Mind-Alliance Test Plan

Author(s)	Quamar Mehmood Siddiqui
Reviewer(s)	Mahesh Kulkarni
Dev Owner(s)	Jean-Francois Cloutier

◎ Document Revision History

Date	Revision	Who	Version
May 6,2011	Initial Draft	Quamar Mehmood	1.0

Table of contents

1. Feature overview and Scope

Mind-Alliance Systems, LLC is a Roseland, New Jersey-based software and management consulting firm that helps public and private sector clients share information and manage knowledge more effectively so they can collaboratively manage risk.

Channels provides planners, consultants, and enterprise architects with the most efficient framework available for creating information sharing plans and communication procedures for homeland security, emergency preparedness, business continuity, and operational risk management. These procedures detail what information needs to be sent and received so multiple organizations or business units can operate jointly.

Current document describes the test approach and test coverage for Channels functionality.

1.1 Assumptions

- It is assumed that the application is stable and updated on regular interval.
- Show stopper issues will be fixed immediately by dev team.
- Dev team will provide release notes along with installation instructions.

1.2 Test Scope

1 Test approach to channels testing
2 Test process elements such as test cycle, release criteria
3 Functional testing of channel administrator features
4 Functional testing of plan and segment bar
5 Functional testing of task and flow panel
6 Functional testing of information flow map
7 Functional testing of all social panels and collaboration panel
8 Functional testing of channels commands
9 Functional testing of display control
10 Functional testing of entities (e.g. Profiling and organization, Defining and event etc.) of channels
11 Strategy of functional testing and nonfunctional testing (performance optimization, compatibility)
12 Test automation of automatable test cases of above test areas
13 Locking behavior of components for multi user access

2. Build Acceptance Criteria & Test Release Criteria

2.1.Build Acceptance Criteria

1. Release notification is available with the build
2. All unit test issues must be fixed before BVT execution. BVT testing is performed by test team on every build
3. Test cases for BVT are selected from existing test case repository
4. The build must be installed successfully and all the BVT test cases must pass before starting test pass execution

2.2.Release Criteria

1. All L0 and L1 test cases are passing
2. All major+ priority bugs are closed
3. All modules are at ZBB (Zero Bug Bounce) for the agreed features in the release

3. Core Test Strategy

Test cases are categorized as following:

1. L0 – BVT test cases
2. L1 – Other functional positive test case variations
3. L2 – Negative test cases and nonfunctional test cases
4. L3 – Boundary test cases
5. L4 – Other test cases (e.g. UI / UX, Browser Compatibility, etc.)

3.1.Functionality

#	Test Area	Scenarios	Verifications
1	Channel Software Installation / Upgrade / Un-installation	• Install Channel on new machine, downloaded from SVN.	• Verify that War file created successfully after running maven command. • Verify Channel Software is installed successfully. • Verify Channel installed without an error.
2		• Upgrade Channel software.	• Verify that Channel gets upgraded. • Verify that data should not be lost.
3		• Un-install Channel software.	• Verify that un-installation removes all the components of Channel.
4	Channel Login.	• Successful Channel login	• Verify that user is able to access Channel URL. • Verify user can login with valid credentials
5		• Unsuccessful Channel Login.	• Verify user cannot login with invalid credential.
6	Channel Admin Management	• Create Plan	• Verify that admin is able to create the new plan.
7		• Delete Plan	• Verify that admin is able to delete the plan.
8		• Productize Plan	• Verify that plan can be productize.
9		• Update Plan	• Verify that newly added plan should get updated

