



# **The Eleventh “Citicup” Financial Innovation Competition**

## **Software Test Documentation**

**Name:** YiBangRen ——A Transparent P2P Platform for Charitable Donations

**Leader:** 汤杰强 (Software Faculty)

**Participants:** 刘畅 (Business Faculty)

李冬 (Software Faculty)

齐盼攀 (Software Faculty)

王博 (Art Faculty)

**Instructor:** 王鹏

**September, 2015**





YIBANGREN

A Transparent P2P Platform for Charitable Donations

## Revision History

Date	Version	Description	People
<10/Aug/15>	<1.0>	First draft	Project Owner and Client: Tang JieQiang Faculty Advisor: Wang Peng Project Group: Li Dong, Liu Chang, Wang Bo, Qi Panpan



# Table of Contents

1. Introduction.....	1
1.1 Purpose.....	1
1.2 Background.....	1
1.3 Scope.....	1
2. Requirements for Test.....	2
3. Test Strategy.....	3
3.1 Testing Types.....	3
3.1.1 Data and Database Integrity Testing.....	3
3.1.2 Function Testing.....	3
3.1.3 Business Cycle Testing.....	9
3.1.4 User Interface Testing.....	10
3.1.5 Performance Profiling.....	11
3.1.6 Load Testing.....	12
3.1.7 Stress Testing.....	13
3.1.8 Volume Testing.....	14
3.1.9 Security and Access Control Testing.....	15
3.1.10 Failover and Recovery Testing.....	16
3.2 Tools.....	18
4. Resources.....	19
5. Project Milestones.....	20
6. Test Summary Report.....	20
6.1 Variances.....	20
6.2 Comprehensiveness assessment.....	20
6.3 Summary of results.....	20
6.4 Evaluation.....	20



# Software Test Documentation

## 1. Introduction

### 1.1 Purpose

This Software Test Documentation for the *P2P Internet Platform for Charitable Donations* supports the following objectives:

- (1) To detail the activities required to prepare for and conduct the system test.
- (2) To communicate to all responsible parties the tasks that they are to perform and the schedule to be followed in performing the tasks.
- (3) To define the sources of the information used to prepare the plan.
- (4) To define the test tools and environment needed to conduct the system test.
- (5) To describe the test preparations, test cases, and test procedures to be used to perform qualification testing of a software item or a software system or subsystem.
- (6) To summarize the results of the designated testing activities and to provide evaluations based on these results.

### 1.2 Background

P2P Internet Platform for Charitable Donations is a platform that intended to provide a secure and reliable platform for spontaneous charitable network. The platform is responsible for receiving the requests submitted by the users and delegating them on the platform to be displayed so that the philanthropic individual can be encouraged to make some donations.

### 1.3 Scope

This test plan covers a full systems test of the corporate payroll system. This includes operator and user procedures, as well as programs and job control. In addition to comprehensively testing multiprogram functionality, external interfaces, security, recovery, and performance will also be evaluated.



## 2. Requirements for Test

The listing below identifies those items—functional requirements, and non-functional requirements—that have been identified as targets for testing. This list represents what will be tested.

### **Functional requirements to be tested:**

- Browse the project on the platform
- Search the project in a fuzzy way or by GPS
- Share the project on the social networking platforms
- Register as a registered user
- Make a donation
- Comment on the project
- Support the project in some way such as pressing “like”
- Be a authenticated user through real-name authentication
- Personal information management
- Share his/her philanthropy on the social networking platforms
- Report suspected fraud project
- Subscribe to the projects to receive project updates
- Make a request for help
- Make a apply for withdrawal
- View the progress of his/her projects
- Process the applications of reporting and complaint
- Process the applications of requesting for help
- Examine and verify the applications for withdrawal
- Examine and verify the authentication

### **Non-functional requirements to be tested:**

- Maximum bug rate: There will be a maximum of 1 bug/KLOC.
- Maximum Time To Repair: This will take less than five minutes.
- Security Considerations: The platform will ensure the privacy of users and projects.
- Response time: The maximum response time for the submission of a job will be 1 second.
- Data sensitivity: Assuming submitted statistics for information of the projects are accurate, the platform will ensure that all data in the database are completed with a 10% error allowance.



