**Test Strategy:**

* 50% tests will be automated, while another 50% will be tested manually.
  + Automated tests will focus on the validation mechanism (e.g. valid input of adding tasks) and use Junit.
  + Manual tests will focus on the functionality of GUI.
  + All tests should be well designed to cover most of the error inputs of users.
* Success criteria – 95% of test cases passed.
* No high priority defects unresolved.
* No runtime error or exception occurs.

**Automated test with Junit**

We use Junit to test the basic function of our program. Tests focus on the validation mechanism, such as valid inputs in adding tasks. We used partition testing.

1. User Test
2. Test the function of initialization of User.

Input: a registered user

Output: return the correct user id

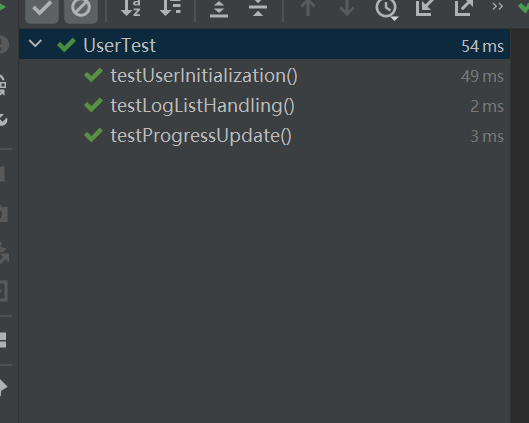
1. Test the progress updating.

Input: a valid progress

Result: return the progress

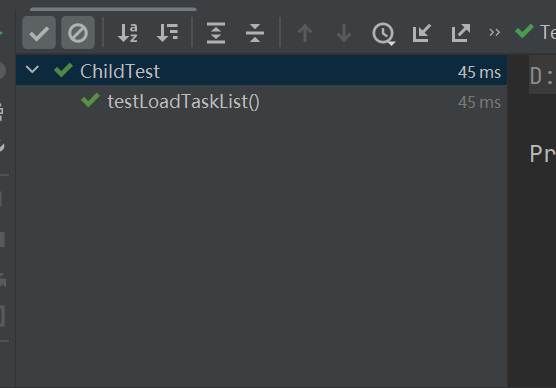
1. Test the log adding.

Input: add a new log item

Output: the size of log list increases by 1.

1. ChildTest
2. Test loading tasks from a json

Input: a valid user containing 1 task

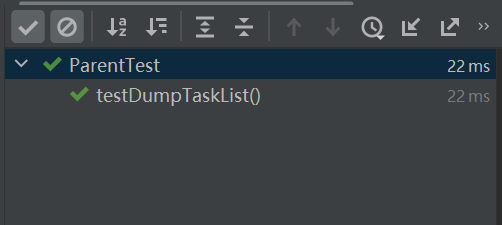
Output: the size of the task list

3. ParentTest

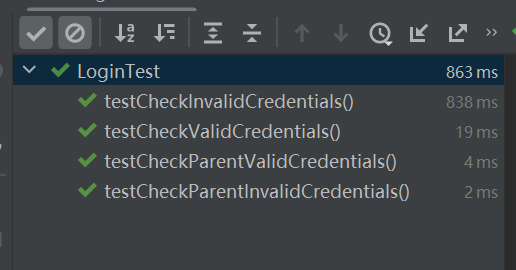
1. Test dumping the task list

Input: a valid parent

Output: found the item written to the json

4. Login Test

Partition testing

1. Invalid Child Account
2. Valid Child Account
3. Valid Parent Account
4. Invalid Parent Account

5. Signup Test

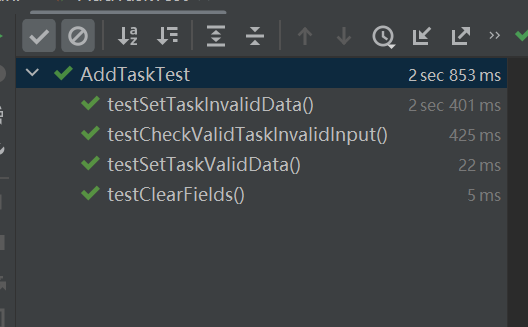
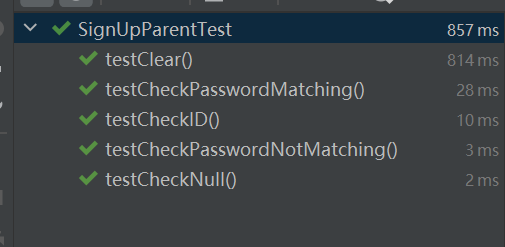
Partition testing

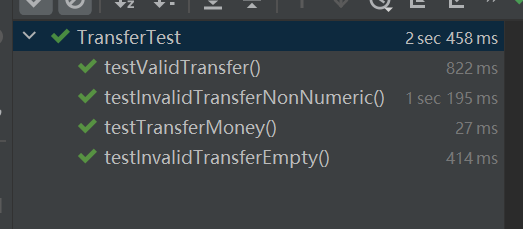
1. Test duplicate ID
2. Test Different Password
3. Test valid Input
4. Test Register
5. Test Clear

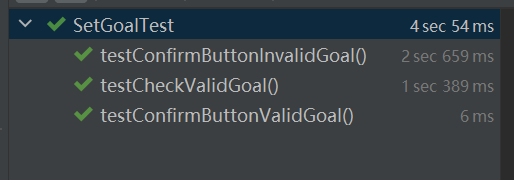
Note: Same test for both child and parent while signing up

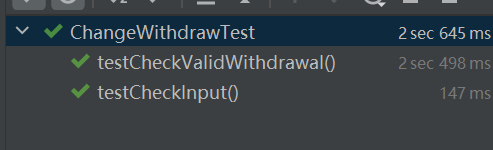
1. 图形用户界面, 文本, 应用程序

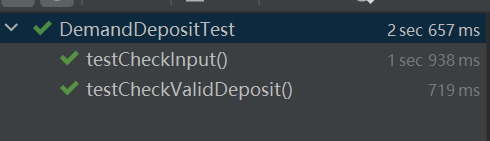
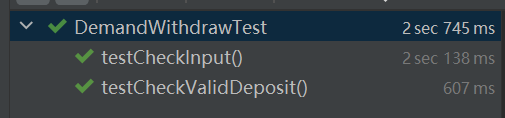
   描述已自动生成For child, test valid parent ID

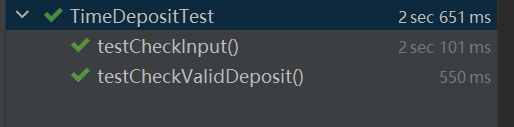
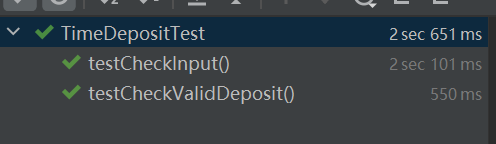
6.AddTask Test

7.Transfer Test

8.SetGoal Test

9. Deposit&Withdraw Test





Manual Test with Scenario-based Testing