# Klaros-Testmanagement Mobile Edition

**Version 4.2.0**

**Publication date June 24 2014**

verit informationssysteme

**Copyright © 2013-2014 verit Informationssysteme GmbH**

# Klaros-Testmanagement Mobile Edition: App for Android User Manual

by Sabrina Gidley, Fabian Klaffke, Patrick Reilly, and Torsten Stolpmann

Version 4.2.0

Publication date June 24 2014
Copyright © 2013-2014 verit Informationssysteme GmbH

**Abstract**

This document serves as the reference documentation for the Klaros-Testmanagement App for Android. It gives a detailed description of the user interface and the provided functionality.

**Legal Notice.** Copyright 2013-2014 verit Informationssysteme GmbH, Europaallee 10, 67657 Kaiserslautern, Germany. All rights reserved. This product or document is protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or documentation may be reproduced in any form by any means without prior written authorization of verit Informationssysteme GmbH and its licensors, if any.

## Trademarks

Android™ , Google™ , Google Play™ and Google Play Store™ are trademarks of Google Inc. and/or their affiliates.

BlueStacks™ and BlueStacks App Player™ are trademarks of BlueStacks and/or their affiliates.

Other names may be trademarks of their respective owners.

# Table of Contents

# List of Figures

# List of Tables

# Chapter 1. Key Features

The key features of the Klaros-Testmanagement App for Android are:

| | |
|---|---|
| Guided Manual Execution of Jobs | The App allows the execution of Jobs containing either Test Cases or Test Suites. |
| Display of assigned Jobs | The App can show all new Jobs from all Projects assigned to the user. |
| Advanced Methods of Test Protocoling | The App uses the integrated hardware of Android devices like the camera or the microphone to capture test results. |
| Offline Execution of Jobs | The App doesn't need to have an active connection to Klaros-Testmanagement in order to execute Jobs. |
| Store Test Artifacts locally | All data relevant to execute Jobs is stored in a local SQLite database on the Android device. |
| Pausing and resuming of Jobs | Jobs can be paused at any time during test execution and resumed at a later point. |
| Notifications | The App optionally issues Android notifications for all Jobs whose due dates are below a specified limit. |
| Synchronization | The App automatically synchronizes it's data every time it starts. |
| Security | Users of the App can only send or retrieve data from Klaros-Testmanagement if they can provide the proper authentication information. |
| Optimized for Smartphones and Tablets | The App uses the screen space more efficiently when running on an Android tablet. |
| Multi-User-Support | Multiple users can work on the same device using the App. |
| Localization | The App automatically changes it's language to English or German depending on the language that is set on the Android device. |

# Chapter 2. Introduction

The Klaros-Testmanagement App for Android is part of the Klaros-Testmanagement Mobile Edition. More information about the server side of Klaros-Testmanagement Mobile Edition is documented in the official documentation of Klaros-Testmanagement which can be found at http://www.klaros-testmanagement.com/files/doc/Klaros-Testmanagement%20User%20Manual.pdf .

# Chapter 3. Installation

The Klaros-Testmanagement App for Android is available for a lot of different types of Android devices. It is acquirable as a download on the Google Play Store. Before the installation can start, a set of requirements have to be fulfilled as described in the following section.

## 3.1. Client Prerequisites

In order to install the App, it is necessary to have a Google Account which grants access to the Google Play Store. Also, the mobile device needs to have at least Android 2.3 installed.

Since Klaros-Testmanagement version 4.1.0, the Klaros-Testmanagement App for Android is also compatible with the BlueStacks App Player ( http://www.bluestacks.com/app-player.html ). Please note that location tracking should be turned off in the Settings Screen (see Section 3.5, “Configuration ” ) when using BlueStacks.

A further prerequisite is an external storage on the device to store attachments such as videos or voice recordings. This external storage can either be an SD-Card or integrated into the device. This external storage shouldn't be removed while the App is running. For more information about external storage on Android devices see http://developer.android.com/guide/topics/data/data-storage.html#filesExternal

| | |
|---|---|
| Display Size | All display sizes are supported |
| Screen Density | ldpi or higher |
| Device Type | Smartphone or Tablet |
| Android Version | 2.3 or higher |

**Table 3.1. Supported Android devices**

## 3.2. Google Play Installation

The App can be downloaded from the Google Play Store. There are two ways to install the App, either from the mobile device or through a Web Browser on a desktop PC. To install from the device, the user should search for *Klaros-Testmanagement App* from *verit Informationssysteme GmbH* in the Google Play Store on the Android Device. To start the installation from a Desktop PC, open a Web Browser and visit https://play.google.com/store/apps/details?id=de.verit.klaros.mobile.android , where the App can be installed directly to the mobile device.

## 3.3. Updates

Updates for the App are delivered through the Google Play Store. If automatic updates are enabled in the device settings, updates will be downloaded and installed automatically. It is also possible to update the App manually through the Google Play Store. Please consult the official Google Play Help at https://support.google.com/googleplay/answer/113412?hl=en&ref_topic=2953009 for more information on updating Android apps.

If the App is obsolete and therefore unable to synchronize with the Server, a Dialog is displayed every time the user tries to synchronize with the Server. This Dialog has a "Go to Google Play" button which takes the user to the Google Play Store page of the App, where the most recent version can be downloaded.

## 3.4. First Start

After the installation is completed, the App can be started by pressing the Klaros-Testmanagement-logo, which can be found in the list of installed Apps. The App then synchronizes with the Server. Progress

is shown by the loading bar. The first time the App synchronizes with the Server, it tries to register and activate the device (See the official Klaros-Testmanagement documentation for more information about managing Mobile Devices in Klaros-Testmanagement).