### 3. Test Strategy

#### 3.1 Testing Types

##### 3.1.1 Data and Database Integrity Testing

The databases and the database processes will be tested as a subsystem within the *P2P Internet Platform for Charitable Donations*. These subsystems should be tested without the target-of-test's User Interface as the interface to the data. Additional research into the DataBase Management System (DBMS) needs to be performed to identify the tools and techniques that may exist to support the testing identified below.

###### 3.1.1.1 Test Objective

Ensure database access methods and processes function properly and without data corruption.

###### 3.1.1.2 Technique

- Invoke each database access method and process, seeding each with valid and invalid data or requests for data.
- Inspect the database to ensure the data has been populated as intended, all database events occurred properly, or review the returned data to ensure that the correct data was retrieved for the correct reasons

###### 3.1.1.3 Completion Criteria

All database access methods and processes function as designed and without any data corruption.

###### 3.1.1.4 Special Considerations

- Testing may require a DBMS development environment or drivers to enter or modify data directly in the databases.
- Processes should be invoked manually.
- Small or minimally sized databases (limited number of records) should be used to increase the visibility of any non-acceptable events.

##### 3.1.2 Function Testing

Function testing of the target-of-test will focus on any requirements for test that can be traced directly to use cases or business functions and business rules. The goals of these tests are to verify proper data acceptance, processing, and retrieval, and the appropriate implementation of the business rules. This type of testing is based upon black box techniques; that is verifying the application and its internal processes by interacting with the application via the Graphical User Interface (GUI) and analyzing the output or results.

The following are the test cases:

No	Name	Brief description	Inputs	Expected Outputs	Actual Outputs	Pass/Fail
STD-001	Browse the list of all projects	Ordinary users browse all items of projects	no	List all items, including title of each project, name of the sponsor, amount of the money requested, contribution deadline, etc.	List all items, including title of each project, name of the sponsor, amount of the money requested, contribution deadline, etc.	Pass



No	Name	Brief description	Inputs	Expected Outputs	Actual Outputs	Pass/Fail
STD-002	Browse the detail of a specific project	Ordinary users browse the detail of a specific project	selected project id	Details of individual projects, including project type, amount of the money requested, requesting reason, project title, the recipient's name, recipient's phone numbers, the recipient's address, name of the emergency contact person, phone number of the emergency contact person, contribution deadlines, etc.	Details of individual projects, including project type, amount of the money requested, requesting reason, project title, the recipient's name, recipient's phone numbers, the recipient's address, name of the emergency contact person, phone number of the emergency contact person, contribution deadlines, etc.	Pass
STD-003	Search the project	Ordinary users search the project in a fuzzy way or by GPS	<b>Fuzzy Search:</b> Project's name or recipient's name or project's type  <b>Place search:</b> GPS location data	List all items that meet the requirements, including title of each project, name of the sponsor, amount of the money requested, contribution deadline, etc.	List all items that meet the requirements, including title of each project, name of the sponsor, amount of the money requested, contribution deadline, etc.	Pass
STD-004	Share the project	Ordinary users Share the project on the social networking platforms such as WeChat, Microblog and so on	selected project id, name of the selected social networking platform	jump to the social platform's interface and show the shared content	jump to the social platform's interface and show the shared content	Pass



No	Name	Brief description	Inputs	Expected Outputs	Actual Outputs	Pass/Fail
STD-005	user registration	Register as a registered user	the user name, password, nickname, real name, phone number,	Show message whether the user is registered successfully or not	Show message whether the user is registered successfully or not	Pass
STD-006	Make a donation	Registered users can contribute to projects through Alipay, China UnionPay and other payment platform	The selected project id, amount of money to be donated, name of the selected payment platform	Show message whether the user has donated successfully or not	Show message whether the user has donated successfully or not	Pass
STD-007	Comment on projects	Registered users can comment on projects	Selected project id, user id, comment content, time	Show message whether the user has commented successfully or not	Show message whether the user has commented successfully or not	Pass
STD-008	Support the projects	Registered users can support the project in some way such as pressing "like"	Selected project id	Show the supporting times of the selected project has been increased by 1	Show the supporting times of the selected project has been increased by 1	Pass
STD-009	Real-name authentication	Registered users become a authenticated user through real-name authentication	Submitted ID information	Show message whether the authentication is successful or not	Show message whether the authentication is successful or not	Pass





