

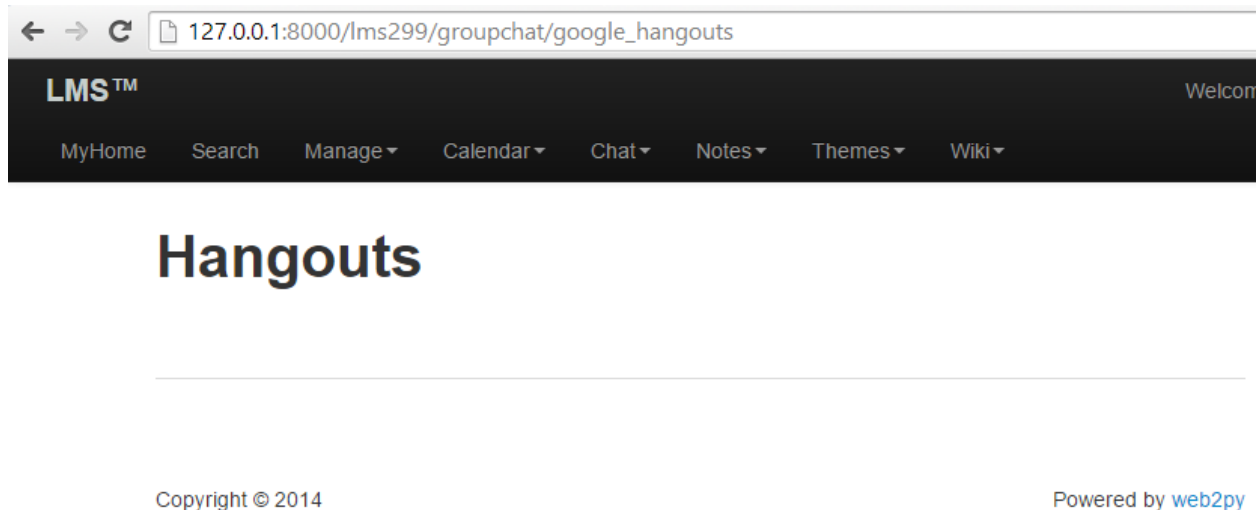
Chat Hangout Documentation

1- Hangout application user guide

The hangout App is a great tool for students to communicate with their peers and discuss course related topics outside class in a more casual environment.

The Google Hangouts page is only accessible for authenticated users. In addition, the student needs to be enrolled in at least one course in order to use the app.

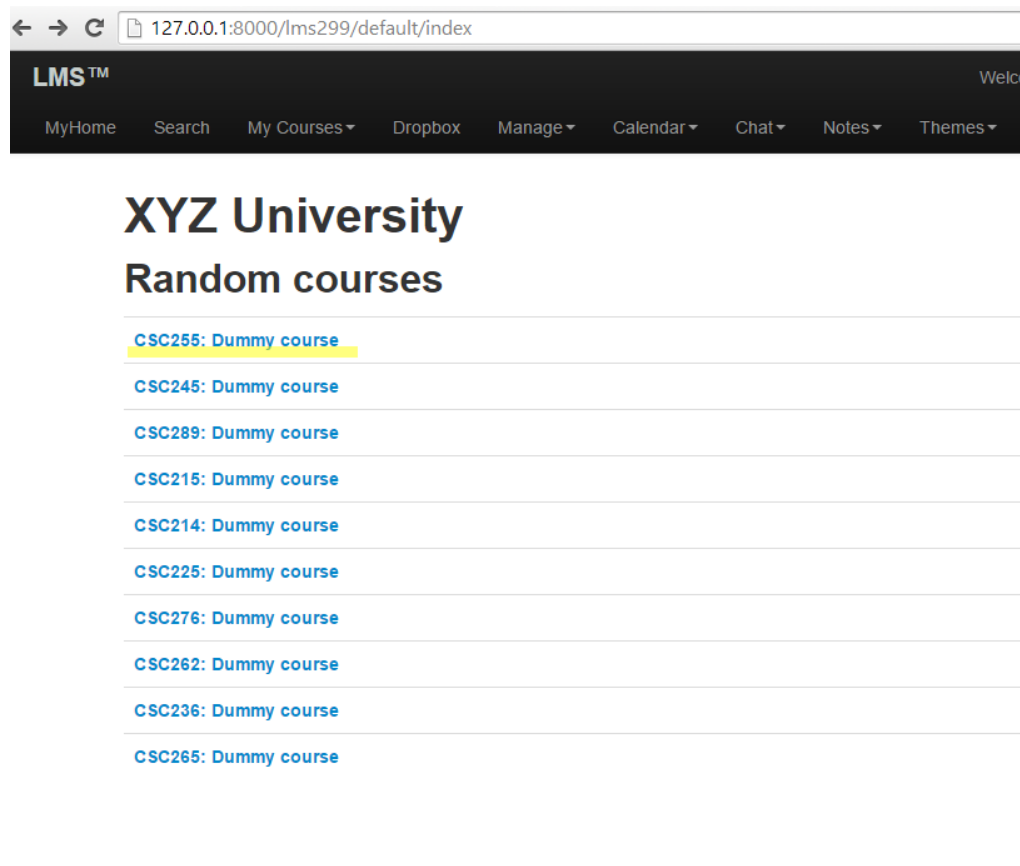
Before signing up for a class the user Hangouts pages is empty



