Group Members:

1. Chenyang Fang(cf2843)
2. Wan Xu(wx2227)
3. Guancheng Ren(gr2625)
4. Tian Niu(tn2415)

Part 1:

1.What does our project do?

Ans: The goal of our project is to create a website where students of Columbia are able to share their study notes with each other.

2. Who will be its users?

Ans: the target users would be the students or even professors of Columbia University. Our current authentication strategy is to allow whoever owns a LionMail affiliated with Columbia University to register/login our website.

3.Demo?

Ans: our demo will be able to show:

1. A user can create an account (only with LionMail ending with @columbia.edu)
2. A registered user can login with email and password
3. A logged-in user can search for notes by specifying department name and course number
4. A user can view previously-uploaded study notes specified by the searching parameters.
5. A user can upload his/her study notes by specifying department name and course number
6. A user can delete the notes uploaded by himself

4. What kind of data do you plan to store?

Ans:

1. User information: account and password.
2. Course material: lecturer notes and their metadata.
3. Courses info: the department name, course number, course name, and the term in which the course is offered

5. APIs and what for?

Ans: We use Django as our RESTful service to handle different requests from the users. We use REACT for front-end development.

Part 2:

1. As a current student of COMSxxxx, I want to view other students' notes for COMSxxxx so that I can better understand course content and prepare for homeworks and exams. I would also like to upload my notes to share my insights with other students so that they have access to more materials that potentially help with study.
    a. Conditions of satisfaction:
        i. The user can register and login using only LionMail affiliated with Columbia University.
        ii. The user can view other shared notes by specifying department name and course number.
        iii. The web app will display all results from the search.
        iv. The user can choose and click on one document to view explicitly.
        v. The user can upload his own class notes by indicating which department and course number the notes belong to.
        vi. After uploading, the files are visible to all users.
        vii. The user can delete the notes that were previously uploaded by himself.
2. As a professor of the course COMSxxxx, I want to provide my students with as much study materials as I could, so that my students can easily get access to more resources relevant to the content of this course.
    a. Conditions of satisfaction:
        i. The user can register and login using only LionMail affiliated with Columbia University
        ii. The user can upload files by indicating which department and course number the files are relevant to.
        iii. After uploading, the files are visible to all users.
        iv. The user can view other shared notes by specifying department name and course number.
        v. The web app will display all results from the search.
        vi. The user can choose and click on one document to view explicitly.
        vii. The user can delete notes uploaded by himself.
        viii. The notes uploaded by the professors have a special label indicating that they are from the instructors.
3. As a former student of COMSxxxx, I want to review the notes from my previous classes, so that I can review the course materials and prepare for my job interviews.
    a. Conditions of satisfaction
        i. The user can register and login using only LionMail affiliated with Columbia University.

ii. The user can view shared notes by specifying department name and course number.
iii. The web app will display all results from the search.
iv. The user can choose and click on one document to view explicitly.

Part 3:

Overall we will conduct the acceptance test in a blackbox testing environment, from the users perspective. Each acceptance test correspond to use cases

1.

Common Case #1:

a. User enters the homepage of airNote
b. Registers an account with Columbia University email
   Sample inputs: rh9921@columbia.edu
c. The system prompts the user to verify email
d. User clicks the link in verification email
e. Returns to airNote and login with correct credentials
f. Searches a course number
   Sample inputs: COMS4156
g. Clicks one entry on the webpage and view the notes
   Sample output: A list of notes in pdf format with preview

Common Case #2 (user already have an account):

a. The user opens the airNote website and login
b. User searches a course number with a corresponding department name
c. The course is found and course notes are displayed
d. User clicks into a note and view it

Common Case #3 (user already have an account):

e. The user opens the airNote website and login
f. The user clicks upload note button and is redirected to file upload page
g. The user uploads the file and fills in necessary information about the course, now the note is available to all users
   Sample input: a pdf file / note in plain text

Special Case #1 (invalid registration email):

a. User enters the homepage of airNote
b. Registers an account with email not from Columbia University
   Sample input: superman@gmail.com
c. The system reject the email address with no verification email sent

Special Case #2 (invalid login credential):

a. (Suppose the user already has an account)
b. User enters airNote and login with wrong password
c. The system reject the request and ask the user to try again

Special Case #3 (course not exist):

a. The user opens the airNote website and login

b. The user searches a course number corresponding to a department name
c. The specified course does not exist.

Special Case #4 (invalid file type):

a. The user opens the airNote website and login
b. The user clicks upload note button and is redirected to file upload page
c. The user uploads the file but the file type is not expected.
   Sample input: exe file, bash file

2.

Common Case #1:

a. User enters the homepage of airNote
b. Registers an account with Columbia University email
   Sample inputs: rh9921@columbia.edu
c. The system prompts the user to verify email
d. User clicks the link in verification email
e. Returns to airNote and login with correct credentials
f. Searches a course number
   Sample inputs: COMS4156
g. Clicks one entry on the webpage and view the notes
   Sample output: A list of notes in pdf format with preview

Common Case #2 (user already have an account):

h. The user opens the airNote website and login
i. The user clicks upload note button and is redirected to file upload page
j. The user uploads the file and fills in necessary information about the course, now the note is available to all users
   Sample input: a pdf file / note in plain text

Special Case #1 (invalid registration email):

d. User enters the homepage of airNote
e. Registers an account with email not from Columbia University
   Sample input: superman@gmail.com
f. The system reject the email address with no verification email sent

Special Case #2 (invalid login credential):

d. (Suppose the user already has an account)
e. User enters airNote and login with wrong password
f. The system reject the request and ask the user to try again

Special Case #3 (course not exist):

d. The user opens the airNote website and login
e. The user clicks upload note button and is redirected to file upload page
f. The user input a course number corresponding to a department name
g. The specified course does not exist.

Special Case #4 (invalid file type):

d. The user opens the airNote website and login
e. The user clicks upload note button and is redirected to file upload page

f. The user uploads the file but the file type is not expected.
   Sample input: exe file, bash file

3.

Common Case #1:

a. User enters the homepage of airNote
b. Registers an account with Columbia University email
   Sample inputs: rh9921@columbia.edu
c. The system prompts the user to verify email
d. User clicks the link in verification email
e. Returns to airNote and login with correct credentials
f. Searches a course number
   Sample inputs: COMS4156
g. Clicks one entry on the webpage and view the notes
   Sample output: A list of notes in pdf format with preview

Common Case #2 (user already have an account):

k. The user opens the airNote website and login
l. The user searches a course number corresponding to a department name
m. The user clicks into a note and view it

Special Case #1 (invalid registration email):

g. The user enters the homepage of airNote
h. The user registers an account with email other than LionMail
   Sample input: superman@gmail.com
i. The system reject the email address with no verification email sent

Special Case #2 (invalid login credential):

g. (Suppose the user already has an account)
h. User enters airNote and login with wrong password
i. The system reject the request and ask the user to try again

Special Case #3 (course not exist):

h. The user opens the airNote website and login
i. The user searches a course number corresponding to a department name
j. The specified course does not exist.

Part 4:

IDE: PyCharm
Package Manager: pip/conda
Unit testing tool: unittest within PyCharm integrated tools
Style Checker: pep8 1.7.1/ pylint 2.6.0
Bug finder: pylint 2.6.0
Code Coverage: coverage 5.3
Persistent data store: DynamoDB/ MongoDB