No	Name	Brief description	Inputs	Expected Outputs	Actual Outputs	Pass/Fail
STD-010	Personal information management	Registered users can view his/her personal information, donating records, supported projects, badges	User id and donating records of the user	Show the personal information, donating records, supported projects and badges and other user related information users	Show the personal information, donating records, supported projects and badges and other user related information users	Pass
STD-011	Share the philanthropy	Registered users share his/her philanthropy on the social networking platforms	User id, donating records of the use, name of the selected social networking platform	Jump to the social platform's interface and show the shared content	Jump to the social platform's interface and show the shared content	Pass
STD-012	Report Projects	Registered users report suspected fraud projects	Selected project id, reporter id, report reason	Show message whether the report is successful or not. If successful, stop the process of the project, the money not transferred will be send back	Show message whether the report is successful or not. If successful, stop the process of the project, the money not transferred will be send back	Pass
STD-013	Subscribe to the projects	Registered users subscribe to the projects to receive project updates	User id, and subscribed project id	Show the updates of the projects	Show the updates of the projects	Pass



No	Name	Brief description	Inputs	Expected Outputs	Actual Outputs	Pass/Fail
STD-014	Make a request for help	Authenticated user make a request for help	Project type, project name, project deadlines, preset amount, donation information, documents and the details of the request	The status of the project turned to "verified"	The status of the project turned to "verified"	Pass
STD-015	Make a apply for withdrawal	Authenticated user make a apply for withdrawal	The amount of withdrawal money, usage and material (such as pay slips), platform applications	Show whether the withdrawal is successful or not. System records the details of capital expenditures and generate donation reports and expenditure reports on time	Show whether the withdrawal is successful or not. System records the details of capital expenditures and generate donation reports and expenditure reports on time	Pass
STD-016	View the progress of the projects	Authenticated users view the progress of his/her projects	Project id	The comment of the project, the data statistic of donations and supports	The comment of the project, the data statistic of donations and supports	Pass
STD-017	Process the applications of reporting and complaint	Administrators process the applications of reporting and complaint	Project id, the materials submitted	Administrators respond to the reporter whether the report is successful. If successful, stop the project.	Administrators respond to the reporter whether the report is successful. If successful, stop the project.	Pass



No	Name	Brief description	Inputs	Expected Outputs	Actual Outputs	Pass/Fail
STD-018	Process the applications of requesting for help	Administrators process the applications of requesting for help	Help sponsor id, the materials submitted	If successful, a new project will be released.	If successful, a new project will be released.	Pass
STD-019	Examine and verify the applications for withdrawal	Administrators examine and verify the applications for withdrawal	Applicant's id, the materials submitted	If successful, send the money to the recipients	If successful, send the money to the recipients	Pass
STD-020	Examine and verify the authentication	Administrators examine and verify the authentication	Help Sponsor id, the materials submitted	If successful, the user types turns to authenticate user.	If successful, the user types turns to authenticate user.	Pass

Identified below is an outline of the testing recommended for each application:

#### 3.1.2.1 Test Objective

Ensure proper target-of-test functionality, including navigation, data entry, processing, and retrieval.

#### 3.1.2.2 Technique

Execute each test case or function, using valid and invalid data, to verify the following:

- The expected results occur when valid data is used.
- The appropriate error or warning messages are displayed when invalid data is used.
- Each business rule is properly applied.

#### 3.1.2.3 Completion Criteria

- All planned tests have been executed.
- All identified defects have been addressed.

#### 3.1.2.4 Special Considerations

Identify or describe those items or issues (internal or external) that impact the implementation and execution of function test



### 3.1.3 Business Cycle Testing

Business Cycle Testing will emulate the activities performed on the P2P Internet Platform for Charitable Donations over time. A period should be identified, such as one year, and transactions and activities that would occur during a year's period should be executed. This includes all daily, weekly, and monthly cycles and, events that are date-sensitive, such as ticklers.

#### 3.1.3.1 Test Objective

Ensure proper target-of-test and background processes function according to required business models and schedules.

#### 3.1.3.2 Technique

Testing will simulate several business cycles by performing the following:

- The tests used for target-of-test's function testing will be modified or enhanced to increase the number of times each function is executed to simulate several different users over a specified period.
- All time or date-sensitive functions will be executed using valid and invalid dates or time periods.
- All functions that occur on a periodic schedule will be executed or launched at the appropriate time.
- Testing will include using valid and invalid data to verify the following:
- The expected results occur when valid data is used.
- The appropriate error or warning messages are displayed when invalid data is used.
- Each business rule is properly applied.