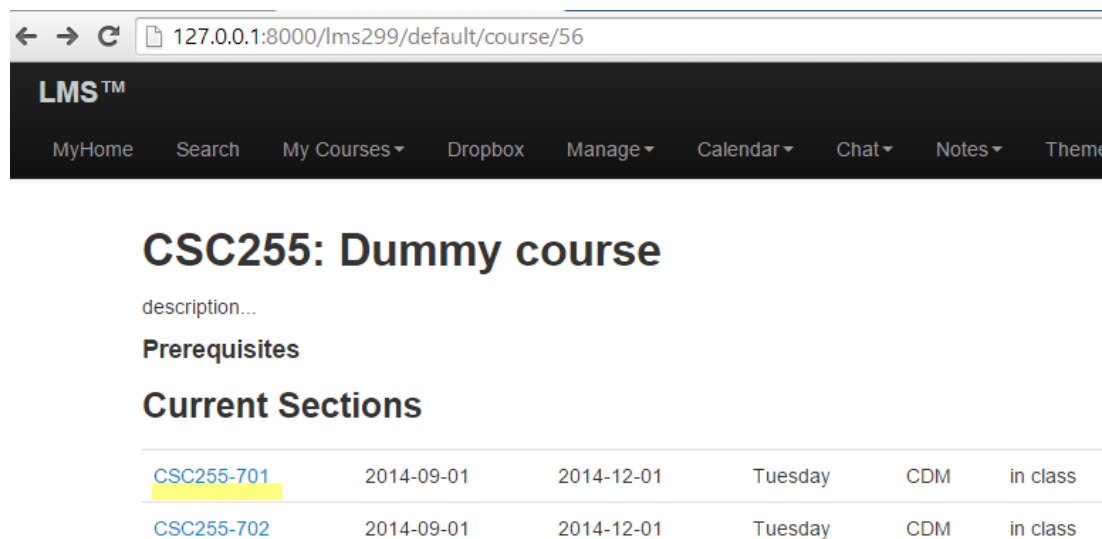
First, the user needs to enroll in a class.

If you wish to enroll in a particular course, go to the */default/index* page, and click one of the available courses.

