ARGO Testing Scanner

Requirements Document

Team 5

Developers

Amanda Bosson

Matthew Freese

Jessica Jennings

Mazen Lawand

Hisham Nabi

Abstract

This requirements document for the ARGO Testing Scanner application project will lay out all functional and nonfunctional requirements to be met upon completion of the project, as well as detail several use cases to aid in the design and implementation of the application. This document will serve as a guideline and provide the basic structure of the project through all remaining phases of planning, design, development, and testing.

Table of Contents

[**1. Introduction**](#_rruooq9dhfns) **4**

[1.1 Introduction to Requirements Document](#_skxmx8bp6p) 4

[1.2 Purpose of the Requirements Document](#_6g4f7lcc1aet) 4

[1.3 Structure of the Requirements Document](#_o60c515a82vu) 4

[1.4 Definitions](#_p7brjj95ksaz) 4

[**2. Functional Requirements**](#_skw1n2o0ezi) **4**

[**3. Use Case Model for Functional Requirements**](#_cf2wi17toe78) **5**

[2.1 Graphic Use Case Model](#_74qymeqsl3he) 5

[2.2 Textual Description](#_64qnd3kmyryz) 5

[**4. Rationale for Use Case Model**](#_tzaqtmr7x03z) **6**

[**5. Non-Functional Requirements**](#_2577ktxjk2aj) **6**

LIST OF FIGURES

# 

# 

# 1. Introduction

## 1.1 Introduction to Requirements Document

The ARGO Testing Scanner is an application that detects and notifies testers of errors and defects. This document provides a brief overview of the requirements for the ARGO Testing Scanner.

## 1.2 Purpose of the Requirements Document

This document serves to describe the functional and nonfunctional requirements for the ARGO Testing Scanner, as well as the use cases associated with the requirements.

## 1.3 Structure of the Requirements Document

The requirements document is divided into 2 major parts: functional and nonfunctional requirements. A rationale and use case model is included under functional requirements.

## 1.4 Definitions

Line of Business - abbreviated LOB, a particular application that is sold by Argo Data. Can be vertical - just one application with a specific goal, or horizontal - multiple applications working together to achieve one or more, more complex goals.

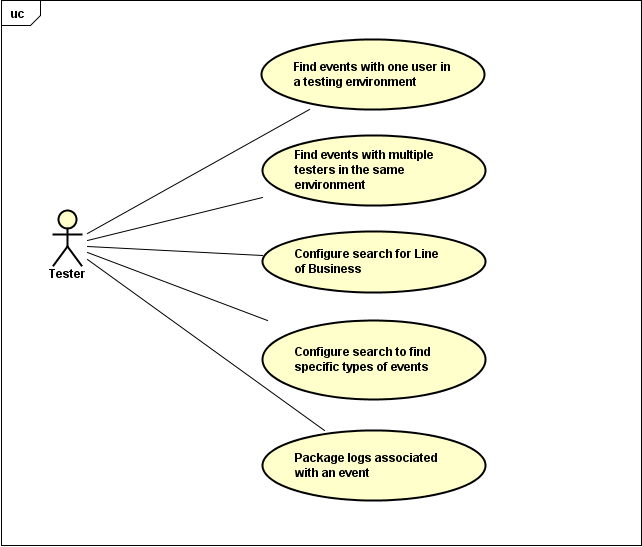
Test Environment - a computer environment that replicates ARGO Data’s operational environment for various LOBs

# 2. Functional Requirements

1. The application shall find events
2. The application shall identify the user who triggered an event
3. The application shall alert the user when an event is found
4. The application will give the user files related to the found event
5. The application shall have
   1. Configurable user to look for
   2. Configurable location to find files
   3. Configurable event severity to look for
   4. Multiple types of alert based on 5.C

# 3. Use Case Model for Functional Requirements

## 3.1 Graphic Use Case Model



## 3.2 Textual Description

UC1 - Find events with one user in a testing environment

Actor: Tester

Entry Condition: User starts the scanning pro

Normal Flow of Events:

1. System scans for events
2. System alerts user about the event
3. User acknowledges alert

Exit Condition: User stops scan

Exceptions (Alternate Flow of Events):

No event occurs

1. System remains in a ready state until event occurs

UC2 - Finding events with multiple testers running in same environment

Actors: Testers

Entry Condition: Each user has started the application

Normal Flow of Events:

1. System scans for events
2. System detects an event
3. System determines which active user caused the event
4. System alerts the specific user about the event
5. User acknowledges alert

Exit Condition: User stops scan

Exceptions (Alternate Flow of Events):

No event occurs

1. System

UC3 - Configure Search for Line of Business

Associated with requirement 5, file locations, severity, etc.

Actors: Testers

Entry Condition:

Normal Flow of Events:

Exit Condition:

Exceptions (Alternate Flow of Events):

UC4 - Configure search to find specific types of events

Actors: Testers

Entry Condition:

Normal Flow of Events:

Exit Condition:

Exceptions (Alternative Flow of Events):

UC5 - Package logs associated with an event

Actors: Testers

Entry Condition:

Normal Flow of Events

Exit Condition:

Exceptions(Alternate Flow of Events):

# 4. Rationale for Use Case Model

The reason for UC1 is to notify the tester about a certain event that is recorded in the log of the test environments application while the application is being tested. The purpose of UC2 is to have multiple users using the application at the same time. The main goal of UC2 is to determine which user produced the activity that was written to the log. Every LOB has different content that is being monitored. Different lines of business include teller payments, sales and service, and retail lending. UC3 requires each LOB to have different notifications for the user. The ability to configure the notifications based on log file specifications defines UC3 and UC4. Logs for a specific event must be packaged into a zip file and tied to a trouble ticket for each event that leads to a notification - this defines UC5.

# 

# 5. Non-Functional Requirements

1. The application shall not slow down system performance by more than 1%
2. The application shall have a graphical user interface
   1. The UI shall allow for configuration
   2. The UI shall allow a search to be started
3. Latency shall not be more than 1.5 seconds
4. The application shall show the user it is in a ready state when no events have been found

6. Evidence the Requirements Have Been Placed Under Configuration Management

7. References

*Astah Professional*. Computer software. *Astah Professional*. Vers. 7.1. Astah, 28 Sept. 2016. Web. 8 Feb. 2017.