#### 3.1.3.3 Completion Criteria

- All planned tests have been executed.
- All identified defects have been addressed.

#### 3.1.3.4 Special Considerations

- System dates and events may require special support activities
- Business model is required to identify appropriate test requirements and procedures.



### 3.1.4 User Interface Testing

User Interface (UI) testing verifies a user's interaction with the software. The goal of UI testing is to ensure that the User Interface provides the user with the appropriate access and navigation through the functions of the target-of-test. In addition, UI testing ensures that the objects within the UI function as expected and conform to corporate or industry standards.

#### 3.1.4.1 Test Objective

Verify the following:

- Navigation through the target-of-test properly reflects business functions and requirements, including window-to-window, field-to-field, and use of access methods (tab keys, mouse movements, accelerator keys)
- Window objects and characteristics, such as menus, size, position, state, and focus conform to standards.

#### 3.1.4.2 Technique

Create or modify tests for each window to verify proper navigation and object states for each application window and objects.

#### 3.1.4.3 Completion Criteria

Each window successfully verified to remain consistent with benchmark version or within acceptable standard

#### 3.1.4.4 Special Considerations

Not all properties for custom and third party objects can be accessed.



### 3.1.5 Performance Profiling

Performance profiling is a performance test in which response times, transaction rates, and other time-sensitive requirements are measured and evaluated. The goal of Performance Profiling is to verify performance requirements have been achieved. Performance profiling is implemented and executed to profile and tune a target-of-test's performance behaviors as a function of conditions such as workload or hardware configurations.

Note: Transactions below refer to “logical business transactions”. These transactions are defined as specific use cases that an actor of the system is expected to perform using the target-of-test, such as add or modify a given contract.

#### 3.1.5.1 Test Objective

Verify performance behaviors for designated transactions or business functions under the following conditions:

- normal anticipated workload
- anticipated worst case workload

#### 3.1.5.2 Technique

- Use Test Procedures developed for Function or Business Cycle Testing.
- Modify data files to increase the number of transactions or the scripts to increase the number of iterations each transaction occurs.
- Scripts should be run on one machine (best case to benchmark single user, single transaction) and be repeated with multiple clients (virtual or actual, see Special Considerations below).

#### 3.1.5.3 Completion Criteria

- Single Transaction or single user: Successful completion of the test scripts without any failures and within the expected or required time allocation per transaction.
- Multiple transactions or multiple users: Successful completion of the test scripts without any failures and within acceptable time allocation.

#### 3.1.5.4 Special Considerations

Comprehensive performance testing includes having a background workload on the server.

There are several methods that can be used to perform this, including:

- “Drive transactions” directly to the server, usually in the form of Structured Query Language (SQL) calls.
- Create “virtual” user load to simulate many clients, usually several hundred. Remote Terminal Emulation tools are used to accomplish this load. This technique can also be used to load the network with “traffic”.
- Use multiple physical clients, each running test scripts to place a load on the system.

Performance testing should be performed on a dedicated machine or at a dedicated time. This permits full control and accurate measurement.

The databases used for Performance Testing should be either actual size or scaled equally.



### 3.1.6 Load Testing

Load testing is a performance test which subjects the target-of-test to varying workloads to measure and evaluate the performance behaviors and ability of the target-of-test to continue to function properly under these different workloads. The goal of load testing is to determine and ensure that the system functions properly beyond the expected maximum workload. Additionally, load testing evaluates the performance characteristics, such as response times, transaction rates, and other time sensitive issues).

Note: Transactions below refer to “logical business transactions”. These transactions are defined as specific functions that an end user of the system is expected to perform using the application, such as add or modify a given contract.

#### 3.1.6.1 Test Objective

Verify performance behavior time for designated transactions or business cases under varying workload conditions.

#### 3.1.6.2 Technique

- Use tests developed for Function or Business Cycle Testing.
- Modify data files to increase the number of transactions or the tests to increase the number of times each transaction occurs.

#### 3.1.6.3 Completion Criteria

Multiple transactions or multiple users: Successful completion of the tests without any failures and within acceptable time allocation.