10		- Create User	- Verify that admin is able to create user.
11		- Delete User	- Verify that user should be get deleted.
12		- Assign Roles	- Verify that Roles can be assigned
13		- Disable User	- Verify that admin can make user as disable. - Verify that disable user should not able to log in.
14		- Assign user to plan	- Verify that user is able to view the plan
15		- Change role from Admin to Planner	- Verify that admin is able to change the user role. - Verify that User cannot view Channel Administration Link
16		- Assign users to all plan	- Verify that user should be assign to the entire plan. - Verify that user is able to view all plans
17		- Create Plan with same name	- Verify that plan should not be created with same name - Valid error message should be generated
18		- Create Plan with special Character	- Verify that plan should not be created with special characters - Valid error message should be generated
19		- Create User with same name	- Verify that user should not be created with same name - Valid error message should be generated
20		- Create user with special Character	- Verify that user should not be created with special characters - Valid error message should be generated
21		- Create user with invalid email id	- Verify that user with invalid email id should not be created. - Valid error message should be generated
22		- Enter valid email id in send planned feedback to	- Verify that feedback should be send
23		- Enter invalid email id in send planner feedback	- Valid error message should be generated. - Verify that feedback should not be send

24		• Create Plan name without Owner	• Verify that Plan should get created
25		• Update Owner name	• Verify that owner name is modified
26		• Delete owner name	• Verify that owner name is blank
		• Navigate to Home page	• Verify that user get navigate to Home page after clicking Home link.
27	Home Page Management	• Click on Channel Administration Link	• Verify that page with Channel Administration opens
28		• Click on Information Sharing model Link	• Verify that page with Channel :<Plan name> opens
29		• Click on Procedure Explorer Link	• Verify that page with All Procedures in <Plan name> opens
30		• Click on Procedures for all Participants Link	• Verify that page with Responder with <Plan name> opens
31		• Send Feedback	• Verify that feedback can be sent to the valid email address
32		• Send Message	• Verify that message can be send
33		• Send message to everyone	• Verify that message can be send to everyone
34		• Send message to All planners	• Verify that message can be send to all planners
35		• Delete Sent Messages	• Verify that sent messages can be deleted
36		• Email sent message	• Verify that message can be emailed
37		• Resend Email	• Verify that email can be resend
38		• Message received	• Verify that message can be received
39		• Survey -Hide completed	• Verify that survey should be hidden
40		• Survey – show all	• Verify that survey should be shown
41		• Calendar	• Verify that calendar can be seen in Channels with details
42		• All Events	• Verify that all events can be seen in calendar
43		• Print Calendar details	• Verify that calendar details can be printed
44		• Edit Calendar details	• Verify that Calendar can be edited
45		• Today date	• Verify that Today's date and

			details can be seen
46		• Learn More	• Verify that Google calendar is opened.
47	Display Controls	• Show flow map legend	• Verify that Flow Diagram legend can be seen
48		• Stretch up form	• Verify that form can be stretch up. • Verify that form can be shrink back
49		• Reduce flow map to fit	• Verify that flow map is reduced to fit within the available space • Verify clicking on the icon once more brings the flow map back to its normal size.
50		• Maximize flow map	• Verify that flow map can be maximize to use all of the screen space allocated to plan editor
51		• Minimize flow map	• Verify that flow map can be minimized. • Verify that flow map also minimize by selecting task or flow within the map
52		• Show / Hide Goals	• Verify that Goals (gains) and risks are displayed on flow map • Verify clicking again icon hides the goals.
53		• Show / Hide Info needs and capabilities	• Verify that when planner clicks on icon it display info needs and capabilities (tasks connected by arrows to square dots). • Verify that clicking again icon hides the info needs and capabilities.
54		• Hide / Show non-operational task and flow	• Verify that non-operation task and flow can be hidden and shown
55	Channel Command Execution	• Verify Undo and Redo command operation.	• Verify Do / Undo / Redo operation for following commands. 1. Add Segment (Update) 2. Remove Segment 3. Add Goal 4. Remove Goal 5. Add New Task 6. Cut Task 7. Paste Task Using Cut