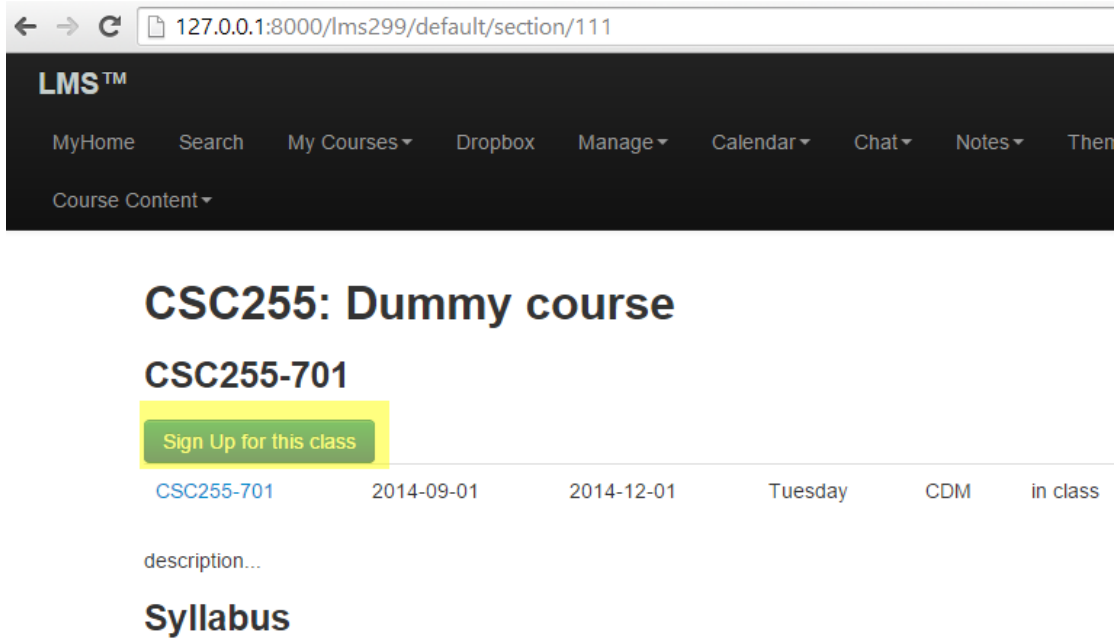
In this example I will select [CSC255](#)



Then click the lick highlighted in yellow



Then click “Sign Up for this class”



The screenshot shows a web browser window with the address bar displaying `127.0.0.1:8000/lms299/default/section/111`. The page header is dark with the LMS logo and navigation links: MyHome, Search, My Courses, Dropbox, Manage, Calendar, Chat, Notes, and Theme. A 'Course Content' dropdown is also visible. The main content area features the course title 'CSC255: Dummy course' and 'CSC255-701'. A green button labeled 'Sign Up for this class' is highlighted with a yellow box. Below this, a table lists course details:

CSC255-701	2014-09-01	2014-12-01	Tuesday	CDM	in class
------------	------------	------------	---------	-----	----------

Below the table is a 'description...' link and a 'Syllabus' link.

Now that the student is successfully enrolled in the [CSC255](#) course. And the `/groupchat/google_hangouts` page will now display the hangouts information related to this specific course.

← → ↻ 127.0.0.1:8000/lms299/groupchat/google_hangouts

LMS™ Welco

MyHome Search My Courses ▾ Dropbox Manage ▾ Calendar ▾ Chat ▾ Notes ▾ Themes ▾

Hangouts

Sends you back to the particular default/section/ for this course

Press this button to start a new hangout

CSC255-701

Start a Hangout

Join Recent Hangouts

Started @ 19:16:11

Says that a hangout was launched by a classmate in the last 5 minutes, at 19:16:11 more precisely and that you are able to join the hangout by clicking the hyperlink

When the “Start a Hangout” button is pressed, the user is directed to this page.

Share the permanent link. Bookmark and come back anytime.

https://plus.google.com/hangouts/_/g5b42yztafxcpjlhd5wgljzy3aa

Send invite as [spinosaurus54321@gmail.com](#)

+ Add names, circles, or email addresses

☐ Require guests to be 18+ to join this video call

☐ Quiet invitation [Learn more](#)

