FaceRecognitionAttendance

Testing plan

Project Name: **FaceRecognitionAttendance**

Project Unit: **Group 9**

Project Time: **2019.2.28 - 2019.5.19**

Version Number: **2.0**

Revision history

Version	Date	AMD	Reviser	Description
1.0	2019-4-28		Group 9	
2.0	2019-4-29	①Add some test strategies ②Modified some test specifications	Group 9	Perfect V1.0

Content

1. Introduction

1.1 Purpose

In order to test whether each functional module in **FaceRecognitionAttendance** can meet user requirements and whether there are functional or logical errors, we have written this test plan document. It will help achieve the following goals:

（1）List the recommended test requirements;

（2）Recommend and explain the testing strategies that can be adopted;

（3）Determine the required resources and estimate the test workload;

（4）List the deliverable elements of the test project.

1.2 Background

This test object is a real-time face check-in system based on C/S architecture. The system can realize the identity registration of the administrator and the object to be investigated, based on the real-time face check-in, basic information increase, modify, delete, search and other functions. At the same time for the administrator provides a complete set of management system and user experience good application.

1.3 Scope

1.3.1 Response time

During the testing process, we will test each module or function under test to obtain the time required for it to respond to the request, which is called response time. Response time, as the main embodiment of software performance from the user's perspective, should be divided into "presentation time" and "system response time".

1.3.2 Concurrent users and concurrent online users

This document distinguishes between the two concepts because they are not really equivalent. Normally, the server can accept multiple users online at the same time, but for a hot business scenario, there may be a large number of users performing the same operation at the same time, we call it concurrent users. So the performance of the most commonly used and focused business operations will be tested in the documentation.

1.3.3 Handling capacity

We will set up corresponding load and strength tests to verify the number of customer requests processed by the system per unit of time and call this throughput. Throughput directly reflects the performance carrying capacity of the system, which is reflected not only in the middleware, but also in the database or hardware.

1.3.4 Operation interface and logic

During the testing process, we need to ensure that the user interface can access the corresponding data through the use of object controls or entries, and that the operation logic conforms to the corresponding specifications or most user operation habits. Secondly, it is necessary to test whether the user interface style meets user requirements, such as whether the interface is beautiful, intuitive, user-friendly and easy to operate.

2. Test reference and submit documents

2.1 Test reference document

The following table lists the documents used to make the test plan and indicates the availability of each document:

Document (version/date)	**Created or available**	**Received or reviewed**	**Author or source**	**Remarks**
Feasibility analysis report	☑ True ☐ False	☑ True ☐ False	Group 9	
Software requirements definition	☑ True ☐ False	☑ True ☐ False	Group 9	
Software system analysis	☑ True ☐ False	☑ True ☐ False	Group 9	
Software summary design	☑ True ☐ False	☑ True ☐ False	Group 9	
Software detailed design	☑ True ☐ False	☑ True ☐ False	Group 9	
Software testing requirements	☑ True ☐ False	☐ True ☑ False	Group 9	
Module development manual	☐ True ☑ False	☐ True ☑ False	Group 9	
Test schedule and staffing	☑ True ☐ False	☐ True ☑ False	Group 9	
Testing report	☐ True ☑ False	☐ True ☑ False	Group 9	

