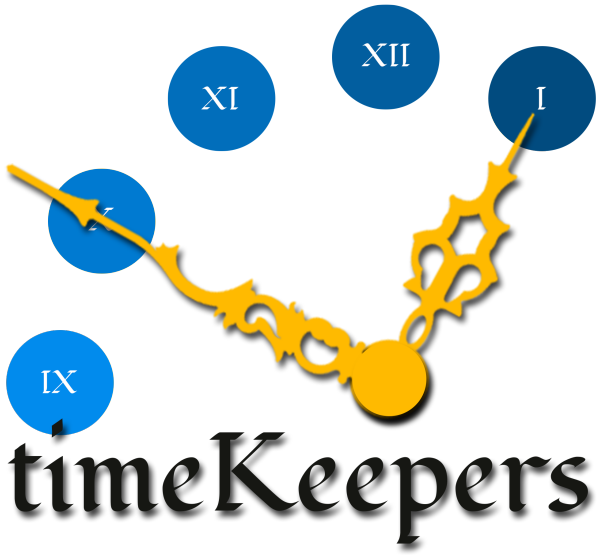
Department of Computer Science and Engineering
The University of Texas at Arlington



Team: TimeKeepers

Project: Volunteer Tracking System

Team Members:

Dineth Hettiarachchi
Damber Khadka
Devkishen Sisodia
Samir Shrestha
Tasneem Devani

Table of Contents

Document Revision History

Revision Number	Revision Date	Description	Rationale
0.1	03/31/2015	Official First Draft	First draft complete
1.0	04/02/2015	Review Ready	Validated the consistency and the formatting of the document; the Draft is ready for review
1.1	04/07/2015	Peer Review Changes	Made corrections based on the feedback received from team Ground Control.
2.0	04/09/2015	Baseline Version	Added remaining test cases and fixed the formatting of the document

List of Figures

List of Tables

1. Introduction

1.1 Test Plan Overview

The System Test Plan document will provide the detail description of various testing procedures incorporated by the team to ensure that Maverick Volunteer Tracking System meets the requirements listed in System Requirement Specification and acceptance criteria set forth by team and customer as well as to preserve the quality of all components of the product. The Test Plan will also make references to previous documents- System Requirement Specification, Architecture Design Specification, and Detailed Design Specification to show the life cycle of development process. The document will also include test items, risks, features to be tested, features to be not tested, overall test strategy, acceptance criteria, test deliverables, test schedules, and approvals.

1.2 Product Concept

In the College of Engineering, there is currently an organization, Maverick Volunteers, which allow the various members of the Board of Advisors to volunteer and participate in different service opportunities. An administrator manually maintains the current system. The Maverick Volunteer Tracking System seeks to solve this problem. The main purpose of this project is to provide a system to the volunteers in the College of Engineering Board of Advisors to input, track and analyze their volunteer activities.

In addition to the website, an Android based mobile app will be developed. The purpose of this app is to provide an ease of access to the Volunteer Tracking System. The app will allow the volunteers to access the same functionality as the website. However, the functionality of the Admin and Facilitators is limited. To access their unique functionality, an Admin or a Facilitator would need to directly access the website. The functionality on the app is limited as the TimeKeepers will be primarily focused on the website.

In the future, the Volunteer Tracking System may be open to students, faculty and staff from the College of Engineering or other departments around the campus.

1.3 Product Scope

The TimeKeepers are designing a website and an Android application that will provide an efficient and interactive way for the Maverick Volunteers to log their volunteer hours as well as to keep track of their volunteer activities. The volunteers will be able to keep themselves updated about the upcoming volunteer opportunities and periodically view their progress report. The system will also provide a means to the facilitators to track volunteer participation and use it as a means to determine strategy for increasing volunteer participation.

Based on our current analysis, the system will have three levels of users, which include admin, facilitator and volunteers. The facilitators will be able to add new events to inform the volunteers and the volunteers will be able to log their time, signup for an event and view their progress. The admin will be able to manage all the users as well as make changes to the system content.

The Volunteer Tracking System can be easily accessed under the uta.edu/engineering/ webpage or through any phone with android version 4.1.2 or above. The app will be available to download from the Google Play Store. Once the app is downloaded onto the phone, the user will start it for the first time. This will require them to enter their Email and password used to create the account. Once they have been validated, the internal database syncs and receives the data from the external database to display the necessary information to the user.

1.4 Test Scope

The System Test Plan is necessary to ensure product design and implementation meets the product as specified in the System Requirements Specification and the Detail Design Specification. It will be used to validate and verify the status of the prototype and its components with respect to whether it is working as expected, it doesn't work or it works with major/minor issues. Therefore, we will be utilizing four different types of testing: unit testing, component testing, integration testing and system validation. The testing environment with respect to the different tests involved in also crucial. Many tests require visual inspection and therefore the team will use different browsers to test the web application and various Android devices to test the Android application.

1.4.1 Unit Testing

Ensures that the lowest level modules explained in the Detailed Design Specification work individually.

1.4.2 Component Testing

Ensures that all the modules work at the subsystem level.

1.4.3 Integration Testing

Ensures that the modules, subsystems work together after the integration.

1.4.4 System Validation

Ensures that the system satisfies requirements and acceptance criteria specified and works with minimal issues.

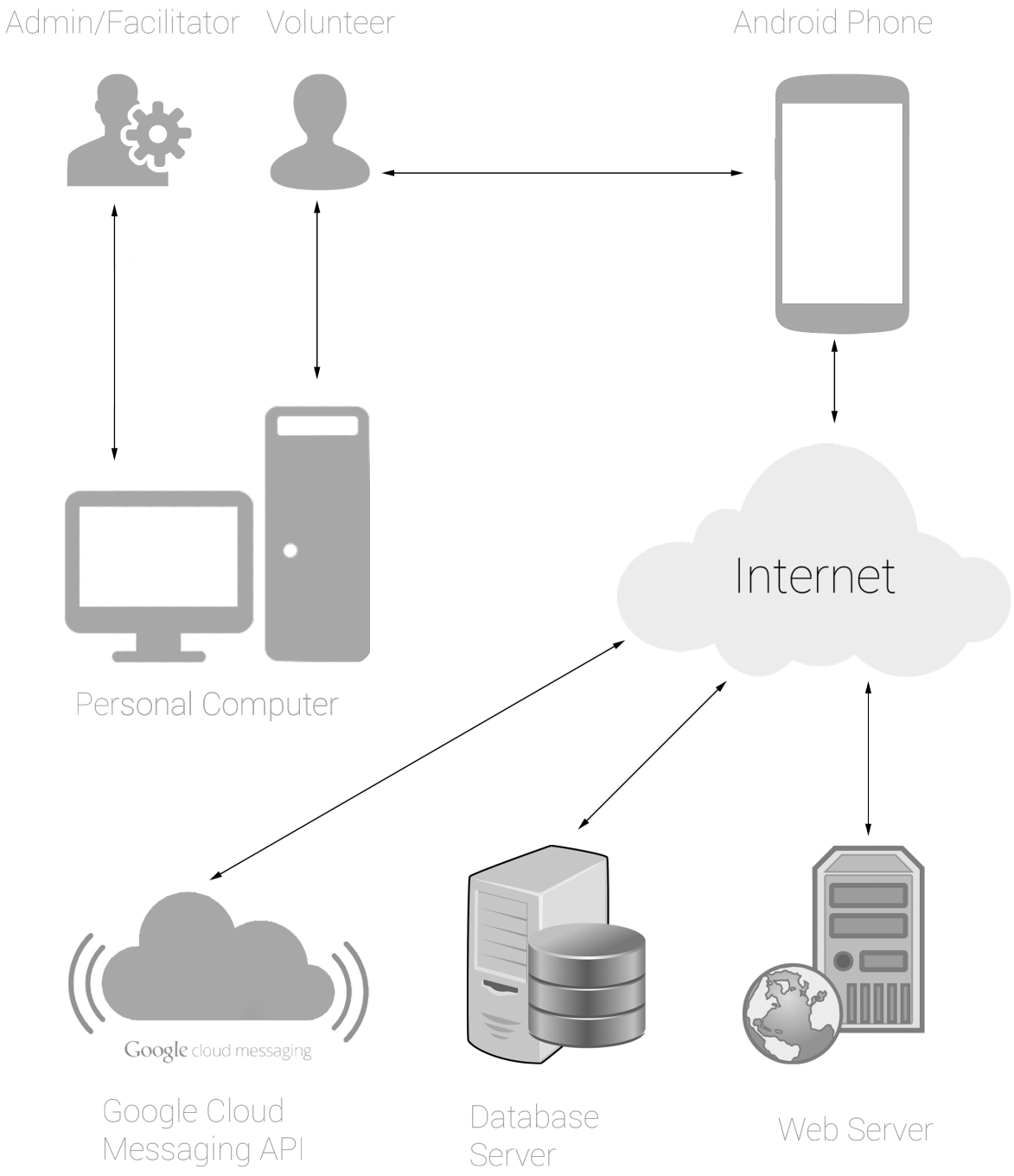


Figure 1.1 High Level System Diagram

2. Test References

This System Test Plan will incorporate previously established documents, namely the System Requirement Specification, Architecture Design Specification, and the Detail Design Specification. These documents form the basis for the testing strategy presented in the document.

2.1 System Requirement Specification

This section lists the requirements listed in the System Requirement Specification. It outlines all the requirements that have been determined by the members of TimeKeepers and the project sponsor, Dr. Linda McCalla. The requirements that will be consider for testing purpose are customer requirements, packaging requirements, performance requirements, safety requirements, and support and maintenance requirements. All these requirements are listed below.

2.1.1 Customer Requirements

Req. No	Req. Title	Description	Priority
3.1	Input Volunteer Hours	The Volunteer Tracking System shall allow a user to input the hours volunteered. To input the hours, the volunteers shall be able to select the name of the opportunity from a dropdown associated with a category and enter the number of hours they volunteered along with comments.	Critical
3.2	Notify Admin	The Volunteer Tracking System shall also notify the admin and the opportunity category facilitator when members input their time volunteered.	High
3.3	Input Hours on Behalf of Volunteers	Upon the request of the volunteer, facilitators must be able to input the volunteer hours on behalf of the volunteer. The facilitators shall be able to see a list of all members and an option to input their volunteer hours. The facilitator will have access to input the volunteer hours of all members without any time limitations or constraints.	Critical
3.4	Add Volunteer Opportunities	The Volunteer Tracking System shall allow facilitators to input the new or upcoming volunteer opportunities. An opportunity may include a title, description, date and time, location and images.	High

3.5	Delete Volunteer Opportunities	The Volunteer Tracking System shall allow facilitators to delete volunteer opportunities previously entered into the System. If volunteers have committed to an opportunity and it is cancelled, the system will notify all volunteers through Email.	High
3.6	Sign Up for Volunteer Opportunities	The volunteers shall be able to see the details of an opportunity such as the date, time, and location and have an option to sign up for an opportunity to indicate they will be volunteering at that opportunity.	High
3.7	Cancel Commitment	The volunteers shall be able to cancel a commitment they previously made. If volunteers previously signed up for an opportunity, the system shall allow them to cancel their commitment to indicate they will no longer be volunteering at that opportunity.	High
3.8	Notify Volunteer	The Volunteer Tracking System shall notify the volunteer and the opportunity facilitator upon the volunteer's acceptance/commitment or cancellation of an opportunity. This notification will be system generated. The volunteer and the facilitator will be able to see this notification on their home page.	High
3.9	Track Progresses	The Volunteer Tracking System shall allow users to track progress of their volunteer activities and the status of different service levels. Service levels are different levels that volunteers can achieve based on the total number of hours. The levels are divided as follows: 30, 60, 90, 150, and 150+.	Critical
3.10	Generate Reports	The Volunteer Tracking System shall generate progress reports for each volunteer upon their request. The progress report should include details such as the categories/types of opportunities volunteered in, and the total number of hours volunteered.	High
3.11	Manage Reports	The Volunteer Tracking System shall allow admin to send progress reports along with comments and attachments to the specified users.	Moderate
3.12	Promote Members	The Volunteer Tracking System shall allow admin to designate or promote a member to a facilitator.	Moderate