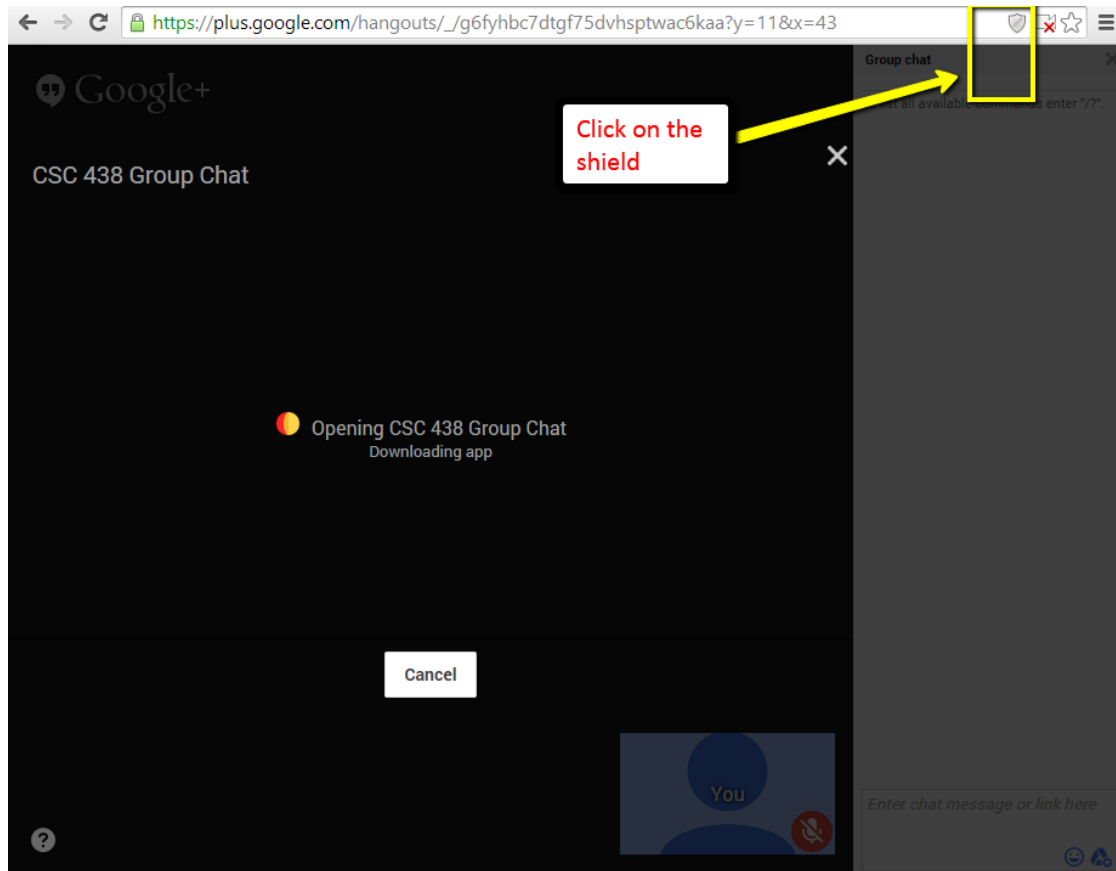
[+ Add telephone](#)