Since only a limited number of devices can be activated at a time for every Klaros-Testmanagement Mobile Edition-Installation, it is possible that the message "Device not activated" is displayed. The device must then be activated by an administrator before it can be used with this Server. The Dashboard opens (See figure Figure 3.1, “The Dashboard” ). If the device is blocked by the Server, a similar message is displayed when trying to synchronize. The device must be unblocked by an administrator in order to synchronize with the Server.


**Figure 3.1. The Dashboard**

## 3.5. Configuration

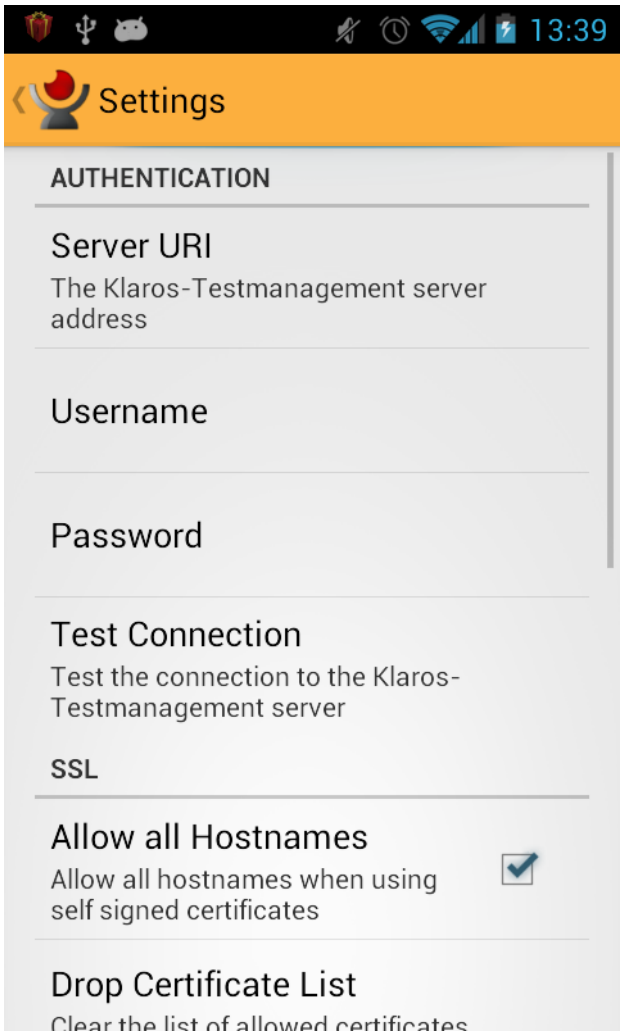


**Figure 3.2. The Settings Screen**

After the App has started, the user is provided with two options, *My Jobs* and *Settings*. After the first start, the settings should be defined so that the App can connect to the specified Klaros-Testmanagement Mobile Edition installation. Initially, the App is configured to connect with a public accessible preview version of Klaros-Testmanagement Mobile Edition. Pressing *Settings* opens the Settings screen (See figure Figure 3.2, "The Settings Screen" ).

### 3.5.1. Authentication

**Authentication Settings**

| | |
|---|---|
| *Server URI* | The address of your Klaros-Testmanagement server. |
| *Username* | The username of a registered user in your Klaros-Testmanagement installation. |
| *Password* | The password of the user. |
| *Test Connection* | Press this button to test the connection to your Klaros-Testmanagement installation. If the server cannot be contacted or if an invalid address has been in the *Server URI* field, an error message will be displayed. |

### Note

Please note that a user who wants to synchronize using his LDAP credentials has to log in via the Klaros-Testmanagement web login at least once in order to synchronize with Klaros-Testmanagement.

## 3.5.2. SSL

**SSL Settings**

| | |
|---|---|
| *Allow all Hostnames* | Allow all hostnames when using self signed certificates. |
| *Drop Certificate List*. | Press this button to clear the list of allowed certificates. |

The Klaros-Testmanagement App for Android supports secure communication with the Klaros-Testmanagement Server via SSL (Secure Socket Layer). If the Tomcat installation is configured to open an SSL port, the Klaros-Testmanagement App for Android can be synchronized using this secure port. The application URL and the port number need to be set (see Section 3.5.1, “Authentication ” ). For example, if your Klaros-Testmanagement installation is available at https://www.example.com/klaros-web on port 8443, the resulting URI would be https://www.example.com:8443/klaros-web.

If an unsigned certificate or an invalid certificate is used, the user will be presented an error message if they test the connection or try to synchronize with the server (See Figure 3.3, “The SSL warning message” ). The user now has the option to store the certificate on the device and allow future synchronization attempts. Certificates stored on the device can be removed by pressing the menu button on the Settings screen and then choosing the option "Delete Certificates".

It is possible that the user will receive an error message when trying to synchronize, even if they have previously stored the certificate on the device. This can occur if the certificate has an invalid hostname set or if the hostname doesn't match the server URI. In this case, it is possible to allow all hostnames for all certificates stored on the device, though it is highly recommended to use a certificate that matches the used server URI.

