Child Safe Test Plan

Team members:

Binwu Peng

Hui Wan

Zeyu Liu

Wenkai Fan

Yuan Feng

Cheng Zhang

Zihui Zheng

Jianwei Du

Methodology

Test Overview

•Testing Stages

In general, a test plan consists of three test stages. They are development testing, release testing, and user testing. Development testing is done during the development process, which contains tests of individual methods and classes. After integrating all components, system tests are carried out to test the interfaces as well. Then, before delivering the product, a release test should be done to check whether it fulfills the requirement documents or not. All combinations of functions with both correct and incorrect inputs should be checked at this stage. Finally, user testing should be conducted by potential users. The two phases of user testing are alpha testing, which is conducted at the developer's site, and beta testing, which is available to the public.

For this project, we plan to spend more effort on development testing, as this part is the basis of the whole project. After finishing all coding assignments, we will conduct release testing together and then, ask our sponsor to finish user testing.

Testing Goals

•Validation Goals

For the validation parts, our goals are to prove that our product complies with our requirement documents. We will use a set of test cases, which are identical with the expected usages of our app, to make sure that our software meets all requirements. Thus, every requirement (both functional and non-functional) should be covered by at least one test case.

•Verification Goals

The verification testing part, which is also called defect testing, is to discover faults or irregular behaviors in the software. Therefore, our test cases should be deliberately malignant and cover as many undefined or malicious behaviors as possible. Specifically, we would design test cases for corner cases like invalid inputs, input/output overflow, undesirable network environment, enormous amount of repeating operations, etc. The task here is to generate more and more unpassed test cases.

Unit testing

•What it does

Unit testing is used to test individual components separately. The component can be a function, a class method, a class itself or even a group of classes with certain interfaces exposed to the outside. We can write relatively simple test cases to test these components fully so that we can be sure that each one of them performs as expected.

•Why it helps

Unit testing is helpful as it is the most basic testing procedure which makes it simple and powerful. By going through unit testing earlier in the development process, we can find bugs with lower cost, structure our program better, as well as writing more detailed documentation.

•How to perform

We should use two types of test cases to ensure our program is working correctly. One is normal input which is used to determine whether our program works as expected. The other is abnormal input which requires proper handling on the program part.

Component Testing

•What it does

Component testing is implemented after unit testing. It tests unit combinations as a single module independently during the code development phase and finds defects for each component module. It ensures further larger testing for the application.

•Why it helps

It helps to check the individual modules in a large application. It can find defects at the early stages. It ensures that each smaller component is tested and bugs are cleared early which saves time for future software development. It will cost a lot if there were hidden component malfunctions during the system testing phase.

•How to perform

We access functions of each independent function module through its interface and give inputs through the interface, check whether the outputs meet the requirement. We pass parameters and make sure data are correctly passed through methods. We test whether modules can share the block of memory, make sure methods are called correctly through encapsulation and message passing correctly. And for components that represent the key functionality of the application, we should pay especially attention to them regardless of their complexity.

System Testing

•What it does

System testing is the critical testing between component testing and Accept testing or user testing| Alpha testing. It systematically tests the integrated system. Especially, it focuses on how the component interacts with each other, how its performances in the application environment, does it meet user requirements. Usually, a third testing team out of the original development department should take this test.

•Why it helps

System testing is the first time to test the software as a whole from the client perspective. It focuses on the behavior of the software, such as the throughput, the latency, and other business requirements. Also, it helps to expose the system design and architecture issue. More importantly, it will also help find the problems during the migration from the development environment from real business situations due to its deployed in the customer's production environment characteristic.

•How to perform

System testing is usually a backbox testing or ending to ending test. It verifies the input to the system produces the expected results. It also will test the non-functional requirement of the system.
There are the following steps:
- Define the purpose or scope of the test
- Confirm the acceptance criteria
- Compile the testing schedule
- Create the test case
- Conduct the test
- Analyze the test result
- Conclude pass or fail

Summary

Testing is an important chain of the software production process. It begins at the requirement specification phase so as to completely understand the software requirement. Finding all the problems within a software system is impossible. However, scientifically and reliably test the software could expose the problem maximally before it is shipped to the customer. The unit testing, component testing, and system testing are all indispensable parts of the testing process. Each testing focus on one level of problem and all they combined to provide satisfying quality assurance. The essence of software testing is to find the problems before the delivery and thus provide software quality assurance. It provides an objective and independent view of the quality of software to all the stakeholders.