#### 3.1.6.4 Special Considerations

- Load testing should be performed on a dedicated machine or at a dedicated time. This permits full control and accurate measurement.
- The databases used for load testing should be either actual size or scaled equally.



### 3.1.7 Stress Testing

Stress testing is a type of performance test implemented and executed to find errors due to low resources or competition for resources. Low memory or disk space may reveal defects in the target-of-test that aren't apparent under normal conditions. Other defects might result from competition for shared resources like database locks or network bandwidth. Stress testing can also be used to identify the peak workload the target-of-test can handle.

Note: References to transactions below refer to logical business transactions.

#### 3.1.7.1 Test Objective

Verify that the target-of-test functions properly and without error under the following stress conditions:

- little or no memory available on the server (RAM and DASD)
- maximum actual or physically capable number of clients connected or simulated
- multiple users performing the same transactions against the same data or accounts
- worst case transaction volume or mix (see Performance Testing above).

#### 3.1.7.2 Technique

- Use tests developed for Performance Profiling or Load Testing.
- To test limited resources, tests should be run on a single machine, and RAM and DASD on server should be reduced or limited.
- For remaining stress tests, multiple clients should be used, either running the same tests or complementary tests to produce the worst-case transaction volume or mix.

#### 3.1.7.3 Completion Criteria

All planned tests are executed and specified system limits are reached or exceeded without the software failing or conditions under which system failure occurs is outside of the specified conditions.

#### 3.1.7.4 Special Considerations

- Stressing the network may require network tools to load the network with messages or packets.
- The DASD used for the system should temporarily be reduced to restrict the available space for the database to grow.
- Synchronization of the simultaneous clients accessing of the same records or data accounts.





### 3.1.8 Volume Testing

Volume Testing subjects the target-of-test to large amounts of data to determine if limits are reached that cause the software to fail. Volume Testing also identifies the continuous maximum load or volume the target-of-test can handle for a given period. For example, if the target-of-test is processing a set of database records to generate a report, a Volume Test would use a large test database and check that the software behaved normally and produced the correct report.

#### 3.1.8.1 Test Objective

Verify that the target-of-test successfully functions under the following high volume scenarios:

- Maximum (actual or physically- capable) number of clients connected, or simulated, all performing the same, worst case (performance) business function for an extended period.
- Maximum database size has been reached (actual or scaled) and multiple queries or report transactions are executed simultaneously.

#### 3.1.8.2 Technique

- Use tests developed for Performance Profiling or Load Testing.
- Multiple clients should be used, either running the same tests or complementary tests to produce the worst-case transaction volume or mix (see Stress Testing above) for an extended period.
- Maximum database size is created (actual, scaled, or filled with representative data) and multiple clients used to run queries and report transactions simultaneously for extended periods.

#### 3.1.8.3 Completion Criteria

- All planned tests have been executed and specified system limits are reached or exceeded without the software or software failing.

#### 3.1.8.4 Special Considerations

What period of time would be considered an acceptable time for high volume conditions, as noted above?



### 3.1.9 Security and Access Control Testing

Security and Access Control Testing focus on two key areas of security:

- Application-level security, including access to the Data or Business Functions
- System-level Security, including logging into or remote access to the system.

Application-level security ensures that, based upon the desired security, actors are restricted to specific functions or use cases, or are limited in the data that is available to them. For example, everyone may be permitted to enter data and create new accounts, but only managers can delete them. If there is security at the data level, testing ensures that "user type one" can see all customer information, including financial data, however, "user two" only sees the demographic data for the same client.

System-level security ensures that only those users granted access to the system are capable of accessing the applications and only through the appropriate gateways.

#### 3.1.9.1 Test Objective:

- Application-level Security: Verify that an actor can access only those functions or data for which their user type is provided permissions.
- System-level Security: Verify that only those actors with access to the system and applications are permitted to access them.

#### 3.1.9.2 Technique:

- Application-level Security: Identify and list each user type and the functions or data each type has permissions for.
  - Create tests for each user type and verify each permission by creating transactions specific to each user type.
  - Modify user type and re-run tests for same users. In each case, verify those additional functions or data are correctly available or denied.
- System-level Access: See Special Considerations below

#### 3.1.9.3 Completion Criteria

For each known actor type the appropriate function or data are available, and all transactions function as expected and run in prior Application Function tests.

