



# **HoopStats Acceptance Test Report**

**Version 1.0**



HoopStats	Version: 1.00
Acceptance Test Report	Date: 2013-01-21

## Revision History

<b>Date</b>	<b>Version</b>	<b>Description</b>	<b>Author</b>
2012-12-30	0.1	Initial Draft This document is not in compliance with the requirements description document. the document will be updated ASAP	Andreas Köhle, Bal Krishna Nyaupane, Predrag Filipovikj, Igor Šarić, Dino Blažeka, Armindo Simões
2013-01-21	1.0	Final version	Predrag Filipovikj

HoopStats	Version: 1.00
Acceptance Test Report	Date: 2013-01-21

## Table of Contents

1.	Introduction	5
1.1	Purpose of this document	5
1.2	Intended Audience	5
1.3	Scope	5
1.4	Acronyms	5
1.4.1	Acronyms and abbreviations	5
1.5	References	6
2.	Test-plan introduction	6
2.1	Test items	6
2.2	Features to be tested	6
2.3	Features not to be tested	6
2.4	Test environment	7
3.	Approach	7
3.1	Approach to configuration and installation	7
3.2	Item pass/fail criteria	7
3.3	Documentation problems	7
3.4	Suspension criteria and resumption requirements	8
3.5	Responsibilities	8
3.5.1	Developers	8
3.5.2	Customer	8
4.	Test procedure	8
4.1	Test case specifications	8
4.1.1	Test Group Web Application (WA)	8
4.1.2	Test Group Android Application (AA)	20
5.	Approvals	26

HoopStats	Version: 1.00
Acceptance Test Report	Date: 2013-01-21

## 1. Introduction

### 1.1 Purpose of this document

The purpose of this document is to provide detailed information how to test the functionality of the developed clients for the HoopStats project. It is used to determine if the defined project requirements have been delivered and to verify that the expected outcomes have been reached.

### 1.2 Intended Audience

The document is intended for the following audience:

- Project Team: to have an overview of the tested functionalities and their outcome
- Customer: To verify that his specified requirements have been met
- Supervisor: Responsible for monitoring the status of the project, its direction and outcomes

### 1.3 Scope

This document provides detailed instructions about the parts and features to be tested as well as the test procedure and the individual test cases. Furthermore it describes the environment and conditions under which the tests had been conducted. At the end of each test, there is evaluation of the test which indicates whether the test has passed or failed.

### 1.4 Acronyms

#### 1.4.1 Acronyms and abbreviations

Acronym or abbreviation	Definitions
APK	Android application package file
UI	User Interface

HoopStats	Version: 1.00
Acceptance Test Report	Date: 2013-01-21

## 1.5 References

- ❖ Project Home
  - <http://www.fer.unizg.hr/rasip/dsd/projects/basketball>
- ❖ Project plan
  - [http://www.fer.unizg.hr/download/repository/Project\\_Plan%5B9%5D.pdf](http://www.fer.unizg.hr/download/repository/Project_Plan%5B9%5D.pdf)
- ❖ Requirements Definition
  - [http://www.fer.unizg.hr/download/repository/Requirements\\_Definition%5B9%5D.pdf](http://www.fer.unizg.hr/download/repository/Requirements_Definition%5B9%5D.pdf)
- ❖ Project Design
  - [http://www.fer.unizg.hr/download/repository/Design\\_Description%5B6%5D.pdf](http://www.fer.unizg.hr/download/repository/Design_Description%5B6%5D.pdf)
- ❖ Web client home
  - <http://www.hoopstats.tk/>
- ❖ Client for mobile platform – Android client
  - Installed locally on device.

## 2. Test-plan introduction

This section introduces the items and features which will be tested in the HoopStats system. Furthermore it describes the test approach as well as testing criteria and limitations.

### 2.1 Test items

The HoopStats project consists of two clients that will be the major test items for the acceptance test and will be shortly described in the following paragraphs:

1. **Web client:** Application which is accessible via the internet and delivers basketball statistics depending on the user input.
2. **Client for mobile platform – Android application:** Used for mobile access of the HoopStats database to retrieve basketball statistics on a android mobile device.

