
Software Test Plan Specification

for

Baggage Tracking Application

Version 1.0

Prepared by

Name	Student ID	E-mail
Tam Minh Chau Bui	1640110	tabui@uw.edu
Vadim M Goncharuk	1668360	vadimg@uw.edu
Binh Hue Hua	1650510	binhhua@uw.edu
Leah M Ruisenor	1569598	lmr81@uw.edu
Norris Spencer	1573221	nisj@uw.edu
Group Name	Lüg-er	

Prepared for

Course	TCSS 360 Software Development and QA
Instructor	Universal RFID Inc. & Dr. Eyhab Al-Masri
Date	12/08/2017

CONTENTS	II
REVISIONS	III
1 INTRODUCTION	
1.1 DOCUMENT PURPOSE	4
1.2 INTENDED AUDIENCE AND DOCUMENT OVERVIEW	4
1.3 DEFINITIONS, ACRONYMS AND ABBREVIATIONS	4
1.4 DOCUMENT CONVENTIONS	4
1.5 ASSUMPTIONS AND DEPENDENCIES	5
1.6 REFERENCES AND ACKNOWLEDGMENTS	5
2 OBJECTIVES AND TASKS	5
2.1 OBJECTIVES	5
3 SCOPE	6
3.1 TEST DOCUMENT SCOPE	6
4 TESTING STRATEGY	6
4.1 DEVELOPMENT TESTING	6
4.1.1 UNIT TESTING	6
4.1.2 TESTING ENVIRONMENT TESTING	6
4.1.3 SYSTEM TESTING	6
5 TESTING ENVIRONMENT	7
5.1 TESTING ENVIRONMENT	7
6 FEATURES TO BE TESTED	7
6.1 DATABASE CONNECTIVITY	7
6.2 CREATE NEW ACCOUNT	7
6.2.1 UPDATE ACCOUNT	7
6.3 TRACK BAGGAGE AND USER LOCATION	8
7 FEATURES NOT TO BE TESTED	8
APPENDIX A – GLOSSARY	8

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
Version 1.0	Tam Bui Vadim Goncharuk Binh Hue Hua Leah Ruisenor Norris Spencer	Initial Document Draft	12/08/17

1 Introduction

Universal RFID Inc. has tasked our group, Lüg-er, to create a passenger application that will allow the passenger to effortlessly track the physical location of their baggage. This application will take advantage of the existing tracking system emplaced in airports and aircraft that utilize RFID technology.

The goal is to provide the passenger with a peace of mind that airline passengers have historically not been able to enjoy. Lüg-er intends to make the interface as user-friendly as possible and, in the process, alleviate a major concern for airline passengers, lost baggage.

1.1 Document Purpose

This document is intended to outline the process for testing the Lüg-er baggage tracking application.

1.2 Intended Audience and Document Overview

This document is primarily intended for testers and developers of the Lüg-er baggage tracking application. Additionally, in the realm of the study of this subject, Dr. Eyhab Al-Masri and personnel deemed by him to be interested are targeted audience members as well. What follows is a detailed plan to test the whole system with identified limited components and basic software functionality of the application.

1.3 Definitions, Acronyms and Abbreviations

The following are terms and abbreviations contained in this document:

- API: Application Programming Interface
- IEEE: Institute of Electrical and Electronic Engineers
- ISP: Internet Service Provider
- LERP: Lüg-er Error Reporting Procedures
- Lüg-er: The name of the group tasked by Universal RFID Inc. to design the application
- PII: Personally Identifiable Information
- RFID: Radio Frequency Identification
- SMD: State Machine Diagram
- SRS: Software Requirement Specification

1.4 Document Conventions

- This document follows IEEE formatting requirements.
- Any reference to the application that is being directly developed by Lüg-er under the direction of Universal RFID Inc. is to be considered the Lüg-er application.

1.5 Assumptions and Dependencies

- It is assumed that the results of the tests are founded on the basis that the databases and related hardware (i.e. RFID antennas, RFID Readers, Software environment, etc.) are all within normal working standards.
- The assumptions that can affect the requirements stated in document involve the use and access of third-party databases and APIs.
- This application will need to have access to its own database in conjunction with access to the airline/airport database and Google Maps database.
- It is assumed that all airports will have RFID hand scanners, RFID antennas and a database that stores the given position of baggage at any given time.
- It is assumed that airplanes will also have RFID antennas installed. The airplane antennas will allow for tracking of baggage away from the airport's fixed antennas.
- It is assumed that all passenger baggage is manufactured with RFID tags.
- It is assumed that the baggage handler will have an RFID reader that can scan the RFID tag of the given bag when the passenger is checking in. The RFID number will be connected to the passenger's flight information which will then be part of the airport/airline database.
- To be able to display this location of the RFID tag (passenger baggage), the application must have access to and take advantage of the Google Maps API. This information, combined with the airport database, will display the passenger's current location and the location of the RFID tag (passenger baggage).
- It is assumed that the Airport database will store and process lost baggage requests.

