

**ASRC**

***Automated Test Tool for GUIs***

**Design Document**



**Prepared by: *The Fire Ants***

**Version: 1.2**

**Date: *12/16/2019***

**Sponsors: *Joe Mcilvaine, Michelle Darby;* ASRC Federal**

**Table of Contents**

[**1.**](#_heading=h.gjdgxs) **Introduction 4**

[1.1. Purpose 4](#_heading=h.30j0zll)

[1.2. Scope 4](#_heading=h.1fob9te)

[1.3. System Environment 4](#_heading=h.tyjcwt)

[1.4. UI Design 4](#_heading=h.3dy6vkm)

[**2. Use Case Diagram 4**](#_heading=h.1t3h5sf)

[**3. Analysis Class Diagrams 4**](#_heading=h.4d34og8)

[3.1. General User 4](#_heading=h.2s8eyo1)

[**4. Class Diagrams 5**](#_heading=h.17dp8vu)

[4.1. Overview 5](#_heading=h.3rdcrjn)

[4.2. Main Activity and User Classes 5](#_heading=h.26in1rg)

[4.3. File Access Classes 6](#_heading=h.lnxbz9)

[**5. Activity Diagram 6**](#_heading=h.35nkun2)

[**6. Sequence Diagrams 6**](#_heading=h.1ksv4uv)

[6.1. User actions 6](#_heading=h.44sinio)

[6.1.1 Start Program 6](#_heading=h.2jxsxqh)

[6.1.2. Set Test Resolution 6](#_heading=h.z337ya)

[6.1.3. Test Pushbutton 6](#_heading=h.3j2qqm3)

[6.2. File Reading and Writing 7](#_heading=h.1y810tw)

[6.2.1. Write Output Files 7](#_heading=h.4i7ojhp)

[6.2.2. Read Input Files 7](#_heading=h.2bn6wsx)

[**7. Glossary of Terms 8**](#_heading=h.1ci93xb)

[**8. Revision History 8**](#_heading=h.3whwml4)

# Introduction

## 1.1. Purpose

This document describes the criteria and plan for the development and design of the *Automated Test Tool for GUIs.* The concept behind the automated test tool that is being developed is to drastically increase the speed and efficiency of the testing abilities of the customer. The purpose of this document is to introduce and explain how this goal can be achieved. This document will outline and explain the projects schedule as it applies to both the scope and the background of the project. Along with an explanation of the project structure and organization, this document will give a cost and benefit analysis to highlight the core reasons for the need to create this test tool.

## 1.2. Scope

The primary goal of the Automated Test Tool for GUIs is to be able to ensure the integrity of all code written for any component within Aegis, more specifically this project is meant to assist in increasing the efficiency of testing for the ASRC display system department. To achieve this goal, we must primarily address our key objectives.

1. Incorporating Motif and Linux Redhat
2. Utilizing GoogleTest and GoogleMock
3. Development of a tool that is universal among AEGIS components
4. Meet or exceed ARSC (customer) requirements
5. Effective delivery of quality software
6. Detail to technical excellence and solid design structure

## In scope

|  |  |
| --- | --- |
| 1 | Design an interface in motif that is able to record a series of clicks and actions and save them in a binary or XML file. |
| 2 | Create a C++ tool that is capable of taking the file as an input and playing a recording of it. The tool will scan through the binary or XML file and perform the actions as if they are physically being done |
| 3 | Using google mock, mock the functionality of the interface we want to test |
| 4 | Using google test, test the application’s GUI without needing to directly access it |
| 5 | Make sure that testing produces the same result regardless of monitor resolution, size or any other attributes. |

## Out of Scope

|  |  |
| --- | --- |
| 1 | Make the recording control the mouse clicks and drags all of the GUIs components before saving them to a binary file |
| 2 | Add a loading bar that tells the user what percent of the tests are complete |
| 3 | Adding an additional textbox that tells the user in live view which of the functionalities have been tested and if they pass or fail, while testing is taking place. |
| 4 | Adding an extra functionality in which the tool will not only tell the user what is wrong, but it will also tell them how to fix the problem. |

## 

## 1.3. System Environment

The minimum version requirement of this program is Red Hat Enterprise Linux (RHEL) release 7, including the X11 or later window system.

