

These are example answers and not the only acceptable answers.

- CabTouch (Riley)
 - Black Box testing with Equivalence Partitions and Boundary Analysis
 - Number Of Drivers:
 - 0 drivers
 - There are no drivers in the area.
 - No drivers have available seats
 - 1 driver
 - Many drivers
 - Phone Number is Invalid:
 - Phone number is too short (shorter than 7).
 - Phone number is too long (longer than 11).
 - Phone number starts with 1 and is not 11 characters.
 - Phone number does not start with 1 and is not 7 or 10 characters.
 - Phone number contains letters.
 - Phone number is disconnected.
 - Phone number is not listed (null).
 - Email is Invalid
 - Email is Null.
 - Test email fails.
 - Location
 - Location is invalid. This will be implementation dependant depending on how location is implemented. Ex: invalid longitude/latitude or location outside of new york.
 - Other
 - No valid contact method available.
 - Black Box testing with metamorphic testing.
 - Given a set of cab drivers within a radius with defined availability. The test should always return the same number of drivers.
 - Vary the locations of the cabs but within the same radius.
 - Vary which drivers are available but keep the number of available drivers constant.
 - Adding one driver within the radius should add one driver to the result.
 - Adding N drivers within the radius should return N new drivers in the results.
 - Grey and White Box Testing.
 - Stress test the api by making a large number of concurrent requests.
 - Stress test the application by making a large number of concurrent requests.
 - Computational stress test by making a large number of cabs returned from the search.

- Test how the application handles the API being unavailable.
 - Assume the application caches contact methods and test cache invalidation.
 - Test the contact functionality by making sure that invalid emails/phone numbers are recognized as invalid.
- Library+ (Rogers)
 - Black Box testing with Equivalence Partitions and Boundary Analysis

User inputs a location(zip code,place name) that has a library

User enters a location that does not have a library

User enters an invalid place(invalid zip code, invalid place name)

 1. User enters a valid location but there are no libraries near by.
 2. User enters a valid location and the list of near by libraries is returned.
 3. User enters an invalid location and an error page is returned.
 4. User enters an empty input is prompted to enter a location
 5. User enters as input numbers
 6. User enters special characters as input locations
 7. User location information with spelling errors
 8. The links returned in the result should redirect to appropriate library home pages
 9. The map should be marked properly with the results
 10. The opening and closing time should be mentioned in the results
 - Black Box testing with metamorphic testing.

Given a set of libraries in a location with the book. The test should always return the same number of libraries.

Vary the search by changing the location of the library. For example if the search for the library in New York has returned 4 libraries in Upper West Side, change the location to Upper West Side; the same 4 libraries have to be returned

Testing for same location using different input formats(location,zip code etc)

Typing the address in multiple formats “NY 129th Street 10027” and “10027 129th Street NY”
 - Grey and White Box Testing.
 - Stress test the app server by making a large number of concurrent requests.
 - Computational stress test by making a large number of libraries returned from the search(give country name and readily available book for search).
 - Test how the application handles the MAP API being unavailable.
 - Test the link functionality by making sure that links are displayed for libraries that have them
 - Ensure that when the app is launched again the input fields have to be

cleared(from code)

- The Social Circle (Tejaswini)
 - Black Box testing with Equivalence Partitions and Boundary Analysis
 - Zip code:
 - Positive 5 digit - Should mention the assumed range
 - Negative
 - Alphabets
 - Blank
 - any number lesser or greater than 5 digits.
 - Options:
 - Click on existing options
 - Simulate negative index or an index greater than the list of options
 -
 - Nutrition value:
 - Should return nutrition value for valid food items
 - Should return error for invalid food items
 - Check if food item falls within the calories that we wish to eat.
 - Black Box testing with metamorphic testing.
 - With defined restaurants, movie theatres etc in a given radius:
 - It should display the same number at any time of the day
 - Within various points in the radius, it should display the same number
 - If we check restaurants after checking movies and bars it should return the same value as bars after restaurants and movies
 - Adding 1 restaurant should increase the number of restaurants by 1
 - with defined nutrition value of a snack:
 - The nutrition value of a healthy snack should be more than that of an unhealthy snack
 - varying the order of checking the nutrition values should not vary the values
 - Should return the same number of reviews even if order is changed
 - Grey and White Box Testing.
 - Test how API handles invalid zipcode
 - Test if API is returning the entire list of restaurants, movie theatres etc
 - Check how application handles API unavailable and invalid authentication
 - Check if the zipcode being entered is the actual zipcode that is passed to the API