Invite

Close

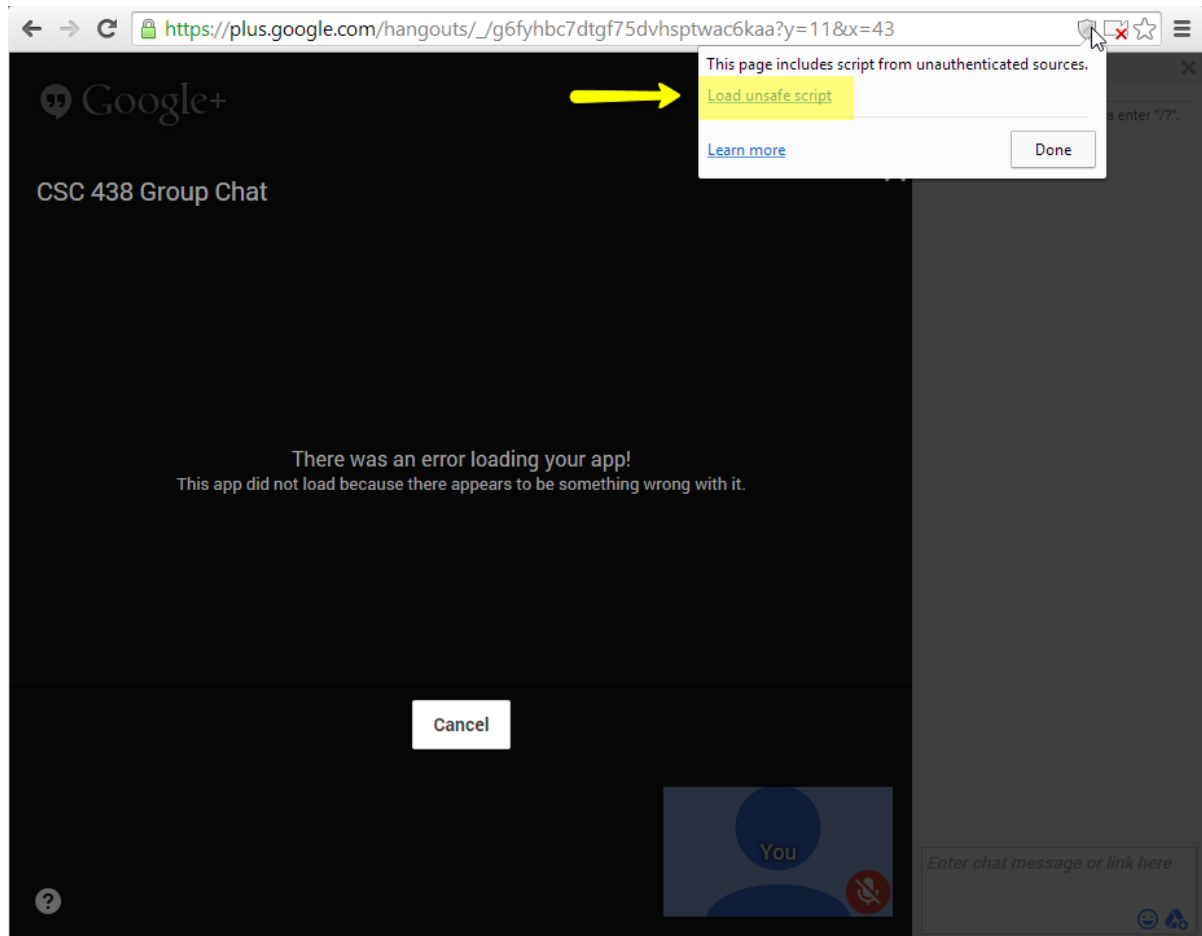


Click the “**Close**” button.

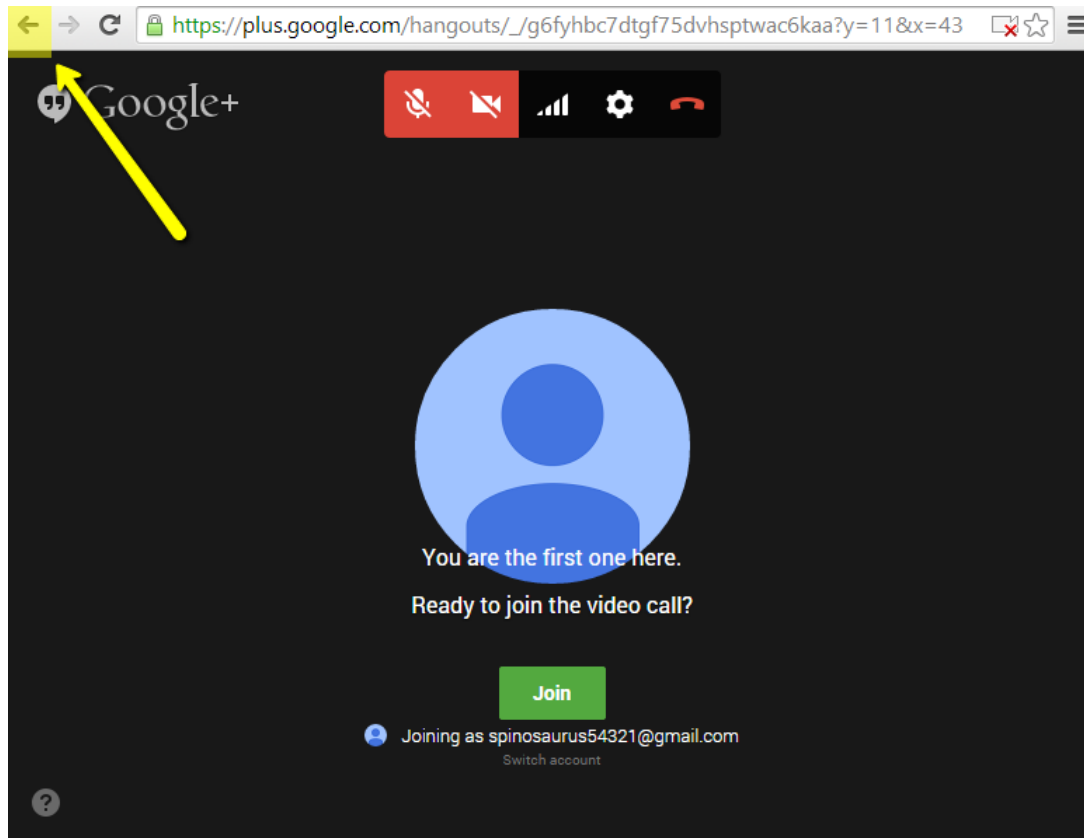


Click on the shield at the top left of the window.