## 

## 1.4. UI Design

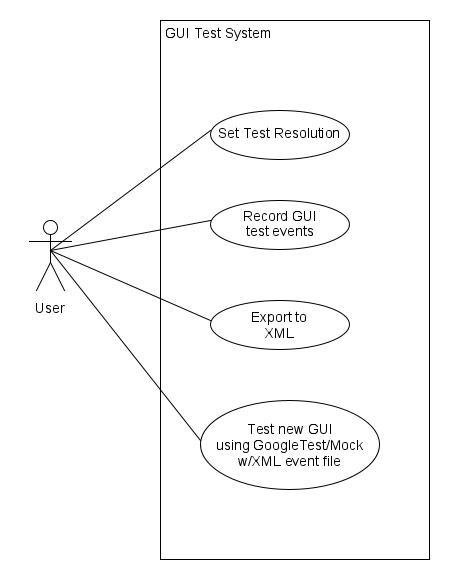
This section will provide snapshots of key interfaces that the user will interact with within the Automated Test Tool for GUIs application.

#### 

# 2. Use Case Diagram

**Actors:** General User

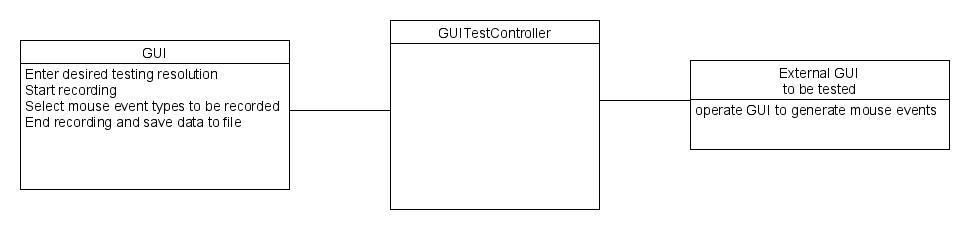
The following use Case Diagram shows all the actions that the Actors above can perform while using the Automated Test Tool for GUIs application.



# 3. Analysis Class Diagrams

## 3.1. General User

The Following Analysis Class Diagram shows all the classes a General User will interact with.