### 2.2 Features to be tested

The test cases defined in this document, are designed to target the implemented features in both clients and to determine whether they fulfill the requirements defined in the respective document. Therefore, the tests are divided in two sections:

1. **User interface (UI) test set:** Tests cases to ensure that the provided UI works as expected.
2. **Business Logic test set:** Test cases to ensure that the correct data is delivered and that inputs are correctly.

### 2.3 Features not to be tested

Under the constraints of the project the following features will not be tested:

1. Performance: e.g. Test cases that target the performance of each of the clients, under concurrent usage.
2. Security: Test cases to ensure that the web application is secured from unauthorized access to the database.

HoopStats	Version: 1.00
Acceptance Test Report	Date: 2013-01-21

## 2.4 Test environment

The following sections describe the environment necessary to conduct the test cases defined in this document.

### Hardware:

- Personal computer, Laptop.
- Mobile device – smartphone.

### Software:

- Operating system (Windows, Linux, OS X)
- Web Browser (e.g. Firefox, Chrome, Safari)
- Android OS (minimum Android version: 3.0 Honeycomb)

### Other:

- Internet access.
- JavaScript must be enabled in web browser.

## 3. Approach

All of the acceptance tests defined in the document have been performed by the development team. Acceptance test report is tailored to enable reproduction of the test cases by third persons.

### 3.1 Approach to configuration and installation

The web application is available via the Internet. The test cases are written based on the live production version, deployed on the project's virtual machine. To be able to access the web application only a web browser and a working internet connection are required. No additional efforts have to be undertaken by the customer and supervisor to install and configure the web application for testing. For easier initial navigation, hyperlinks to the web application have been provided for the external stakeholders.

For the Android application an Android application package file (APK) will be provided, which is used to install the application on the mobile device of the customer and supervisor. The installation is automated by default, so the users (client and the supervisor) should be able to install it by themselves.

In terms of the Android application it has to be ensured, that the operating system version matches the defined version of section 6. With older mobile devices that do not support the minimal required OS version the application will not start. Furthermore the Internet connection has to be enabled and internet access for the HoopStats has to be granted. Without this features enabled, the application is not testable on a mobile device.

### 3.2 Item pass/fail criteria

The following table shows all the possible test types for a test case. Overall each of the provided test cases in this document has to pass in the described way to successfully pass the acceptance test.

Test Type	Description
<b>Positive</b>	Test case is passed when the defined input delivers the expected outcome. Otherwise it fails
<b>Negative</b>	An error should be reported for each disallowed and unsupported input then passes. If the error goes unnoticed without any error message or warning (an unhandled error of any sort) then it fails.

### 3.3 Documentation problems

As all of the related documents are already written, no major problems should occur that can influence the contents of the related documents. If the test cases reveal severe mismatches in the design of the application according to the requirements and changes have to be applied to the applications, then the corresponding documents will be rewritten.

HoopStats	Version: 1.00
Acceptance Test Report	Date: 2013-01-21

### 3.4 Suspension criteria and resumption requirements

The testing process can be paused and continued in any point of time. The developed test cases have a short runtime and can be performed rather quickly. Therefore no specific process for pausing and resuming testing has to be developed. If bugs are discovered during the testing phase, the test process will continue. After the bug is fixed, the test case where the bug occurred is run again.

### 3.5 Responsibilities

In this section the responsibilities of the developers and the customer during the test are described.

#### 3.5.1 Developers

The developers of the HoopStats project have to react to bugs identified during the test run and fix them. Furthermore they have to assist the customer if questions occur during the test.

#### 3.5.2 Customer

The customer has to inform the development in case of bugs occur during the test of the application.

## 4. Test procedure

In the following section all the test cases are described and grouped together for the different items and functionalities. The cases are divided in the two mayor groups, tests related to the android application and test related to the web application.