3.13	Demote Facilitators	The Volunteer Tracking System shall allow admin to demote a facilitator to a member.	Moderate
3.14	Customize Preferences	The Volunteer Tracking System shall allow volunteers to customize their preferences. Preferences include setting the date of availability along with level of interest in different opportunity categories.	High
3.15	Login	The Volunteer Tracking System shall allow users to login with their Email and password. When a user logs in to the system for the first time, the system shall allow them to enter their Email for validation. When the Email is validated, the system shall ask the user to establish their password. When a user logs in to the system again, they will be required to enter their Email and Password for validation.	Critical
3.16	Logout	The Volunteer Tracking System shall allow volunteers to logout of the system. When the user is logged out, the system shall redirect to the login page.	Critical
3.18	Register Volunteers	The Volunteer Tracking System shall allow Admin to register volunteers and allow access into the system.	Critical
3.22	Ease of Use	The Volunteer Tracking System shall provide a user-friendly interface. The system shall also limit the number of clicks to allow a user to reach their desired page easily.	High
3.23	Android Application	The Volunteer Tracking System shall be available in the form of an Android Application. The Application will be available in the Google Play Store to download for free.	Low

Table 2.1 Customer Requirements

2.1.2 Packaging Requirements

Req. No	Req. Title	Description	Priority
4.1	Website URL	Website will be hosted under a subdirectory of http://www.uta.edu.	Critical
4.2	Page URLs	Website URLs will be human readable and search engine friendly.	Moderate

4.3	Google Play Publication	Android app will be released into Google Play as a free download.	High
4.4	Installation Script	A PHP installation script that will populate the necessary database tables shall be provided.	Low

Table 2.2 Packaging Requirements

2.1.3 Performance Requirements

Req. No	Req. Title	Description	Priority
5.1	Application Response Time	Response time between user interaction and result should be less than 8 seconds in both the website and the Android app.	Moderate
5.2	Dynamic Page Update	Only the necessary parts of the web page will be updated upon the user interaction instead reloading the page completely.	Moderate
5.3	File Compression	JavaScript and CSS files will be compressed to reduce the file size. Size of JPEG images should be less than 3 MB.	Low
5.4	Third Party Libraries and Frameworks	JavaScript and CSS libraries will be directly accessed from the CDN servers, thereby, improving the access time in distant locations.	Low
5.5	Serve Scaled Image	Differently scaled images will be used in different scenarios. i.e. thumbnails, full-screen images	Low

Table 2.3 Performance Requirements

2.1.4 Security and Privacy Requirement

Req. No	Req. Title	Description	Priority
7.1	Website Cache	Age of the website cache will be restricted to 7 days.	Low
6.2	Password Encryption	All user password shall be encrypted in the MySQL database.	Critical
6.3	Malicious Input Protection	System shall validate all the input data to ensure that the entered data is correct and/or user has not entered any malicious code in any input fields.	Moderate

Table 2.4 Security and Privacy Requirements

2.1.5 Maintenance and Support Requirements

Req. No	Req. Title	Description	Priority
7.1	Source Code Documentation/Availability	All the documentation prepared by team TimeKeepers including System Requirements Specification, Architectural Design Specification, Detail Design Specification, and System Testing Plan will be made available to future senior design students. The source code shall be well documented with comments and details about functionality. The code shall help anyone who want to further develop this product in future.	Moderate
7.2	Password Encryption	The team TimeKeepers shall not be responsible to maintain the system or source code after completion of project. The College of Engineering website Developer, Christopher Woods, will continue to maintain the website as it will be hosted under uta.edu/engineering.	Critical
7.3	PHP Version Support	The UTA servers are running PHP version 5.1. Therefore, the web application shall be compatible with PHP version 5.1.	Critical
7.4	Android Version Support	The mobile version of the system will be Android based. The application will support a minimum API level of 16, which corresponds to version 4.1.2 (Jelly Bean).	Low
7.5	User Manual	The team will provide user manual that describes the different functionality of product and instructions on how to use product. This user manual shall support system administrator for any problems in future.	Moderate
7.6	Training	The team shall provide training to system manager on how to use and manage the system. The team will demo the product upon completion and explain functionality of the system that shall help manager to understand system better.	Moderate

Table 2.5 Maintenance and Support Requirements

2.1.6 Others Requirements

Req. No	Req. Title	Description	Priority
8.1	Web Browser Compatibility	The web interface shall be accessible via various popular browsers such as Safari, Google Chrome, Mozilla Firefox, and Internet Explorer.	Moderate
8.2	Web Source Compatibility	All The source code of the web functionality shall be compatible and portable with various platforms such as Windows, Mac, and Linux.	Moderate
8.4	Responsive Design	The website shall reflow its layout to fit in for the screen resolution or the window size.	Future
8.5	Testing	The features and functionality of Volunteer Tracking System will be thoroughly tested with all requirements and acceptance criteria before handing system to the customers.	Critical

Table 2.6 Other Requirements

2.2 Architecture Design Specification

The Architectural Design Specification (ADS) documents the design of Maverick Volunteer Tracking System and the Android app. ADS provides a concept of the website and the Android app and provides a high-level overview and interaction between each layer and their subsystems. The test plan is designed to verify each component and the data flows between them. In order to ensure that the system functions properly, and that it is producing acceptable results, the team must test these components to verify that each one is working as it should, and that the interactions between the components are behaving correctly.

2.2.1 Architecture Design Diagram

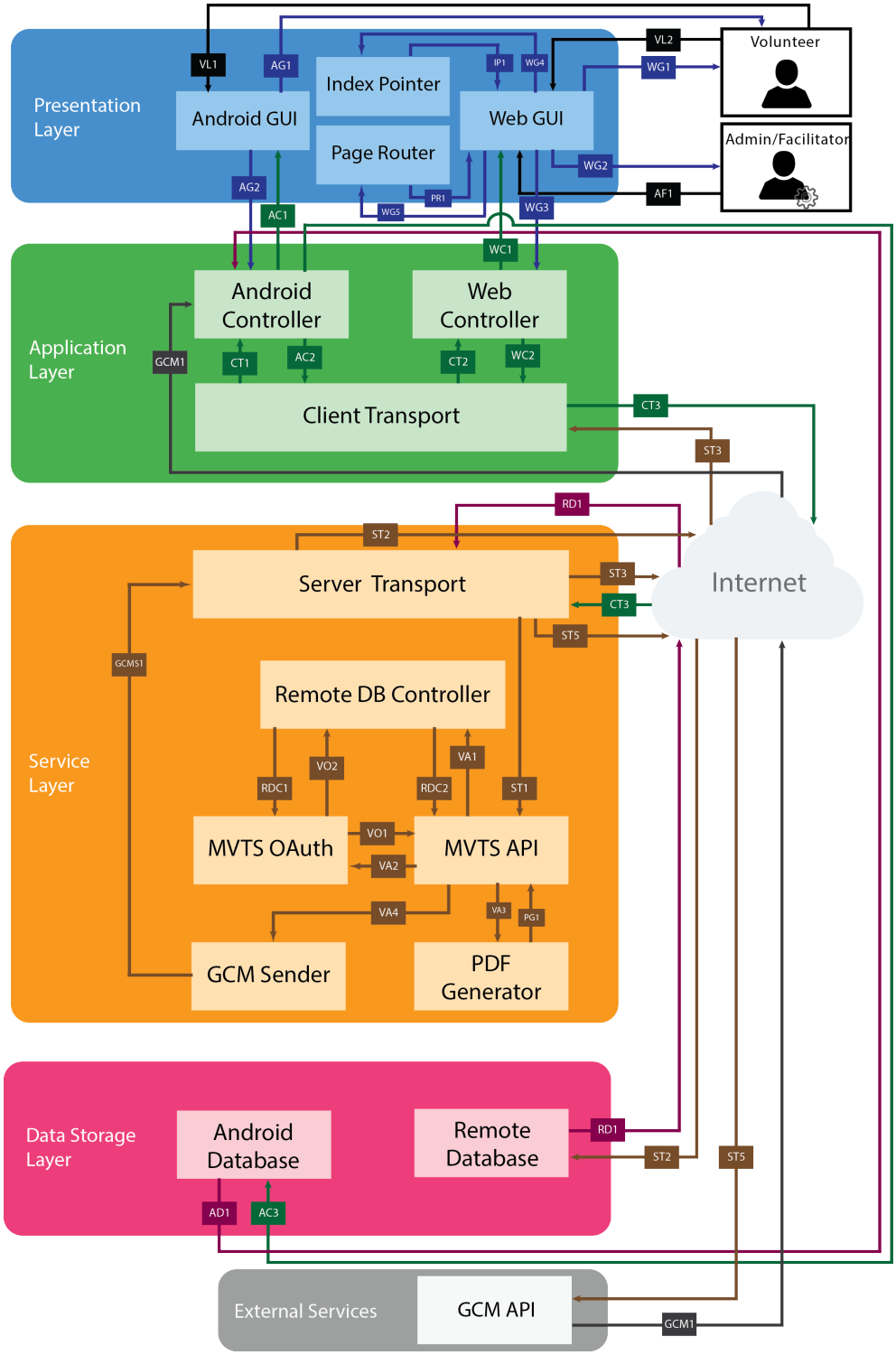


Figure 2.2 Architecture Diagram

2.2.2 Data Flow Definition

Data Element	Description	Source	Sink
AC1	The Android Controller will relay what needs to be displayed to the user in the Android GUI	Android Controller	Android GUI
AC2	The Android Controller will format and send user input data to the Client Transport which will then be processed by the Service Layer	Android Controller	Client Transport
AC3	The Android Controller will input formatted user input data into the Android Database	Android Controller	Android Database
AD1	The Android Controller will read data from the Android Database	Android Database	Android Controller
AF1	The Admin/Facilitator will input data into the Web GUI	Admin/Facilitator	Web GUI
AG1	The Android GUI will display information to the Volunteer	Android GUI	Volunteer
AG2	The Android GUI will relay user input data to the Android Controller	Android GUI	Android Controller
CT1	The Client Transport will send data to the Android Controller from the Service Layer	Client Transport	Android Controller
CT2	The Client Transport will send data to the Web Controller from the Service Layer	Client Transport	Web Controller
CT3	The Client Transport will send data from either the Android Controller or Web Controller to the Server Transport	Client Transport	Server Transport
GCM1	The GCM API will send an alert to the Android Controller notifying it of a change in the Remote Database	GCM API	Android Controller
GCMS1	The GCM Sender sends an alert to the GCM API via the Server Transport notifying it of a change in the Remote Database	GCM Sender	Server Transport
IP1	Depending on the user input data (user request) received from the Web GUI, the Index Pointer will redirect the user request or process the request	Index Pointer	Web GUI

PG1	The PDF Generator sends the report, in PDF format, to the MVTS API	PDF Generator	MVTS API
PR1	The Page Router will return the relevant HTML content to the Web GUI.	Page Router	Web GUI
RD1	The Remote Database Controller will read data from the Remote Database via the Server Transport	Remote Database	Server Transport
RDC1	The Remote Database Controller will send formatted data to the MVTS O-Auth for user authentication	Remote Database Controller	MVTS O-Auth
RDC2	The Remote Database Controller will send formatted data to the MVTS API for further processing	Remote Database Controller	MVTS API
RDC3	Data read from and inputted into the Remote Database by the Remote Database Controller will travel via the Server Transport	Remote Database Controller	Server Transport
ST1	The MVTS API receives user request data from the Server Transport	Server Transport	MVTS API
ST2	Data sent to the Remote Database from the Service Layer will travel via the Server Transport	Server Transport	Remote Database
ST3	The Application Layer and Service Layer will send and receive data to and from each other via the Server and Client Transports	Server Transport	Client Transport
ST5	The Server Transport sends the alert created by the GCM Sender to the GCM API (see GCMS1)	Server Transport	GCM API
ST6	Data sent to the Remote Database Controller from the Remote Database will travel via the Server Transport	Server Transport	Remote Database Controller
VA1	The MVTS API sends data or requests to read data to and from the Remote Database Controller	MVTS API	Remote Database Controller
VA2	MVTS API sends data or requests to read data to and from the Remote Database Controller	MVTS API	MVTS O-Auth
VA3	The MVTS API sends the necessary data to the PDF Generator in order to generate a report	MVTS API	PDF Generator

