***Justification for choosing the three-layer architecture pattern***

As a team, we chose the three-layer architecture software design pattern. A three-tier architecture is can be described as a “client-server architecture in which the functional process logic, data access, computer data storage and user interface are developed and maintained as independent modules on separate platforms.” (Techopedia, 2017). Due to the fact that the independent modules are developed on separate platforms, there can be multiple sets of developers simultaneously developing each component of the system (Hagos, D. 2008).

This architecture pattern is advantageous over one-layer and two-layer architectures as the data can be managed independently from the physical stored data and the data is overall more secure than 1 and 2-layer architectures because clients can’t directly access the database business logic (Hagos, D. 2008). Having this security is prevalent for the smart city project where there are user accounts being created with valuable information (phone numbers, addresses).

Furthermore, changes can be made on the presentation level without affecting the business or data access layer in any way (Hagos, D. 2008).

Another benefit of using this architecture pattern is that when one tier fails, there is no data loss because you are always secure by accessing the other tier. The last advantage of using the three-tier architecture is that it accelerates the process of migration to a new graphical environment.

Due to all the benefits of using a three-layered architecture, we will be using this pattern to develop our project.

***Reference List***

Hagos, D. (2008) *Advantages and disadvantages of 3-Tier Architecture in Web Development.* Available at: http://des-megan.blogspot.com.au/2008/11/advantages-and-disadvantages-of-3-tier.html [Accessed 1/09/2017].

Techopedia (2017). *Three-Tier Architecture.* Techopedia. Available at: https://www.techopedia.com/definition/24649/three-tier-architecture [Accessed 1/09/2017].