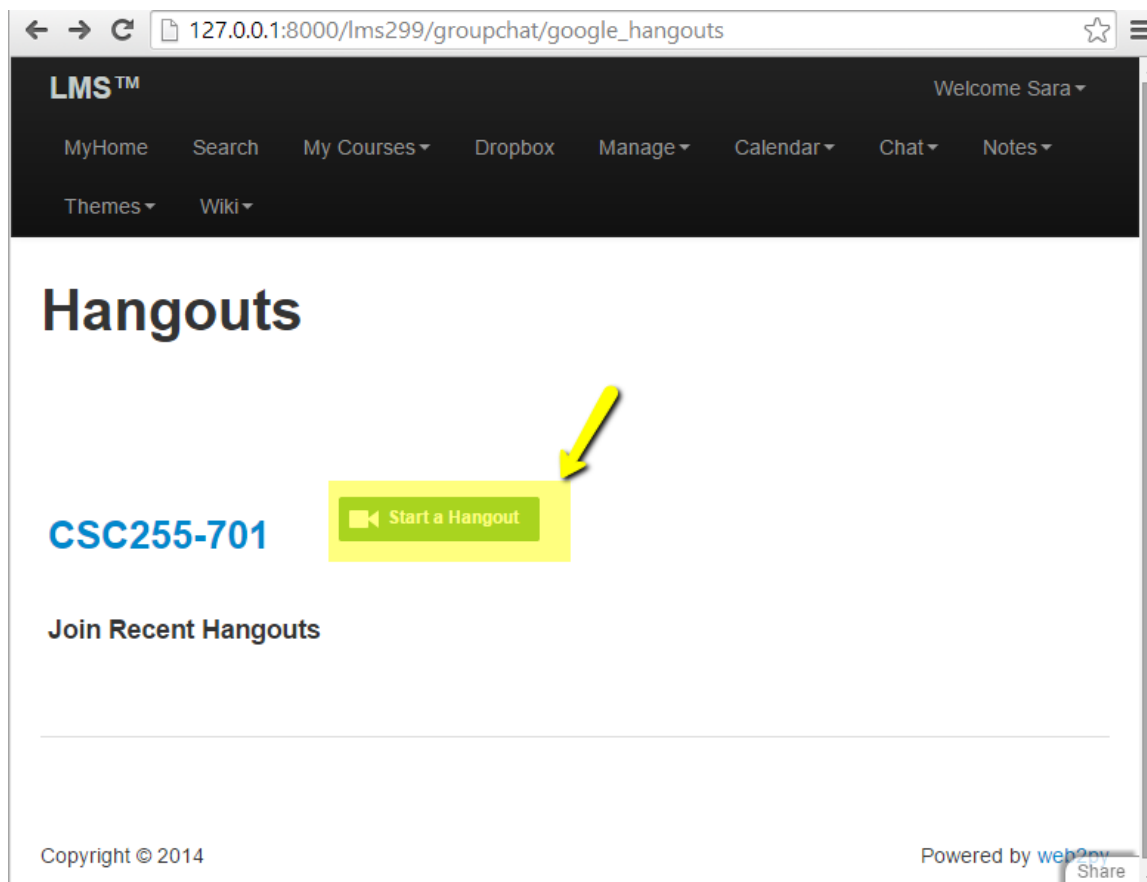
Note: This is needed currently because our web2py servicing is not secured and all of Google+ Hangouts runs on https://. This is only a workaround, the ideal solution would be to secure the web2py server.



Then click **“Load unsafe script”**.



The script from the google project has now been loaded. Click the back arrow to return to the previous page.