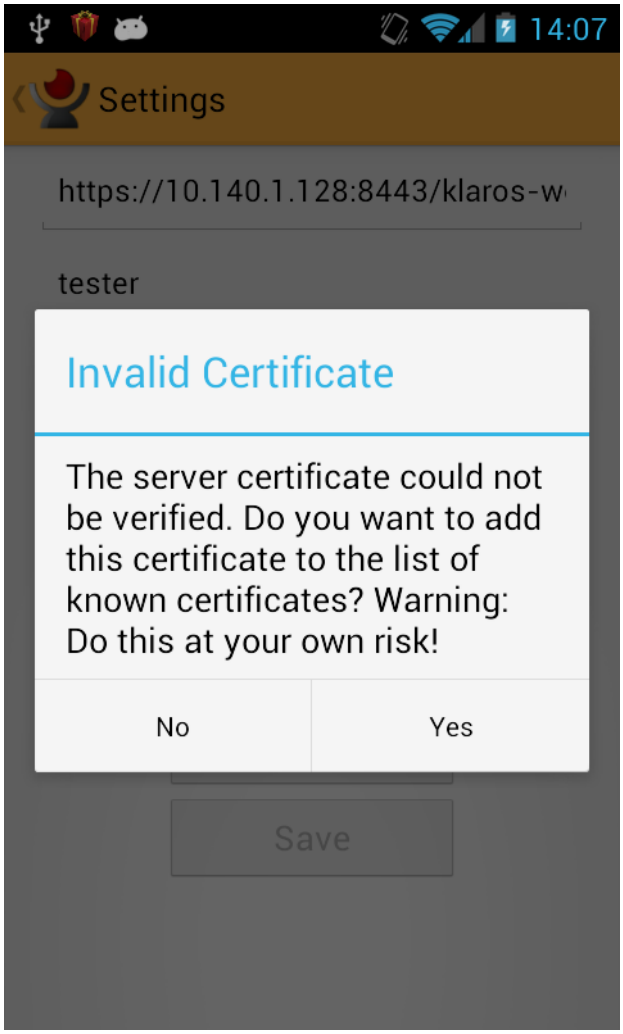


**Figure 3.3. The SSL warning message**

## 3.5.3. Usage Data

The Klaros-Testmanagement App for Android uses Google Analytics. In accordance with the Measurement Protocol / SDK / User ID Policy of Google Inc., we will now clarify which data is sent to Google Analytics. Only data relating to visited screens and crash reports is sent. No personal data is sent and all data is anonymized. This data helps us to improve the app. These settings can be changed at any time in the settings menu.

**Usage Data Settings**

| | |
|---|---|
| *Send Crash Reports* | Send anonymous, non-personalized crash reports to improve the App. |
| *Provide Usage Data* | Send anonymous, non-personalized usage data to improve the App. |

## 3.5.4. Application Settings

**Application Settings**

| | |
|---|---|
| *Location Tracking* | If this checkbox is enabled, the actual location information received via GPS or network location is appended to every test case step result. To use this feature, either GPS or Network Location must be enabled on the mobile device. |

| | |
|---|---|
| *Notifications* | This checkbox turns Android Notifications on or off (See chapter Chapter 6, *Notifications* for more information about Notifications in the App). Notifications are shown when a job reaches its due date. |
| *Delete Data* | Press this button to open a popup window from which you can delete all test related data from the device. |
| *Reset Settings* | Press this button to open a popup window from which you can reset all application settings to their default values. |

## Note

The App can manage data from multiple Klaros-Testmanagement Servers. It is possible to switch between different Servers by changing th URL in the settings.

# Chapter 4. The Job List

## 4.1. My Jobs

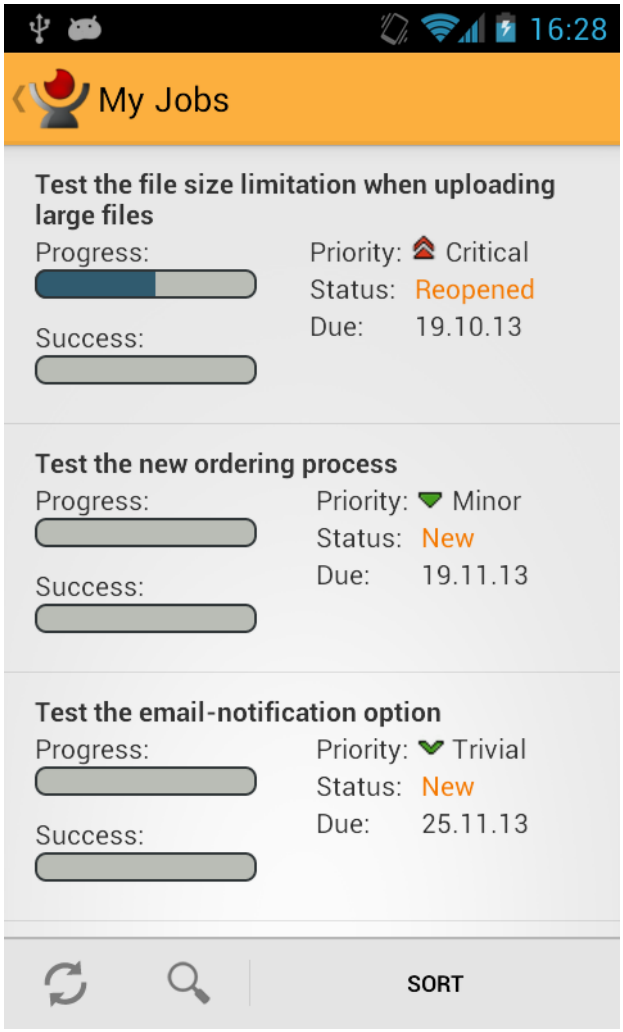


**Figure 4.1. The Job List**

In this screen, the user can see all Jobs that are assigned to him and also all unassigned Jobs, which can be executed by all users. The job list may be scrolled up and down, as shown by the scroll bar on the right. Each Element in the list represents a Job with the following attributes:

- Summary

  The Summary of the Job. This is usually a short description of the Job's content.