VA4	The MVTS API notifies the GCM Sender to send an alert to the GCM API whenever an update in the Remote Database should affect the Android Database	MVTS API	GCM Sender
VL1	The Volunteer will input data (including touch data) into the Android GUI	Volunteer	Android GUI
VL2	The Volunteer will input data into the Web GUI	Volunteer	Web GUI
VO1	The MVTS O-Auth Subsystem provides information back to the MVTS API on the state of validation	MVTS O-Auth	MVTS API
VO2	The MVTS O-Auth Subsystem checks user's credentials with credentials stored in the Remote Database via the Remote Database Controller	MVTS O-Auth	Remote Database Controller
WC1	The Web Controller will relay what needs to be displayed to the user in the Web GUI	Web Controller	Web GUI
WC2	The Web Controller will format and send user input data to the Client Transport which will then be processed by the Service Layer	Web Controller	Client Transport
WG1	The Web GUI will display information to the Volunteer	Web GUI	Volunteer
WG2	The Web GUI will display information to the Admin/Facilitator	Web GUI	Admin/Facilitator
WG3	The Web GUI will send user request and user input data to the Web Controller for processing	Web GUI	Web Controller
WG4	The Web GUI will send user request and user input data to the Index Pointer which will either process the request or redirect the request	Web GUI	Index Pointer
WG5	Based on the user request from the Web GUI, the Page Router determines if there is a URL change (and injects the requested page content back into the Web GUI)	Web GUI	Page Router

Table 2.7 Data Flow Definition

2.2.3 Presentation Layer

The Presentation Layer consists of the Web App GUI, Android App GUI. This layer is responsible for gathering input from the user and displaying processed information back to the user. The Web App GUI consists of all the website GUI's that will allow the user to interact with the system through their computer. Similarly, the Android App GUI consists of all the GUI's that will allow the user to interact with the system through their Android smart phone. The layer below the presentation layer is Application Layer. The Presentation Layer will pass on the raw data collected from the user as input to the Application Layer. Depending on the task, the Application Layer will pass the processed data back to the Presentation Layer, which will then be able to display the information back to the user either through the Web GUI or the Android GUI.

2.2.4 Application Layer

The Application Layer is the next layer in our hierarchy of layers. This layer is composed of Android Controller, Web Controller, and Client Transport subsystems. The Application Layer communicates with the presentation layer to get the user input and events associated with user actions. Depending on the nature of the input, this layer either needs to request information from the database or store data in the database through the Service Layer. The Android Controller will be responsible for handling all the events generated by the user through the Android GUI whereas the Web Controller will be responsible for handling all the events generated by the user through the Web GUI. The Controller will then pass the data collected to the Service Layer through the Client Transport subsystem. The Client Transport acts like a bridge between the Application Layer and the Service Layer.

2.2.5 Service Layer

The Service Layer is the next layer in our hierarchy of layers. The Service Layer is composed of the Server Transport, the VTS API, VTS OAuth subsystems. This layer is responsible for doing all the server side processing and direct database access. The Application Layer transfers the data to the server layer through the server transport subsystem which then uses the VTS OAuth subsystem to verify the session before doing any processing. After the session has been verified the VTS API which houses all the protocols for data input and retrieval will access the database through the Server Transport. The Server Transport acts like the main communication line between the client and the server side to either pass data to the VTS API and VTS OAuth or to send the JSON response back to the Application Layer.

2.2.6 Data Storage Layer

The next layer in our layer hierarchy is the Data Storage Layer. This layer consists of the Android and remote databases. The Application Layer communicates with the Data Storage Layer through the Service Layer.

The web application will only communicate with the remote database that will be implemented as a relational MySQL database. However, the Android application will communicate with both the Android database that will be implemented as relational SQLite database and the remote MySQL database.

2.3 Detailed Design Specification

This Detailed Design Specification (DDS) documents the different modules for the Maverick Volunteer Tracking System and the Android app. These modules describe the finer details of how the system will be implemented. This document will divide each subsystem from each layer into modules and describe the details of how these modules interface with each other or with external services. Because these modules are the smallest components of the system, it is crucial that each module is tested to ensure that they are functioning properly. The larger components are built from the modules. Therefore, it is impossible to ensure that the larger components of the system work without first proving that the modules that make them up work.

.3.1 Detailed Design Diagram

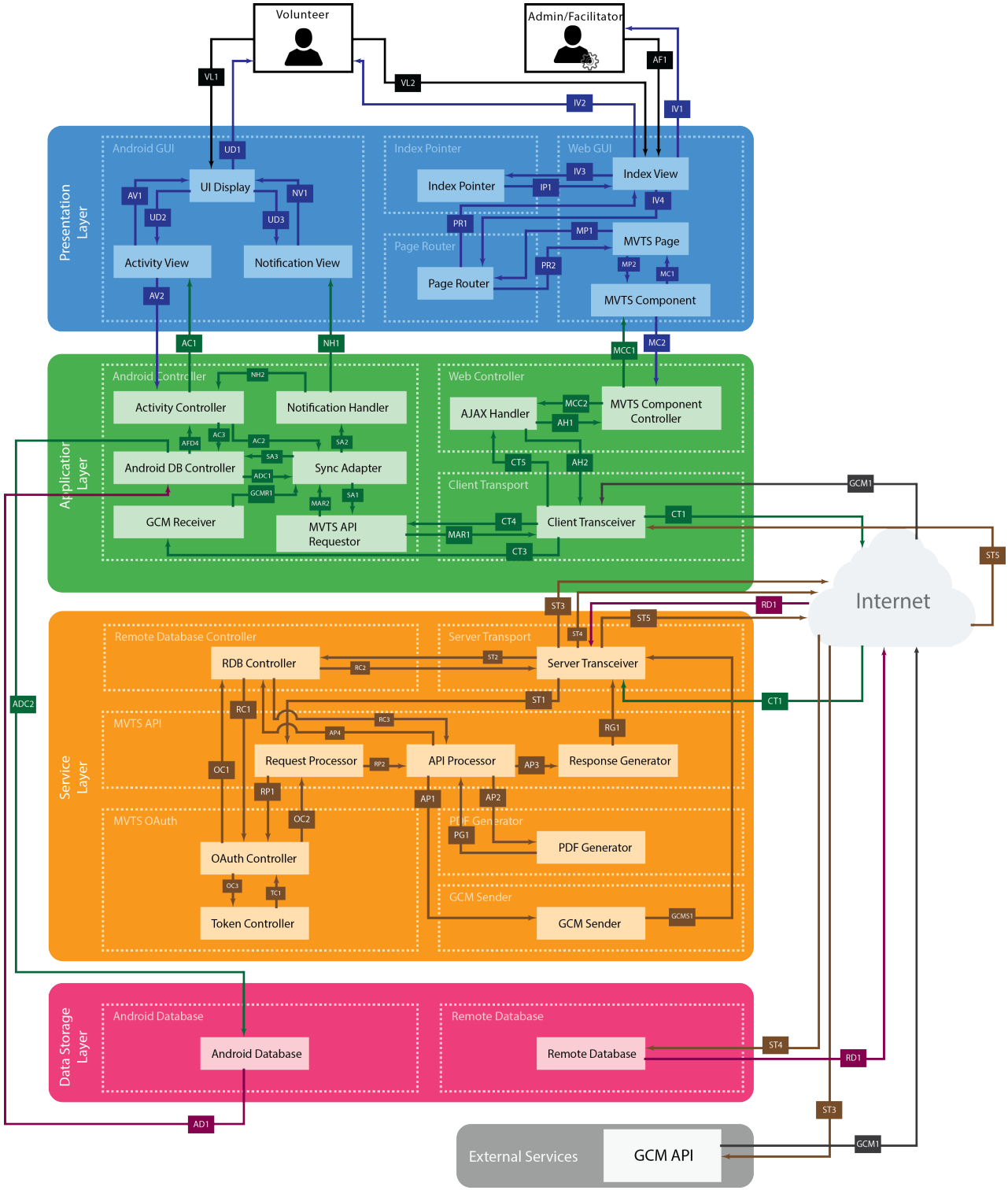


Figure 2.3 Module Decomposition Diagram

2.3.2 Producer-Consumer matrix

	Volunteer	Admin/Facilitator	UI display	Activity View	Notification View	Index Pointer	Index View	Page Router	MVTS Page	MVTS Component	Activity Controller	Notification Handler	Android DB Controller	GCM Receiver	Sync Adapter	MVTS API Requestor	AJAX Handler	MVTS Component Controller	Client Transceiver	Server Transceiver	RDB Controller	Request Processor	API Processor	Response Generator	OAuth Controller	Token Controller	PDF Generator	GCM Sender	Android Database	Remote Database	GCM API
Volunteer			VL1				VL2																								
Admin/Facilitator							AF1																								
UI display	UD1			UD2	UD3																										
Activity View	AV1										AV2																				
Notification View	NV1											NV2																			
Index Pointer							IP1																								
Index View	IV2	IV1				IV3																									
Page Router						PR1			PR2																						
MVTS Page							MP1			MP2																					
MVTS Component									MC1									MC2													
Activity Controller				AC1									AC3		AC2																
Notification Handler					NH1																										
Android DB Controller											AFD4				ADC1														ADC2		
GCM Receiver															GCMR1																
Sync Adapter												SA2	SA3			SA1															
MVTS API Requestor															MAR2				MAR1												
AJAX Handler																		AH	AH												

MVTS Component Controller										MCC1								MCC2													
Client Transceiver														CT3		CT4	CT5			CT1											
Server Transceiver																			ST5		ST2	ST1							ST4	ST3
RDB Controller																				RC2					RC1						
Request Processor																							RP2		RP1						
API Processor																								APS3			APS2	APS1			
Response Generator																				RG1											
OAuth Controller																					OC1	OC2				OC3					
Token Controller																									TC1						
PDF Generator																							PG1								
GCM Sender																				GCMS1											
Android Database													AD1																		
Remote Database																				RD1											
GCM API																			GCM1											

Table 2.8 Producer Consumer Matrix

2.3.3 Requirement Traceability Matrix

2.3.3.1 Presentation Layer

Req. No.	Requirement Name	Module							
		UI display	Activity View	Notification View	Index Pointer	Page Router	Index View	MVTS Page	MVTS Component
3.1	Input Volunteer Hours				X		X	X	X
3.2	Notify Admin			X				X	
3.3	Input Volunteer Hours on behalf of user				X		X	X	X
3.4	Add Volunteer Opportunities				X		X	X	X
3.5	Delete Volunteer Opportunities				X		X	X	
3.6	Sign-Up for Volunteer Opportunities				X	X	X	X	X
3.7	Cancel Commitment				X		X	X	
3.8	Notify Volunteer			X				X	
3.9	Track Progress				X	X	X	X	
3.10	Generate Reports				X	X	X	X	X
3.14	Customize Preferences							X	
3.15	Login				X	X	X		X
3.16	Logout				X	X	X		
3.18	Register Volunteers				X		X		X
3.23	Android Application	X	X	X					

Table 2.9 Presentation Layer Modules Traceability Matrix

2.3.3.2 Application Layer

Req. No.	Requirement Name	Modules							
		Activity Controller	Android DB Controller	GCM Receiver	Notification Handler	Sync Adapter	MVTS API Requestor	AJAX Handler	MVTS Component Controller
3.1	Input Volunteer Hours			X		X	X	X	X
3.2	Notify Admin			X	X				
3.3	Input Volunteer Hours on behalf of user			X		X	X	X	X
3.4	Add Volunteer Opportunities						X	X	X
3.5	Delete Volunteer Opportunities			X		X	X	X	X
3.6	Sign-Up for Volunteer Opportunities			X			X	X	X
3.7	Cancel Commitment			X		X	X	X	X
3.8	Notify Volunteer	X		X	X				
3.9	Track Progress						X	X	X
3.10	Generate Reports						X	X	X
3.14	Customize Preferences						X	X	X
3.15	Login	X	X				X	X	X
3.16	Logout	X	X				X	X	X
3.18	Register Volunteers					X	X	X	X
3.23	Android Application	X	X			X	X	X	

Table 2.10 Application Layer Modules Traceability Matrix

2.3.3.3 Service Layer & Data Storage Layer

Req No.	Requirement Name	Modules									
		RDB Controller	Request Processor	API Processor	Response Generator	OAuth Controller	Token Controller	PDF Generator	GCM Sender	Android Database	Remote Database
3.1	Input Volunteer Hours	X	X	X		X	X		X	X	X
3.2	Notify Admin		X	X		X	X			X	X
3.3	Input Volunteer Hours on behalf of user	X	X	X		X	X		X		X
3.4	Add Volunteer Opportunities	X	X	X		X	X		X		X
3.5	Delete Volunteer Opportunities	X	X	X		X	X		X		X
3.6	Sign-Up for Volunteer Opportunities	X	X	X		X	X			X	X
3.7	Cancel Commitment	X	X	X		X	X		X	X	X
3.8	Notify Volunteer		X	X		X	X		X	X	X
3.9	Track Progress	X	X	X	X	X	X			X	X
3.10	Generate Reports	X	X	X	X	X	X	X		X	X
3.14	Customize Preferences	X	X	X	X	X	X			X	X
3.15	Login	X	X			X	X			X	X
3.16	Logout					X	X			X	X
3.18	Register Volunteers	X	X	X					X		X
3.23	Android Application		X	X	X				X	X	

Table 2.11 Service Layer and Data Storage Layer Modules Traceability Matrix

3. Test Items

3.1 Unit Testing

3.1.1. Description

The Maverick Volunteer Tracking System will be tested with 4 different phases. The different phases are Unit Test phase, Component Test phase, Integration Test phase and System Validation test phase. The Unit Test phase will test the individual modules of all the subsystems in the system. This is the most detailed of the test phase.