			8. Paste Task Using Copy 9. Duplicate Task 10. Intermediate Task 11. Disintermediate Task 12. Move Task 13. Connect Flow 14. Update Set 15. Add Issue 16. Remove Issue 17. Attach Document 18. Copy Attachment 19. Cut Attachment 20. Paste Attachment 21. Copy Task 22. Set Task From Copy 23. Add Info Need 24. Add Info Capability 25. Add Flow 26. Duplicate Flow 27. Break Up Flow 28. Copy Flow 29. Transfer Job 30. Create Entity 31. Update Add 32. Update Move 33. Update Remove
56	Collaboration Panel	• Hide collaboration Panel	• Verify that panel can be hidden
57		• Show collaboration panel (Planner list)	• Verify that collaboration panel can be seen
58		• Presence tab	• Verify that list shows all the logged in users
59		• Show / Hide inactive users	• Verify that list shows all users (logged in as well as logged off) when user click on “Show all user” link • Verify that only logged in users are visible when user select “Hide inactive user”
60		• Send Message to inactive user from Presence tab	• Verify that message can be sent to inactive user
61		• Activities	• Verify that list of activities done by all the logged in users
62		• Hide my activities	• Verify that all activities performed by active user can be hidden

63		• Show all activities	• Verify that all activities of all users can be seen
64		• Show more activities	• Verify that list shows all activities performed by user
65		• Show few activities	• Verify that activities can be fit to page and shows recent activities
66		• Send Message to user from Activities tab	• Verify that a message can be send
67		• Send message to selected user from Message tab	• Verify that message can be send to selected user • Verify that the recipient received the message
68		• Send & Email message to selected user from message tab	• Verify that message can be send and emailed to selected user • Verify that recipient received the massage and email
69		• Send message to all planners	• Verify that message can be send to all planners • Verify that all planners should receive the message
70		• Send & Email message to all planner	• Verify that message can be send and emailed to all planners • Verify that all planners should receive the message and email
71		• Send message to everyone	• Verify that message can be send to everyone i.e. (To admin/ lanner/User) • Verify that everyone should receive the message
72		• Send & Email message to everyone	• Verify that message can be send and emailed to every • Verify that everyone received the message
73		• Hide broadcasts (sent messages)	• Verify that the broadcast message can be hidden
74		• Show all messages	• Verify that all the message are shown
75		• Delete this message (Selected message)	• Verify that message can be deleted • Verify that message can also be deleted from the recipients screen
76		• Show received messages	• Verify that received message can be seen

77		• Show sent messages	• Verify that sent messages can be seen
78		• Reset	• Verify that message text box reset to blank
79		• Email this message (Selected message)	• Verify that message can be emailed
80		• Resend email	• Verify that message can be emailed
81	Plan and Segment Bar	• Show menu	• Verify that show menu give access to collaboration panel if collapsed, about plan, about plan segment, survey, procedure report, mapped procedure, all segments, all issues, all types, index, help
82		• Action menu	• Verify that action menu enables the planner to act on the segment as a whole by-sending a message, paste task, attachment ,add issue, segment, move segment, remove segment and logout user.
83		• Help icon	• Verify that help page opens
84		• Back and Forward button	• Verify that planner can back or forward opened window
85		• Plan Segment link on plan page	• Verify that the about plan segment window opens • Verify that details can be added
86		• About plan	• Verify that window about plan opens • Verify that about plan details can be modified • Verify that About plan show menu gives access to details, all events, all organizations ,all types, all issues, index, participants, versions , who's who, bibliography, all tags, evaluations, procedure map
87		• About plan action	• Verify that action menu enables to send message and add new issue, undo, redo, paste
88		• About plan segment	• Verify that window about plan segment opens • Verify that plan segment can be

			modified • Verify that show gives access to details, goal, organizations, map, task mover
89		• About plan segment action	• Verify that action menu enables to send message and add new issue, undo, redo, paste
90		• Plan Events	• Verify that new events can be added • Verify that details for event can be added
91		• Plan Organizations	• Verify that new organization can be added • Verify that details for organization can be added
92		• Plan Procedure Map, All Segment, Who's Who, All Issue, Bibliography, All Index, All Types, All Tags, Evaluation, Participation, Versions, Action	• Verify that planner is able to access mentioned components
93		• Plan Survey	• Verify that survey window opens • Verify that planner is able to access plan survey
94		• Procedure Report	• Verify that Procedure-anyone from organization opens • Verify that planner is able to access Procedure report
95		• Mapped Procedure	• Verify that All procedure in <plan name> • Verify that planner is able to access Mapped Procedure
96		• Help form about plan	• Verify that help page opens
97		• Move task to segment	• Verify that window with task mover opens • Verify that planner can move task from one segment to other segment
99		• Add new issue	• Verify that issue can be added
100		• Add new segment to plan	• Verify that segment can be added • Verify that details can be entered • Verify that about plan segment

