System and Unit Test Report

Accent

November 23, 2016

Sprint 1 User Stories:

1. User story 1 from sprint 1: As a user, I want the app to be able to recognize what I’m saying.
2. User story 2 from sprint 1: As a user, I want the app to be able to synthesize my speech
3. User story 3 from sprint 1: As a user, I want to be able to create an account.

Scenario:

1. Start Accent app; select ‘new user’; type
   1. First name = ‘enter first name here’;
   2. Last name = ‘enter last name here’;
   3. Email = ‘enter email here’;
   4. Password = ‘enter password here’;
   5. Press enter key
   6. User should see that account is created=====this was only working for frontend without actually storing in backend?============
2. Select button to speak
3. Speak and see that there is speech on the screen recognize what was spoken.

Sprint 2 User Stories:

1. As a user, I want to be able to hear strings
2. As a developer, I want to be able to query a python script via a POST request
3. As a developer, I want to be able to train my n-gram replacement model
4. As a developer, I want to be enable R/W access to backend while running through a daemon.

Scenario:

1. ===================no idea if we can hear strings ===================
2. ===================wonder how script is being run ===================
3. =====================idk about n-gram replacement model=============
4. Type “<http://159.203.233.58:8000/accent/default/api/login/email@email.com/password>” in a browser
   1. Information in the database should come up like “{"content": {"firstname": "agadberr", "lastname": "agadberr", "id": 11, "password": "password", "email": "email@email.com"}, "status": "success"}”
5. Type “curl -d "firstname=hi&lastname=hello&password=12345&email=hello@gmail.com" <http://159.203.233.58:8000/accent/default/api/acc>” in our server.
   1. Store’s information into the database and comes success like “{"acc": {"errors": {}, "id": 3}, "status": success"}”

Sprint 3 User Stories:

1. As a developer, I want to be able to make a POST request to correct text input
2. As a user, I want to be able to save my past speech queries
3. As a user, I want to be able to create an account
4. As a user, I want to be able to login
5. As a developer, I want to be able to query for joint probabilities of bigrams and trigrams.
6. As a developer, I want access to the NUCLE grammar correction dataset.

Scenario:

1. Start Accent app; select login; type
   1. Login = ‘user login here’;
   2. Password = ‘user password here’;
   3. Press button in the app
   4. Speak a string in the app
   5. Look and see string corrected =================not sure if this is how its corrected or by a button=================================
2. Start Accent app; select login; type
   1. Login = ‘user login here’;
   2. Password = ‘user password here’;
   3. Press button in the app
   4. See past speeches in the app. ===============dont think this is implemented yet=============================================
3. Start Accent app; Select ‘new user’; type
   1. First name = ‘enter first name here’;
   2. Last name = ‘enter last name here’;
   3. Email = ‘enter email here’;
   4. Password = ‘enter password here’;
   5. Press enter key
   6. User should see that account is created
4. ========Procedure for query’ing joint probabilities of bigrams and trigrams =========
5. =====Procedure for NUCLE grammar correction data set=============

• Unit tests (25 points): Include a file/directory named ‘Testing’ in your Git Repository. There should be details (can be in a separate file in the directory) provided by each team member about the module and the functional testing they have done. Each team member picks a module or module and lists the equivalence classes and the test cases selected to cover all equivalence classes