Touch.inc

Wi-Fi Communication platform Master Test Plan

Version 1.0

Table of Contents

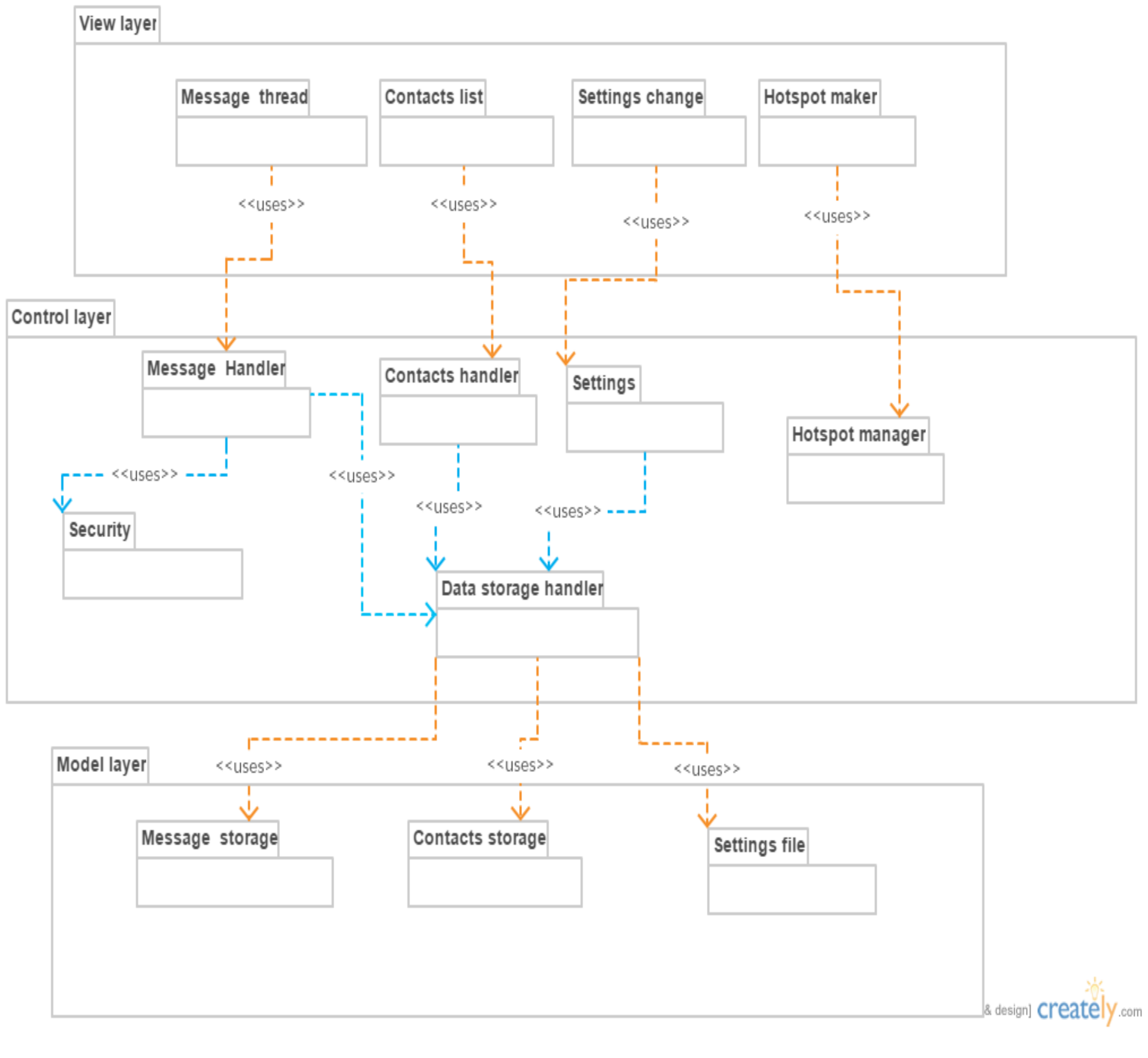
1. Evaluation Mission and Test Motivation

The mission of the testing process is to identify the development errors and logic errors in the application and solve them for a better and reliable system.