Moving up the hierarchy we have Component Test phase which will test the system on the subsystem level. This test will check if all the subsystems are functioning properly or not. The Integration testing will test the different layers of the system. The layers will be isolated with each other and tested separately to identify any faults in the layer interfaces or the layer itself.

Lastly we have the System Validation test which will verify if the system correctly fulfills all the system requirements. The test cases in this phase will map to each of the critical requirements. The diagram below shows how the testing phases are broken down.

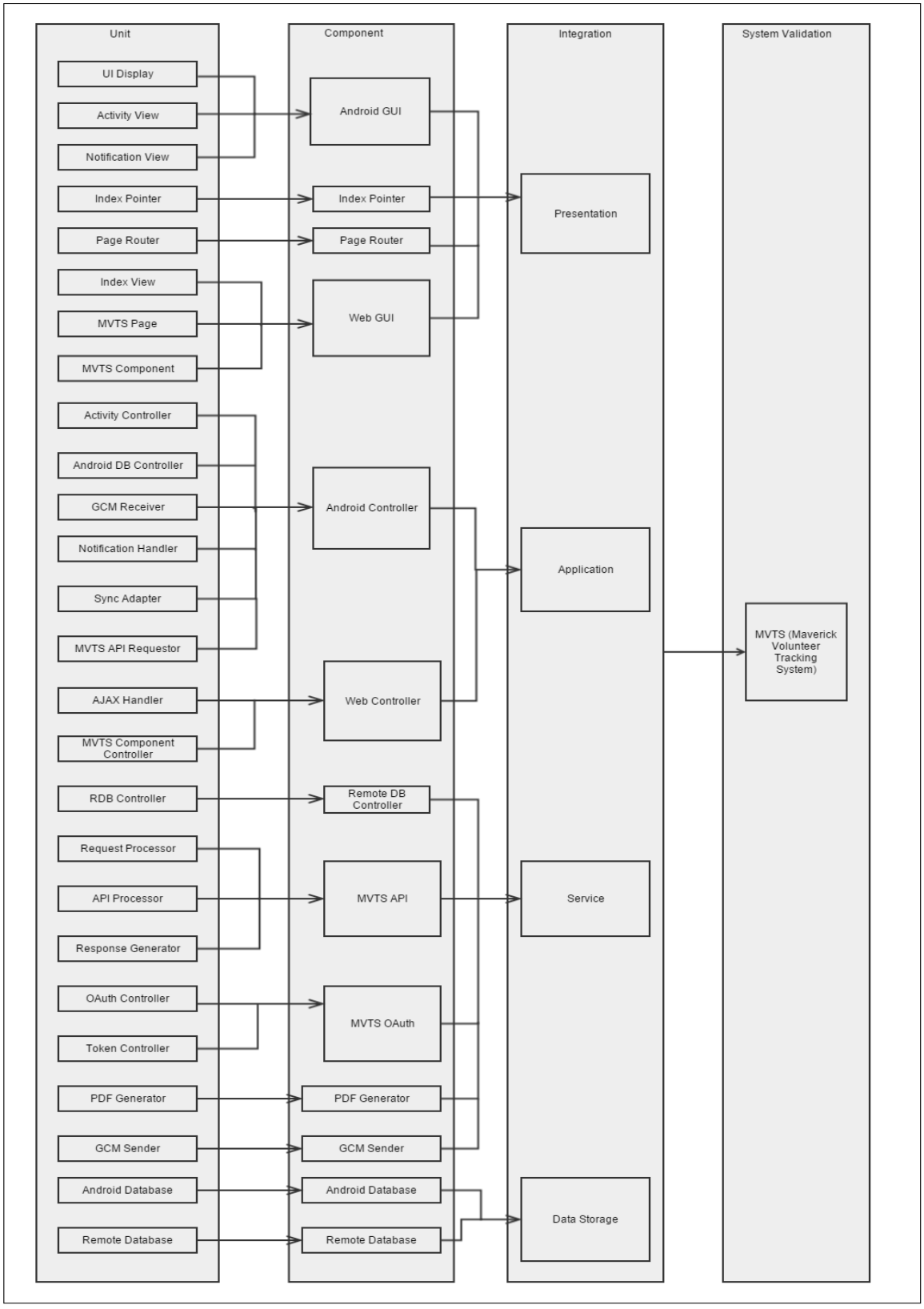


Figure 3.1 Testing Phases

3.1.2 Android GUI Unit Tests

The table below describes the unit tests for the different modules in the Android GUI subsystem.

Test ID	Module	Purpose	Input	Test Detail	Expected Result	Priority
A1	UI display	To test if all the different views are rendered properly in the Android App.	User Tap Request	User will touch the various buttons in the Android App	The view (login, home, profile etc.) corresponding to the user action should be rendered on the screen	High
A2	Activity View	To test if the Activity View successfully returns an XML data to the UI display	UI element	Test UI element data corresponding to a user action will be passed to the Activity View module	The module should return an XML data layout containing all the data for rendering a view by the UI display	High
A3	Notification View	To test if notification view successfully returns a Notification object with notification data	UI element	Test Android UI element data corresponding to a user pressing the notification icon will be used to call the Notification View module	The module should return a Notification object. This object should contain all the data needed by the UI display to render the past and new notification	Medium

Table 3.1 Android GUI Unit Tests

3.1.3 Index Pointer Unit Tests

The table below describes the unit tests for the different modules in the Index Pointer subsystem.

Test ID	Module	Purpose	Input	Test Detail	Expected Result	Priority
I1	Index Pointer	To test if the index pointer returns the correct Full path URL upon receiving a subdirectory URL request	Subdirectory URL requests	This module will be called with a subdirectory URL request for one of the pages in MVTS	Index pointer should return the complete URL request corresponding to the Subdirectory URL request as a String	High

Table 3.2 Index Pointer Unit Tests

3.1.4 Page Router Unit Tests

The table below describes the unit tests for the different modules in the Page Router subsystem.

Test ID	Module	Purpose	Input	Test Detail	Expected Result	Priority
P1	Page Router	To test if the page router returns the relevant HTML content to the index view	URL request	The page router module will be invoked with a URL request for a HTML page	The module should return HTML content corresponding to the URL request which can be parsed by the index view module and rendered	High

Table 3.3 Page Router Unit Tests

3.1.5 Web GUI Unit Tests

The table below describes the unit tests for the different modules in the Web GUI subsystem.

Test ID	Module	Purpose	Input	Test Detail	Expected Result	Priority
W1	Index View	To test if the web interface is displayed consistently	User input/click action	The user will interact with the web interface through the various buttons and forms in the display	The module should render the web pages consistently corresponding to the user specific action	High
W2	MVTS Page	To test if the MVTS Page can combine all the MVTS Components and return it as an HTML content to the page router	HTML import request	HTML import request for an MVTS Page will be sent to this module through a test script	Combined HTML content containing all the requested MVTS Component will be generated	High
W3	MVTS Component	To test if the MVTS Component can return the HTML component requested from the MVTS Page	HTML import request	HTML import request for a MVTS Component will be send to the MVTS Component module through a test script	HTML content data for the requested MVTS Component should be returned back as a HTML String data.	High

Table 3.4 Web GUI Unit Tests

3.1.6 Android Controller Unit Tests

The table below describes the unit tests for the different modules in the Android Controller subsystem.

Test ID	Module	Purpose	Input	Test Detail	Expected Result	Priority
AC1	Activity Controller	To test if the Activity Controller returns the Activity Object containing all the details for constructing an Activity View	User Input data	A sample user input data request such a user pressing a profile button will be used to invoke the Activity Controller module	The module should return an object containing all the necessary information to create an activity view which can be rendered by the UI display module	High
AC2	Android DB Controller	To test if the controller receives the result set data from the Android Database	Parsed String data	String array containing query string will be send to the module	The module should return a String array containing data for the requested query	High
AC3	GCM Receiver	To test if the GCM receiver notifies the SYNC adapter when GCM receiver receives database update notification from GCM Sender	GCM API response data	GCM API response data containing the information about update made to the remote database will be used to invoke the GCM receiver module	The module should parse the response and call the SYNC adapter module with the parsed data and initiate a synchronization process via sync adapter	High

AC4	Notification Handler	To test if a notification handler passes the notification data to Notification View	Notification String data	A test Notification string data will be send to the Notification handler via a test script	The Notification Handler should parse the Notification data and return the notification object to the Notification View	Medium
AC5	Sync Adapter	To test if the Sync Adapter updates the Android database when Remote database is updated	JSON object containing synchronization data	JSON object containing test synchronization data will be used to call the Sync Adapter module. The synchronization data will contain an update request to the remote database	The Sync Adapter should update the Android database to synchronize with remote database	High
AC6	MVTS API Requestor	To test if this module returns a JSON Object containing updated rows in remote DB to Sync adapter	JSON Object	A Synchronization request will be send to the MVTS API requestor to check for any updated rows in Remote DB	A JSON Object with all the updated data rows pertaining to the API call should be returned	High

Table 3.5 Android Controller Unit Tests

3.1.7 Web Controller Unit Tests

The table below describes the unit tests for the different modules in the Web Controller subsystem

Test ID	Module	Purpose	Input	Test Detail	Expected Result	Priority
WC1	AJAX Handler	To test if AJAX Handler returns a JSON response for loading a page	JSON Request	A JSON request for an MVTS Component data will be made to this module	The module should return a JSON Response containing the requested MVTS Component data	High
WC2	MVTS Component Controller	To test if it returns a formatted JSON data whenever a JSON request for a MVTS Component data is made	JSON Request	A JSON request for an MVTS Component will be made to this module with a sample test script	The module should return a formatted JSON response containing all the necessary data to load an MVTS Component in the GUI	High

Table 3.6 Web Controller Unit Tests

3.1.8 Remote DB Controller Unit Tests

The table below describes the unit tests for the different modules in the Remote DB Controller subsystem.

Test ID	Module	Purpose	Input	Test Detail	Expected Result	Priority
RC1	Remote DB Controller	To test if SQL queries can be performed to the Remote database	Parsed String data for creating SQL queries	String array containing data for SQL queries will be send to the various database access methods within this module	String array as a result set for the requested SQL query should be returned back	High

Table 3.7 Remote DB Controller Unit Tests

3.1.9 MVTS API Unit Tests

The table below describes the unit tests for the different modules in the MVTS API subsystem.