### 4.1 Test case specifications

#### 4.1.1 Test Group Web Application (WA)

##### 4.1.1.1 Player regular seasons and playoff data flexible querying (WAPRSPOFQ)

<i>Test Name</i>	<b>Player regular seasons and playoff search without conditions</b>	<b>WAPRSPOFQ001</b>
<i>Description</i>	Test case is to validate the functionality of the flexible querying feature when no conditions are imposed by the user	
<i>Test type</i>	Positive	
<i>Preconditions</i>	<ol style="list-style-type: none"> <li>1. User is connected to the internet</li> <li>2. The web application is up and running</li> <li>3. User is on <a href="http://hoopstats.tk/Player/PlayerAdvancedSearchQuery">http://hoopstats.tk/Player/PlayerAdvancedSearchQuery</a> page</li> <li>4. JavaScript must be enabled</li> </ol>	
<i>Input definition</i>	<ol style="list-style-type: none"> <li>1. User selects at least one column as result of the query.</li> <li>2. User presses execute button</li> </ol>	
<i>Output definition</i>	Results from the query are displayed in two tables. First table is holding the data for the regular seasons and the second for the playoff data. The same projections and potential conditions are imposed on the both tables.	
<i>Remarks:</i>	No remarks	
<i>Execution Status:</i>	Pass.	

<i>Test Name</i>	<b>Player regular seasons and playoff search with conditions</b>	<b>WAPRSPOFQ002</b>
<i>Description</i>	Test case is to validate the functionality of the flexible querying feature when user imposes conditions	
<i>Test type</i>	Positive	
<i>Preconditions</i>	<ol style="list-style-type: none"> <li>1. User is connected to the internet</li> <li>2. The web application is up and running</li> <li>3. User is on <a href="http://hoopstats.tk/Player/PlayerAdvancedSearchQuery">http://hoopstats.tk/Player/PlayerAdvancedSearchQuery</a> page</li> <li>4. JavaScript must be enabled</li> </ol>	

HoopStats	Version: 1.00
Acceptance Test Report	Date: 2013-01-21

<i>Input definition</i>	<ol style="list-style-type: none"> <li>1. User selects at least one column as result of the query</li> <li>2. User select at least one column as a condition</li> <li>3. User select value for that column (property) <ol style="list-style-type: none"> <li>a. If the column is numeric – user can also enter the range less than/ equal to/ greater than the specified value</li> <li>b. If the column is non-numeric (player name) the value is exactly matched upon the corresponding column.</li> </ol> </li> </ol>
<i>Output definition</i>	Results from the query are displayed in two tables. First table is holding the data for the regular seasons and the second for the playoff data. The same projections and potential conditions are imposed on the both tables.
<i>Remarks:</i>	No remarks
<i>Execution Status:</i>	Pass.

<i>Test Name</i>	<b>Player regular seasons and playoff search without selecting projections.</b>	<b>WAPRSPOFQ003</b>
<i>Description</i>	Test case when user tries to execute the query without selecting any column as projection.	
<i>Test type</i>	Negative	
<i>Preconditions</i>	<ol style="list-style-type: none"> <li>1. User is connected to the internet</li> <li>2. The web application is up and running</li> <li>3. User is on <a href="http://hoopstats.tk/Player/PlayerAdvancedSearchQuery">http://hoopstats.tk/Player/PlayerAdvancedSearchQuery</a> page</li> <li>4. JavaScript must be enabled</li> </ol>	
<i>Input definition</i>	<ol style="list-style-type: none"> <li>1. There are no columns selected</li> <li>2. Imposing the conditions is not affecting the behavior</li> <li>3. User presses the execute button.</li> </ol>	
<i>Output definition</i>	JavaScript validation is present for the form. If no columns are selected as output there is popup message that warns user to select columns. The form is not submitted and the user stays on the current page.	
<i>Remarks:</i>	Validation is JavaScript based. In case of JavaScript being disabled in the browser, form will be submitted, but query will not yield any results.	
<i>Execution Status:</i>	Pass.	

