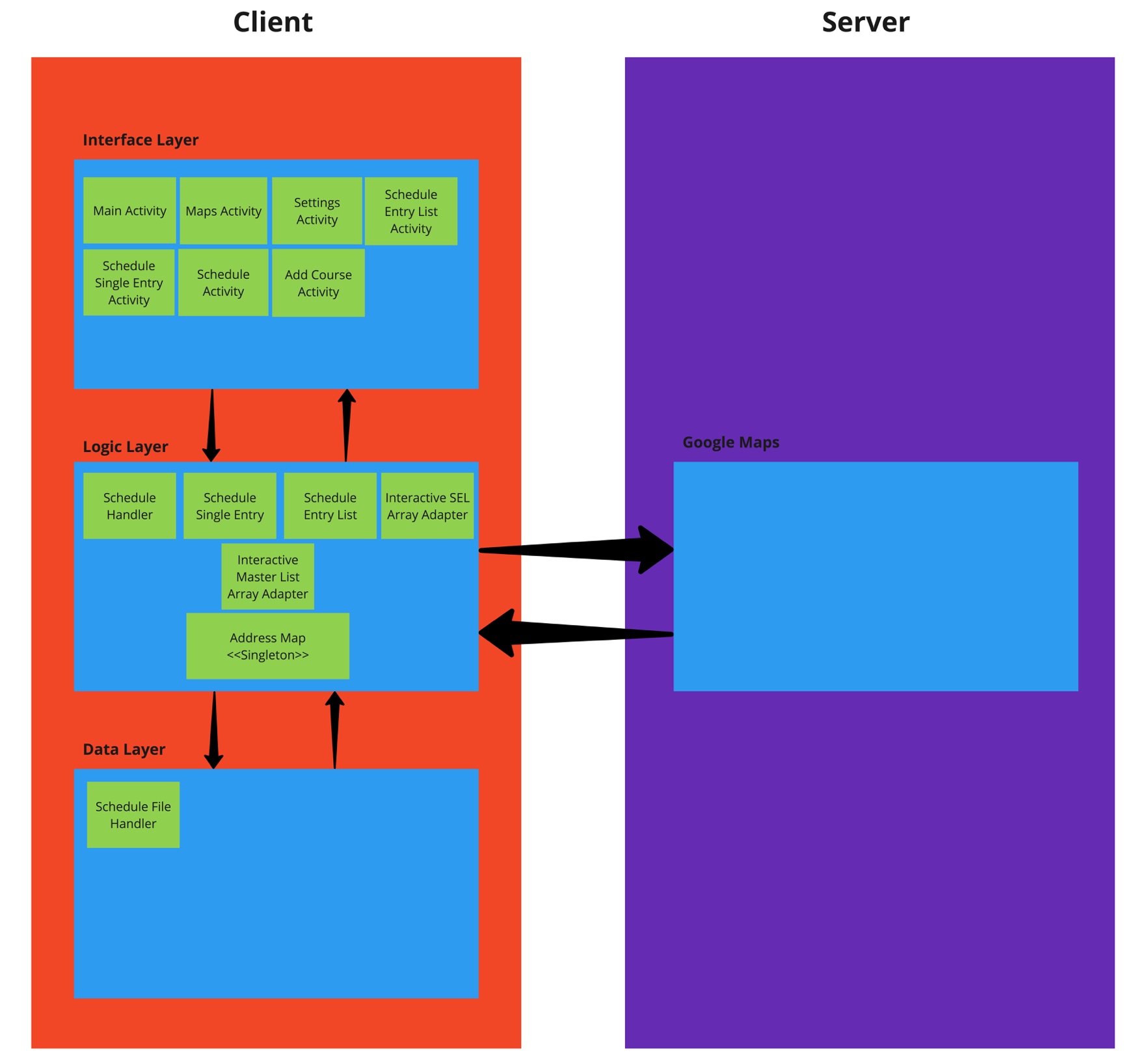
Software Architecture & Design Pattern Document

The software architecture for the *RaiderNAV* app is based on a simple hybrid of the layered and client-server architectural patterns as detailed below.

The *RaiderNAV* software located on a target Android device is itself comprised of three layers. The first is an interface layer, composed of Android activities, which facilitates communication between the user and the underlying system. Next is a logic layer, implemented in the form of Java classes, which interprets user input and processes it according to the app’s design. Finally, at the lowest level is a data layer, similarly formed by Java classes, that is responsible for manipulation and storage of processed data. True to the layered pattern, the interface and data layers interact directly only with the logic layer. The logic layer, in the middle of the three-tiered design, interacts with both other layers.

Altogether, these three software layers on the device constitute the client component of the encompassing client-server model. The server module of this model is represented in the form of the backend functionality of the Google Maps API, which is necessary for navigation and location features. Communication between server and client components is limited to the logic layer of the client-side implementation. The functionality of *RaiderNAV* is dependent on both the user interaction and information processing done on the client-side as well as the data retrieved from the Google Maps server.

Figure 1: Software Architecture