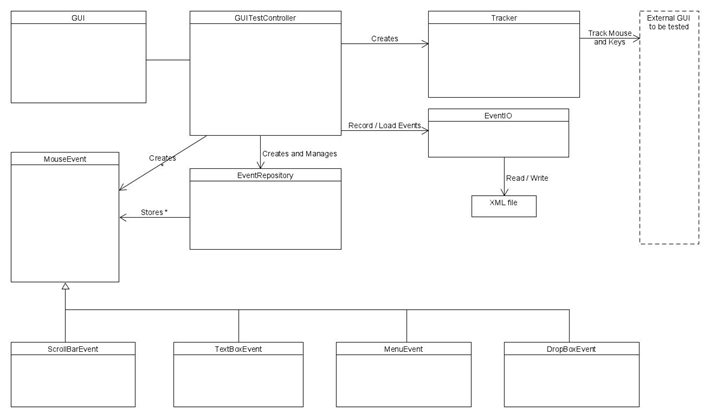
# 4. Class Diagrams

## 4.1. Overview

Gives an overview of all the Classes used within the Automated Test Tool for

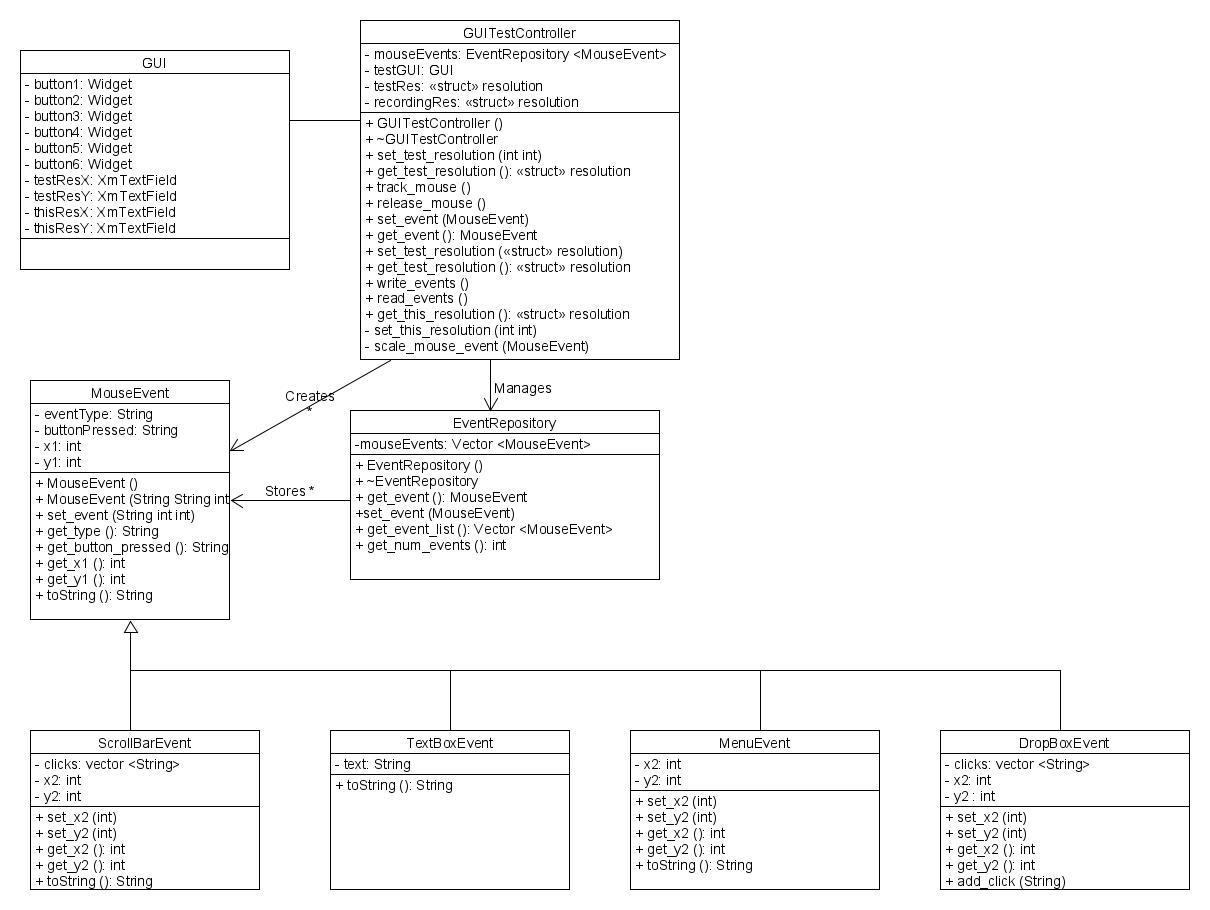
GUIs Application. This overview does not include Fields and Functions to give a

quick insight on how the classes interact.