			have access to goals, details, organizations, map, task mover
101		• Add new task to segment	• Verify that task can be added • Verify that details of task can be entered
102		• Remove Segment from plan	• Verify that segment can be removed
103		• Logout from plan page	• Verify that user can logout from the channels
104	Task and Flow Panel	• Receives panel	• Verify that Receives panel is present
105		• Sends panel	• Verify that Send panel is present
106		• Add info receives	• Verify that info can be received • Verify that details can be entered • Verify that details can be modified
107		• Add info sent	• Verify that info can be send • Verify that details can be entered • Verify that details can be modified
108		• Create Flow	• Verify that flow can be created • Verify that flow can be modified
109		• Expand task summery	• Verify that summaries can be expanded in form
110		• Color of bullet	• Verify that the color of bullet signified the how important the task is
111	Entities	• Verify CRUD (Create / Read / Update and Delete) operation on entities of plan	• Verify CRUD operation for following Entities 1. Agent 2. Places 3. Transmission media 4. Event 5. Phase 6. Organizations 7. Network entity 8. Issue View
112	Information Flow map	• Start of segment event phase	• Verify that event is green
113		• End of segment	• Verify that event is red
114		• Information	• Verify that information can be seen on the arrow
115		• Intent-Only if task fails	• Verify that flow is dotted

116		• Sender's task-Terminates	• Verify that flow contains stop information
117	Procedure for all participants	• Procedure for all participant page	• Verify that page with Responder list for <plan name>
118		• Other planners	• Verify list of all planners who can edit this plan • Verify that email can be sent to the list of user from the outlook to the specified users
119		• Other plans	• Verify list of all the plans which logged user can edit • Verify that when click on <Plan name> the name of agent comes if available • Verify that when double clicked the agents get disabled
120		• Agent	• Verify that list of task assigned to the agent • Verify that person defined in plan can be assigned to users • Verify that person can remove the assigned user
121	Lock functionality	• Lock on Channel components	• Verify lock on following Channel components when multiple user accessing the same component 1. About Plan Details 2. About Plan Events 3. About Plan Organization 4. About Plan Segment Details 5. About Plan Segment Goal 6. Update Task 7. Update Flow 8. Update Phase 9. Update Agent 10. Update Place 11. Update Transmission Media 12. Update Organization 13. Update Role 14. Move Task

3.2.Unit Tests

TBD: Need to review code for preparing unit test cases.

3.3.Error/ Exception Handling

TBD: Need to review code for preparing scenarios and test cases.

3.4.Browser Compatibility Testing

OS / Browser	Windows 7	Unix Platform (Ubuntu Server)
Internet Explorer 8	✓	✓
Firefox 3.6	✓	✓
Google Chrome	✓	✓

LEGEND

Symbol	**Description**
✓	Fully compatible
P	Partially compatible (some features may not work)
N	Not Tested
×	Not compatible

