Team Members: Lukas Bush, Bao Nguyen, Megan McGinnis, Weiqiang Wang, Jack Humphrey

Project Title: WTW(What To Wear)

User Acceptance Tests:

- Search Bar
- Weather API
- Comments
- Weather Man

Search bar:
When users type something into the search bar, it's going to suggest cities based on what the users typed into the search bar. We are going to use bootstrap on the drop-down search bar so that it's not going to take up an entire page.

As users are typing in the specific city name, the search bar is going to get smaller until the user has fully typed in the city name. For example, if a user types "spring ", the search bar should have all city names begin with "spring", (spring, Springfield, Spring Hill, etc.). When the user types more characters, the search bar should match the input string automatically.

Since some cities have the same name and belong to different states or countries if the user does type that city name in, the search bar suggestion must display all the different places with the same city name. If the user accidentally misspells the city name, then the search bar suggestion will be empty since the misspelled city name wouldn't be in our array. If the user tries to click on search with a misspelled city name, then there will be an alert saying that the city doesn't exist. The user then has to re-enter the city name with the correct spelling.
The drop-down search bar should have a format like (City, State), (Boulder, CO) to identify the weather of the city. For example, if a user types "Spring", the down bar should have (Spring, TX), (Springfield, MO),(Springfield, MA),(Springfield, OH), etc.
The test case can be :

- Correct City names with incorrect State name(expect: an error alert)
- State name only (expect: city names begin with the State name like 'CO' will have (Colora, MD), (Colorado Spring, CO), etc.)
- Incorrect city names(expect: empty search bar, and an error alert if user want to check the weather)
- City names only (expect: no response, until the user chooses a state name)

Weather Man:
The weatherman will consist of our front end and back end data. This is going to be on the WTW page of the website. After users have typed in their city name, a suggestion for what the user should wear will be shown based on the current weather for that specific location. The basic pictures(hat, shirt, gloves, pants, socks, shoes) will be over the stickman image and once

the user clicks on a specific picture, there will be a display showing what they can wear for that day. For example, if the user chooses a city where it's snowing, then clicking on the picture of the gloves will display winter gloves for them. If the user chooses another city where it's hot, clicking on the picture of the gloves will either be empty or have some sort of statement saying "you don't need gloves today". This feature must work with the weather API. This means that if the city is hot, the pictures on the weatherman should only suggest clothes for the hot weather and not winter clothes.
Test case:

- Change the clothes in the stickman(expect: if the user chooses a new hat, the new one will replace the old one.)
- The suggested clothes should have a default setting for different. If there is no suggestion about wearing, the stickman will have a default picture for it. (expect: the default image should base on the weather API)
- Click on the different clothes of the stickman, if the pop-out display opens normally, and close normally (expect: If we click in the image of the gloves for 100 times, it should only have one pop-out display, until we close the pop-out display and click in the glove again, a new pop-out display will come out.)

Weather API:
The weather API will be tested by running multiple forecasts across different locations. The success of this test will be dependent on each location that is tested produces the correct image, high temperature, low temperature, humidity, and the chance of precipitation. The correctness of each of these deliverables will be checked against an outside weather program such as weather.com to ensure that the correct values are inserted for each day. There should be a five day forecast as well as the current day information. This API information should then also be available to the weatherman feature in order to produce the correct images on the man based on the data received from the API call. This will be checked by ensuring all temperature ranges are met with the proper images.

- Test two cities name from each State, such as (Denver, CO) (Boulder, CO)(expect: it should return the correct information)
- Test cities have the same names but different locations.
- The weatherman image should change based on the temperature from the API information. (If the API return a hot temperature, it weatherman should have clothes for hot weather)

Comments:
The comments section will include one database where all comments are stored but each location will have its own comment section. This will be tested by creating test comments into the database for several different locations then loading that locations page to ensure that only those comments for that location appear in the comments. We will also test to ensure that no

comments that should be included in the query are excluded for any reason or placed into the wrong location. The last test that will be conducted on the comments feature will be that all fields required to submit a comment have been filled out otherwise the user will not be able to submit the comment and will be prompted to fill in the missing field before resubmitting and passing the comment into the database.
Test case:

- Incomplete form in comment page (expect: the user should not be able to submit the comment, and incomplete part will be marked)
- Comment with too many characters (expect: When the user types too many characters, the input field will not allow it.)
- Submit different comments to the database (different location, temperature), and those comments for that location should appear.