- Progress

  The progress bar indicates the overall progress of the job in percent.

- Success Rate

  The Success Rate bar indicates the overall success rate of the job in percent.

- Priority

  The priority of the Job ( *Blocker* , *Critical* , *Major* , *Minor* or *Trivial* ).

- Status

  The status of the Job ( *New* , *Reopened* , *In Progress* , *Resolved* , *Closed* or *Rejected* ).

- Due Date

  The due date of the Job, by which the job should be executed by the user.

## 4.2. Refreshing the Job List

Touching the *Synchronization* button in the ActionBar starts a synchronization process with the Klaros-Testmanagement Mobile Edition installation, which retrieves any new jobs assigned to the user since the last synchronization and also all unassigned Jobs.

## 4.3. Sorting the Job List

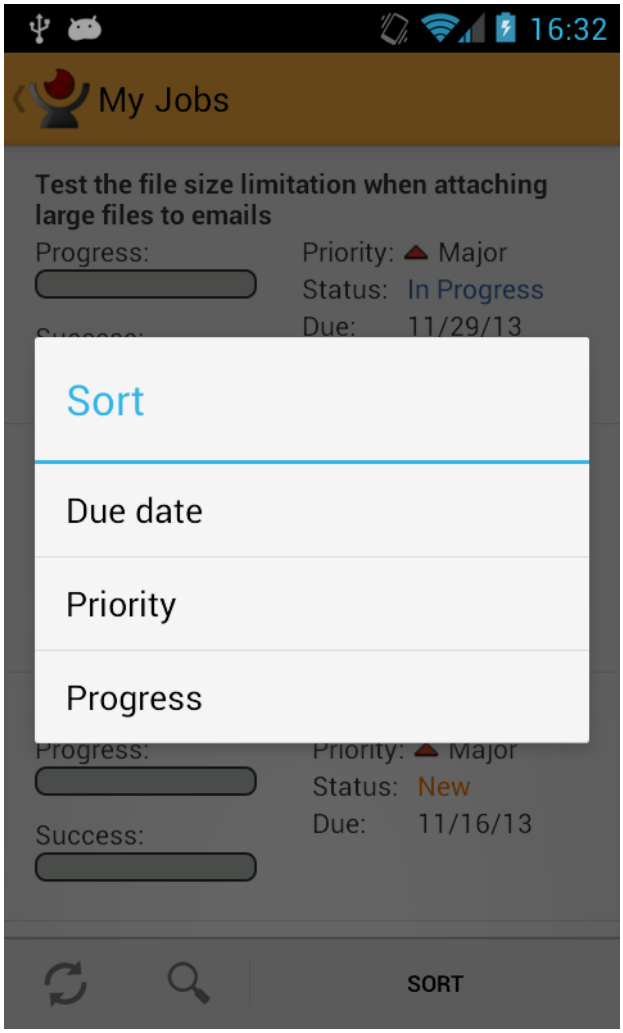


**Figure 4.2. Sorting the Joblist**

Touching the *Sort* button in the ActionBar displays three options for sorting the Jobs, *Sort by Due Date* , *Sort by Priority* and *Sort by Progress* . Choosing one of these options sorts the list of Jobs in ascending order according to the chosen sorting method. Choosing the same sorting method again sorts the Jobs in descending order.

## 4.4. Searching the Job List

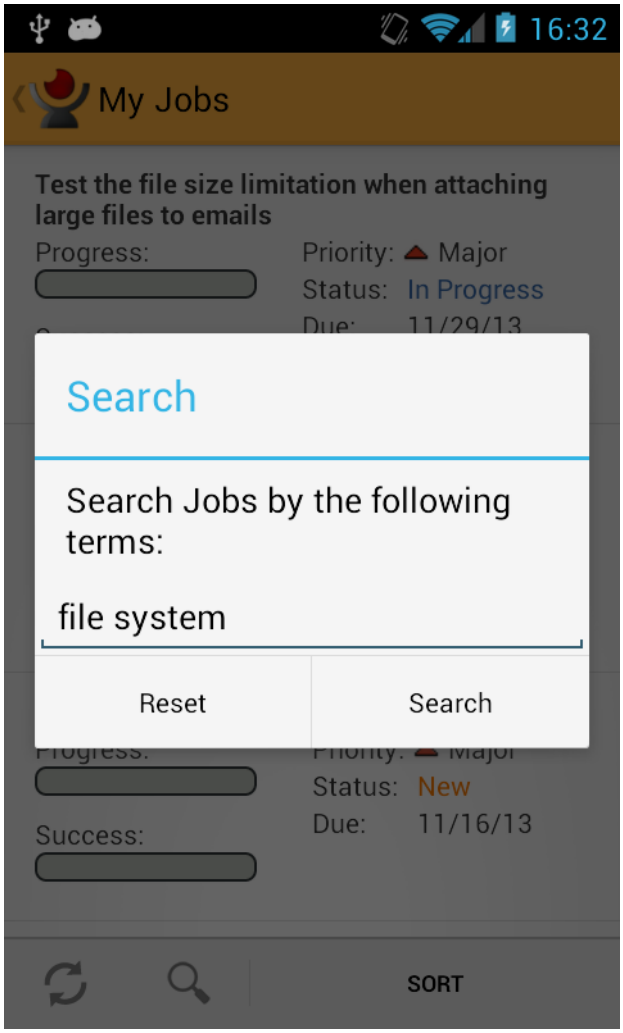


**Figure 4.3. Searching the Joblist**

By touching the *Magnifying Glass* icon in the ActionBar, the user can search the list of Jobs by title and description. The search can be resetted by touching the icon again.