#### 3.1.9.4 Special Considerations

Access to the system must be reviewed or discussed with the appropriate network or systems administrator. This testing may not be required as it may be a function of network or systems administration.



### 3.1.10 Failover and Recovery Testing

Failover and Recovery Testing ensures that the target-of-test can successfully failover and recover from a variety of hardware, software or network malfunctions with undue loss of data or data integrity.

Failover testing ensures that, for those systems that must be kept running, when a failover condition occurs, the alternate or backup systems properly “take over” for the failed system without loss of data or transactions.

Recovery testing is an antagonistic test process in which the application or system is exposed to extreme conditions, or simulated conditions, to cause a failure, such as device Input/Output (I/O) failures or invalid database pointers and keys. Recovery processes are invoked and the application or system is monitored and inspected to verify proper application, or system, and data recovery has been achieved.

#### 3.1.10.1 Test Objective

Verify that recovery processes (manual or automated) properly restore the database, applications, and system to a desired, known, state. The following types of conditions are to be included in the testing:

- power interruption to the client
- power interruption to the server
- communication interruption via network servers
- interruption, communication, or power loss to DASD and or DASD controllers
- incomplete cycles (data filter processes interrupted, data synchronization processes interrupted).
- invalid database pointer or keys
- invalid or corrupted data element in database

#### 3.1.10.2 Technique

Tests created for Function and Business Cycle testing should be used to create a series of transactions. Once the desired starting test point is reached, the following actions should be performed, or simulated, individually:

- Power interruption to the client: power the PC down.
- Power interruption to the server: simulate or initiate power down procedures for the server.
- Interruption via network servers: simulate or initiate communication loss with the network (physically disconnect communication wires or power down network servers or routers).
- Interruption, communication, or power loss to DASD and DASD controllers: simulate or physically eliminate communication with one or more DASD controllers or devices.

Once the above conditions or simulated conditions are achieved, additional transactions should be executed and upon reaching this second test point state, recovery procedures should be invoked.

Testing for incomplete cycles utilizes the same technique as described above except that the database processes themselves should be aborted or prematurely terminated.

Testing for the following conditions requires that a known database state be achieved. Several database fields, pointers, and keys should be corrupted manually and directly within the database (via database tools). Additional transactions should be executed using the tests from Application Function and Business Cycle Testing and full cycles executed.

#### 3.1.10.3 Completion Criteria

In all cases above, the application, database, and system should, upon completion of recovery procedures, return to a known, desirable state. This state includes data corruption limited to the known corrupted fields, pointers or keys, and reports indicating the processes or transactions that were not completed due to



interruptions.

#### 3.1.10.4 Special Considerations

- Recovery testing is highly intrusive. Procedures to disconnect cabling (simulating power or communication loss) may not be desirable or feasible. Alternative methods, such as diagnostic software tools may be required.
- Resources from the Systems (or Computer Operations), Database, and Networking groups are required.
- These tests should be run after hours or on an isolated machine.

#### 3.1.11 Installation Testing

Installation testing has two purposes. The first is to insure that the software can be installed under different conditions (such as a new installation, an upgrade, and a complete or custom installation) under normal and abnormal conditions. Abnormal conditions include insufficient disk space, lack of privilege to create directories, and so on. The second purpose is to verify that, once installed, the software operates correctly. This usually means running a number of the tests that were developed for Function Testing.

##### 3.1.11.1 Test Objective

Verify that the target-of-test properly installs onto each required hardware configuration under the following conditions:

- new installation, a new machine, never installed previously with *P2P Internet Platform for Charitable Donations*
- update, machine previously installed *P2P Internet Platform for Charitable Donations*, same version
- update, machine previously installed *P2P Internet Platform for Charitable Donations*, older version

##### 3.1.11.2 Technique

- *Manually or develop automated scripts, to validate the condition of the target machine—new - P2P Internet Platform for Charitable Donations never installed; P2P Internet Platform for Charitable Donations same version or older version already installed).*
- Launch or perform installation.
- Using a predetermined sub-set of function test scripts, run the transactions.

##### 3.1.11.3 Completion Criteria

*P2P Internet Platform for Charitable Donations* transactions execute successfully without failure.