This test plan is describing the techniques that used to test the system functionalities and non-functional tasks. Those are defined here for further usage and have an idea about the how the test being done. Then it provides the details about what tests this system has passed so that this can use a measurement of how reliable the solution is.

This project is Wi-Fi Communication platform which is mainly targets file transferring and chatting. It is supposed to communicate without data charges between the network by hiding the complexity of the network.

Planed architecture of this project is here.



In this iteration evaluation effort is based under these reasons.

- find as many bugs as possible
- find important problems, assess perceived quality risks
- verify a specification (requirements, design or claims)
- advise about perceived project risks

2. Target Test Items

wi-fi device
file share part of the software
chat part of the software
control message part of the software
encoding decoding part of the data

the products are relay on Windows 7 or above operating system and dotNet 4.

3. Test Approach

This test plan has done according to these testing types.

- Data and Database integrity types
- Function Testing
- User Interface Testing
- Performance Profiling
- Load Testing
- Security and Access Control Testing
- Failover and Recovery Testing
- Configuration Testing

3.1 Testing Techniques and Types

3.1.1 Data and Database Integrity Testing

This system is not using a DBMS. It is only using a serialized data objects for data saving.

Technique Objective:	• Ensure Data access methods and processes function properly and without data corruption
Technique:	• Invoke each data 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 data events occurred properly, or review the returned data to ensure that the correct data was retrieved (for the correct reasons)
Oracles:	• Unit test in MSTest will be used Assert.AreEqual(expected, resultFfromTestFunction) will be used for to accurately check the result.
Required Tools:	• Visual studio 2017 • MSTest
Success Criteria:	• All data access methods and functions work as expected without any data corruption and pass the test cases.
Special Considerations:	• Processes should be invoked manually.

3.1.2 *Function Testing*

According to requirements this system is intended to test following main functionalities.

- Hotspot maker
- Server creation.
- Connection making process.
- Text sending.
- File sending.
- Closing application.

Technique Objective:	• Ensure proper application functioning. • Ensure Data sending without crashing or data losses as it is the main objective of the application.
Technique:	• Invoke functions with expected values and result • Testing the system in real environments because it needs the same kind of application for some testing.
Oracles:	• Unit test in MSTest will be used • Assert.AreEqual(expected, resultFfromTestFunction) will be used for to accurately check the result.
Required Tools:	• Visual studio • MSTest
Success Criteria:	• All planned tests have been executed • All test methods pass the test cases. • All identified defects have been addressed.
Special Considerations:	• Some functions need LAN connection for working

3.1.3 User Interface Testing

In the first version of the application it comes with a Windows form UI. It is expected to provide a WPF UI in future versions. The testing in here have specially target Windows Form UI

Technique Objective:	• Ensure proper execution of UI navigations. • Event handling functionality checking
Technique:	• Create / modify tests for each window to verify proper navigation. • Manual testing for each scenario in UI
Oracles:	• Check all UI pages and make sure proper UI for the application.
Required Tools:	• Visual studio • MSTest
Success Criteria:	• Each window successfully verified to remain consistent with expected behaviors.
Special Considerations:	• Required theme should be followed in the UI

3.1.4 Performance Profiling

Technique Objective:	• Determining the speed of the application processes and the utilization or resources (Ram, CPU)
Technique:	• Run application and notice the performance and utilization
Oracles:	• Using Visual studio integrated tools identify the utilization (Debugger) it can be identified accurately
Required Tools:	• Visual studio • Performance profiler • Diagnostic tools
Success Criteria:	• Memory(RAM) usage of the application is under 100mb.
Special Considerations:	• Performance can be varied with the device and the situation of testing of device.

3.1.5 Load Testing

Technique Objective:	• Verify System Response time for designated transactions or business cases under varying workload conditions.
Technique:	• Run application and notice the performance and utilization under various loads of inputs and data transfer capacities.
Oracles:	• Using Visual studio integrated tools identify the utilization (Debugger) it can be identified accurately
Required Tools:	• Visual studio • Performance profiler • Diagnostic tools
Success Criteria:	• Successful completion of the tests without any failures and within acceptable time allocation. • No crashes
Special Considerations:	• Load testing should be performed on a dedicated machine or at a dedicated time. This permits full control and accurate measurement.