## 4.5. Job Details

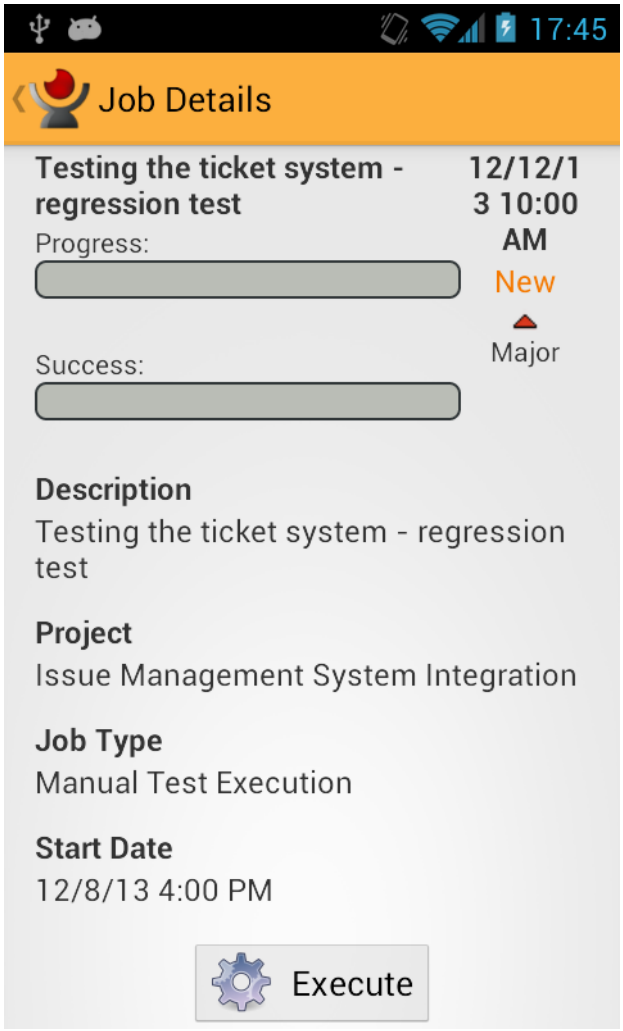


**Figure 4.4. The Details of a Job**

Touching one of the entries in the Jobs list opens the Job Details screen. In addition to the details in the Job List, the following details are displayed:

- Type

  The execution type of the Job ( *manual* or *automated* ).

- Estimated Time

  The time estimated for the Job to be executed.

- Test Case

  The Description and Summary of the Test Case included in the Job. A Job can either contain a Test Case or a Test Suite.

- Test Suite

  The Description and Summary of the Test Suite included in the Job. A Job can either contain a Test Case or a Test Suite.

- Test Environment

The Test Environment set for this Job. If there is no Test Environment set for the Job, the user will have to set one before the Job may be executed. A Test Environment may be selected using a drop-down box.

- System under Test

  The System under Test set for this Job. If there is no System under Test set for the Job, the user will have to set one before the Job may be executed. A System under Test may be selected using a drop-down box.

If both Test Environment and a System under Test are selected, the execution of the Job can be started by touching the button labeled *Execute this Job* .

# Chapter 5. Executing Jobs

## 5.1. Executing Jobs with Test Cases

When starting the Execution of a Job which contains a Test Case, the user is directly brought to the first Test Case Step of the Test Case in the *Test Case Step Screen* . This is the main screen in which the user can capture the results of the test execution.

## 5.2. Executing Jobs with Test Suites

When the user starts the Execution of a Job which contains a Test Suite, they are brought to the first Test Case in the Test Case Screen (See figure Figure 5.1, “The Test Case Screen” ).

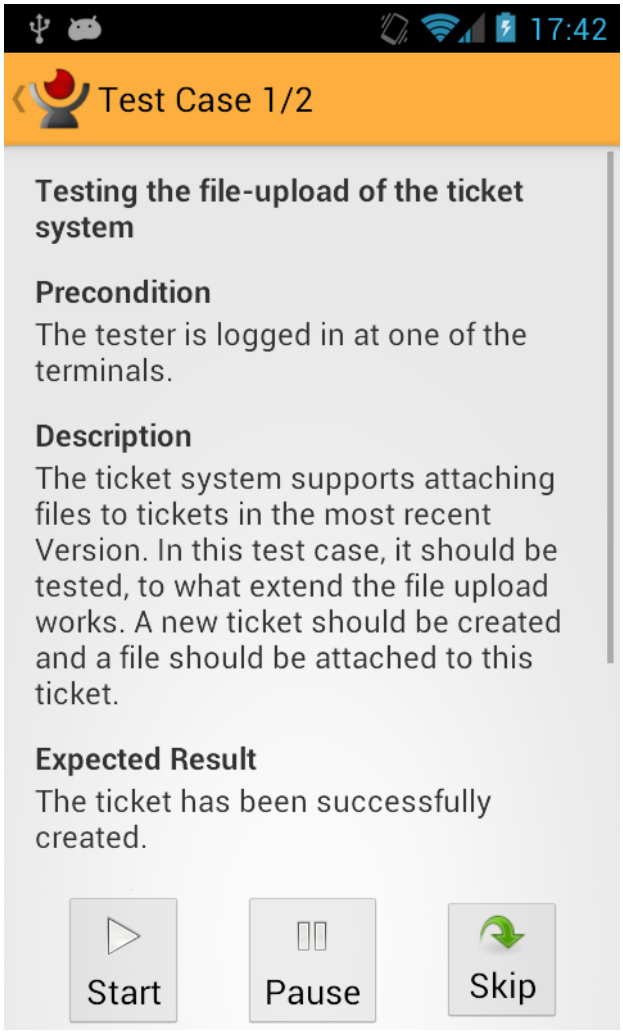


**Figure 5.1. The Test Case Screen**

This screen acts as an overview of the progress of the Test Suite execution. In the Action Bar at the top, the current Test Case number and the total number of Test Cases in this Test Suite is displayed. Below the Action Bar is the name of the Test Case and the list of Test Case Properties (See the official Klaros-Testmanagement documentation at http://www.klaros-testmanagement.com/files/doc/html/User-Manual.Define.Main.Test-Cases.Details.Edit-TestCases.html ). If the value of a property is empty, this property isn't shown in the list.