<i>Test Name</i>	<b>Player regular seasons and playoff search with conditions without values</b>	<b>WAPRSPOFQ004</b>
<i>Description</i>	Test case when user tries to execute the query without entering value for the imposed condition(s).	
<i>Test type</i>	Negative	
<i>Preconditions</i>	<ol style="list-style-type: none"> <li>1. User is connected to the internet</li> <li>2. The web application is up and running</li> <li>3. User is on <a href="http://hoopstats.tk/Player/PlayerAdvancedSearchQuery">http://hoopstats.tk/Player/PlayerAdvancedSearchQuery</a> page</li> <li>4. JavaScript must be enabled</li> </ol>	
<i>Input definition</i>	<ol style="list-style-type: none"> <li>1. User selects output columns.</li> <li>2. User select columns as filtering condition(s) but is not entering values for the selected columns.</li> <li>3. User presses the execute button.</li> </ol>	
<i>Output definition</i>	JavaScript validation is present for the form. If there are columns in the conditions part that have no value (empty), the form will not be submitted and the columns with no values will be properly marked.	
<i>Remarks:</i>	Validation is JavaScript based. In case of JavaScript being disabled in the browser, form will be submitted, but all the conditions that have no value will be ignored by the query parsing engine. If all the conditions are without values, query will be executed as <b>WAPRSPO001</b> .	

HoopStats	Version: 1.00
Acceptance Test Report	Date: 2013-01-21

<i>Execution Status:</i>	Pass.
--------------------------	-------

<i>Test Name</i>	<b>Player regular seasons and playoff search with conditions with non-numerical value for numerical columns.</b>	<b>WAPRSPOFQ005</b>
<i>Description</i>	Test case when user tries to enter non numeric value for numeric field.	
<i>Test type</i>	Negative	
<i>Preconditions</i>	<ol style="list-style-type: none"> <li>1. User is connected to the internet</li> <li>2. The web application is up and running</li> <li>3. User is on <a href="http://hoopstats.tk/Player/PlayerAdvancedSearchQuery">http://hoopstats.tk/Player/PlayerAdvancedSearchQuery</a> page</li> <li>4. JavaScript must be enabled</li> </ol>	
<i>Input definition</i>	<ol style="list-style-type: none"> <li>1. User selects columns as projections.</li> <li>2. User selects columns as conditions</li> <li>3. User enters non – numeric value for numeric column</li> <li>4. User presses the execute button.</li> </ol>	
<i>Output definition</i>	JavaScript validation is present for the form. User will not be allowed to enter non-numeric character in the numeric column field.	
<i>Remarks:</i>	Validation is JavaScript based. In case of JavaScript being disabled in the browser, form will be submitted, but the query parser validator will ignore non-numeric inputs for the numeric columns and treat them as invalid. If all the imposed conditions are non-valid the query will be executed as <b>WAPRSPO001</b> .	
<i>Execution Status:</i>	Pass.	

<i>Test Name</i>	<b>Player regular seasons and playoff search with non-reasonable input.</b>	<b>WAPRSPOFQ006</b>
<i>Description</i>	Test case when user enters non-reasonable value for the conditions.	
<i>Test type</i>	Negative	
<i>Preconditions</i>	<ol style="list-style-type: none"> <li>1. User is connected to the internet</li> <li>2. The web application is up and running</li> <li>3. User is on <a href="http://hoopstats.tk/Player/PlayerAdvancedSearchQuery">http://hoopstats.tk/Player/PlayerAdvancedSearchQuery</a> page</li> <li>4. JavaScript must be enabled</li> </ol>	
<i>Input definition</i>	<ol style="list-style-type: none"> <li>1. User selects columns as projections</li> <li>2. User selects conditions</li> <li>3. User enters non-reasonable values for input (very large numeric or some illogical string for non-numeric values)</li> <li>4. User presses the execute button.</li> </ol>	
<i>Output definition</i>	Input is valid and the form will be submitted. Because of the non-reasonable input the query is going to produce empty result set. Results from the query are displayed in two tables. First table is holding the data for the regular seasons and the second for the playoff data. The same projections and potential conditions are imposed on the both tables.	
<i>Remarks:</i>	No remarks.	
<i>Execution Status:</i>	Pass.	