Test ID	Module	Purpose	Input	Test Detail	Expected Result	Priority
M1	Request Processor	To test if the request processor sends a token validation request to the OAuth Controller	Unauthenticated API Call Request	API call with user login information will be made to the Request Processor module via PHP unit test script	Validated API call request will be received back from OAuth controller module	High

M2	API Processor	To test if API Processor generates a string array containing data for the respective API call	Authorized API call request	Validated API call request will be send to the API Processor module via PHP unit test script	A string array containing all the relevant data for that particular API request should be returned by this module	High
M3	Response Generator	To test if a correct HTTP response is generated for an API call	Raw String data containing user requested information	String Array containing user action specific data will be send to this module	HTTP response message containing the user action requested data should be generated by this module	High

Table 3.8 MVTS API Unit Tests

3.1.10 MVTS OAuth Unit Tests

The table below describes the unit tests for the different modules in the MVTS OAuth subsystem.

Test ID	Module	Purpose	Input	Test Detail	Expected Result	Priority
MO1	OAuth Controller	To test if OAuth Controller returns a Boolean upon receiving user name and password	User validation request	Username and password string will be send to this module	Boolean variable indicating authenticati on status will be returned	High
MO2	Token Controller	To test if a token is generated upon user authentication by OAuth Controller	Token update request	User login session information such as username and password will be send to this module	MO2	Token Controller

Table 3.9 MVTS OAuth Unit Tests

3.1.11 GCM sender Unit Tests

The table below describes the unit tests for the different modules in the GCM sender subsystem.

Test ID	Module	Purpose	Input	Test Detail	Expected Result	Priority
G1	GCM Sender	To test if GCM sender sends a String data containing remote database update information to the GCM Server.	Raw String data containing information about the database insert/update request	This module will be invoked with an insert/update request to the remote database in order to trigger the GCM Servers	JSON Object containing information about updated rows in the Remote DB should be generated	High

Table 3.10 GCM Sender Unit Tests

3.1.12 PDF Generator Unit Tests

The table below describes the unit tests for the different modules in the PDF Generator subsystem.

Test ID	Module	Purpose	Input	Test Detail	Expected Result	Priority
PD1	PDF Generator	To test if a PDF document is generated in the Server upon receiving user request.	Raw String data containing all the information for the requested report	Test report data necessary for creating a report will be passed as a string to the PDF Generator module as a function call	URL for the generated PDF report document should be returned. Upon entering the URL in a web browser the PDF document should be opened	Medium

Table 3.11 PDF Generator Unit Tests

3.1.13 Android Database Unit Tests

The table below describes the unit tests for the different modules in the Android Database subsystem.

Test ID	Module	Purpose	Input	Test Detail	Expected Result	Priority
AD1	Android Database	To test if data can be stored and retrieved properly from the android database	SQL lite database query	Various SQL lite queries will be performed with test scripts	SQL lite result-sets as String array corresponding to the queries should be returned	High

Table 3.12 Android Database Unit Tests

3.1.14 Remote Database Unit Tests

The table below describes the unit tests for the different modules in the Remote Database subsystem.

Test ID	Module	Purpose	Input	Test Detail	Expected Result	Priority
RD1	Remote Database	To test if data can be stored and retrieved properly from the remote database	MYSQL database query	Various MySQL queries will be performed with database test scripts	Result-sets as String array corresponding to the respective queries should be returned.	High

Table 3.13 Remote Database Unit Tests

3.2 Component Testing

3.2.1 Description

Component Testing will be performed on each component to ensure that all components are functioning properly after completing unit test for each module. The Component Testing will be performed at subsystem levels of the system. The subsystems will be supplied with test data and compared to the expected outcome to ensure that modules with in the subsystem are functioning together properly and as expected.

3.2.2 Presentation Layer Component Test

The table below describes the component tests for the different subsystems in the Presentation Layer.

Test ID	Subsystem	Purpose	Input	Test Description	Expected Result	Priority
AG1	Android GUI	To test if Android GUI can display different views and visual changes with the user tap to display to the user.	User Tap Request	Various views in the app will be tapped to check if all the pages are displayed correctly.	The Android GUI should display different views corresponding to the user actions.	High
AG2	Android GUI	To test if Android GUI is successfully relaying user input data to the Android Controller.	User Input	Methods will be called to send the user inputs information and check if it is sending data correctly.	The Android GUI should relay user input data to Android Controller.	High
IP1	Index Pointer	To test if the index pointer returns the correct Full path URL upon receiving a subdirectory URL request.	Subdirectory URL requests	Index Pointer will be called with a subdirectory URL request for one of the pages in MVTS pages.	Index pointer should point all the subdirectory URL requests to the base directory/path.	High
PR1	Page Router	To test if the page router returns the relevant HTML content to the index view.	URL request	The page router will be invoked with a URL request for a HTML page.	The Page Router should return HTML content corresponding to the URL request.	High
WG1	Web GUI	To test if Web GUI retrieves the user interaction through various buttons and forms and display results	User input/click action	Various pages will be clicked and various inputs will be entered to check if all the pages and	The Web GUI should display different views and results corresponding to the user actions.	High

		to the user.		results are displayed correctly.		
WG2	Web GUI	To test if Web GUI retrieves the user interaction through various buttons and forms and pass it to Web Controller.	User input/click action	Methods will be called to send the user request and check if it is making successful request.	The Web GUI should relay user input data to Web Controller.	High
WG3	Web GUI	To test if Web GUI retrieves all the MVTS component needed for each specific page in the form of HTML content.	HTML import request	HTML import request for a MVTS Component will be send through a test script to check if correct MVTS component is returned.	The Web GUI should receive all the MVTS components in the form of HTML content needed for each specific page.	High

Table 3.14 Presentation Layer Component Tests

3.2.3 Application Layer Component Test

The table below describes the component tests for the different subsystems in the Application Layer.

Test ID	Subsystem	Purpose	Input	Test Description	Expected Result	Priority
AC1	Android Controller	To test if the Android Controller retrieves the Activity Object containing all the details for constructing an Activity View.	User Input data	A sample user input data request such a user pressing a profile button will be used to invoke the Activity Controller.	The Android Controller should receive and return an object to containing all the necessary information to	High

					create an activity view.	
AC2	Android Controller	To test if Android controller can successfully request and receive the result set data from the Android Database.	Parsed String data	Sample string of array containing SQLite query will be send to check if Android Controller is retrieving information correctly from Android database.	The Android Controller receives result set from Android Database depending on the request.	High
AC3	Android Controller	To test if Android Controller can successfully update Android Database upon receiving notification about changes in Remote Database.	GCM API response data and JSON object containing synchronization data	Sample SQLite queries for updating the database will be sent to Android Database to update Android Database.	The Android Controller should update Android Database upon receiving notification about changes in Remote Database.	High
AC4	Android Controller	To test if Android Controller successfully pushes API calls via HTTP Request to send and receive data from the MVTS API subsystem.	JSON Object	A Synchronization request will be made to MVTS API to check if API calls are successfully pushed.	The Android Controller should receive a JSON Object with result pertaining to the API call.	High
WC1	Web Controller	To test if Web Controller is relaying results pertaining to user request back to the user.	JSON Response	A set of JSON response will be provided to see if Web Controller is converting to HTML and passing results back to the Web GUI.	The Web Controller Should convert JSON response to HTML content and relay results back to the Web GUI.	High
WC2	Web Controller	To test if Web Controller successfully responds to all the events generated by user through Web GUI and relay it to the	HTTP Request	A HTTP request pertaining to the events generated by user will be sent to check if correct HTTP request is made to MVTS API.	The Web Controller should successfully responds to all the events generated by user through	High

		MVTS API.			Web GUI and relay it to the MVTS API.	

Table 3.15 Application Layer Component Tests

3.2.4 Service Layer Component Test

The table below describes the component tests for the different subsystems in the Service Layer.

Test ID	Subsystem	Purpose	Input	Test Description	Expected Result	Priority
RDBC1	Remote DB Controller	To test if SQL queries can be performed to the Remote database.	Parsed String data for creating SQL queries	Sample string array containing data for SQL queries will be send to the various database access methods.	String array as a result set for the requested SQL query should be returned back.	High
MA1	MVTS API	To test if the MVTS API is fetching the required data from the Remote Database and fulfilling the API request coming from the Application layer.	API response and String Array	Various API calls pertaining to the request from Android and Web Controller subsystem will be made to check if MVTS API is responding correctly.	The MVTS API should respond to the various API calls pertaining to the request and provide the requested data.	High
MO1	MVTS OAuth	To test if MVTS OAuth validates the user and session.	User and Session validation request	User credentials will be checked with remote	The MVTS OAuth should receive either	High

				database to validate user.	successful user validation or unmatched user credentials.	
PG1	PDF Generator	To test if a PDF document is generated in the Server upon receiving user request.	Raw String data	A request will be made to generate PDF through test script.	URL for the generated PDF report document should be returned.	Medium
GS1	GCM Sender	To test if GCM sender sends a notification regarding update on the remote database to the GCM Server.	Notification Response	This GCM Sender will be invoked with an update in the remote database in order to trigger the GCM Servers.	The GCM Senders should notify GCM Server about changes in remote database.	High

Table 3.16 Service Layer Component Tests

3.2.5 Data Storage Layer Component Test

The table below describes the component tests for the different subsystems in the Data Storage Layer.

Test ID	Subsystem	Purpose	Input	Test Description	Expected Result	Priority
AD1	Android Database	To test if data can be stored and retrieved properly from the android database	SQLite database query	Various SQL lite queries will be performed with test scripts.	SQLite result-sets as String array corresponding to the queries should be returned	High

RD1	Remote Database	To test if data can be stored and retrieved properly from the remote database	MYSQL database query	Various MySQL queries will be performed with database test scripts	Result-sets as String array corresponding to the respective queries should be returned.	High

Table 3.17 Data Storage Layer Component Tests

3.3 Integration Testing

3.3.1 Description

Integration Testing will be performed to ensure the overall system is functioning after integrating all the modules and components into the layers. Integration Testing will verify the functions specified in the ADS and DDS. Integration Testing will be conducted at the system level. The Systems will be supplied with test data and compared to the expected outcome to ensure that subsystems within the system are functioning together properly and as expected.

3.3.2 Integration Test

The table below describes the integration tests for the Maverick Volunteer Tracking System.

Test ID	System	Purpose	Input	Test Description	Expected Result	Priority
I1	Presentation Layer	To test if the Presentation Layer is displaying all pages for Android App and listening and responding to user request.	User Tap Request	Various actions will be performed on the various pages to check if those pages are displaying correct results.	The Presentation Layer should respond correctly to all the user requests.	High
I2	Presentation Layer	To test if the Presentation Layer is displaying all pages for MVTS website and listening and	User Input/ Click Action	Various actions will be performed on the various pages to check if those pages are displaying correct results.	The Presentation Layer should respond correctly to all the user requests.	High

		responding to user request				
I3	Application	To test if the Application Layer is responding to the user request in Presentation Layer	User Request	The Application Layer will either stores or retrieves data in the database based on the user request through service layer.	The data is retrieved or stored in the database and result is shown to the user.	High
I4	Application	To test if the Application Layer is making HTTP request to Service layer and receiving HTTP response from Service layer pertaining to the user request.	JSON Request JSON Response	Various JSON requests will be send to the Service Layer to check if it is responding correctly with relevant JSON response.	Relevant JSON response should be returned to correspondi ng JSON request.	High
I5	Application	To test if the Application Layer is correctly updating Android Database after updates on Remote Database.	GCM API response data and JSON object containing synchronizati on data	The request for updating Android Database with sample SQLite queries will be sent after getting notification about updates in Remote Database.	The Application Layer should update Android Database after Remote Database is updated.	High
I6	Service Layer	To test if the user and session is verified before doing any processing for any requests from Application Layer.	User and session validation requests	User's credential is checked from database to verify user.	The user verification status and session status is returned.	High
I7	Service Layer	To test if the service layer is responding to the various	JSON Request and API Calls	Various API calls will be made to check if Service layer is responding	The Service layer should provide Application	High

		requests made by Application Layer.		to the request from the Application Layer.	Layer with relevant JSON response correspondi ng to the request.	
I8	Service Layer	To test if the Service layer is inserting/updat ing/deleting data in remote database.	MySQL queries	Various sample queries for insertion/deletion/u pdate will be sent to check if database is updated corresponding to the relevant request.	The Service Layer should successfully insert/delete/ update data in remote database.	High
I9	Data Storage	To test if data can be stored and retrieved properly from the remote database and Android Database	SQLite queries MySQL queries	Various SQLite and MySQL queries will be performed to check if the data is inserted/updated/de leted successfully in the databases.	The Data Storage Layer should store and retrieved properly from Remote and Android Databases.	High

Table 3.18 Integration Tests

3.4 System validation Testing

3.4.1 Description

System Validation Testing will be performed to meet the requirements stated in the System Requirement Specification the acceptance criteria formulated by team and customer. The test will ensure that layers are communicating with each other and performing the required task.