At the bottom of the screen are three buttons, labeled *Execute* , *Pause* and *Skip* . Touching *Execute* starts the execution of the Test Case and brings the user to the first Test Case Step in the *Test Case Step Screen*

. Touching *Pause* pauses the execution of the Job and brings the user back to the Job List. It is possible to resume the execution of the Job at a later point of time. Touching *Skip* skips this Test Case. The user is presented the option to skip the Test Case permanently or to execute it later (See figure Figure 5.2, "The Test Case Screen")

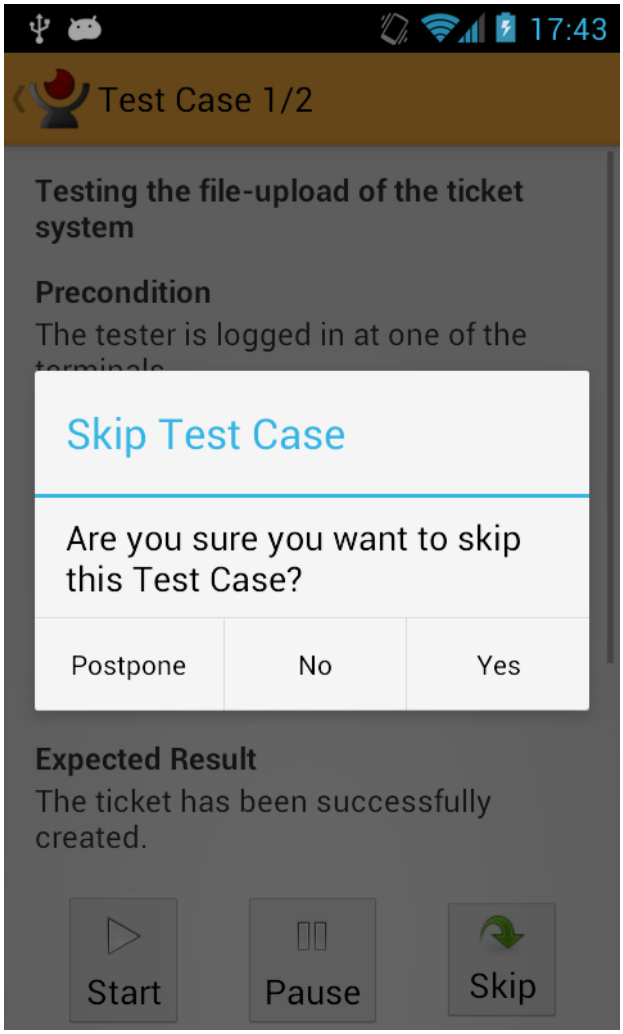


**Figure 5.2. The Test Case Screen**

## 5.3. Executing Test Case Steps

In the *Test Case Step Screen* the user can capture the results of the test execution. On the left of the Action Bar the current Test Case and the total number of Test Cases for this Job are displayed. Below is the position of the Test Case Step in the current Test Case. Beneath the total progress of the Test Case execution in form of a progress bar is shown.

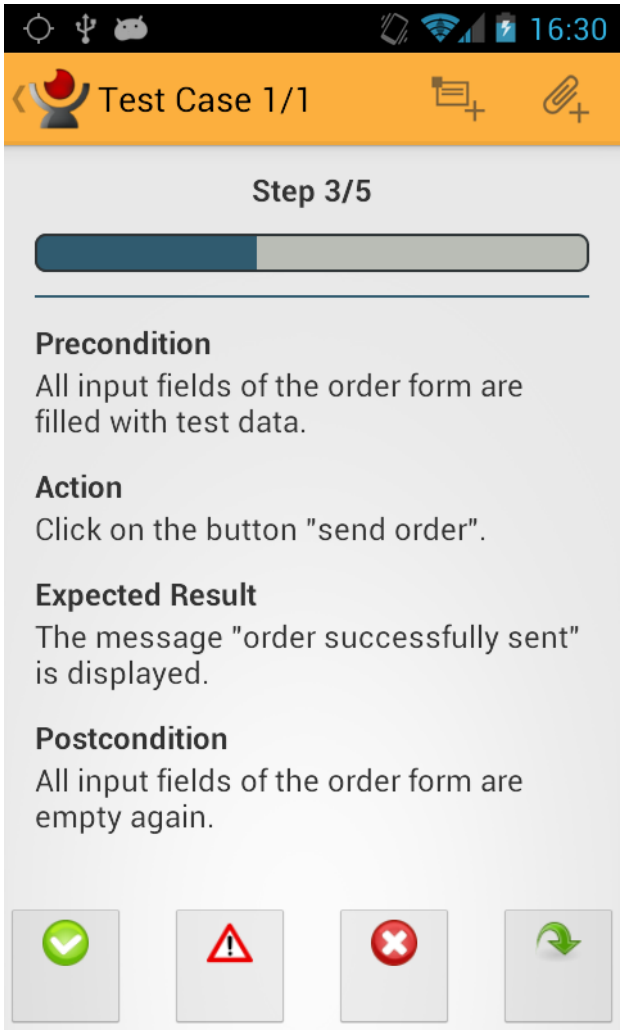


**Figure 5.3. The Test Case Step Screen**

The center of the *Test Case Step Screen* has the following information:

- Precondition

  The Precondition has to be true before the Test Case Step is executed.