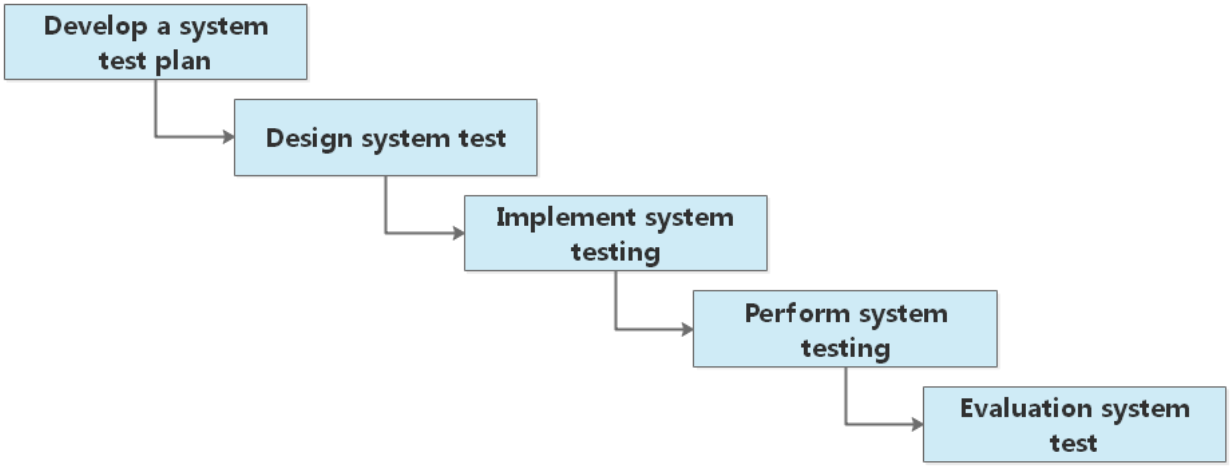
Test analysis report	☐ True ☑ False	☐ True ☑ False	Group 9	
User operation manual	☐ True ☑ False	☐ True ☑ False	Group 9	
Installation guide	☐ True ☑ False	☐ True ☑ False	Group 9	

2.2 Test submission

Test analysis report, appendix A and B

3. Test schedules

System test flow chart is as follows:



The following table lists the scheduling of test times for each phase:

Test activity	**Planned start date**	**Actual start date**	**End date**
Develop a test plan	2019.4.24	2019.4.24	2019.4.28
Design test	2019.4.24	2019.4.24	2019.4.28
Integration Testing	2019.4.30		
System test	2019.5.3		
Performance Testing	2019.5.7		
Installation test	2019.5.11		
User acceptance test	2019.5.13		
Evaluate the test	2019.5.16		
Product release	2019.5.18		

4. Test resource

4.1 Human resources

The following table lists the staffing configurations made in this project:

Role	**Minimum recommended resources**	**Specific duties or notes**
Chief in charge	1	Responsible for the management of all transactions during the test phase
Unit testing team	2	Within the team, one person was elected to be responsible for white-box interface testing and another for white-box testing of local data structures and basic database operations.
Integration test team	3	The general responsible person shall be responsible for data transmission between modules and functional conflict testing between modules; The other two are responsible for testing the functional correctness of module assembly and the global data structure.
System test team	2	The team elects one person for functional and performance testing and another person for interface, reliability, ease of use, and compatibility testing.
Acceptance test team	1	Black box test for all functions of the whole system to ensure the accuracy and reliability of all functions.

4.2 Testing environment

The following table lists the system environment under test:

Software environment
Android mobile operating system, version: V5.0 ~ V9.1
Windows desktop operating system, version: Windows7, Windows8, Windows8.1 and Windows10
Internet Explorer 8 and above
Google Chrome 52.0.2743.82 and above

Hardware environment
Intel(R) Core(TM) i5-3470 CPU @ 3.20GHz quad-core CPU desktop
Qualcomm Snapdragon 660 and above mobile processor platform

5. System risk & priority

System risk	Priority
Unable to accurately display and capture valid faces	High
Server downtime due to high concurrency	High
The application cannot communicate with the server for data	High
Unable to respond to application requests in a timely manner	Medium
The information portion is lost when the application data is not cached locally in response to an application request	Low
Form files transfer slowly	Low

6. Testing strategy

6.1 Data and database integrity test

Because the data storage of this system almost all depends on the database storage technology, so it is very necessary to design a set of complete data and database integrity test.

Test objective	Ensure that the methods and processes that access the database are working and that the data is not corrupted
Test scope	Mobile application, web administrator interface
Technology	Requests are sent to the server database on the mobile end and the web end respectively to fully invoke the database access methods and processes. Verify database access success by populating the database with valid and invalid data. In addition, you still need to verify that the RUD operation of the database is responded to and that the data returned is real and valid.
Starting standard	Null
Completion criteria	The methods and processes that access the database are running without data corruption
Test focus and priority	Null
Special issue	Testing may require the DBMS development environment or driver to enter or modify data directly in the database. The database needs to judge the response of the effective or invalid data randomly filled in the test stage, reject the invalid data in time and give information feedback.