3.4.2 System Validation Test

The table below describes the system validation tests for the Maverick Volunteer Tracking System.

Test ID	System Test	Req.	Input	Test Description	Expected Result	Priority
V1	Verify Login and Logout	3.15, 3.16	Username and Password	The correct and incorrect data for username and password will be submitted to verify that correct actions are taken in each step. The user clicks on logout button to logout of the system.	The System logs the user in after successful username and password validation. The system logouts user after clicking logout button.	Critical
V2	Register Volunteers	3.18	User credential for registration	In the admin page, required credentials to register volunteers into the system are entered to verify that new user is added to the system.	System shall register new volunteers.	Critical
V3	Verify input volunteer hours and Track Progress	3.1, 3.3	User requests to input number of hours	The various number of hours volunteered are entered in the specific page.	The System modifies/updates relevant data in the database.	Critical
V4	Verify Notification	3.2, 3.8	User creating new events or adding volunteered hours.	After new events are created or numbers of volunteered hours is added, System should send notification to relevant users.	System notifies user about relevant changes.	High
V5	Verify Add/Edit/Delete/Sign up Opportunities/ commitment	3.4, 3.5, 3.6, 3.7	User requests	The opportunities/commitment will be added, edited, deleted, and signed up to verify that the information in the database is changed.	The System modifies/updates relevant data in the database.	High
V6	Generate/Manage Reports	3.10, 3.11	User requests	The request will be provided to generate and manage reports to	The System shall generate/manage relevant reports.	High

				verify that relevant report is generated.		
V7	Verify Ease of Use	3.22	User Inspection	The user will try to visit all pages to ensure that they can reach anywhere on the webpage with maximum of 3 clicks.	The system allows user to reach anywhere on the website with maximum of 3 clicks.	High
V8	Promote Members/Demote Facilitators	3.11, 3.12	User Request	In the admin page, the members are promoted or demoted to verify that information is modified in the database.	The System modifies/updates relevant data in the database.	Moderate
V9	Verify Web Browser Compatibility	8.1	None	The website will be opened in various browsers to ensure that it can be opened.	The System is opened in various browsers namely Google Chrome, latest Internet Explorer, Google Chrome, and Safari.	Moderate

Table 3.19 System Validation Tests

4. Risks

4.1 Overview

The following section contains a list of risks that may be encountered during the testing phase of MVTS. This section will also cover the strategies to mitigate these risks.

4.2 Risk Table

The table below identifies the potential risks that the team may come across while testing the Maverick Volunteer Tracking system. Each risk is associated with a risk ID, risk description, risk impact, severity, and a mitigation strategy for handling or avoiding the risk. Each risk is ranked on a severity of low (unlikely to occur), medium (likely to occur) or high (expected to occur).

Risk ID	Risk	Impact	Severity	Mitigation Strategy
R1	Web Server connection fails	User will not be able to connect to the website and the Android App will not be able to connect to the website	High	Ensure UTA's server is live and functioning
R2	Database access fails	MVTS is unable to connect to the database to retrieve or send information.	High	Ensure that database connection is properly set up and each request is authorized
R3	Inaccurate Data	Incorrect data being sent to the database can result in inaccurate data being represented	Medium	Ensure the data being passed is accurate by allowing the admin to verify the input
R4	GCM Server connection fails	Android App may not be able to accurately send notifications to the device	Medium	Verify if the App is able to properly connect to the GCM server to send notifications to the device
R5	Adding new requirements towards the end of the project	Can result in new bugs in the system	Medium	Test all affected functions and continue the component and integration testing after each change

Table 4.2 Risks

5. Features To Be Tested

This section covers the features to be tested. Features are associated with a risk level that is described below along with the testing approach. The risk levels are described below.

5.1 Risk Definitions

5.1.1 High:

This feature's implementation is in the process of development.

5.1.2 Medium:

This feature's process of implementation is established but is still undergoing development.

5.1.3 Low:

This feature's has been accurately functioning during the stage II of the prototype phase of development and testing.

5.2 Customer Requirements

5.2.1 Input Volunteer Hours

Description: The Volunteer Tracking System shall allow a user to input the hours volunteered. To input the hours, the volunteers shall be able to select the name of the opportunity from a dropdown associated with a category and enter the number of hours they volunteered along with comments. If a volunteer does not commit to an opportunity, but still volunteers at that opportunity, the system shall allow them to input the hours they volunteered.
Risk: Medium
Testing Approach: Login to the system as a volunteer and input volunteer hours to ensure the data in terms of number of hours volunteered is correctly analyzed by the system. Also ensure that no commitment is given to an event but the user is still able to enter the number of hours they volunteered.

5.2.2 Notify Admin

Description: The Volunteer Tracking System shall also notify the admin and the opportunity category facilitator when members input their time volunteered. This notification will be system generated. The admin and the facilitator will be able to see the notification upon logging in to the system.
Risk: Medium
Testing Approach: Login to the system as a volunteer and input the time volunteered for an opportunity. Login to the system again as a facilitator and ensure that notifications are visible for the volunteered hours for that particular member.

5.2.3 Input Volunteer Hours on Behalf of User

Description: Upon the request of the volunteer, facilitators must be able to input the volunteer hours on behalf of the volunteer. The facilitators shall be able to see a list of all members and an option to input their volunteer hours. The facilitator will have access to input the volunteer hours of all members without any time limitations or constraints
Risk: Medium
Testing Approach: Login as a facilitator into the system and select a member. Ensure that the system allows an option to input volunteer hours on behalf of the volunteer.

5.2.4 Add Volunteer Opportunities

Description: The Volunteer Tracking System shall allow facilitators to input the new or upcoming volunteer opportunities. An opportunity may include a title, description, date and time, location and images.
Risk: Low
Testing Approach: Login to the system as a facilitator and input a new volunteer opportunity. Ensure that the system accepts a new opportunity by the facilitator.

5.2.5 Delete Volunteer Opportunities

Description: The Volunteer Tracking System shall allow facilitators to delete volunteer opportunities previously entered into the System. If volunteers have committed to an opportunity and it is cancelled, the system will notify all volunteers through Email.
Risk: Medium

Testing Approach: Login to the system as a facilitator. Ensure the system allows the deletion of volunteer opportunities. Cancel an existing opportunity in the system and ensure that a notification is sent to all volunteers though email.

5.2.6 Sign Up for Volunteer Opportunities

Description: The volunteers shall be able to see the details of an opportunity such as the date, time, and location and have an option to sign up for an opportunity to indicate they will be volunteering at that opportunity.
Risk: Medium
Testing Approach: Login to the system as a volunteer and click the opportunities page to see the lit of all available opportunities. Select and opportunity and sure all the details regarding this particular opportunity is available and the system allows the volunteer to commit to the event.

5.2.7 Cancel Commitment

Description: The volunteers shall be able to cancel a commitment they previously made. If volunteers previously signed up for an opportunity, the system shall allow them to cancel their commitment to indicate they will no longer be volunteering at that opportunity.
Risk: Medium
Testing Approach: Login to the system as a volunteer and commit to an opportunity. Cancel the commitment just made. Ensure that the system indicates the volunteer is no longer attending this event.

5.2.8 Notify Volunteer

Description: The Volunteer Tracking System shall notify the volunteer and the opportunity facilitator upon the volunteer's acceptance/commitment or cancellation of an opportunity. This notification will be system generated. The volunteer and the facilitator will be able to see this notification on their home page.
Risk: Medium
Testing Approach: Login to the system as a volunteer and commit to an opportunity. Ensure that a notification of acceptance is seen. Cancel the commitment to the same opportunity. Ensure that a notification of cancellation is seen. Login to the system as a facilitator and ensure that both notifications are seen.

5.2.9 Track Progress

Description: The Volunteer Tracking System shall allow users to track progress of their volunteer activities and the status of different service levels. Service levels are different levels that volunteers can achieve based on the total number of hours. The levels are divided as follows: 30, 60, 90, 150, and 150+.
Risk: Medium
Testing Approach: Login to the system as a volunteer and input the listed hours for an opportunity to track the changes in the volunteer levels.

5.2.10 Generate Reports

Description: The Volunteer Tracking System shall generate progress reports for each volunteer upon their request. The progress report should include details such as the categories/types of opportunities volunteered in, and the total number of hours volunteered.
Risk: Medium
Testing Approach: Login to the system as a volunteer and click the generate report button. Open the report and ensure that all details such as categories of volunteer opportunities and total number of hours volunteered are included.

5.2.11 Customize Preferences

Description: The Volunteer Tracking System shall allow volunteers to customize their preferences. Preferences include setting the date of availability along with level of interest in different opportunity categories.
Risk: Medium
Testing Approach: Login to the system as a volunteer and click on the profile tab. Select customization options and set preferences to include the dates of availability and rank the level of interest in certain categories. Ensure that the system is generating a new notification based on the possible matches.

5.2.12 Login

Description: The Volunteer Tracking System shall allow users to login with their Email and password. When a user logs in to the system for the first time, the system shall allow them to enter their Email for validation. When the Email is validated, the system shall ask the user to establish their password. When a user logs in to the system again, they will be required to enter their Email and Password for validation.
Risk: Medium

Testing Approach: Enter the Email address for validation. Ensure the system requests to establish a new password. Login to the system and enter both the Email and password for validation. Ensure that system allows the user to view the home page after logging in.

5.2.13 Logout

Description: The Volunteer Tracking System shall allow volunteers to logout of the system. When the user is logged out, the system shall redirect to the login page.
Risk: Medium
Testing Approach: Login to the system as a volunteer and logout of the system. Ensure that when logged out, the system is redirected back to the login page.

5.2.14 Register Volunteers

Description: The Volunteer Tracking System shall allow Admin to register volunteers and allow access into the system.
Risk: Medium
Testing Approach: Login to the system as an Admin. Register a new user by entering in their Email address. Login to the system as the newly registered volunteer and ensure the system is able to redirect the volunteer to the home screen after verification.

5.2.15 Ease of Use

Description: The Volunteer Tracking System shall provide a user-friendly interface. The system shall also limit the number of clicks to allow a user to reach their desired page easily.
Risk: Medium
Testing Approach: Login to the system as a volunteer. Select a page as a starting point and another page as the destination. Keep a counter to track the number of click it takes the volunteer to get from the start to the destination. Ensure that total number of clicks is less than three.

5.2.16 Android Application

Description: The Volunteer Tracking System shall be available in the form of an Android Application. The Application will be available in the Google Play Store to download for free.
Risk: Medium

Testing Approach: The app will be tapped to open the application and ensure that it launches the system without any issues. Navigate through various screens to verify all the options and data is seen properly. The specific application functionalities will be tested as described in Customer Requirements section of this document.

5.3 Packaging Requirements

5.3.1 Website URL

Description: Website will be hosted under a subdirectory of http://www.uta.edu
Risk: Low
Testing Approach: Ensure that all files are properly uploaded to the UTA server through a secure port. In a browser, type in the site address and ensure that it works as expected.

5.3.2 Page URLs

Description: Website URLs will be human readable.
Risk: Low
Testing Approach: Type in a sub URL that directs to another page in the system. Ensure that the browser address seen in the website is readable by a member of the team.

5.3.3 Installation Script

Description: A PHP installation script that will populate the necessary database tables shall be provided.
Risk: Low
Testing Approach: Request a member from another team to use this installation script on their local server to ensure the necessary database tables are properly populated.

5.4 Performance Requirements

5.4.1 Application Response Time

Description: Response time between user interaction and result should be less than 8 seconds in both the website and the Android app.
Risk: Medium
Testing Approach: Login to the system as a volunteer and test any input a user may make in the application to ensure the response time is less than 8 seconds.