- Action

  The Action that has to be performed by the user.

- Expected Result

  The Result that is expected for the Test Case Step to be successfully executed.

- Postcondition

  The Precondition has to be true after the Test Case Step has been executed.

On the bottom of the screen are four buttons with which the user can capture the test result. The buttons are labeled:

- Passed

  If the Postcondition and Expected Result match with the results after the user has executed the Action, the Test Case Step should be marked as passed.

- Failed

  If the Postcondition and Expected Result *don't* match with the results after the user has executed the Action, the Test Case Step should be marked as failed. Note that if a single Test Case Step is marked as failed, the result of the whole Test Case counts as failed.

- Error

  If the user is unable to execute the action, the Test Case should be marked as an error. Note that if at least one Test Case Step is marked as an error, the whole Test Case Result is marked as an error, too.

- Skip

  When touching the *Skip* button the user is provided with the option to either *skip this step* or to *skip all steps without a result* . If the latter is chosen, all Test Case Steps which haven't been executed yet are set to skipped.

The user has also the option to capture test data by using the internal devices of the Android Device. This option can be accessed by pressing the *Add Attachment* option in the Action Bar, represented by a paperclip. In the popup that opens, the user can choose between different options, depending on the hardware integrated into the device:

- Capture Photo

  Pressing this option opens up the Camera application installed on the device. The back camera is preferred. If no back camera is available on the device, the front camera is used instead.

- Capture Video

  Pressing this option opens up the Camera application installed on the device. The back camera is used preferred. If no back camera is available on the device, the front camera is used instead.

- Capture Audio

  This option opens up the *Capture Audio Screen* , in which the user can record audio (See figure Figure 5.4, “The Capture Audio Screen” ).

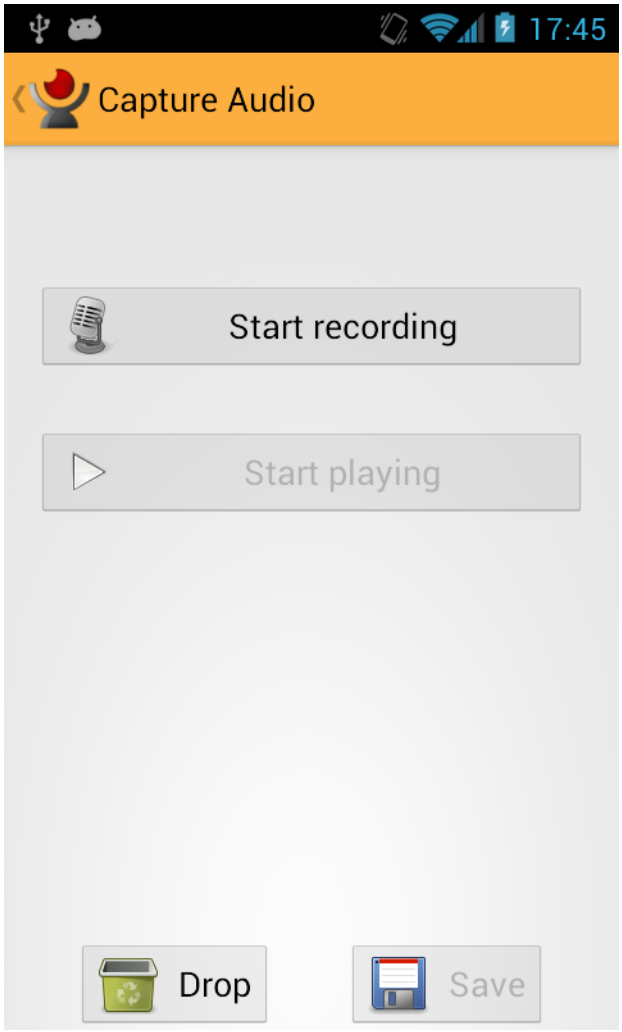


**Figure 5.4. The Capture Audio Screen**

After each Test Case Step, the Result is saved into the local database on the mobile device. The user is then brought to the next Test Case Step of the current Test Case. If all Test Case Steps have been marked with a result, the user is provided the option to finish the current Test Case. Choosing to finish the execution triggers different things, depending if the Job holds only this Test Case or a Test Suite (which contains multiple Test Cases). First, the overall Result of the Test Case is calculated from the results of the single Test Case Steps. When executing a Test Suite, the user is then brought to the next Test Case in the Test Case Screen.

If all Test Cases have been executed, the status and success rate of the Job is updated, and the updated Job and the results of the execution are sent to the Klaros-Testmanagement Mobile Edition installation. If no connection can be set up, the user can start a new synchronization attempt later by pressing the synchronization button on the Job List Screen. The user is then brought back to the Job List.

# Chapter 6. Notifications

The Klaros-Testmanagement App for Android supports Android notifications. If the due date of a Job assigned to the user is in 24 hours or less, a notification is displayed on the Android Status Bar. Touching this notification brings the user directly to the Job Details Screen of the corresponding Job. If the user is currently executing another Job, this job has to be paused or completed first. If a Job with a notification has been completed, the corresponding notification disappears. Job notifications can also be turned on or off in the Settings Screen (See section Section 3.5, “Configuration ” ).