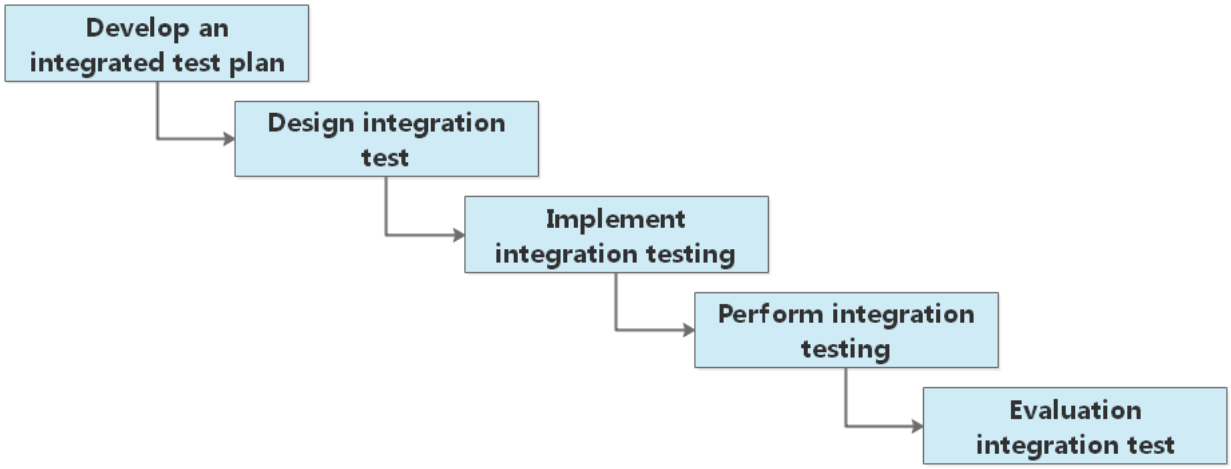
6.2 Interface test

The face recognition algorithm in this system relies on a self-developed algorithm interface, so it must be tested to verify the timeliness of the interface to return data under normal and special circumstances.

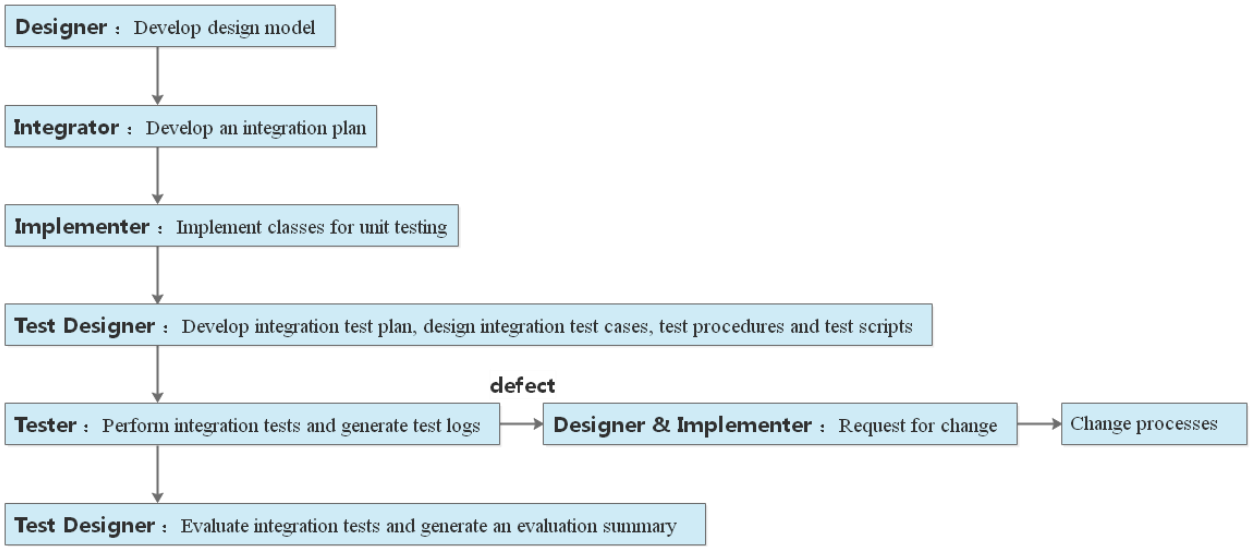
Test objective	Ensure that the interface calls are correct
Test scope	Face recognition interface
Technology	Design a sample program to verify the authenticity and validity of the data returned by the interface
Starting standard	Interfaces can be instantiated
Completion criteria	The interface is called correctly and the data returned is real and valid
Test focus and priority	The two most important methods in the priority judgment interface are getting facial feature vectors and comparing to determine whether the face can be called correctly and effectively.
Special issue	As for the input parameters that may be wrong during the development process, an additional verification interface is needed to verify whether these errors can be detected and explained in the exception feedback.