3.5.UI / UX Testing

#	Test Area	Description
1	All Views (Channels web pages)	Ensure that all the views are covered and verified. Refer Mind-AllianceTestCaseSheet.ods or Mind-AllianceViewsCoverageSheet.ods.
2	Target page	Ensure that maximum 3 mouse click should be performed to access any page.
3	Required Fields	If the screen requires data entry on a specific field, designers should identify the required fields with a red asterisk and generate a friendly warning if the data is left blank.
4	Data Type Errors	If the screen contains dates, numeric, currency or other specific data types, ensure that only valid data can be entered.
5	Field Widths	If the screen contains text boxes that allow data entry, ensure that the width of data entered does not exceed the width of the table field (e.g. a title that is limited to 100 characters in the database should not allow more than 100 characters to be entered from the user interface).
6	Onscreen Instructions	Any screen that is not self-explanatory to the casual user should contain onscreen instructions that aid the user.
7	Keep Onscreen Instructions Brief	While onscreen instructions are great, keep the wording informative, in layman's terms, but concise.
8	Progress Bars	If the screen takes more than 5 seconds to render results, it should contain a progress bar so that the user understands the processing is continuing.
9	Same Document Opened Multiple Times	If the application opens the same document multiple times, it should append a unique number to the open document to keep one document from overwriting another. For example, if the application opens a document named Minutes.txt and it opens the same document for the same user again, consider having it append the time to the document or sequentially number it (Minutes2.txt or Minutes_032321.txt).
10	Cosmetic Inconsistencies	The screen look, feel, and design should match the other screens in your application. Creating and using a style guide is a great way to ensure consistency throughout your application.
11	Abbreviation Inconsistencies	If the screens contain abbreviations (e.g. No for number, Amt for amount, etc), the abbreviations should be consistent for all screens in your application. Again, the style guide is key for ensuring this.
12	Save Confirmations	If the screen allows changing of data without saving, it should prompt users to save if they move to another record or screen.
13	Delete	If a person deletes an item, it is a good idea to confirm the

	Confirmations	delete. However, if the user interface allows deleting several records in a row, in some cases developers should consider allowing them to ignore the confirmation as it might get frustrating to click the confirmation over and over again.
14	Type Ahead	If the user interface uses combo boxes (drop down lists), be sure to include type ahead (if there are hundreds of items in a list, users should be able to skip to the first item that begins with that letter when they type in the first letter).
15	Grammar and Spelling	Ensure the test cases look for grammar or spelling errors.
16	Table Scrolling	If the application lists information in table format and the data in the table extend past one page, the scrolling should scroll the data but leave the table headers intact.
17	Error Logging	If fatal errors occur as users use your application, ensure that the application writes those errors to a log file, event viewer, or a database table for later review. Log the routine the error was in, the person logged on, and the date/time of the error.
18	Error Messages	Ensure that error messages are informative, grammatically correct, and not condescending.
19	Shortcuts	If the application allows short cut keys (like CTRL+S to save), test each shortcut to ensure it works in all different browsers (if the application is web based).
20	Invalid Choices	Do not include instructions for choices not available at the time. For example, if a screen cannot be printed due to the state of the data, the screen should not have a Print button.
21	Invalid Menu Items	Do not show menu items that are not available for the context users are currently in.
22	Dialog Box Consistency	Use a style guide to document what choices are available for dialog boxes. Designers should not have Save/Cancel dialog on one screen and an OK/Cancel on another. This is inconsistent.
23	Screen Font Type	Ensure that the screen font family matches from screen to screen. Mismatching fonts within the same sentence and overuse of different fonts can detract from the professionalism of your software user interface.
24	Screen Font Sizes	Ensure that the screen font sizes match from screen to screen. A good user interface will have an accompanying style guide that explicitly defines the font type and size for headers, body text, footers, etc.
25	Colors	Ensure that screens do not use different color sets as to cause an inconsistent and poorly thought-out user interface design. Your style guide should define header colors, body background colors, footer colors, etc.
26	Icons	Ensure that icons are consistent throughout your application by using a common icon set. For example, a BACK link that contains

		an icon next to it should not have a different icon on one screen versus another. Avoid free clip-art icons; opt for professionally designed icons that complement the overall look and feel of your screen design.
27	Narrative Text	Having narrative text (screen instructions) is a great way to communicate how to use a specific screen. Ensure that narrative text appears at the same location on the screen on all screens.
28	Brevity	Ensure that narrative text, error messages and other instructions are presented in laymen's terms but are brief and to-the-point.
29	Dialog Box Consistency	Use a style guide to document what choices are available for dialog boxes. You should have not have Save/Cancel dialog on one screen and an OK/Cancel on another, this is inconsistent.
30	Links	If your application has links on the screen (e.g. Save as Spreadsheet, Export, Print, Email, etc.), ensure that the links have consistent spacing between them and other links, that the links appear in the same order from screen to screen, and that the color of the links are consistent.
31	Menus	If your application has menu items, ensure that menu items that are not applicable for the specific screen are disabled and the order in which each menu item appears is consistent from screen to screen.
32	Buttons	If your application has buttons (e.g. Submit, OK, Cancel, etc.), ensure that the buttons appear in a consistent order from screen to screen (e.g. Submit then Cancel).