Plan

Sprint1

•Unit tests

Task ID	Test Name	Owner	Start Date	Finish Date	Test Steps	Expected Result
0	User	Wenkai Fan	3/5	3/6	The classes are tested with various string/int type inputs.	1. The inputs are handled correctly. 2. The server can connect with no problem.
1	ServerAdapter	Wenkai Fan	3/5	3/6		
2	MainActivity interface	Hui Wan	3/5	3/6	We navigate back and forth between those interfaces. We also use the home/back button to test if the app works correctly.	1. The interfaces behave as expected. 2. There is no crashing during the running of the app.
3	MainFragment interface	Hui Wan	3/5	3/6		
4	Authentication interface	Hui Wan	3/5	3/6		

•Component tests

Task ID	Test Name	Owner	Start Date	Finish Date	Test Steps	Expected Result
T1	Sign up	Zihui Zheng	3/1	3/2	1) Register with Email 2) Login in with registered accounts	A valid account can be created and found in the user information database. If the registration information is correct: 1) correct email; 2) email is verified by clicking the

						link in the verification email 3) password is valid.
T2	Log in	Zihui Zheng	3/1	3/2	Login in with registered accounts	Can log in the app if the account name/ password are correct, otherwise, log in should fail
T3	Reset password	Zihui Zheng	3/1	3/2	1) Using registered account to change its password 2) Using "forget your password" function to change the password	Changing password successfully for authenticated accounts if the new password is valid; Can change password via email if an account is registered with the email and new password is valid
T4	Binding/Unbinding parents and children accounts	Jianwei Du	3/2	3/3	Crosswise bind one parent account with several children, also bind one child account with several parents; Unbind one child account and check the correctness of display on each parent bound with these children	List of children bound for each parent is correct when we unbind one child account, the name of this child should be removed from every parent account which bound with this child
T5	Launching lost/found child event	Zeyu Liu	3/4	3/5	Two emulators are used. We declare lost in	When we declare lost in the child

					the child account and observe whether the corresponding parents get the notification or not. Then, declare found in the parent acc Also, we test corner cases including repetitive declaring, declaring on other devices and declaring in irregular order(declare found before declaring lost).	account, the corresponding parent account will show a lost label. Then, after declaring found, the label will change to safe. Only the first time we declare an event will change the status. When the child is safe/lost, declaring safe/lost is useless.
T6	Calling 911	Zihui Zheng	3/4	3/5	Press the calling button under a lost/safe status.	911(in fact, an alternative phone number is used instead of 911 for testing convenience) will always be called, no matter what the current status is.
T7	Location tracking and data uploading module	Wenkai Fan	3/7	3/8	A real phone and an emulator are used for testing. We close/kill the app, reboot the phone, put it into idle mode to test this module.	The location recording and uploading stream work correctly as long as the user does not log himself out/status change to safe. Even kill the app or reboot the phone does not affect this module which is what we want to

Task ID	Test Name	Owner	Start Date	Finish Date	Test Steps	Expected Result
						achieve.
T8	Single device sign-in functionality	Hui Wan	3/7	3/8	A real phone and emulator are used for testing. We test sign in on the two devices with different order/with the signed in a device being in a different status.	The previous signed-in device is automatically signed out as long as it is connected to the internet. No matter what activity it is in. This is again what we expect.

•*System tests*

Task ID	Test Name	Owner	Start Date	Finish Date	Test Steps	Expected Result
0	Server Stability	Cheng Zhang	3/6	3/7	We try to send various data to the server on both a real device and an emulator. Both valid and abnormal data are used.	The server can handle different types of data without problems.
1	App stability	Hui Wan	3/6	3/7	The app is run many times. All the clickable UI components are tested multiple times. We also tried to input different strings in all the input fields.	The app is running correctly and can handle all kinds of inputs.

Sprint2

•*Unit tests*

Task ID	Test Name	Owner	Start Date	Finish Date	Test Steps	Expected result
1	User Profile Interface	Wenkai Fan	3/17	3/18	The user profile interface is linked to the main interface. We try to navigate back and forth. We also test if the back and home button performs correctly in this interface.	The navigation of these interfaces is working as expected. The various clickable UI components on these interfaces are also working properly.
2	Settings Interface	Cheng Zhang	3/17	3/18		

•*Component tests*

Task ID	Test Name	Owner	Start Date	Finish Date	Test Steps	Expected result
T9	Day/night mode switch	Binwu Peng	3/17	3/18	A real phone and emulator are used for testing. Open the app and zoom to the setting page. We change the night model switch from on to off and from off to on.	The app automatically changes the background. When the switch is on, it shows a black background. When it is off, it shows a white background.
T10	Notification entrance	Binwu Peng	3/17	3/18	A real phone and an emulator are used for testing. The phone opens and signed the app as parents and run the app in the backend,	The app automatically receives a notification. And by tab the message, the child safe app will be opened.

Task ID	Test Name	Owner	Start Date	Finish Date	Test Steps	Expected result
					with at least one binding child. Any child status change will send notifications to all parent app.	
T11	Child status change disabled on parent end when offline	Yuan Feng	3/17	3/18	A simulator is used for testing. Try to change the status of the child when the Internet is off.	When the Internet is off, the parent end cannot see the information of children
T12	Not implemented yet					
T13						
T14						

• *System tests*

(The same aspect of the system is tested, but now we have more features and functionalities and we want to see if the system is still working as a whole)

Task ID	Test Name	Owner	Start Date	Finish Date	Test Steps	Expected result
0	Server Stability	Cheng Zhang	3/18		We try to send various data to the server on both a real device and an emulator. Both valid and abnormal data are used.	The server can handle different types of data without problems.
1	App stability	Hui Wan	3/18		The app is run many times. All the clickable UI components are tested multiple times. We also tried to input different strings in all the input fields.	The app is running correctly and can handle all kinds of inputs.