# Glossary

## A

| | |
|---|---|
| Admin | See Administrator. |
| Administrator | User role that has access to all functionalities in Klaros-Testmanagement. |
| Artifact | An Artifact is a definable object like a Project, Iteration, Requirement, Test Environment, System under Test, Job, Test Case, Test Suite or Test Case. |

## B

| | |
|---|---|
| Bugzilla | Bugzilla is an open source bug tracking and testing tool. |

## C

| | |
|---|---|
| Category | Artifacts can be assigned to any number of user-defined categories. Using Categories Users can group related Artifacts together. |

## D

| | |
|---|---|
| Database | A database is a collection of information organized into interrelated tables of data and specifications of data objects. |

## E

| | |
|---|---|
| E-Mail | Electronic mail, often abbreviated as e-mail, is any method of creating, transmitting, or storing primarily text-based communications with digital communications systems. |
| Error | An error is the inability of the system to perform a test correctly. Not to be confused with Failure. |

## F

| | |
|---|---|
| Failure | A failure is a discrepancy between a computed, observed, or measured value or condition and the true, specified, or theoretically correct value or condition. Not to be confused with Error. |

## G

| | |
|---|---|
| GUIdancer | GUIdancer is an Eclipse-based tool for automated functional testing through the Graphical User Interface (GUI). |

## I

| | |
|---|---|
| Issue | The term Issue is a unit of work to accomplish an improvement in a system. An Issue could be a bug, a requested feature, task, missing documentation, and so forth. |
| Issue Management System | An Issue Management System (Issue Tracking System) is a software to manage Issues. |

| | |
|---|---|
| Issue Tracking System | See Issue Management System. |

# J

| | |
|---|---|
| Java | Java is a programming language. Most often, Java is used as a abbreviation for Java Runtime Environment, which needs to be installed in order to run Klaros-Testmanagement. |
| Java Runtime Environment | The Java Runtime environment needs to be installed in order to execute applications programmed in the Java programming language. |
| JavaScript | JavaScript is a scripting language most often used to add functionality to web pages. Most newer Web browsers can process JavaScript generated code. |
| Java Runtime Environment | See Java Runtime Environment. |
| JIRA | JIRA is a bug tracking, Issue tracking, and project management system by Atlassian Software . |
| Job | Jobs may consist of the execution of Test Cases, Test Suites or any other possible task. Jobs can be nested and assigned to individual users. The executions and results of jobs can be tracked by Klaros-Testmanagement. |
| Jubula | Jubula provides automated functional GUI testing for various types of applications. |
| JUnit | JUnit is a unit testing framework for the Java programming language. |

# M

| | |
|---|---|
| Manager | User role that has access to create, edit, delete and search for objects, run test cases and test suites, show results and generate reports. |
| Mantis | Mantis (MantisBT, Mantis Bug Tracker) is an open source bug tracking system. |

# O

| | |
|---|---|
| Operating System | An operating system (commonly abbreviated to either OS or O/S) is an interface between hardware and applications. It is responsible for the management and coordination of activities and the sharing of the limited resources of the computer. Common contemporary operating systems include Microsoft Windows, Mac OS, Linux, BSD and Solaris. |
| OS | See Operating System. |

# P

| | |
|---|---|
| Postcondition | Environmental and state conditions that must be fulfilled after the execution of a test or test procedure. |
| Precondition | Environmental and state conditions that must be fulfilled before the component or system can be executed with a particular test or test procedure. |
| Project | A project is the main unit that contains all other Artifacts that are needed to execute Test Cases. |

# Q

| | |
|---|---|
| QFTest | QF-Test is a professional tool for automated testing of Java and Web applications with a graphical user interface from Quality First Software . |

# R

| | |
|---|---|
| Redmine | Redmine is an open source bug tracking, Issue tracking, and project management system. |
| Role | A User can either be an Administrator, Manager or Tester. |

# S

| | |
|---|---|
| Selenium | *Selenium* is a web browser automation tool primarily used for automated testing of web apps. Selenium is able to produce *JUnit*-compatible test results, which can be imported into Klaros-Testmanagement. |
| SUT | See System under Test. |
| System Account | A System Account is a user that is not able to login at the login page and interactively control the application. System Accounts may be used for automated tasks like importing test results. |
| System under Test | A System under Test is used to represent a version of a software product that can be tested. |

# T

| | |
|---|---|
| Test Case | A Test Case is a set of input values, execution preconditions, expected results and execution post-conditions, developed for a particular objective or test condition, such as determine whether an application or software system meets its specifications. |
| Test Environment | Test Environments represent the extrinsic settings that may influence the test result. Examples for components of a Test Environments are the Operating System or an application server (e.g. Tomcat 7 on Ubuntu 12.10). |
| Tester | User role that has access to search for objects, run test cases and test suites and show results. |
| Test Execution | The process of running a test by the component or system under test, producing actual result(s). |
| Test Suite | A test suite is a set of test cases and the test cases can be executed in groups. The test suite will be used to verify and ensure that a product or system meets its design specifications and other requirements. |
| Trac | Trac is an open source bug tracking and Issue tracking system. |

# U

| | |
|---|---|
| URL | A Uniform Resource Locator (URL) specifies where an identified resource is available and the mechanism for retrieving it. Examples for URLs are http://www.klaros-testmanagement.com/" or file:///C:/klaros.txt |

# W

| | |
|---|---|
| Web browser | A Web browser is a software application which enables a user to display and interact with text, images and other information typically located on a Web page at a Web site on the World Wide Web or a local area network. |
| Windows | Windows is the name of several Operating Systems by Microsoft. |