Justification

A one-tier architecture is a stand-alone application run only on the client tier and stored locally or on a shared drive. The problem with a one-tier architecture is access cannot be distributed, as it’s not supported when all layers of the architecture are in the client tier. The cost is also high to run the central mainframe and the code requires a large amount of maintenance to keep the application running smoothly. The two-tier web-based architecture is different to the three-tier architecture because it offers a direct communication between the client server and database. The client application writes the code and saves it to the database server communicating to the server via request. This can create a data integrity issue because when you have multiple users the server cannot respond and process multiple requests at the same time. The developers need to be disciplined or the logic for the business and database can be confused, which creates duplicates. The three-tier architecture implements the middle tier server that controls business and database logic to eliminate the problems that occur with two-tier data integrity issues. This means the three-tier architecture has an added level of security so clients do not have direct access to the database. If Additional tier layers were to be added to the three-tier architecture it would be unnecessary as we are only focusing a single software application versus integrating with other 3rd party applications. Therefore, the correct architecture pattern to use for Team Twelve’s smart city project is the three-tier web-based architecture.