1. Automated Unit Test

Our Unit test is focusing on the functionality verification which is implemented by gtest, the scope of our Unit test’s targets include add, delete, update, search, and undo operations, all of them are tested by examining the Calendar entry list and general entry list. In the Unit test, all the system requirements are given to ensure those functions work in a “safe environment”, so only the functionality are tested in the Unit Test.

1. Systematic Test

Being contrast to Unit Test, the systematic test is against the system specification and we take a whole system to test instead of a part of the system.

* 1. Performance testing
     1. When the user input a 50-characters-long command, how long user will wait for the software responds, and is there any bug?
     2. When the user input a 255-characters-long command, how long user will wait for the software responds, and is there any bug?
     3. When the user input a 500-characters-long command, how long user will wait for the software responds, and is there any bug?
     4. When there are 10 entries in the storage file, how long we need wait for the software responds?
     5. When there are 100 entries in the storage file, how long we need wait for the software responds?
     6. When there are 1000 entries in the storage file, how long we need wait for the software responds?
     7. When there are 10000 entries in the storage file, how long we need wait for the software responds?
  2. Load Testing
     1. When there are 10 entries in the storage file, is it loadable?
     2. When there are 100 entries in the storage file, is it loadable?
     3. When there are 1000 entries in the storage file, is it loadable?
     4. When there are 10000 entries in the storage file, is it loadable?
  3. Security Testing
     1. Wrong-format command Testing
        1. Random strings eg. “xxxxx”
        2. The command keyword correct but with random string following.

Eg.1) Add xxxx 2) delete xxxx 3) search xxxx 4) undo xxxx 7) update xxxx

* + - 1. The command correct, but details following is incomplete.(only applicable for calendar entry as the general list would take all things as entry description)

Eg. add 10:00-01:00 (data missing) have lunch at soc priority high

* + - 1. Both of command and details are correct.
    1. Sensitive number/words Testing

As our codes get some sensitive words lists:

For instance, ‘#’,” at” and ”priority”. When these three words are included in the command, it will be errors.

* + 1. Modify the storage files.

1. Remove the storage files.
2. Modify the content inside the storage files.
   1. Portability Testing
      1. Run in Windows 7
      2. Run in Windows XP, NT, Vista.
   2. Usability Testing
      1. Get the feedback for our software from different knowledge-background people
   3. Compatibility testing

As our YourDay is independent from other software, we do not need to test this.

Note: All the test cases can be found under/testing/System Test Case/, but when you are using them to test, please put them under /source/YourDay/ directory and change their name ”YourDayCEntry-XXX.txt” to “YourDayCEntry.txt” and ”YourDayGEntry-XXX.txt” to “YourDayGEntry.txt”.

1. User Acceptance Testing

: The testing will be against the requirements specification, not system specification.

For the Acceptance test, we launch it after the System testing, here are requirements for our final project, and we need to test whether the system does what it is intended to do, which means meets the requirements.