3.6.Performance Testing / Concurrent User Test

Performance test / concurrent user testing is performed to measure / observe the Channel behavior under heavy load. Jmeter tool is used for developing the scripts and conducting test. Test is conducted with number of threads with mix of transactions as per the required percentage.

Test is conducted to make sure that following high level features should not break:

a. Lock behavior. I.e. Multiple planners should not modify the same components at the same time. If multiple planner are accessing the channels components (e.g. segment, task) then a lock should be generated on these components and only one planner can access the channel components.
b. Multiple planners should be able to access different components of the channels. E.g. If planner A is updating plan details other planner can add segments to the plan.

Refer "**MindAllianceConcurrentTestcaseSheet.ods**" file for scenarios.

3.7.Security Testing

1. Exposed interfaces of the system will be identified
2. Threat modeling of each interface will be done using STRIDE Threat Model
3. Report of security and privacy related vulnerabilities will be identified
4. Controlled attacks will be done to verify system security against the identified vulnerabilities

3.8.Build Upgrading.

Test Area	Key Verification
Upgrade major releases	• Verify that Channel gets upgraded. • Verify that core functionality is intact • Verify that one can access old plan and retrieve all the components (Users / Segments / Tasks / Organizations, etc.) and the data associated with the components (No data loss)
Upgrade minor releases	• Verify that Channel gets upgraded. • Verify that core functionality is intact • Verify that one can access old plan and retrieve all the components (Users / Segments / Tasks / Organizations, etc.) and the data associated with the components (No data loss)

4. Communication and Reporting

Document Type	What For	Who Sends and When
Daily Status Report	Daily Activity Status and Issues / Queries / Concerns	SDET 1 daily at the end of day (IST)
Weekly Status Report	Goals Achieved in Current Week Goals Planned for Next Week Issues / Queries / Concerns Reference to Test Status Report	SDET 1 at the end of every week.
Monthly Status Report	Goals achieved in current month Goals Planned for Next Month Issues / Queries / Concerns	SDET 1 at the end of month.
Test Status Report	Top 15 Defects Test Pass Statistics Defect Status Automated Test Execution Statistics	SDET 1

5. Roles and Responsibilities

Role	Responsibilities
SDET 1	Test Planning Test Design Test Elaboration and Test Execution Defect Logging and Tracking Test Results Reporting Test Case Management Test Automation Performance Test Development Performance Test Analysis Communication through Calls / Demos and Emails Daily Status and Weekly Status Reporting
SDET 2	Test Design Test Elaboration and Test Execution Defect Logging and Tracking Test Results Reporting Test Case Management Test Automation Performance Test Development

6. Defect Management

1. All defects will be logged by test team in trac.
2. Bugs will be closed by test team members after verification of issues resolved by development team members
3. All defects logged will have a test case documented in test case repository. Few exceptions will be made where test case is not needed (Such as typo error or alignment bugs)

7. Test Case Management

All documents related to QA activities are stored in SVN repository with latest changes.

Below is the list of documents:

a. UI automation test case spread sheet.
b. Concurrent user/ performance test spread sheet.
c. Automation framework design document.
d. Automation framework user guide document.

e. Channels views test coverage sheet.
f. Channels command test coverage sheet.
g. Functional test case sheet.

A test case sheet should contain following sections.

a. Test case id: Unique number or name. E.g. MAV0001_ChannelLogin
b. Test case name / Title: Test case title (feature to be tested)
c. Priority: the importance of this test case w.r.t functionality
d. Pre-condition necessary tasks to do before starting the test execution
e. Test procedure / Steps: a step by step procedure to execute this test case
f. Expected Result
g. Actual Result
h. Test case pass / fail criteria (Status): when this case is "pass", when this case is "fail"
i. Comment

