**Architecture Analysis**

**1. Overview of the Application Architecture**

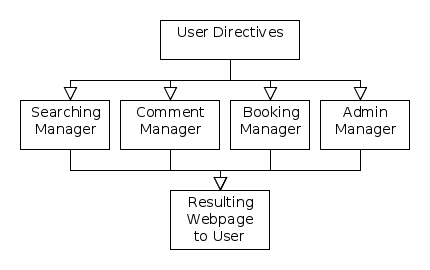
Here we use a simplified MVC(Model-View-Controller) architecture pattern as the top-level design.

**1.1.** A controller takes directives from users and send corresponding commands to the model.

**1.2.** A model takes input from the controller then change it's state if neccesary, and notify the view to generate its output.

**1.3.** A view gets information from the model and produce the corresponding output to users.

Then let's have a nearer look at our application. Incoming user requests are handled by the corresponding mamagers, as the following figure shows.



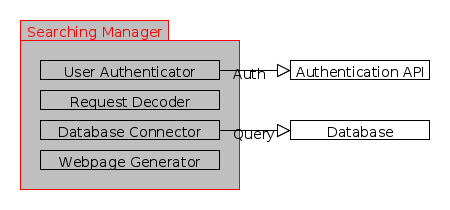
**2. Subroutines in A Snapshot**

**2.1. The Searching Manager**

The searching manager handles searching requests, including hotel/flight search, comment search, and booking record search.

**2.1.1 Package Component**

There're four components come with this manager, which are user authenticator, request decoder, database connector, and webpage generator respectively.



**2.1.2. Architecture Context Diagram**

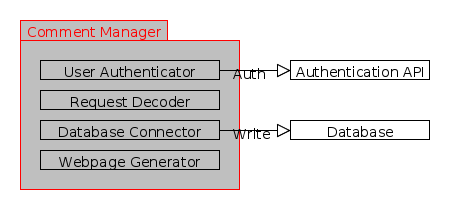
****

**2.2. The Comment Manager**

The comment manager handles commenting requests.

**2.2.1 Package Component**

There're four components come with this manager, which are user authenticator, request decoder, database connector, and webpage generator respectively.



**2.2.2. Architecture Context Diagram**

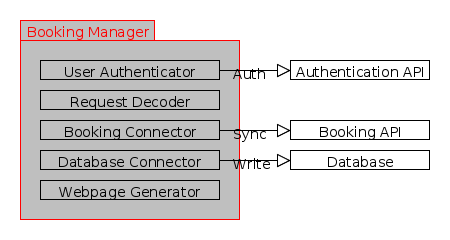


**2.3. The Booking Manager**

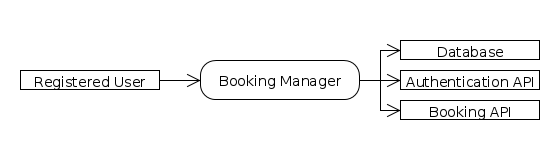
The booking manager handles booking requests.

**2.3.1 Package Component**

There're five components come with this manager, which are user authenticator, request decoder, booking connector, database connector, and webpage generator respectively.



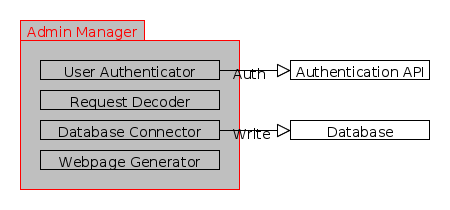
**2.3.2. Architecture Context Diagram**



**2.4. The Admin Manager**

The admin manager handles admin requests, including data insertion, data removal, and data modification.

**2.4.1 Package Component**

There're four components come with this manager, which are user authenticator, request decoder, database connector, and webpage generator respectively.

**2.4.2. Architecture Context Diagram**



**3. Data Hierarchy**

Carefully we designed the data hierarchy to reduce redundancy, which would apply to all the components of our architecture.

See the figure below.