##### 3.1.11.4 Special Considerations

What *P2P Internet Platform for Charitable Donations* transactions should be selected to comprise a confidence test that *P2P Internet Platform for Charitable Donations* application has been successfully installed and no major software components are missing?

#### 3.1.12 Integration Testing

Integration Testing, also known as Assembly Testing or Joint Testing. On the basis of Unit testing, all the modules are built as subsystem or system according to the design requirements, and then perform integration testing. It has been proved that the module can work individually, but it cannot be guaranteed that it can still be functioning after linking.

**3.1.12.1 Test Objective**

- Verify whether the data through the module interface will be lost when connecting modules
- Verify whether it is able to meet expectations of the parent features when all child features are combined,
- Verify whether the function of a module will have an adverse effect on another module functions
- Verify whether the accumulated error of a single module will enlarge, thus reaching unacceptable levels

**3.1.12.2 Technique**

- Combine unit testing and functional testing
- The module code in accordance with the specifications prepared

**3.1.12.3 Completion Criteria**

- The successful implementation of all integration tests in the test plan
- Fixed errors found
- Passed the test review

**3.1.12.4 Special Considerations**

None

**3.2 Tools**

The following tools will be employed for this project:

	Tool	Vendor/In-house	Version
ASQ Tool for performance testing	Test Data Generator	Vendor	
Project Management	Microsoft Project	Vendor	
DBMS tools	Database Auditor	Vendor	



#### 4. Resources

This table shows the staffing assumptions for the project.

Human Resources		
Worker	Minimum Resources Recommended (number of full-time roles allocated)	Specific Responsibilities or Comments
Test Manager, Test Project Manager	1	Provides management oversight. Responsibilities: <ul style="list-style-type: none"> <li>• provide technical direction</li> <li>• acquire appropriate resources</li> <li>• provide management reporting</li> </ul>
Test Designer	2	Identifies, prioritizes, and implements test cases. Responsibilities: <ul style="list-style-type: none"> <li>• generate Software Test Documentation</li> <li>• generate test model</li> <li>• evaluate effectiveness of test effort</li> </ul>
Tester	2	Executes the tests. Responsibilities: <ul style="list-style-type: none"> <li>• execute tests</li> <li>• log results</li> <li>• recover from errors</li> <li>• document change requests</li> </ul>
Test System Administrator	2	Ensures test environment and assets are managed and maintained. Responsibilities: <ul style="list-style-type: none"> <li>• administer test management system</li> <li>• install and manage access to test systems</li> </ul>
Database Administrator, Database Manager	2	Ensures test data (database) environment and assets are managed and maintained. Responsibilities: <ul style="list-style-type: none"> <li>• administer test data (database)</li> </ul>
Designer	3	Identifies and defines the operations, attributes, and associations of the test classes. Responsibilities: <ul style="list-style-type: none"> <li>• identifies and defines the test classes</li> </ul>



		<ul style="list-style-type: none"> <li>identifies and defines the test packages</li> </ul>
Implementer	2	<p>Implements and unit tests the test classes and test packages.</p> <p>Responsibilities:</p> <ul style="list-style-type: none"> <li>creates the test classes and packages implemented in the test model</li> </ul>

## 5. Project Milestones

Milestone Task	Effort	Start Date	End Date
Plan Test	4	2015/8/16	2015/8/17
Design Test	5	2015/8/17	2015/8/17
Implement Test	3	2015/8/18	2015/8/18
Execute Test	3	2015/8/19	2015/8/20
Evaluate Test	4	2015/8/21	2015/8/21

## 6. Test Summary Report

### 6.1 Variances

Conditions identified during testing resulted in enhancements to the set of invalid conditions described in the original functional design. This in turn resulted in the specification of 2 additional test cases. All of these changes are included in the current documentation.

### 6.2 Comprehensiveness assessment

The attached (but not included with example) checklists and execution trace reports demonstrate that the minimum comprehensiveness requirements specified in the test design specification have been satisfied.

### 6.3 Summary of results

Two of the test cases (006 and 008) exposed faults involving insufficient logic. Additional logic was added, some new test cases were defined, and the test set was rerun. All features passed their tests.

### 6.4 Evaluation

The *P2P Internet Platform for Charitable Donations* passed comprehensive testing with only two faults being detected. No more than one additional fault in the first six months of use is specified.