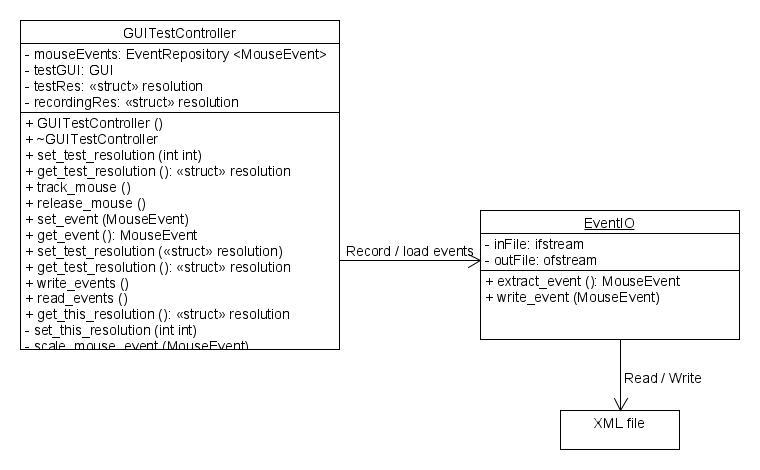
4.2. Main Activity and User Classes

Shows Class Diagrams for the Main Activity and all Classes associated with a User to show how these classes interact.