HoopStats	Version: 1.00
Acceptance Test Report	Date: 2013-01-21

<i>Test Name</i>	<b>Player regular seasons and playoff search select all projections with button click.</b>	<b>WAPRSPOFQ006</b>
<i>Description</i>	Test case when user selects all columns as results, with one button click.	
<i>Test type</i>	Positive.	
<i>Preconditions</i>	<ol style="list-style-type: none"> <li>1. User is connected to the internet</li> <li>2. The web application is up and running</li> <li>3. User is on <a href="http://hoopstats.tk/Player/PlayerAdvancedSearchQuery">http://hoopstats.tk/Player/PlayerAdvancedSearchQuery</a> page</li> <li>4. JavaScript must be enabled</li> </ol>	
<i>Input definition</i>	<ol style="list-style-type: none"> <li>1. User clicks the Select All button</li> </ol>	
<i>Output definition</i>	All of the columns in the selection part are selected and moved to the adequate container.	
<i>Remarks:</i>	No remarks.	
<i>Execution Status:</i>	Pass.	

<i>Test Name</i>	<b>Player regular seasons and playoff search clear all projections with button click.</b>	<b>WAPRSPOFQ008</b>
<i>Description</i>	Test case when user clears all previously selected columns as results, with one button click.	
<i>Test type</i>	Positive.	
<i>Preconditions</i>	<ol style="list-style-type: none"> <li>2. User is connected to the internet</li> <li>3. The web application is up and running</li> <li>4. User is on <a href="http://hoopstats.tk/Player/PlayerAdvancedSearchQuery">http://hoopstats.tk/Player/PlayerAdvancedSearchQuery</a> page</li> <li>5. JavaScript must be enabled</li> </ol>	
<i>Input definition</i>	<ol style="list-style-type: none"> <li>1. User clicks the Clear Selection button.</li> </ol>	
<i>Output definition</i>	All of the columns in the selection part are marked as non-selected and moved to the initial container.	
<i>Remarks:</i>	No remarks.	
<i>Execution Status:</i>	Pass.	

<i>Test Name</i>	<b>Player regular seasons and playoff search drag and drop functionality.</b>	<b>WAPRSPOFQ009</b>
<i>Description</i>	Test case for validating the drag and drop functionality	
<i>Test type</i>	Positive.	
<i>Preconditions</i>	<ol style="list-style-type: none"> <li>2. User is connected to the internet</li> <li>3. The web application is up and running</li> <li>4. User is on <a href="http://hoopstats.tk/Player/PlayerAdvancedSearchQuery">http://hoopstats.tk/Player/PlayerAdvancedSearchQuery</a> page</li> <li>5. JavaScript must be enabled</li> </ol>	
<i>Input definition</i>	<ol style="list-style-type: none"> <li>1. User select column and drag it to the adequate container.</li> <li>2. User drops the column over the adequate container.</li> </ol>	
<i>Output definition</i>	The column is moved to the projections/conditions container.	
<i>Remarks:</i>	The drag and drop functionality applies to the projections and conditions. There is validation that projection can be dropped only in the projections container and condition can be dropped only in the conditions container. If user tries to drop property (column) in inadequate container, it will be returned back to the initial container.	
<i>Execution Status:</i>	Pass.	

HoopStats	Version: 1.00
Acceptance Test Report	Date: 2013-01-21

<i>Test Name</i>	<b>Player regular seasons and playoff search pin and unpin conditions.</b>	<b>WAPRSPOFQ009</b>
<i>Description</i>	Test case for validating the pin and unpin functionality as an alternative to drag and drop functionality.	
<i>Test type</i>	Positive.	
<i>Preconditions</i>	<ol style="list-style-type: none"> <li>1. User is connected to the internet</li> <li>2. The web application is up and running</li> <li>3. User is on <a href="http://hoopstats.tk/Player/PlayerAdvancedSearchQuery">http://hoopstats.tk/Player/PlayerAdvancedSearchQuery</a> page</li> <li>4. JavaScript must be enabled</li> </ol>	
<i>Input definition</i>	1. User clicks pin/unpin on some of the column.	
<i>Output definition</i>	The column is moved in the initial container (unpin) or in the selection container (pin).	
<i>Remarks:</i>	No remarks.	
<i>Execution Status:</i>	Pass.	

