* **DESIGN PATTERNS IMPLEMENTED IN THE APP:**

1. **Separation of concerns principle**
2. **Open – close principle**
3. **Null object principle**
4. **Dependency Inversion principle**
5. **Adapter pattern**

* **APP STRUCTURE:**

**As requested, I followed MVVM with Adapter Pattern to isolate modules.**

* **Things that I focused on:**

1. **Design Patterns**
2. **MVVM structure**
3. **Designing a reusable Network**
4. **Other coding features (Mocked features) to help show my knowledge and ability to make a structured cleaned code.**

**I have added a screen for university flow to demonstrate how code is dynamic using Feature Flagging, and reusable with one UITableViewController.**

**To activate the flow, please modify** executeNewFlow attribute **in loadAppConfiguration method in AppConfiguration.swift file.**

**I have not focused on UI/UX, so I did simple views that reflects API data fetched out of endpoints.**

**Note:**

**Schools that are fetched from first API don’t have SAT data. So, I decided to still look up a school detail of second API using school\_name as a parameter. If a school has a SAT data, then I show it. Otherwise, I show N/A as Not Available.**