6.3 Integration test

The integration test flow chart is as follows:



And the integration test workflow is as follows:



Test objective	Check the process of each stage, the correctness of data flow, and verify the collaboration between modules.
Test scope	The mobile application and the webpages administrator interfaces respectively define the business process and verify the working state of the integrated functions obtained by combining different functional modules.
Technology	Use valid and invalid data to execute each use case, use case flow, or function to verify the following: Expect results when using valid data. Displays an error message or warning message when invalid data is used. The business rules are applied correctly.
Starting standard	Each module must meet its own standards before integration testing
Completion criteria	①All planned tests have been executed. ②All defects found have been resolved.
Test focus and priority	Null
Special issue	This stage requires the identification of internal or external factors affecting the implementation and execution of functional tests.

6.4 Function test

This phase of the test is based on black box technology, interacting with the application through a graphical user interface, and analyzing the output or results of the interaction to verify the application and its internal processes.

Test target	Make sure the tests are functioning properly, including page navigation, data input and output, processing and retrieval.
Test Scope	Mobile app and web-side administrator interface.
Technology	Use valid and invalid data to execute individual use cases, use case flows, or features to verify the following: ①Get the expected results when using valid data. ②Displays an appropriate error or warning ③Message when invalid data is used. Every business rule has been applied correctly.
Starting standard	Null
Completion criteria	All tests functioned properly and met expectations
Test focus and priority	Priority is given to the registration and check-in functions of mobile applications, as well as the data CRUD function of the web-side administrator interface, as these functions are the basic functions of the system. Second, test additional features, such as switching between interfaces and some additional features attached to the mobile app.
Special matters	This phase requires explicit internal or external factors that affect the implementation and implementation of functional testing.

6.5 User interface test

First, you need to ensure that the user interface can access the test of the corresponding data by using the object control or the portal. Secondly, it is necessary to test whether the user interface style meets the user's requirements, such as whether the interface is beautiful, intuitive, user-friendly, user-friendly, and easy to operate.

Test target	The size of the mobile terminal, the position is appropriate, and the operation logic conforms to the operating habits of most people. The objects and features (menu, size, location, status, and center) of the web-side interface window are compliant.
Test Scope	Mobile application, web-side administrator interface
Technology	Instantiate each functional interface or window and create separate tests for each function-related interface to verify that each application window and object is properly viewed and in a normal object state.
Starting standard	Each interface or window can be successfully run and displayed, responding to click events.
Completion criteria	Each interface or window is consistent with the design's baseline or meets acceptable standards.
Test focus and priority	Null
Special matters	The mobile interface needs to consider whether the interface can be rotated in the direction of the carrier machine (such as horizontal screen or vertical screen). The web interface needs to consider whether the interface can adapt to the size of the window.

6.6 Performance test

In order to test the expected performance of the current system under normal use conditions, as well as the extreme performance under extreme conditions, it is necessary to test the system's performance against response time, transaction rate and other time-related requirements.

Test target	Verify the performance of the face recognition function on the mobile side and the multi-user registration and data access functions of the web page in the following cases: ①Normal expected workload ②Expected heavy workload
Test Scope	Mobile application, web-side administrator interface
Technology	The mobile terminal simulates a scene of a large number of user face registrations by modifying the data file, wherein the face registration part automatically captures the face from the CMU Multi-PIE face database and extracts the feature vector by the script program to simulate the real extraction of the user face. Feature extraction. The web-side scripting simulates multiple users who are admins to register and access relevant management data. The script runs on a single terminal machine with a single user, a single transaction, and is repeated on multiple terminal machines (virtual or actual terminals, see [Special Considerations] below).
Starting standard	Both the mobile and web pages can respond to the basic operations of at least one administrator user.