5.4.2 Dynamic Page Update

Description: Only the necessary parts of the web page will be updated upon the user interaction instead reloading the page completely.
Risk: Medium
Testing Approach: Login to the system as a facilitator and enter a new opportunity into the opportunities page. Ensure that only the opportunities are refreshed and not the entire page.

5.4.3 File Compression

Description: JavaScript and CSS files will be compressed to reduce the file size. Size of JPEG images should be less than 3 MB.
Risk: Medium
Testing Approach: When uploading the files to the server, compress the JavaScript and CSS files. Check the compression size and of the files and ensure that it is not more than 3 MB.

5.5 Security and Privacy Requirements

5.5.1 Password Encryption

Description: All the user passwords shall be encrypted in the MySQL database.
Risk: Low
Testing Approach: Access the database and request the password of the accounts created. Ensure that passwords are encrypted when returned.

5.5.2 Malicious Input Protection

Description: System shall validate all the input data to ensure that the entered data is correct and/or user has not entered any malicious code in any input fields.
Risk: Medium
Testing Approach: Insert incorrect data and ensure that the system is able to detect and respond with an appropriate notification to the user.

5.6 Maintenance and Support Requirements

5.6.1 PHP Version Support

Description: The UTA servers are running PHP version 5.1. Therefore, the web application shall be compatible with PHP version 5.1.
Risk: Medium
Testing Approach: Run the application on PHP version 5.1 on the local server and ensure that the application works as expected without any issues

5.6.2 Android Version Support

Description: The mobile version of the system will be Android based. The application will support a minimum API level of 16, which corresponds to version 4.1.2 (Jelly Bean).
Risk: Medium
Testing Approach: The system will be tested on a different Android device running version 4.1.2 and above. The team will ensure that MVTS runs on this device without any issues.

5.6.3 User Manual

Description: The team will provide user manual that describes the different functionality of product and instructions on how to use product. This user manual shall support system administrator for any problems in future.
Risk: Medium
Testing Approach: Review the manual thoroughly to ensure that it specifies the details on system installation, system functionalities and system specifications.

5.7 Other Requirements

5.7.1 Web Browser Compatibility

Description: The web interface shall be accessible via various popular browsers such as Safari, Google Chrome, Mozilla Firefox, and Internet Explorer.
Risk: Medium
Testing Approach: The system will be tested on Safari, Chrome, Firefox and IE to ensure all the functionality is intact and performing as expected.

5.7.2 Web Service Code Compatibility

Description: All The source code of the web functionality shall be compatible and portable with various platforms such as Windows, Mac, and Linux.
Risk: Low
Testing Approach: The system will be tested on Windows, Mac and Linux to ensure all the functionality is intact and performing as expected on the different platforms.

5.7.3 Responsive Design

Description: The website shall reflow its layout to fit in for the screen resolution or the window size.
Risk: Low
Testing Approach: The system will be tested on an Android device to ensure the layout fits in the fit resolution. For the web browsers, the team will resize the windows to various lengths and widths to ensure the layout still fits in the current size.

6. Features Not To Be Tested

This section below covers features that will not be tested. Some features in this section cannot be directly tested, or they may not have been implemented due to time constraints.

6.1 Customer Requirements

6.1.1 Generate Newsletter

Description: The Volunteer Tracking System shall generate a newsletter upon the addition of new volunteer opportunities.

Rationale: This is a future requirement and it will not be implemented in this version of the release.

6.1.2 Generate Appreciation Letter

Description: When a volunteer reaches a service level, the system shall notify the admin. This notification will be a system-generated notification, which they can view on their homepage. Upon this notification, the admin shall be able to generate a generic appreciation letter for that particular volunteer as a PDF.

Rationale: This is a future requirement and it will not be implemented in this release.

6.1.3 Volunteer Stories

Description: The Volunteer Tracking System shall provide a social aspect to the interface where the volunteers can input and share their stories.

Rationale: This is a future requirement and it will not be implemented in this release.

6.1.4 iOS Mobile Application

Description: The Volunteer Tracking System shall be available in the form of an iOS Application. The Application will be available in the App Store to download for free.

Rationale: This is a future requirement and it will not be implemented in this release.

6.2 Performance Requirements

6.2.1 Third-Party Code Libraries and Frameworks

Description: JavaScript and CSS libraries will be directly accessed from the CDN servers, thereby, improving the access time in distant locations.

Rationale: This feature is not testable. When retrieving third party code libraries, the performance time depends on the response it receives from the third party servers.

6.3 Security and Privacy Requirements

6.3.1 Website Cache

Description: Age of the website cache will be restricted to 7 days.

Rationale: Due to time constraints, this requirement will not be tested.

6.4 Packaging Requirements

6.4.1 Google Play Publication

Description: Android app will be released into Google Play as a free download.

Rationale: No test is needed for the requirement. If it passes Google's quality assurance process of app publishing then it will be published.

6.5 Maintenance and Support Requirements

6.5.1 Source Code Documentation/Source Code Availability

Description: All the documentation prepared by team TimeKeepers including System Requirements Specification, Architectural Design Specification, Detail Design Specification, and System Testing Plan will be made available to future senior design students. The source code shall be well documented with comments and details about functionality. The code shall help anyone who wants to further develop this product in future.

Rationale: This requirement is not testable

6.5.2 System Maintenance

Description: The team TimeKeepers shall not be responsible to maintain the system or source code after completion of project. The College of Engineering website Developer, Christopher Woods, will continue to maintain the website as it will be hosted under uta.edu/engineering.

Rationale: This requirement is not testable. Christopher Woods may continue to maintain the system in the future.

6.5.3 Training

Description: The team shall provide training to system manager on how to use and manage the system. The team will demo the product upon completion and explain all the functionality of the system, which shall help manager to understand system better.

Rationale: This maintenance requirement is not testable

6.6 Other Requirements

6.6.1 Tablet Support

Description: The Android app will be available on Android tablets, supporting Android version 4.1.2 or higher.

Rationale: This is a future requirement and it will not be implemented in this release.

6.6.2 Testing

Description: The features and functionality of Volunteer Tracking System will be thoroughly tested with all requirements and acceptance criteria before handing system to the customers.

Rationale: This maintenance requirement is not testable**.** Other metrics such as unit, component, integration and system verification testing will used instead to ensure the system is performing and functioning as expected.

7. Overall Test Strategy

7.1 Testing Phases

This section will cover the overall strategy for testing Maverick Volunteer Tracking System to ensure that it meets the requirements defined in System Requirement Specification and to verify that the construction of the product is consistent with the architecture defined in the Architecture Design Specification and Detail Design Specification.

7.1.1 Unit Testing Phase

The Unit Testing will be performed in each unit and module to test the function and prevent any system errors and malfunction before integrating all the modules to component. Unit testing phase will be performed by each developer developing specific module and will cover the following modules.

- UI Display
- Activity View
- Notification View
- Index Pointer
- Page Router
- Index View
- MVTS Page
- MVTS Component
- Activity Controller
- Android DB Controller
- GCM Receiver
- Notification Handler
- Sync Adapter
- MVTS API Requestor
- Ajax Handler
- MVTS Component Controller
- RDB Controller
- Request Processor
- API Processor
- Response Generator
- OAuth Controller
- Token Controller
- PDF Generator
- GCM Sender
- Android Database
- Remote Database

7.1.2 Component Testing Phase

Component Testing will be performed on each component after completing unit test for each modules that are combined with in that component to ensure that all components are functioning properly. The Component Testing will be performed by team members and will test the following subsystems.

- Android GUI
- Index Pointer
- Page Router
- Web GUI
- Android Controller
- Web Controller
- Remote DB Controller
- MVTS API
- MVTS OAuth
- PDF Generator
- GCM Sender
- Android Database
- Remote Database

7.1.3 Regression Testing Phase

Regression Testing will be used to test the system and uncover new bugs when changes are made in the software. Regression Testing will be conducted as each stage of our delivery schedule. After each modules are integrated into the component, we will perform regression test to ensure that it still functions as expected. Some of the testing that will be included in Regression Testing will be:

- Run all previous test
- Run new tests relating to the added feature
- Record test results

7.1.4 Integration Testing Phase

Integration Testing will be performed to ensure the overall system is functioning after integrating all the modules and components into the layers. Integration Testing will verify the functions specified in the ADS and DDS. Integration Testing will include following layers.

- Presentation Layer
- Application Layer
- Service Layer
- Data Storage Layer

7.1.5 System Validation Testing Phase

System Validation Testing will be performed to meet the requirements stated in the System Requirement Specification the acceptance criteria formulated by team and customer.

7.2 Tools

The team is planning to use the following tools to aid in testing.

- JUnit – For unit testing of individual functions and features in the Android App.
- Expresso – For testing Android App UI elements
- PHPUnit – For unit testing individual functions and features in the website and backend integration testing.
- Web-Component-Tester (based on Mocha & Chai) – For unit testing polymer components.
- GitHub – Following the testing at each stage, bugs will be reported in the GitHub repository.

7.3 Test Metrics

Priority	Description	Pass Criteria	Fail Criteria
Critical	Features that are essential to the system. Any defects that render the system nonfunctional or prevent other tests to run successfully must be fixed immediately.	100%	<100%
High	Features that are important to the system but it can still function without them. Any defects that affect the critical functions of the system must be fixed in current release cycle.	≥ 90%	< 90%
Moderate	Features that help polish and refine the system. The system will still function properly without these features. The defects can be fixed in the next release cycle.	≥ 75%	< 75%
Low	Features that are listed in future requirements or are extra add on functionality to existing system. Any defects that have little or no impact on the system can be fixed in upcoming future releases.	≥ 30%	< 30%

Table 7.3 Test Metrics

8. Acceptance Criteria

This section specifies the acceptance criteria. The different tables state the pass and fail criteria for Unit, Component, Integration and System Validation testing.

8.1 Unit Testing

Test ID	Module	Pass Criteria	Fail Criteria
A1	UI Display	The different views corresponding to the user action is rendered on the screen	• The different views are not rendered or not properly rendered on the screen
A2	Activity View	The module returns an XML data layout containing all the data for rendering a view by the UI display	• Activity View fails to return an XML data layout containing the data for rendering a view
A3	Notification View	The module returns a Notification object. This object contain all the data needed by the UI display to render the past and new notification	• Module fails to return a notification object • Module does not contain the data needed for the UI display to render the notifications
I1	Index View	Index pointer returns the complete URL request corresponding to the Subdirectory URL request as a String	• Module fails to return the URL request as a String
P1	Page Router	The module returns HTML content corresponding to the URL request which is parsed by the index view module and rendered	• Module fails to return HTML content
W1	Index View	The module renders the web pages consistently corresponding to the user specific action	• Module fails to render web pages corresponding to the user specific action
W2	MVTS Pages	Combined HTML content containing all the requested MVTS Component is generated	• HTML content is not generated
W3	MVTS Components	HTML content data for the requested MVTS Component returns back as a HTML String data.	• HTML content data for the requested MVTS component is not returned as a HTML string
AC1	Activity Controller	The module returns an object containing all the necessary information to create an activity view which is rendered by the UI display module	• Module fails to return an object that contains the necessary information to create an activity view.