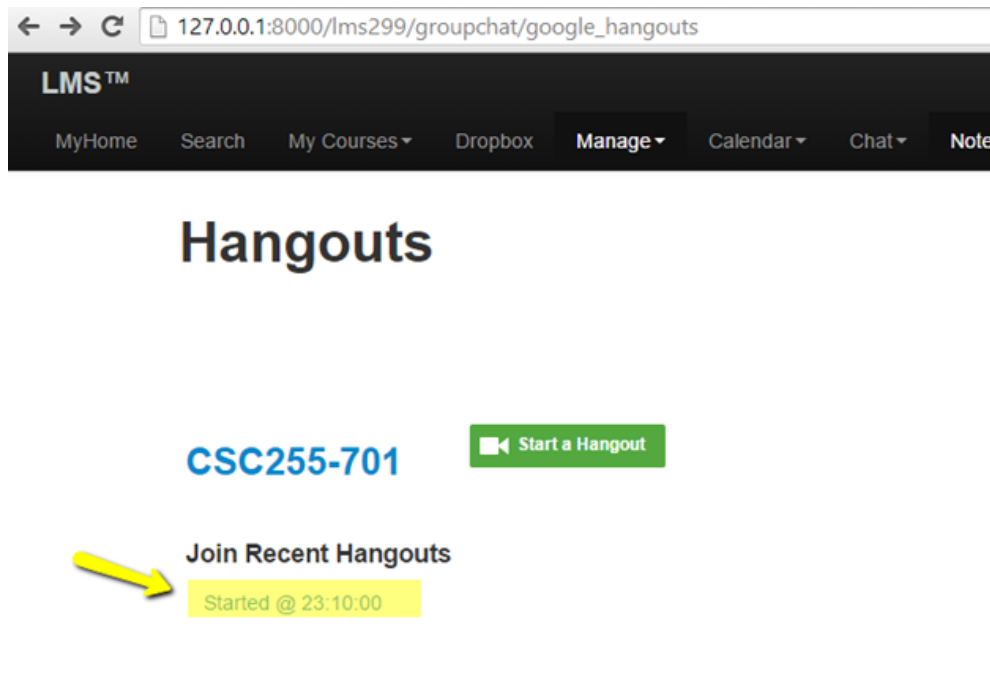
Click **“Start a Hangout”** once again on this page.

Since the script has been loaded on the page, the app will now start with the correct settings.

On the next page enter a person you wish to invite and the hangout will start right away. The groupchat/google_hangouts notifies other people enrolled in the class that a new hangout associated with this course has been started and provides them with the hangout link in case they want to take part.