<i>Test Name</i>	<b>Player regular seasons and playoff search player name autocomplete</b>	<b>WAPRSPOFQ010</b>
<i>Description</i>	Test the functionality of autocomplete plugin.	
<i>Test type</i>	Positive.	
<i>Preconditions</i>	<ol style="list-style-type: none"> <li>1. User is connected to the internet</li> <li>2. The web application is up and running</li> <li>3. User is on <a href="http://hoopstats.tk/Player/PlayerAdvancedSearchQuery">http://hoopstats.tk/Player/PlayerAdvancedSearchQuery</a> page</li> <li>4. JavaScript must be enabled</li> <li>5. User starts to type first name of the player</li> <li>6. User selects one of the options using keyboard or mouse.</li> </ol>	
<i>Input definition</i>	1. User clicks enter or selects with mouse.	
<i>Output definition</i>	The player name input has been given the right value.	
<i>Remarks:</i>	No remarks.	
<i>Execution Status:</i>	Pass.	

<i>Test Name</i>	<b>Player regular seasons and playoff search results paging plugin</b>	<b>WAPRSPOFQ011</b>
<i>Description</i>	Test the functionality of paging custom plugin.	
<i>Test type</i>	Positive.	
<i>Preconditions</i>	<ol style="list-style-type: none"> <li>1. User is connected to the internet</li> <li>2. The web application is up and running</li> <li>3. User is on <a href="http://hoopstats.tk/Player/PlayerAdvancedSearchQuery">http://hoopstats.tk/Player/PlayerAdvancedSearchQuery</a> page</li> <li>4. JavaScript must be enabled</li> <li>5. User selects projections</li> <li>6. User optionally imposes constraints</li> <li>7. User executes query</li> <li>8. User clicks on the paging buttons</li> </ol>	
<i>Input definition</i>	1. User clicks enter or selects with mouse.	
<i>Output definition</i>	The result tables for the regular seasons and playoffs have been displayed.	
<i>Remarks:</i>	No remarks.	
<i>Execution Status:</i>	Pass.	

HoopStats	Version: 1.00
Acceptance Test Report	Date: 2013-01-21

#### 4.1.1.2 Test Group Web Application Team Flexible Querying (WATFQ)

<i>Test Name</i>	<b>Feasible query</b>	<b>WATFQ001</b>
<i>Description</i>	Team Flexible Querying allows users to make queries about team seasons. User makes a feasible query where he selects the columns that he is interested in and imposes conditions (filters) to limit the search to a specific team or specific (numeric) conditions.	
<i>Test type</i>	Positive	
<i>Preconditions</i>	<ol style="list-style-type: none"> <li>1. User is connected to the internet.</li> <li>2. Team seasons table data is available in the database.</li> <li>3. The web application is up and running.</li> <li>4. JavaScript must be enabled.</li> </ol>	
<i>Input definition</i>	<ol style="list-style-type: none"> <li>1. User clicks the "Team" tab on the home page to open the Team Flexible Querying.</li> <li>2. User selects the properties (desired columns) by dragging them into "Selected columns" area or by clicking the pin on the property.</li> <li>3. User selects the conditions for the search. For any numeric condition, user selects one of the (in)equalities from the combo box on the condition – ET (Equal To), LT (Less Than), or GT (Greater Than), and inputs the desired number into the text box on the condition. For team name condition, user inputs the desired team name (team names are auto completed). Then, user applies the desired conditions by dragging them into "Conditions" area or by clicking the pin on the condition.</li> <li>4. User clicks the "Execute" button to start the search.</li> </ol>	
<i>Output definition</i>	<ol style="list-style-type: none"> <li>1. The results are shown in form of a team seasons table for seasons which comply to the specified conditions. Two predefined columns include team name and season year, and the rest are the columns that user initially selected in the query. The seasons are sorted by year.</li> <li>2. If a user wants to view the details of any of the listed teams, user clicks the team name in the results table – a team details page is shown with basic team info and a team seasons table for that specific team (with all available columns). The seasons are sorted by year.</li> </ol>	
<i>Remarks:</i>	WADQ001 test has no known bugs and delivers results as expected.	
<i>Execution Status:</i>	Pass.	