Completion criteria	①Normal expected workload: Quick response and complete a series of specified operations during the test, the test script can be successfully completed within the expected time range of each firm without any failure. ②Expected heavy workload: Complete a series of specified operations during the test process quickly or within an acceptable timeframe, successfully completing the test script without any failure.
Test focus and priority	The priority of this part of the test should be to verify that the system can at least meet the basic operations of 1 to 5 administrator users, that is, to meet the normal expected workload.
Special matters	There are several ways to do this when adding background workloads to the server, including: ①Directly assign "transactions to the server" directly, which is usually implemented as a "structured language" (SQL) call. ②Simulate a number of (here set to 5 to 10) clients by creating a "virtual" user load. This load can be achieved through the Remote Terminal Emulation tool. This technology can also be used to load "traffic" in the network. ③Performance testing should be performed on a dedicated computer or in a dedicated machine for complete control and accurate evaluation. The database used for performance testing should be a database of actual size or the same scaling.

6.7 Load test

As a performance test, in addition to being able to test the performance behavior of the face sign-in system under different workload conditions and the ability to continue to operate normally, the load test phase can ensure that the system can still exceed the maximum expected workload. Normal operation.

Since the performance and heavy performance of the system have been tested and verified in previous performance tests, this phase only needs to test the system's operating performance under the maximum expected workload.

Test target	Verify the performance of the specified functions on the mobile and web pages under extreme workload conditions.
Test Scope	Mobile application, web-side administrator interface.
Technology	The mobile terminal simulates a scene of a large number of user face registrations by modifying the data file. The web-side scripting simulates a large number of users who are registered as administrators to access relevant management data. The script runs on a single terminal machine with a single user, a single transaction, and runs repeatedly on multiple terminal machines.
Starting standard	Both the mobile terminal and the web page can respond to the basic operations of 10 to 15 administrator users.
Completion criteria	A series of specified operations during the test was completed quickly or within an acceptable time frame, and the test script was successfully completed without any failure.
Test focus and priority	The priority of this part of the test should be to verify that the system can at least meet the basic operations of more than 15 administrator users, that is, the minimum workload

	of the simulation system.
Special matters	For possible system anomalies caused by overload, it is necessary to prepare a reasonable solution in advance to prevent the loss of important data due to flashback of the application or suspended animation of the webpage.

6.8 Strength test

Because the system relies heavily on data storage technology and involves a large number of data storage operations in the work process, it is necessary to design and execute strength tests to minimize errors caused by insufficient resources or resource contention.

Test target	Verify that the test object is working properly under the following strength conditions without any errors: ①There is little or no memory available on the server (RAM and DASD) ②Connect or simulate the largest actual (actually allowed) number of clients ③The most cumbersome transaction volume or the worst transaction combination (see the previous [Performance Test] section).
Test Scope	Mobile application, web-side administrator interface
Technology	For testing with limited resources, you should reduce or limit the RAM and DASD on the server. For strength testing, you should use multiple clients to run the same or complementary tests on the basis of [Performance Test] and [Load Test] to produce the heaviest transaction volume or the worst transaction combination.
Starting standard	Null
Completion criteria	The planned tests have all been performed, and no software failures have occurred or exceeded the specified system limits, or the system has failed conditions that are not within the specified conditions.

Test focus and priority	The system should still be able to satisfy the basic operations of a user with one or two administrators when there is little or no memory available.
Special matters	When verifying the load strength caused by a large amount of data uploaded by the mobile terminal to the server (or downloaded from the server to the mobile terminal), it is necessary to increase the network work intensity, and may need to use network tools to add messages or packet load to the network. This verifies the response processing performance of the server in the case of multiple data. The DASD for the system should be temporarily reduced to limit the growth of database free space.

6.9 Failover and recovery test

Failover and recovery testing Cocoa ensures that test subjects successfully complete the transfer and recover from various hardware and software network failures that result in unexpected data loss or data integrity breaches.

In this phase, we expect to put the application or management system under extreme simulation conditions to generate faults, then call the inspection process or recovery process to monitor and inspect the application and management system. Finally, observe whether the program or system can respond in time and prevent the process from running further, or manually check the database to verify that the application or system and data have been properly restored.

