* **Architectural Design Specifications**
  + The following document specifies the architectural design planning considerations for FlyAway, an application developed by Grey Matter.
* **Justification for a Layered Architectural Scheme**
  + FlyAway will primarily implement a layered architectural paradigm. The software will be divided into a user layer and three functional layers, with data and information unable to pass through more than one layer at a time without moving through intermediate layers first.
    - The top layer will consist of a “Users” which will contain the different kinds of users that can access the application.
    - The first functional layer will consist of a “Presentation Layer” which will contain the user interface, as well as graphical elements which display public-facing account information.
    - The second functional layer will consist of an “Authentication Layer” which will contain the user authentication, as well as a secure payment authentication which will make sure all data entered into our database is stored in the “Account Information Database”.
    - The thirdd functional layer will consist of a “Logical Layer” which will host important functions that make the application usable. These include the methods that allow accounts to communicate with each other via text, schedule & book reservations, and coordinate payments.
    - Lastly, the lowest level will be a “Data Layer” which will primarily host two components: an account information database with information for each user account, and a system that handles monetary transactions through systems such as PayPal. The latter component would usually exist in the Logical Layer and not the Data Layer, but for the purpose of security it is placed in the Data Layer so it is more difficult for outside users to tamper with it.
  + Layered architecture is the best architecture design for FlyAway because it prevents users from accessing the databases directly without negotiating through the main server first. This should prevent users from accessing sensitive account information without obtaining the proper authorization first, either accidentally or as a product of malfeasance. This is important because user personal information security is a high priority for FlyAway.
  + FlyAway will not directly implement any features contained in the “client server” architecture paradigm. Because FlyAway is a web application, it is not necessary to create distinct user clients installed on each individual user’s mobile device. However, because each individual user uses the internet to communicate with a central server, there is a “hub and spoke” element to the architecture of FlyAway that is somewhat similar to applications with a clear client-server architecture framework, therefore client-server architecture is a secondary architectural design used by FlyAway.
* **Summary of Architectural Features**
  + **Account Authentication**
    - All account information is stored in a database that shall henceforth be referred to as the “Account Information Database''. This database will be constructed using Firestorm and will only interface directly with the logical layer of the application. Each account will have a unique username and password, and all other account information can only be obtained when the server submits the correct authentication token.
    - Users that are not already logged in are referred to as “public facing users” - they are presented with the FlyAway website for informational purposes, but must log in or create a new account to access the application’s features.
      * When generating a new account, the user will be faced with a web page which will allow them to submit all the necessary information for creating a new account, after which their account will be added to the database and they will be given the option to log in.
      * When logging in to an existing account, the user will submit their username and password through the user interface to the Logical Layer. This layer will then search for the appropriate username in the database, generate and match the authentication token with the associated account, and retrieve all of the necessary account information. This information will then be visible to the user on the presentation layer, allowing them to use the application’s services. The user will also be permitted to edit the information related to their account, thus overwriting information in the database.
  + **Browsing**
    - Once logged in, the passenger user’s presentation layer will send a request to the data layer (via the logical layer) for the public facing fields attached to all nearby pilot accounts. This information will be presented to the passenger in the form of a long list, which can be narrowed down through a variety of search parameters.
    - The passenger user will also have access to the application’s “map” feature. This will consist of a map of New England (or America, if the app is scaled nationwide) and will show every available pilot account next to their home airport. To do this, the map will make a request through the logical layer to access each pilot account’s home airport, generate the location of that airport, and render this location directly to the map.
  + **Communications Between Accounts**
    - The logical layer will also be able to facilitate the appropriate protocols with regard to communications between passenger and pilot accounts. Once a passenger finds a pilot that they would like to get in touch with, they will have the option of sending a message request, which the pilot must then accept.
    - Once both accounts have authorized direct communications, the logical layer will be able to facilitate the secure transfer of text messages between the two accounts.
    - The logical layer will then allow the two accounts which have the ability to communicate with each other to establish flight reservations and store these reservations in the personal planner associated with each account.
    - After the flight has been completed, passengers will have the ability to indicate whether or not they recommend the pilot to other passengers using FlyAway’s “like/dislike” review system.
    - Lastly, the logical layer will be responsible for facilitating the secure transfer of payments between pilot and passenger accounts. To do this, it will implement a custom PayPal button. When a payment request is made, the application will utilize the appropriate channels of communication with PayPal to ensure that the payment arrives at the pilot account in an amount that is equal both to the amount requested by the pilot and the amount sent by the passenger. Any calculations regarding sensitive monetary information will take place in the Data Layer for security purposes, even though this process does not strictly involve the storage of data in the long term.
  + **Other Features**
    - Features that are specific to the user accounts, such as location, contact information, pilot credentials, aircraft type, and the reservations planner will be wholly contained within either a user’s presentation layer or the account database and thus are not explicitly detailed in this description of software architecture.

**UML Diagram**

**Presentation Layer**

**Users**

**Data Layer**

**Logical Layer**

**Authentication Layer**