

INT ELLIGENCE FIRST ITERATION DEMO

C Sai Srujan – sc4401

G. Chandana Priya – cg3111

Varun Chandra – vc2491

Jiayi Wang- jw3704

- 1) We completed this demo on 7th November (Wednesday) at 4:00 pm. The main challenge we faced during the demo was a minor bug that occurred during the demo. However, we were able to debug on the spot and demonstrate the correct functionalities. In particular, this was a database error which occurred due to inconsistent data.
- 2) The user stories that we completed are:
 - Create Account
 - Sign In
 - Create Groups
 - Create Meetings

The conditions of satisfaction are:

User Story	Rule	Given	When	Then
Create Account	Does not accept duplicate usernames	Database contains user "ABC1234"	User types "ABC123" as username and "123456789" as password	New user with username "ABC123" is created and stored into the database AND User "ABC123" is logged into the account
		Database contains user "ABC1234"	User types "ABC1234" as username and "123456789" as password	No new user is created
Sign in	Username and password must match to sign into the account	Database contains user "ABC1234" with password "123456789"	User types "ABC123" as username and "123456789" as password	Sign in is not authorized AND Prompt user to re-enter username

				and password
		Database contains user "ABC1234" with password "123456789"	User types "ABC1234" as username and "123456789" as password	Sign in is authorized AND User dashboard is shown
Create Groups (Organizer)	User creates a group and sends invitation to the attendees.	User creates a group and adds a list of attendees using their IDs.	User enters all valid user names	A group is created with the given users as participants and invitations are sent out to them.
		User creates a group and adds a list of attendees using their IDs.	User enters one or more invalid IDs	Group is not created and no invitations are sent out

3) We used travis CI for continuous integration.

We connected our GitHub repo with our travis account. The .travis.yml file contains scripts to run our program and test cases.

The .travis.yml file can be divided into 3 parts. First, the requirements.txt file contains libraries that need to be installed. By running the setup.py file, our Meeting Scheduler library is installed. Next, the main script runs the flask server and prepares the data in our database. Then, pytest runs the test cases in test folder. At last, log files for bug finder and test cases are generated and pushed back to our repo.

Writing script for installing library, and running test cases and bug finders is easy. The more challenging part for us was pushing back to GitHub repo. We don't have previous experience and we had some trouble pushing to remote repository. For the first few tries although the script runs, no file is added to our repo. We changed our script several times to make it work.

For bug finder we used prospector, a python library that integrated tools such as pylint. We used unittest and pytest library for testing.

4) Git Hub link with first iteration tagged

link: https://github.com/ASE-Int-Elligence/Meeting_Scheduler/releases/tag/First-Iter