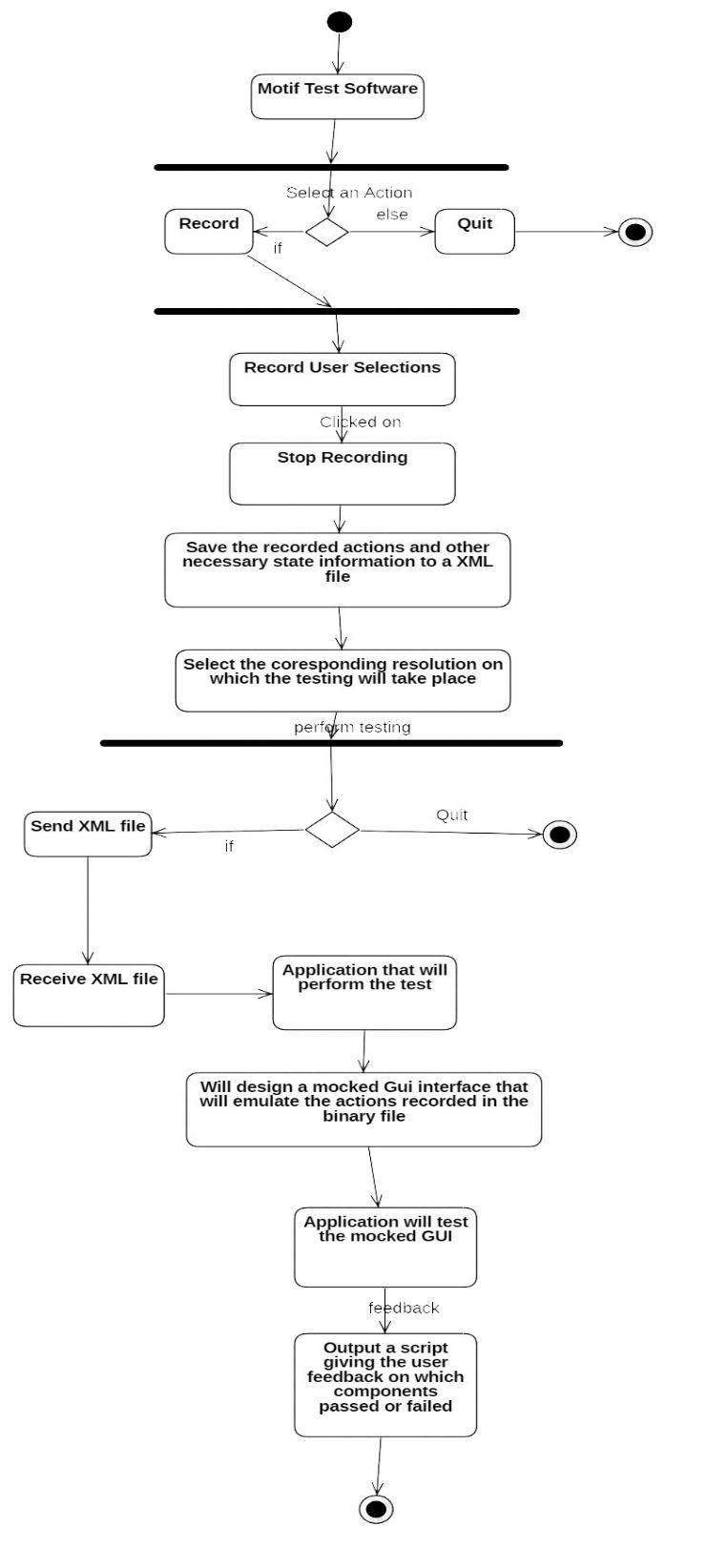
## 4.3. Report Log and File Classes

Shows Class Diagrams for all Classes associated with the Report Log to show how these classes interact.



# 5. Activity Diagram

This Activity Diagram is used to model system behaviors throughout the Automated Test Tool for GUIs application. Shows step-by-step activities a user can perform while using this application.



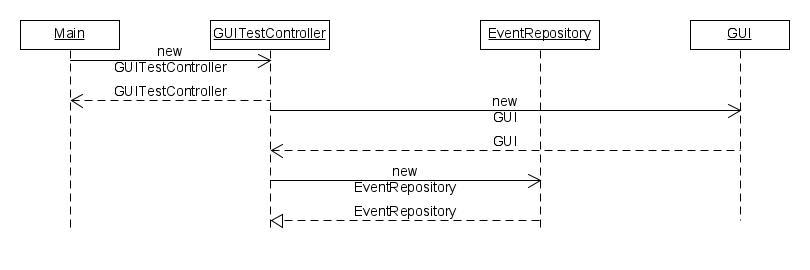
# 6. Sequence Diagrams

## 

## 6.1. User actions

### 6.1.1 Start Program

The following Sequence Diagram shows the program launch process.



### 6.1.2. Set Test Resolution

The following Sequence Diagram shows the set test resolution process. This resolution

will be used to scale mouse coordinates to desired test resolution.

### .

#### 

### 

### 6.1.3. Test PushButton

The following Sequence Diagram shows the Test Pushbutton process.

This process is similar for each type of widget tested, and will simply change the callback

function referenced based on the button type clicked.

#### 

## 6.2. File Reading and Writing

### 6.2.1. Write Output Report

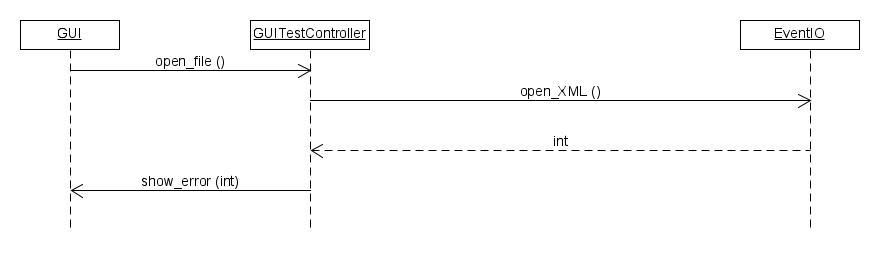
The following Sequence Diagram shows the Write Output File process within the Automated Test Tool for GUIsapplication.

#### 

### 

### 6.2.2. Read Input File(s)

The following Sequence Diagram shows the Read Input File process within the Automated Test Tool for GUIs application.

.

# 7. Glossary of Terms

|  |  |  |
| --- | --- | --- |
| **#** | **Term** | **Description** |
| 1 | ***Motif*** | ***API used to develop windowed applications for Linux*** |
| 2 | ***RHEL*** | ***Red Hat Enterprise Linux OS*** |
| 3 | ***X11*** | ***Windowing system for Linux systems*** |

# 8. Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Updated By** | **Comments** |
| 1.0 | ***10/25/2019*** | ***George Clelland*** | First Draft |
| 1.1 | ***10/28/2019*** | ***George Clelland*** | Final Draft (subject to revision) |
| 1.2 | ***12/16/2019*** | ***George Clelland*** | Updated UML overview |
|  |  |  |  |
|  |  |  |  |