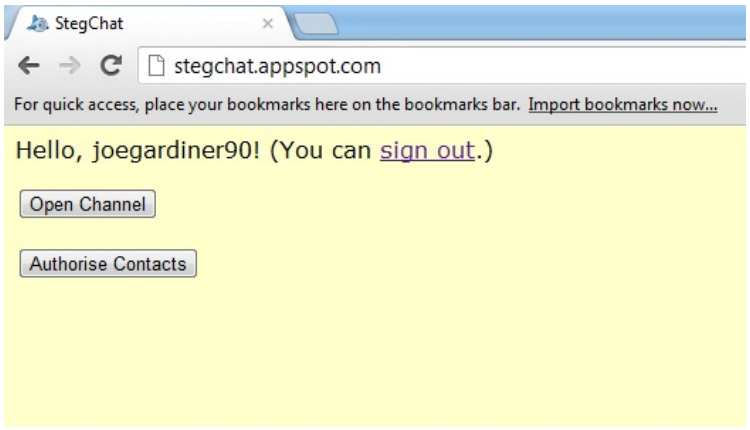

Figure 4.1: Home Screen. User signed in using Google User Service. User clicks "Open Channel" to go to chat screen

4.1.1 Resources

The application was developed using the Eclipse IDE, using the Google App Engine plugin. The website was developed using Chrome and Firefox.

4.2 Dictionary Storage

One of the biggest challenges with developing the system in a technical sense was the issue of how to store and access the WordNet library. For a local Java application, this is very simple. You install the WordNet database on your computer, and then there are a number of libraries that can be used to access the database by pointing them to the database files. The database files are simply large text files, ranging from a few kilobytes in size to fourteen megabytes. There are two primary types, index files, which for each word list the senses which that word belongs to, and data files, which contain the senses. There is one of each of these files for each of the four word types. The data files are accessed via offset values, which are contained in the file, and the index file simple lists these offsets.
When it came to transporting this model of a library and database files onto the App Engine, a number of issues arose. Firstly, the files had to be stored on the App Engine. The App Engine provides a file store called the BlobStore which is primarily used for hosting content such as video or images to be delivered to users of the application. There is no upper limit to the file size, but there is a maximum fetch size of one megabyte (anything larger has to be fetched sequentially). With this being the only viable way to store the files themselves on the App Engine, this also caused a problem for the Java libraries used to access the database. Because of this, it was decided that, as the files were relatively easy to parse and process manually, that libraries would not be used. This was tested with the application,