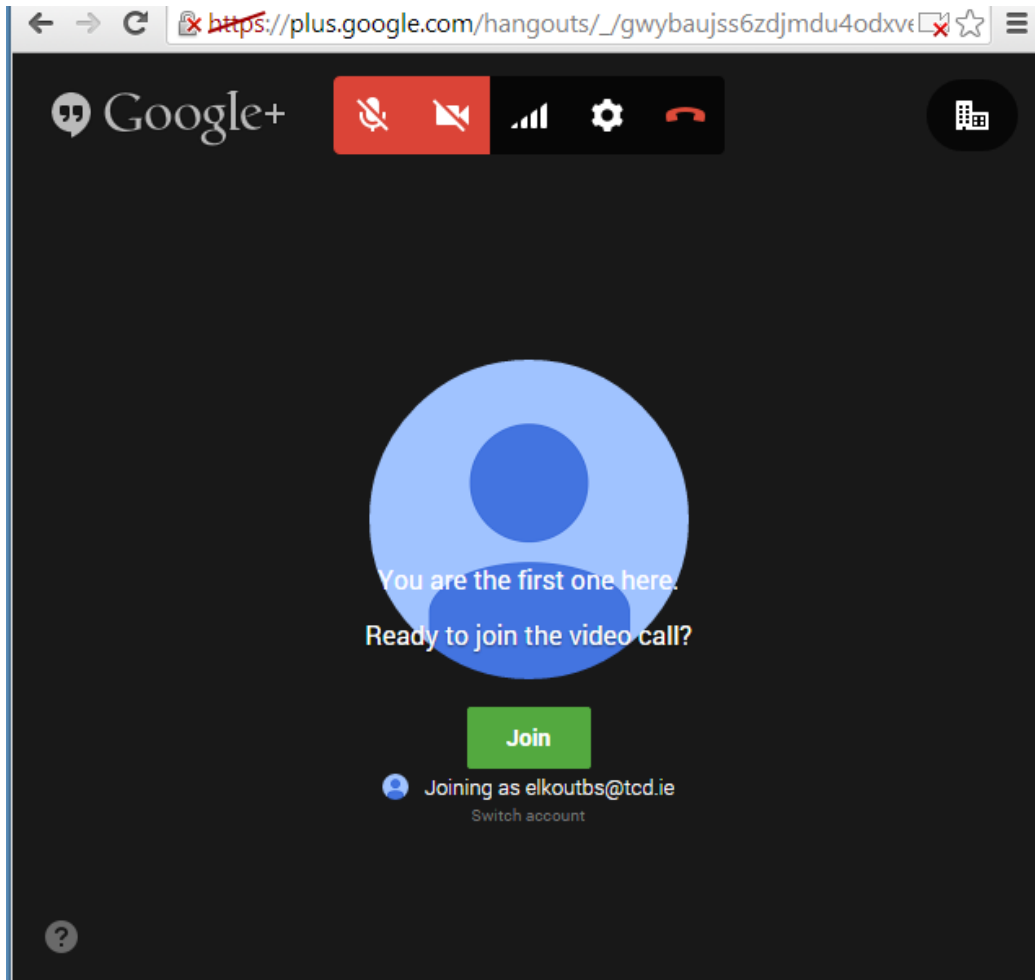
Previous Hangouts

It is also possible to join in a Google Hangout that is already in progress. To do this, click on a link from within **Join Recent Hangouts** section.



This will launch the previous Hangout.

This will direct the user to the ongoing hangout.



Hangout user chat settings

Clicking on either the microphone or webcam icon within the Google Hangout will trigger an AJAX call back to web2py to update the chat preferences for that user. The settings will then be stored in the database. Unfortunately, we were unable to then load these chat settings on future Hangouts as it appears that Google will disable these settings by default when a Hangout is loaded (probably for security reasons).

<< insert pic with hangout user chat settings >>

2- Hangout application developer guide

Our team created a Google Code project (named csc-438-group-chat) to host the application's XML file because it needs to be public and accessible for the users. XML file can be viewed at:

https://code.google.com/p/csc-438-group-chat/source/browse/group_chat_i_frame.xml

The hangout identifier, called gid, is passed to the app via an ajax onclick call when the user hits the "Start a Hangout" button.

The Hangout is launched in an Iframe so that parameters can be passed from our lms299 app to the Google Hangout (in our case, we pass the "gd" parameter. After the Google Hangout API is finished loading the following occurs: JavaScript callbacks are setup for when a user mutes their mic/webcam (note: settings are stored in database but Google overrides the settings when loading the Hangout app so settings are not persisted between Hangouts), the hangout url is fetched along with the "gd" parameter, and then stored into the db.group_chat_session database table via an jQuery AJAX call.

A student enrolled in a particular course will be able to view the hangouts started over the last 5 minutes because the urls for these records are retrieved from the table and displayed to the *groupchat/google_hangouts* page.

Files that the chat group created or modified by the Chat_Group

Controllers

- google_chat.js
 - o def group_chat():
 - o def google_hangouts():
Retrieves the courses the student is enrolled in and the hangout urls associated with each course.
 - o def hangouts_url_for_session():
 - o def update_user_settings_microphone():
 - o def update_user_settings_camera():
 - o def insert_new_hangout(course_section_id, user_id, hangout_url):
Inserts a new row for the newly created hangout
 - o def update_existing_hangout(session_id, url):
 - o def history():
 - o def add_user_to_chat(session_id):
-

Models

- menu_chat.py
 - o added a dropdown menu for two pages: "History" and "Google Hangouts"
- group_chat.py defines the following tables:
 - o group_chat_session
 - o group_chat_user_settings
 - o group_chat_user_session
 - o init_group_chat_session
 - o update_group_chat_session
 - o add_user_to_group_chat_session
 - o update_user_group_chat_settings
 - o update_user_setting_mic
 - o update_user_setting_cam
 - o get_user_group_chat_settings
 - o get_group_chat_sessions_for_user
 - o get_group_chat_sessions_for_user

Static/css

- group-chat.css
- chat.css
 - o contains styling for the google_hangouts page

Static/js

- group_chat.js
- google_hangouts.js

Static/css

- start-hangout.png
 - o google hangouts Image where the user can start the hangout

Static

- Chat Hangout Documentation.docx

Views

- google_hangouts.html
 - Main page where the user launches and joins hangouts
- google_chat.html
- history.html

Future improvements

- Uploading the email addresses of the students enrolled in the course into the app so that the users are able to retrieve the names of their classmates quickly when sending the invitation to the hangout.
- Secure the web2py server with https:// as this would allow us to house all of our code within web2py and not have to use google code for a secured URL page.
- Determine if there is another way to load the user's Hangout chat settings into a new Hangout without Google overriding them.