8. Information and Knowledge Management

Artifacts	Repository
Test Plan	http://wush.net/trac/mindalliance/browser/TestPlan
Test Case sheet	http://wush.net/trac/mindalliance/browser/TestCases
Test Coverage sheet	http://wush.net/trac/mindalliance/browser/TestCoverage
Defects logging and tracking	http://wush.net/trac/mindalliance/newticket
Test automation scripts	http://wush.net/trac/mindalliance/browser/TestAutomation
Performance test scripts	http://wush.net/trac/mindalliance/browser/PerformanceTests
Framework design and user guide	Documents will be placed under docs folder under respective repositories

9. Test Automation

Test Area	Approach	Tools and Technologies
UI Automation	Test Cases of Views are targeted and automated to ensure sanity and robustness of UI. This provides insurance against breaking of parts of UI due to changes to the Channels. Test Coverage sheet is reviewed time to time and test cases are added and automated for new Views.	Selenium with Webdriver and core java for framework development through Eclipse IDE.

	Reference Documents: Mind-AllianceTestAutomationFrameworkDesign.pdf Mind-AllianceTestAutomationUserGuide.pdf	
Functional Test Automation	In phase 1 test cases of Plan and Command Execution targeted for test automation Test Coverage sheet is reviewed time to time and test cases are added and automated for changes to the plans and command execution functionality. In phase 2 functional test cases will be documented and automated in order of priority of test cases. This will serve as insurance against regression to the core functionality. Reference Documents: Mind-AllianceTestAutomationFrameworkDesign.pdf Mind-AllianceTestAutomationUserGuide.pdf	Selenium with Webdriver and core java for framework development through Eclipse IDE.
Performance Test Automation	In phase 1, concurrent user tests, for up to 10 simultaneous users, were automated and executed for verification of component locking behavior. In phase 2, concurrent user tests, for up to 100+ simultaneous users were automated and executed for verification of system behavior under load. Iterations of above tests will be executed iteratively on every week to ensure that system does not introduce performance bottleneck or locking issues due to functional changes in the system. Reference Documents: MindAllianceConcurrentTestcaseSheet.ods	Jmeter is used for developing scripts and conducting test.
API Test Automation	List of key APIs will be created after code walk through and signed-off. API / Unit tests will be created for the key functions. These tests will be executed every week to ensure interface lock down and consistent behavior of API across builds.	JUnit or Custom Test Harness, Core java framework development through Eclipse IDE.

10. About AFour Technologies

AFour Technologies (AFour) is an Independent Testing Services organization with a singular focus on software testing and quality engineering. The emphasis on “Testing Only” as opposed to “Testing Also” has enabled us to develop deep expertise in all aspects of quality engineering, and has given us the required agility in meeting the business and quality objectives of our customers. AFour has offices in Pune, India, and in Bellevue, USA. AFour’s niche services include Test Strategy and Consulting, Test Automation, Performance, Security and Compatibility testing in various domains.

AFour has executed engagements for various companies from start-ups to large organizations and has a sustainable customer base. Some of the areas we have deep experience in are Test Automation, Performance, Security, Compatibility and Interoperability testing. We have expertise in all major 3rd party test automation tools such as QTP, SilkTest, RFT, ALM Platforms like VSTS, and Open source test automation tools and libraries including Selenium. In the mobility space, we have developed test automation frameworks for applications on handsets as well as on the server side, using tools like M-Eux, Robotium and iMacros. In addition, we have also developed custom tools and frameworks in mobility. We also have experience working with a range of Performance Testing tools like LoadRunner, JMeter, VSTS Load Test and AppPerfect LoadTest, and in Lab set up and test infrastructure management.

AFour has a strong test team of 30 expert test engineers, each with automation experience and specialized skills in testing. With our agility and nimbleness, we have been a trusted QA partner for several high-tech startups.

11. AFour’s Differentiators

- “Expert Testing” corporate culture and mindset
- Test focused career paths enabling us to attract and retain top talent in testing
- Innovative training-based recruitment model
- Strong emphasis on Automation, Tool Development, and non-functional test attributes like Performance, Security, Reliability, Interoperability and Compatibility
- “A4Agile” Engagement Methodology
- Creative service models for test management infrastructure in the cloud