1.6 References and Acknowledgments

- Google API terms of service, <https://developers.google.com/maps/terms> for details. Date Last Accessed 30 Oct 2017

2 Objectives

2.1 Objectives

The objective for the testing of this application is to ensure that the product is functioning at peak efficiency. In order to ensure this, these areas need to be verified: account creation, update user information, database connectivity, track user baggage, and user location.

3 Scope

3.1 Test Document Scope

The scope of this document is limited to the testing of the Track Baggage and related functions, to test the connectivity of the related servers/databases, and to validate the data returned from these servers. Anything outside this scope should be referred to the appropriate document for further guidance.

4 Testing Strategy

4.1 Development Testing

The subsequent tests will be primarily tested through batch testing to automatically verify the integrity of the software application, the connectivity of the databases, and the system as a whole. As the batch files are run, 1/10 processes will be system tested. Any errors should be logged as per LERP.

4.1.1 Unit Testing

The batch file to test the Account Creation must be designed to repeatedly create “n” number of accounts and subsequently distribute emails to mock accounts to verify that only upon verification of the email accounts, are the user accounts activated in the Lüg-er database.

4.1.2 Component Testing

The batch file testing will be created to run both database connectivity and the Track Baggage process. The purpose is to verify the testing components interactions is successful under repeated processes.

4.1.3 System Testing

The batch files from the unit and component testing will be combined and used to test the system as a whole. The overall purpose is to verify the interconnectivity required between the various databases and the application.

5 Testing Environment

5.1 Testing Environment

The testing for this document is primarily going to be done on the simudroid (Android app simulator) and simiOs (iOS simulator). While these will be used to run the system tests, the appropriate databases, including the Lüg-er database, will actually be pinged in order to determine if the appropriate results are being returned where required, and to validate the connectivity test to the databases.

6 Features “To” be Tested

6.1 Database Connectivity

- Precondition:
 - Develop component test designed to access Google, Lüg-er, and targeted airport databases to determine if connectivity is successful.
 - Internet connectivity must be established and operational.
 - Validation from Google and targeted airports that their databases are functional.
- Test summary:
 - Test name - Database Connectivity.
 - Test type - Test will access Google, Lüg-er, and targeted airport databases to verify the connection.
 - Verifying database connectivity - pass or fail.

6.2 Create New Account

- Precondition:
 - Develop a Java unit test program that will call createUser() and input the batch file containing the mock account input to the Lüg-er database.
 - Develop a Java program to provide email confirmation for the new account to be created.
- Test summary:
 - Test name - Create Account.
 - Test Detail - Test generates account information used to create mock account.
 - Verify database account creation (verify that the mock account was properly stored) - pass or fail.

6.2.1 Update User Information

- Precondition:
 - Develop unit test designed to randomly update “n” number of mock accounts by calling updateAcct().
- Test summary:
 - Test name - Update User Information.
 - Test Detail - Test generates account information used to randomly update mock accounts.
 - Verify database updates (verify user information was properly updated) - pass or fail.

6.3 Track Baggage and User Location

- Precondition:
 - Develop mock “Airport Database” to store RFID Lat/Long locations related to the mock accounts. These locations will be used to simulate the gathering of RFID locations from the database to allow the application to process the returned data.
 - Track Baggage must be capable of tracking and displaying a single baggage and its associated user’s location.
- Test summary:
 - Test name - Database Connectivity.
 - Test type - Test will access airport database to verify the connection.
 - Verifying database connectivity - pass or fail.

7 Features “Not” to be Tested

7.1 Features

Any aspect of the application not listed in the previous section, will not be the focus of this testing process. If an aspect of the application causes errors in the results of the testable areas, log the results per LERP.

Appendix A – Glossary

- (API) Application Programming Interface: A set of subroutine definitions, protocols, and tools for building application software.
- (IEEE) Institute of Electrical and Electronic Engineers: A professional association whose objectives are the educational and technical advancement of electrical and electronic engineering, telecommunications, computer engineering and allied disciplines.

- (LERP) Lüg-er Error Reporting Procedures
- (PII) Personally Identifiable Information: this includes username, addresses, phone number, email, password, and the number of baggage.
- (RFID) Radio Frequency Identification Device: it is a small electronic device that consists of a small chip and an antenna.