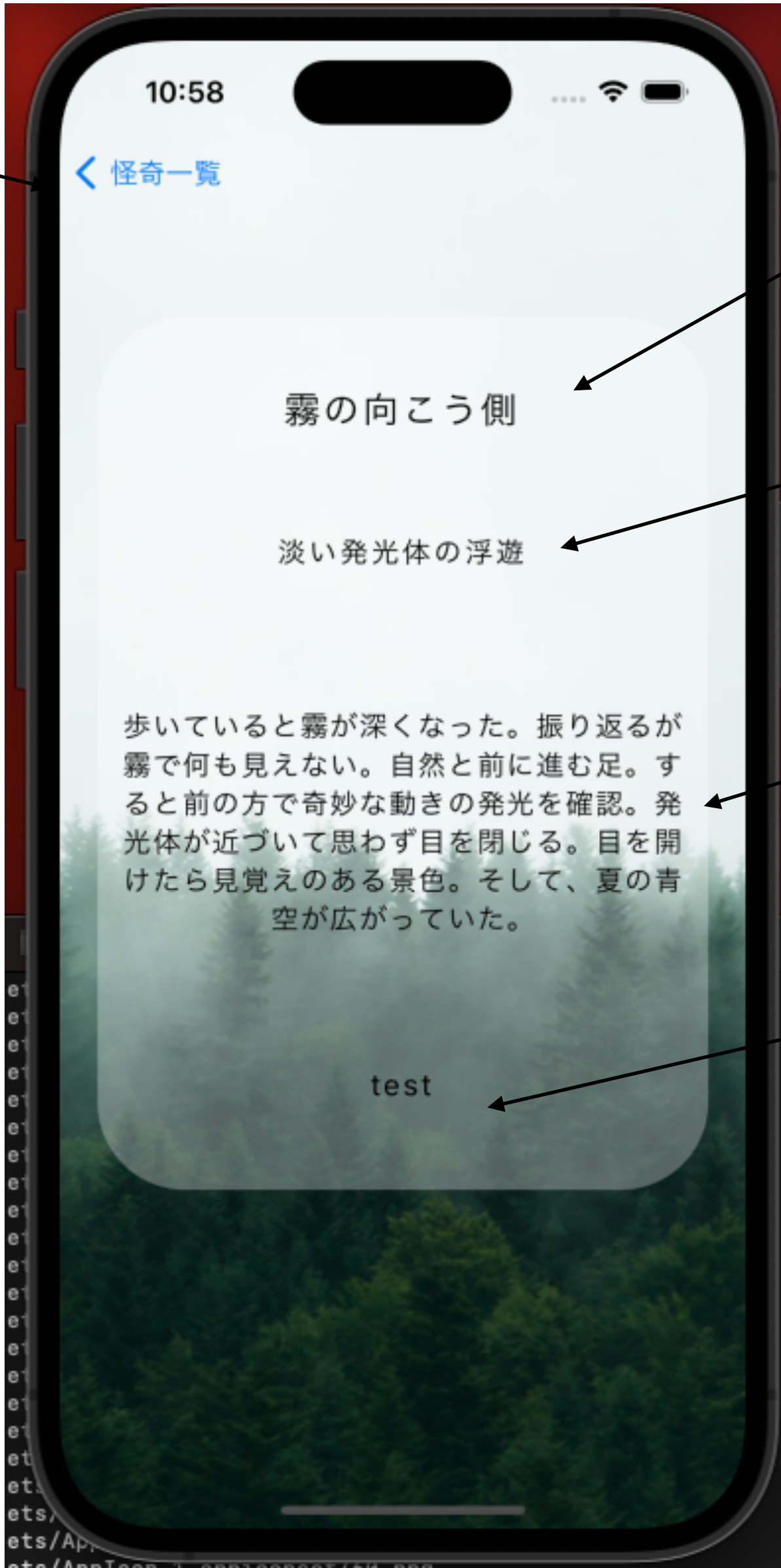
記事一覧



初期登録してる記事と新規投稿された記事が表示されます

アプリについて -詳細画面-

記事一覧に戻る



タイトル

現象

詳細

位置情報

アプリについて -新規画面-

記事一覧に戻る

投稿

タイトル記入

現象記入

詳細記入

写真選択

位置情報自動入力
※現状正常に取得できて
いません

The screenshot shows a mobile application interface for creating a new post. At the top, there's a status bar with the time 11:09 and signal indicators. Below that, a navigation bar contains a back arrow labeled '怪奇一覧', the title '遭遇した怪奇', and a '追加' (Add) button. The main content area is a form with a light green background. It has a 'タイトル' (Title) field, a '緯度: 37.785834, 経度: -122.406417' (Latitude: 37.785834, Longitude: -122.406417) field, a '現象' (Phenomenon) field, a '詳細' (Details) field, and a '写真を選択' (Select Photo) button. The background of the app is a painting of a tree in a field. Arrows point from labels to the corresponding UI elements: '記事一覧に戻る' points to the back arrow, '投稿' points to the '追加' button, 'タイトル記入' points to the title field, '現象記入' points to the phenomenon field, '詳細記入' points to the details field, '写真選択' points to the photo selection button, and '位置情報自動入力 ※現状正常に取得できていません' points to the location field.

コードについて

メインview:他のviewを読んだり、このviewから他のviewへ画面遷移します。

```
var body: some View {
    NavigationStack(path: $path) {
        ZStack {
            Image("背景6")
                .resizable()
                .ignoresSafeArea()
            List {
                ForEach(recordStore.records) { record in
                    HStack {
                        NavigationLink(destination: RecordsDetail(record: record)) {
                            record.image
                                .resizable()
                                .scaledToFill()
                                .clipShape(Circle())
                                .frame(width: 50, height: 50)
                        }
                        Text(record.title)
                            .padding(.leading, 20)
                    }
                }
            }
        }
        listRowBackground(Color.secondary)
    }
}
```

背景画像

記事一覧

ZStackで画像とリストをネストして重ね合わせている。

1週間で学んだこと

位置情報の取得(実装途中)

- ・ SwiftUIとStoryboardでは作られるファイルが異なる為、アプローチが変わる。

NavigationViewとNavigationStack

- ・ iOS 16.1以降NavigationViewは非推奨。
- ・ どちらも画面遷移のナビゲーションスタックを管理。
- ・ 画面遷移処理させる実装の煩雑さが軽減。
- ・ 検証しきれていない部分もあるが、viewの重ね合わせで透過さて背面のviewを可視化させることが難しかった。
viewを分けて土台となるviewでスタックを指定することが良いと思った。(Reactのコンポーネント的思考)