**Design Goals:**

1. User Friendliness: The application should be easy to navigate, even for non-tech-savvy users. It should have a clean, clear, intuitive user interface.

2. Scalability: The system should be able to accommodate an increasing number of users and data over time without losing performance.

3. Security: User data, especially their private bucket list, should be stored securely and with respect to privacy laws like GDPR.

4. Personalization: Profile personalization allows users to modify their profile pictures and customize information.

5. Accessibility: The system should be designed as a web application to be easily accessible from various devices with different screen sizes.

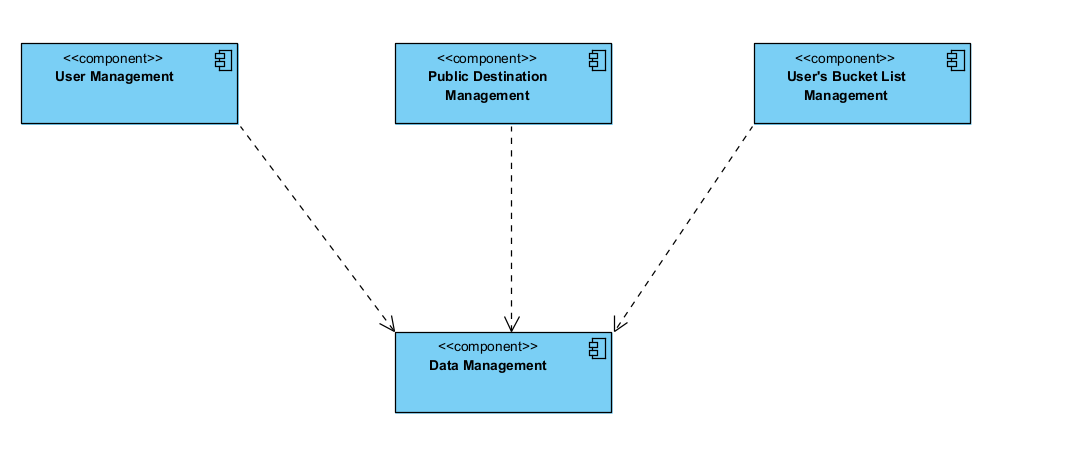
6. Reliability: The system should be reliable, ensuring data is not lost and providing uptime as close to 100% as possible.

7. Performance: The system should be responsive and handle requests in a timely manner.

**Subsystem Decomposition:**

1. User Management Subsystem: This subsystem is responsible for user registration, login, profile management, and account cancellation.

2. Public Destination Management Subsystem: This is an admin-controlled subsystem for managing the public list of destinations.

3. User's Bucket List Management Subsystem: This subsystem allows users to manage their private vacation destination lists.

**System Architecture (UML Component Diagram):**

The chosen type of layering is the Transparent Layering (Open), where each layer can interact with all the layers beneath it. This will allow for more flexibility in the interactions between layers.

The chosen architectural style is the Model-View-Controller (MVC) due to the following reasons:

1. Separation of Concerns: The MVC architecture separates data access, business logic, and user interface concerns, making it easier to manage and maintain.

2. Scalability: This architecture allows the system to be scaled more easily. For example, if the system needs to handle more requests, additional controllers can be added.

3. Flexibility: This architecture allows changes in one component to have minimal impact on the others. For instance, changes in the View component will not affect the Model component.

**Interdependent Subsystems:**

1. User Management Subsystem

2. Public Destination Management Subsystem

3. User's Bucket List Management Subsystem

The User Management Subsystem is responsible for the overall user management, and it interacts with both Public Destination Management Subsystem and User's Bucket List Management Subsystem. Users add destinations from the Public Destination Management Subsystem to their bucket list managed by User's Bucket List Management Subsystem.

**Subsystem Interface (Object Design):**

1. **User Management Subsystem**

* Attributes: UserID, UserName, Password, Email, UserProfile
* Operations: RegisterUser (UserDetails), LoginUser (UserCredentials), UpdateProfile (UserProfile), CancelAccount (UserID)

2. **Public Destination Management Subsystem**

* Attributes: DestinationID, DestinationTitle, Image, Description, Geolocation
* Operations: AddDestination (DestinationDetails), UpdateDestination (DestinationDetails), DeleteDestination (DestinationID), GetDestination (DestinationID)

3. **User's Bucket List Management Subsystem**

* Attributes: UserID, BucketList, DestinationID
* Operations: AddToBucketList (UserID, DestinationID), RemoveFromBucketList (UserID, DestinationID), ViewBucketList (UserID), UpdateBucketList (UserID, BucketList)