Test target	Ensure that the recovery process (either manually or automatically) properly restores the database, applications, and systems to the expected known state. The test will include the following: ①Forced exit from mobile terminal or web terminal ②Communication interruption generated by the web server ③Database pointer or keyword is invalid ④Invalid or corrupted data elements in the database
Test Scope	Mobile application, web-side administrator interface
Technology	In this phase of testing, we prepared and periodically created a series of transactions to perform or simulate the following operations: ①Forced exit of mobile terminal or web terminal: Click the back button, home button or close button when entering relevant information to achieve the purpose of simulating forced exit, and also use the task manager to force the

	process to end to simulate the possible pole. The application flashes back in special circumstances. ②Communication interruption generated by the web server simulates or initiates communication interruption of the network: The connection of the communication line is actually disconnected, such as closing the WIFI connection or the cellular data connection in the setting tab. ③The database pointer or keyword is invalid: By randomly creating some invalid fields and having the application read them. Verify that the application can detect, determine, and even process related exception pointers or keywords in a timely manner. ④The data elements in the database are invalid or corrupted: Enter the background database, manually delete some valid fields (the type of deleted fields should cover all kinds of data stored in the database), and verify whether the application can detect, judge or even process related abnormal pointers or keywords in time. Once the above situation (or simulation) is achieved, other transactions should be performed to minimize or reduce losses.
Starting standard	Null
Completion criteria	In all of the above cases, the application, database, and management system should block further operations of the process and make some emergency processing, or return to a known expected state as soon as the recovery process is complete.

Test focus and priority	The focus of this phase of the test is to determine that the relevant test object can prevent the process from performing the next step or recovering the data information until the exception occurs before the exception occurs, and at least the former can be completed.
Special matters	Null

6.10 Installation test

The purpose of this phase of testing is to ensure that mobile applications can be installed under normal and abnormal conditions.

The normal situation includes the first installation, upgrade, etc., and the abnormal situation includes the lack of directory, without granting the application corresponding permissions, etc.

Secondly, this stage also needs to verify that the software can run normally immediately after installation.

Test objective	Verify that the test object is correctly installed into the various required hardware configurations when: ①First installation：Mobile devices that have never had mobile applications installed before. ②Update：Mobile device that has previously installed the same version of a mobile application.
Test scope	Mobile application
Technology	Null
Starting standard	Null
Completion criteria	The first installation and subsequent update installation transactions were successfully executed without any failure.
Test focus and priority	Null
Special issue	Null

7. Problem severity description

This section documents some of the issues detected during development to the current stage:

Problem severity	Description	Reporter	Processor	The cause of the problem	Response time
High	Calling algorithm interface is not responding	ZhangWei	WanHongda	Since the way of loading the OpenCV visual library in Android has changed, the user's request cannot be effectively responded.	In 1 week
Medium	The mobile app opens the rear camera by default on non-millet brand phones.	ZhangWei	WanHongda	Since Google has abandoned the original Camera class in Android 5.0 and above and switched to the Camera2 class, the mobile phone device manufacturers	2 weeks

				have severely fragmented the interface.	
High	Probability loss of face feature data storage.	ZhangWei	WanHongda	When using the LitePal database to cache information, the activity does not get a valid Uid to store face information to the specified location.	In 1 day
Low	Some logic or display errors on the application interface.	ZhangJiaqing	ZhangWei	Increase judgment and modify logic.	In 3 days

Appendix A

Test strategy design sheet

Test strategy name					
Test case ID	CS_FRA-GN001				
Test tracking	“Face Recognition Attendance Test Plan” CS_FRA-GN001				
Test instruction					
Test case initialization					
Premise and constraints					
Termination condition					
Testing process					
Serial number	Input and operating instructions	Expected test results	Evaluation criteria	Actual test results	Results of the
Testers		Test date		Implementation	

Appendix B

Software problem report

Numbering： Serial number：

Report number		project name	
Problem name			
Number of problems		Software version	
Development stage	☐Project planning ☐Demand analysis ☐Detailed design ☐System implementation ☐Software test ☐Software delivery and maintenance		
Problem category	☐Design problem	☐Program problem	☐Documentation problem	☐Other problems
Problem level	☐Level 1 problem	☐Level 2 problem	☐Level 3 problem	☐Level 4 problem
Problem Description	Descriptor： Date：			

Developer comments and signatures	Signature：　　Date：