Find Friends on Flights

Millions of people travel on airplanes every year. However, if you've ever been on a flight, you'll notice that it is usually a very solitary affair, with no conversation between you and the people sitting next to you for several hours. But the truth is, there are many interesting people with many interesting life stories on board any given flight. We want to break the awkwardness barrier between airline passengers, and encourage conversation and friendship.

Find Friends on Flights is a web-based application that connects like-minded travelers, whether they share a common interest, are going to the same destination, or are looking to fly on the same day. Users of Find Friends on Flights can find other flyers by flight, date of travel, name, or by common interests and hobbies. Two flyers that are interested in talking to each other can then use an external service to book adjacent seats on the flight. Our target audience would thus be airline passengers looking for conversation in an otherwise awkward environment.

Currently, there are two competing services that offer similar functionality as our proposal. The first is the Airsocial.co app, which is available on web and mobile. This app allows users to add their flights after they buy their tickets, and see who will be flying with them. However, we are targeting an audience who want to find friends *before* they buy their plane ticket. Whereas Airsocial is more of a passive system, Find Friends on Flights is more of an active system. The second competing service is Planely, which offers very similar functionality to Airsocial. However, their web app seems to be broken, and once again we want to target those who want to find flight friends before they buy their tickets.

To build a system like this, we would utilize front-end frameworks like AngularJS to manage the data and display it to the user. On the back end, we would use Java and associated frameworks (e.g. Spring), in order to talk to the relevant airliners' API's and store users' information. The high-level architecture involved in this application would be MVC, where the model consists of user data, flight data, and user preferences; and where the controller and view are built in HTML, CSS, JavaScript and handle user interactions. Additionally, the model will retrieve data from API's such as Google Flights, in order to expose information such as upcoming flights to users.

The primary challenge to any social networking application is receiving enough data and user registrations to make interesting interactions. To combat this, a large user base can be simulated by a computer in order to breathe life into the application. Another secondary challenge could be integrating with the airline industries' tech stack, especially if their API's are out of date or not well-documented. Given the time constraints in this project, it may be feasible to have one or two group members specialize in learning such an API. If this is not possible, then it is always possible to pursue an alternative design where users manually enter information about their flights.