AC2	Android DB Controller	The module returns a String array containing data for the requested query	• Module fails to return a String array containing data for the requested query
AC3	GCM Receiver	The module parses the response and calls the Sync adapter module with the parsed data and initiates a synchronization process via sync adapter	• Module fails to parse response • Fails to call the Sync Adapter module with the parsed data • Fails to initiate a synchronization process via the sync adapter
AC4	Notification Handler	The Notification Handler parses the Notification data and returns the notification object to the Notification View	• Fails to parse the notification data • Fails to return the notification object to the notification view
AC5	Sync Adapter	The Sync Adapter updates the Android database to synchronize with remote database	• Fails to update the android database sync with remote database
AC6	MVTS API Requestor	A JSON Object with all the updated data rows pertaining to the API call is returned	• Fails to return the JSON Object containing updated rows
WC1	AJAX Handler	The module returns a JSON Response containing the requested MVTS Component data	• Fails to return a JSON response containing the requested MVTS component data
WC2	MVTS Component Controller	The module returns a formatted JSON response containing all the necessary data to load an MVTS Component in the GUI	• Fails to return a formatted JSON response containing all the necessary data
RC1	Remote DB Controller	String array as a result set for the requested SQL query is returned	• Fails to return String array as a result set
M1	Request Processor	Validated API call request is received back from OAuth controller module	• Fails to receive the validated API call from the OAuth Controller
M1	API Processor	A string array containing all the relevant data for that particular API request is returned	• Fails to return the string array containing all the relevant data
M3	Response Generator	HTTP response message containing the user action requested data is generated by this module	• Fails to generate a HTTP response containing the user action requested
MO1	OAuth Controller	Boolean variable indicating authentication status is returned	• Fails to return a Boolean variable
MO2	Token Controller	The module will generates a cookie file containing user session information	• Fails to generate a cookie file containing user session information
G1	GCM Sender	JSON Object containing information about updated rows in the Remote DB is generated	• Fails to generate the JSON object that contains information about updated rows
PD1	PDF Generator	URL for the generated PDF report document is returned. Upon entering the URL in a web browser	• Fails to generate URL for the PDF • URL is incorrect

		the PDF document is opened	
AD1	Android Database	SQL lite result-sets as String array corresponding to the queries is returned	• Fails to return result sets as String array
RD1	Remote Database	Result-sets as String array corresponding to the respective queries is returned	• Fails to return result set as String array

Table 8.1 Unit Testing Acceptance Criteria

8.2 Component Testing

Test ID	Subsystem	Pass Criteria	Fail Criteria
AG1	Android GUI	The Android GUI displays different views corresponding to the user actions.	• Fails to display different views corresponding to different actions
AG2	Android GUI	The Android GUI relays user input data to Android Controller.	• Fails to pass input data to Android Controller
IP1	Index Pointer	Index pointer returns the complete URL request corresponding to the Subdirectory URL request as a String.	• Fails to return URL as a String corresponding to the Subdirectory URL
PR1	Page Router	The Page Router returns HTML content corresponding to the URL request.	• Fails to return the HTML content corresponding to the URL request
WG1	Web GUI	The Web GUI displays different views and results corresponding to the user actions.	• Fails to display different views corresponding to the user actions
WG2	Web GUI	The Web GUI relays user input data to Web Controller.	• Fails to send user input data to the Web Controller
WG3	Web GUI	The Web GUI receives all the MVTS components in the form of HTML content needed for each specific page.	• Fails to receive HTML content for the specific MVTS components
AC1	Android Controller	The Android Controller receives and returns an object to containing all the necessary information to create an activity view.	• Fails to receive an object that contains the input data • Fails to receive an object that contains the necessary information to create an activity view
AC2	Android Controller	The Android Controller receives result set from Android Database depending on the request.	• Fails to receive result set from Android Database
AC3	Android Controller	The Android Controller updates Android Database upon receiving	• Fails to update Android Database

		notification about changes in Remote Database.	
AC4	Android Controller	The Android Controller receives a JSON Object with result pertaining to the API call.	• Fails to receive a JSON object with result pertaining to the API call.
WC1	Web Controller	The Web Controller formats and relays results back to the user	• Fails to format the result appropriately • Fails to send the result back to the Web GUI
WC2	Web Controller	The Web Controller sends a JSON request pertaining to the events generated by user and receive a JSON response corresponding to the JSON request.	• Fails to send a JSON request pertaining to the events generated by the user • Fails to receive a JSON response
RDBC1	Remote DB Controller	String array as a result set for the requested SQL query is returned	• Fails to return a result set for the requested SQL query
MA1	MVTS API	The MVTS API responds to the various API calls pertaining to the request and provides the requested data.	• Fails to respond to the API calls • Fails to provide requested data
MO1	MVTS OAuth	The MVTS OAuth receives either successful user validation or unmatched user credentials.	• Fails to receive successful user validation or unmatched user credentials
PG1	PDF Generator	URL for the generated PDF report document is returned Upon entering the URL in a web browser the PDF document is opened	• Fails to generate URL for the PDF • URL is incorrect
GS1	GCM Sender	JSON Object containing information about updated rows in the Remote DB is generated	• Fails to generate the JSON object that contains information about updated rows
AD1	Android Database	SQL lite result-sets as String array corresponding to the queries should be returned	• Fails to return result sets as String array
RD1	Remote Database	Result-sets as String array corresponding to the respective queries should be returned.	• Fails to return result set as String array

Table 8.2 Component Testing Acceptance Criteria

8.3 Integration Testing

Test ID	System	Pass Criteria	Fail Criteria
I1	Presentation Layer	The Presentation Layer responds correctly to all the user requests.	• Fails to respond correctly to all user requests

I2	Presentation Layer	The Presentation Layer responds correctly to all the user requests.	• Fails to respond correctly to all user requests
I3	Application Layer	The Application Layer updates Android Database after Remote Database is updated.	• Fails to update Android Database
I4	Application Layer	The data is retrieved or stored in the database and result is shown to the user.	• Fails to retrieve data stored in the database
I5	Service	The user verification status and session status is returned and requested actions from Application Layer are processed.	• Fails to return user verification status and session information • Fails to process requested actions from the Application Layer
I6	Service	The Service layer provides Application Layer with JSON response pertaining to the request.	• Fails to provide the Application Layer with JSON responses corresponding to the requests.
I7	Data Storage	The Data Storage Layer stores and retrieves properly from Remote and Android Databases.	• Fails to store data properly in the Remote and Android Databases • Fails to retrieve data properly from the Remote Database.

Table 8.3 Integration Testing Acceptance Criteria

8.4 System Validation

Test ID	System Test	Pass Criteria	Fail Criteria
V1	Verify Login and Logout	The System logs the user in after successful username and password validation. The system logouts user after clicking logout button.	• System does not Login • System does not Logout
V2	Register Volunteers	System shall register new volunteers.	• System does not register new volunteers
V3	Verify input volunteer hours and Track Progress	The System modifies/updates relevant data in the database.	• System does not update or modify relevant data in the database
V4	Verify Notification	System notifies user about relevant changes.	• System does not notify user about relevant changes
V5	Verify Add/Edit/Delete/Sign up Opportunities/commitment	The System modifies/updates relevant data in the database.	• System does not update or modify relevant data in the database related to adding, editing, deleting and signing up for opportunities.

V6	Generate/Manage Reports	The System shall generate/manage relevant reports.	• System does not generate relevant reports
V7	Verify Ease of Use	The system allows user to reach anywhere on the website with maximum of 3 clicks.	• System takes more than 3 clicks to reach anywhere on the website
V8	Promote Members/Demote Facilitators	The System modifies/updates relevant data in the database.	• System does not update or modify the relevant data in the database related to promoting and demoting facilitators
V9	Verify Web Browser Compatibility	The System is opened in various browsers namely Internet Explorer 10+, Safari 6+, Google Chrome and Mozilla Firefox	• System does not function properly in browsers such as Chrome, Firefox, IE 10+, and Safari 6+.
V10	Verify Android App Availability	Android App is downloaded and installed in Android devices running Android version 4.1.2+	• Android App cannot be downloaded and/or installed in Android devices running Android version 4.1.2+

Table 8.4 System Validation Acceptance Criteria

9. Test Deliverables

The sections below describe the test deliverable that will be made available to the stakeholders. A list of bug reports will also be maintained for tracking errors in the future.

9.1 System Test Plan

The System Test Plan includes our overall strategy in detail for testing the system. It describes our approach for testing the system in parts and a whole in different phases.

9.2 Test Cases

The following information will be available for each of the test cases:

- Test ID – The Unique ID for the particular test.
- Test item – The module/subsystem name that will be tested.
- Purpose – The reason for conducting this test.
- Input – The Input that the test item will receive either through a test script or manually.
- Test Description – a brief description on how the test will be conducted.
- Expected Results – The expected behavior of the module upon receiving the input.
- Priority – Test priority of low, medium or high.

9.3 Test Report

The following report will be provided for each of the test that is conducted.

- Test ID – The Unique ID for the particular test.
- Tester – The name of the member who performed the test.
    - Date/Time Stamp – The date and time for the test performed.
    - Outcome – Test status of Passed or Failed.
    - Comments – comments concerning any issues with the testing process.

9.4 Bug Report

Based on the GitHub issues page, a bug report will be provided along with the documentation that will include the following details:

- Bug ID – The unique bug Identifier.
- Date/Timestamp – The date and time when the bug was found.

- Priority – Bug priority to be fixed: low, medium or high.
- Tester – Team member who performed the test.
- Fixer – Team member who fixed the bug, if applicable.
- Bug Description – Brief description of the bug.
- Status – Current status of the bug fix effort: Not Started, In Progress or Completed.
- Resolution Date/Timestamp – The date and time the bug was fixed, if applicable.

10. Test Schedule

WBS	Task Name	Planned Start	Planned Finish
3.4.4	**Testing Phase**	**4/19/15**	**5/07/15**
3.4.4.1	**Phase-1 Unit Testing**	**4/19/15**	**4/25/15**
3.4.4.1.1	UI display	4/19/15	4/25/15
3.4.4.1.2	Activity View	4/19/15	4/25/15
3.4.4.1.3	Notification view	4/19/15	4/25/15
3.4.4.1.4	Index pointer	4/19/15	4/25/15
3.4.4.1.5	Index view	4/19/15	4/25/15
3.4.4.1.6	MVTS page	4/19/15	4/25/15
3.4.4.1.7	MVTS component	4/19/15	4/25/15
3.4.4.1.8	Activity controller	4/19/15	4/25/15
3.4.4.1.9	Android DB controller	4/19/15	4/25/15
3.4.4.1.10	GCM Receiver	4/19/15	4/25/15
3.4.4.1.11	Notification Handler	4/19/15	4/25/15
3.4.4.1.12	Sync Adapter	4/19/15	4/25/15
3.4.4.1.13	MVTS API requestor	4/19/15	4/25/15
3.4.4.1.14	AJAX Handler	4/19/15	4/25/15
3.4.4.1.15	MVTS Component Controller	4/19/15	4/25/15
3.4.4.1.16	RDB Controller	4/19/15	4/25/15
3.4.4.1.17	Request Processor	4/19/15	4/25/15
3.4.4.1.18	API Processor	4/19/15	4/25/15
3.4.4.1.19	Response Generator	4/19/15	4/25/15
3.4.4.1.20	OAuth Controller	4/19/15	4/25/15
3.4.4.1.21	PDF Generator	4/19/15	4/25/15
3.4.4.1.22	Token Controller	4/19/15	4/25/15
3.4.4.1.23	GCM Sender	4/19/15	4/25/15
3.4.4.1.24	Android Database	4/19/15	4/25/15
3.4.4.1.25	Remote Database	4/19/15	4/25/15
3.4.4.1.26	Page Router	4/19/15	4/25/15
3.4.4.2	**Phase-2 Component Testing**	**4/25/15**	**4/30/15**
3.4.4.2.1	Android GUI	4/25/15	4/30/15
3.4.4.2.2	Web GUI	4/25/15	4/30/15
3.4.4.2.3	Android Controller	4/25/15	4/30/15
3.4.4.2.4	Web Controller	4/25/15	4/30/15
3.4.4.2.5	MVTS API	4/25/15	4/30/15
3.4.4.2.6	MVTS OAuth	4/25/15	4/30/15
3.4.4.3	**Phase-3 System Verification and validation**	**4/30/15**	**5/05/15**

	testing		
3.4.4.3.1.1	Presentation Layer Testing	4/30/15	5/05/15
3.4.4.3.1.2	Application Layer Testing	4/30/15	5/05/15
3.4.4.3.1.3	Service Layer Testing	4/30/15	5/05/15
3.4.4.3.1.4	Data Storage Layer Testing	4/30/15	5/05/15
3.4.4.3.2	**Validation**	**5/05/15**	**5/07/15**

Table 10.1 Test Schedule

11. Approvals

Name	Title	Signature	Date
Mr. O'dell	Project Supervisor		/ /
Dr. Linda McCalla	Project Sponsor		/ /
Dineth Hettiarachchi	Team Leader		/ /
Devkishen Sisodia	Team Member		/ /
Tasneem Devani	Team Member		/ /
Damber Khadka	Team Member		/ /
Samir Shrestha	Team Member		/ /

Table 11.1 Approval Signatures