HoopStats	Version: 1.00
Acceptance Test Report	Date: 2013-01-21

<i>Test Name</i>	<b>Unfeasible query (no properties selected)</b>	<b>WATFQ002</b>
<i>Description</i>	Team Flexible Querying allows users to make queries about team seasons. User tries to makes an unfeasible query by not selecting any properties (columns).	
<i>Test type</i>	Negative	
<i>Preconditions</i>	<ol style="list-style-type: none"> <li>1. User is connected to the internet.</li> <li>2. Team seasons table data is available in the database.</li> <li>3. The web application is up and running.</li> <li>4. JavaScript must be enabled.</li> </ol>	
<i>Input definition</i>	<ol style="list-style-type: none"> <li>1. User clicks the "Team" tab on the home page to open the Team Flexible Querying.</li> <li>2. User selects <b>none</b> of the properties (columns) for the search.</li> <li>3. User selects the conditions for the search. For any numeric condition, user selects one of the (in)equalities from the combo box on the condition – ET (Equal To), LT (Less Than), or GT (Greater Than), and inputs the desired number into the text box on the condition. For team name condition, user inputs the desired team name (team names are auto completed). Then, user applies the desired conditions by dragging them into "Conditions" area or by clicking the pin on the condition.</li> <li>4. User clicks the "Execute" button to start the search.</li> </ol>	
<i>Output definition</i>	<ol style="list-style-type: none"> <li>1. A message box is shown, with a message "You must select at least one column."</li> </ol>	
<i>Remarks:</i>	The WATFQ002 test has no known bugs and delivers results as expected.	
<i>Execution Status:</i>	Pass.	

<i>Test Name</i>	<b>Unfeasible query (invalid numeric conditions)</b>	<b>WATFQ003</b>
<i>Description</i>	Team Flexible Querying allows users to make queries about team seasons. User tries to makes an unfeasible query by inputting a non-numeric values to numeric conditions.	
<i>Test type</i>	Negative	
<i>Preconditions</i>	<ol style="list-style-type: none"> <li>1. User is connected to the internet.</li> <li>2. Team seasons table data is available in the database.</li> <li>3. The web application is up and running.</li> <li>4. JavaScript must be enabled.</li> </ol>	
<i>Input definition</i>	<ol style="list-style-type: none"> <li>1. User clicks the "Team" tab on the home page to open the Team Flexible Querying.</li> <li>2. User selects the properties (desired columns) by dragging them into "Selected columns" area or by clicking the pin on the property.</li> <li>3. User selects the conditions for the search. For numeric conditions, user tries to input non-numeric values in the text field on the condition, such as letters or a minus character.</li> </ol>	
<i>Output definition</i>	<ol style="list-style-type: none"> <li>1. The text field on the condition does not accept non-numeric input. User cannot define a numeric condition.</li> </ol>	
<i>Remarks:</i>	The WATFQ003 test has no known bugs and delivers results as expected.	
<i>Execution Status:</i>	Pass.	