3.1.6 Security and Access Control Testing

This system is not intended to provide a user login because one of its main target is simplicity. And it is a standalone application. The main concern is message encoding.

Technique Objective:	• Verify the proper encoding and decoding for sending messages through network for security purpose.
Technique:	• Invoke encoding functions and decoding functions in test cases.
Oracles:	• Unit test in MSTest will be used Assert.AreEqual(expected, resultFfromTestFunction) will be used for to accurately check the result.
Required Tools:	• Visual studio • MSTest
Success Criteria:	• pass the test case for given expected message. And encrypted decrypting process become successful.
Special Considerations:	• Control messages and user messages should be divided after decryption.

3.1.7 *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.

Here the main concern is network breakdown or disconnecting situations.

Technique Objective:	• Verify the system behaviors and develop for recover data when the network break down situations and power down situations.
Technique:	• The tests already created for Function and Business Cycle testing can be used as a basis for creating a series of transactions to support failover and recovery testing, primarily to define the tests to be run to test that recovery was successful. • Power interruption to the client: power the PC down. • Network breakdown or disconnecting • 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.
Oracles:	• Manual power off and network breakdown can be used for this with handled risk.
Required Tools:	• Visual studio debugger
Success Criteria:	• Proper communication even in power failure or network failures without data losses or application crashes.
Special Considerations:	• It should handle the risk of these breakdown by proper backup mechanism.

3.1.8 Configuration Testing

For the simplicity, this application tries it best to work the application without additional configuration.

Technique Objective:	• Validate and verify that the client Applications function properly on the prescribed client workstations.
Technique:	• Use Integration and System Test scripts • Open / close various PC applications, either as part of the test or prior to the start of the test. • Execute selected transactions to simulate user activities into and out of various PC applications. • Repeat the above process, minimizing the available conventional memory on the client.
Oracles:	• Testing the application on various machines as its normal usage can be used for accurate test.
Required Tools:	• base configuration imager and restore • installation monitoring tools (registry, hard disk, CPU, memory, and so on)
Success Criteria:	• For each combination of the Prototype and PC application, transactions are successfully completed without failure.
Special Considerations:	• The entire systems, network servers, etc. should also be documented as part of this test. • Users should be free of complex configuration setting.

4. Deliverables

Deliverables for various stakeholders are stated here.

4.1 Test Evaluation Summaries

This system has been testing from the beginning. It is done with every function adding. After completion of the system it is expected to test performance testing and failover testing.

- Testing is done according to these topics. Data and Database integrity types
- Function Testing
- User Interface Testing
- Performance Profiling
- Load Testing
- Security and Access Control Testing
- Failover and Recovery Testing
- Configuration Testing

4.2 Reporting on Test Coverage

Testing report will be included testing techniques, results and bugs that identified during the test. Also, the fixed bug details will be included. Test iteration will be released a test report including this information.

Iteration period will be normally with after every build or after new function adding.

5. Risks, Dependencies, Assumptions, and Constraints

As this system is a generic one there is no customer character in the development and testing phases. So the developer will act as the customer too.

Risk	Mitigation Strategy	Contingency (Risk is realized)
Prerequisite entry criteria is not met.	Tester will define the prerequisites that must be met before Load Testing can start. Developer will endeavor to meet prerequisites indicated by Tester in the development.	• Meet outstanding prerequisites • Consider Load Test Failure
Test data proves to be inadequate.	Developer will ensure a full set of suitable and protected test data is available. Tester will indicate what is required and will verify the suitability of test data.	• Redefine test data • Review Test Plan and modify • components (that is, scripts) • Consider Load Test Failure
Data store requires refresh.	Developer will endeavor to ensure the Database is regularly refreshed as required by Tester	• Restore data and restart • Clear Database
Network failure or hardware failures of networking.	Check testing devices and make sure the devices satisfy the required qualities for the system.	• Use quality hardware and devices