Our team utilized the top-down integration for implementing MapKU. Top-down integration is a code integration strategy that initializes with one main artifact and proceeds down to lower level branches for implementation and integration. The head artifact of our top-down integration would be the initialization of the map and its direction services. Team members would use this head artifact to add additional features to the map. The stubs in the lower level are implementations of additional features such as the sidebar, markers, info panel and other user interface related elements. After a team member has implemented a new feature on a separate branch, we then merge it into one. When working on these branches, our sessions were usually split into groups of 2-3 and we did not have a lot of merges since most of the code being pushed was worked together on Visual Studio Code’s LiveShare. As the stubs traverse down to the lower levels, the artifacts become more operational, such as adding current location to the map or adding a building. This makes it easy to add and remove any functional features that is map related.

We chose this top-down integration strategy because it’s complementary to the structure of our codebase. This is because we were using the map and its direction service as the foundation and then additional implementations become new branches. We were also able to have the core functionality of our project at the early stage of development. This integration structure helped us to see flaws early in development as well as having a centralized system for interchanging features.