HoopStats	Version: 1.00
Acceptance Test Report	Date: 2013-01-21

<i>Test Name</i>	<b>Unfeasible query (non existing team name)</b>	<b>WATFQ004</b>
<i>Description</i>	Team Flexible Querying allows users to make queries about team seasons. User makes an unfeasible query where he selects the columns that he is interested in and imposes a condition with a non-existing team name (among other conditions).	
<i>Test type</i>	Negative	
<i>Preconditions</i>	<ol style="list-style-type: none"> <li>1. User is connected to the internet</li> <li>2. Team seasons table data is available in the database.</li> <li>3. The web application is up and running.</li> <li>4. JavaScript must be enabled.</li> </ol>	
<i>Input definition</i>	<ol style="list-style-type: none"> <li>1. User clicks the "Team" tab on the home page to open the Team Flexible Querying.</li> <li>2. User selects the properties (desired columns) by dragging them into "Selected columns" area or by clicking the pin on the property.</li> <li>3. User selects the conditions for the search. For any numeric condition, user selects one of the (in)equalities from the combo box on the condition – ET (Equal To), LT (Less Than), or GT (Greater Than), and inputs the desired number into the text box on the condition. For team name condition, user inputs a <b>non-existing</b> team name. Then, user applies the desired conditions (including non-existing team name condition) by dragging them into "Conditions" area or by clicking the pin on the condition.</li> <li>4. User clicks the "Execute" button to start the search.</li> </ol>	
<i>Output definition</i>	<ol style="list-style-type: none"> <li>1. An empty Team Seasons page is displayed.</li> </ol>	
<i>Remarks:</i>	The WATFQ004 test has no known bugs and delivers results as expected. It is possible to introduce an interface improvement by showing the user a friendly message that the team with the name specified in the condition does not exist.	
<i>Execution Status:</i>	Pass.	

<i>Test Name</i>	<b>Unfeasible query (no results match the imposed conditions)</b>	<b>WATFQ005</b>
<i>Description</i>	Team Flexible Querying allows users to make queries about team seasons. User makes an unfeasible query where he imposes conditions that produce no results.	
<i>Test type</i>	Negative	
<i>Preconditions</i>	<ol style="list-style-type: none"> <li>1. User is connected to the internet.</li> <li>2. Team seasons table data is available in the database.</li> <li>3. The web application is up and running.</li> <li>4. JavaScript must be enabled.</li> </ol>	
<i>Input definition</i>	<ol style="list-style-type: none"> <li>1. User clicks the "Team" tab on the home page to open the Team Flexible Querying.</li> <li>2. User selects the properties (desired columns) by dragging them into "Selected columns" area or by clicking the pin on the property.</li> <li>3. User selects the conditions for the search. For any numeric condition, user selects one of the (in)equalities from the combo box on the condition – ET (Equal To), LT (Less Than), or GT (Greater Than), and inputs the desired number into the text box on the condition. For team name condition, user inputs the desired team name (team names are auto completed). <b>Conditions that user has inputted are values that do not match any of the results.</b> Then, user applies the desired conditions by dragging them into "Conditions" area or by clicking the pin on the condition.</li> <li>4. User clicks the "Execute" button to start the search.</li> </ol>	
<i>Output definition</i>	<ol style="list-style-type: none"> <li>1. An empty Team Seasons page is displayed.</li> </ol>	
<i>Remarks:</i>	The WATFQ005 test has no known bugs and delivers results as expected. It is possible to introduce an interface improvement by showing the user a friendly message that no results match the imposed conditions.	

HoopStats	Version: 1.00
Acceptance Test Report	Date: 2013-01-21

<i>Execution Status:</i>	Pass.
--------------------------	-------

<i>Test Name</i>	<b>Unfeasible query (conditions selected but not defined)</b>	<b>WATFQ006</b>
<i>Description</i>	Team Flexible Querying allows users to make queries about team seasons. User tries to makes an unfeasible query by selecting conditions and not defining their values.	
<i>Test type</i>	Negative	
<i>Preconditions</i>	<ol style="list-style-type: none"> <li>1. User is connected to the internet.</li> <li>2. Team seasons table data is available in the database.</li> <li>3. The web application is up and running.</li> <li>4. JavaScript must be enabled.</li> </ol>	
<i>Input definition</i>	<ol style="list-style-type: none"> <li>1. User clicks the "Team" tab on the home page to open the Team Flexible Querying.</li> <li>2. User selects the properties (desired columns) by dragging them into "Selected columns" area or by clicking the pin on the property.</li> <li>3. User selects the conditions for the search, but doesn't input any of the condition values in the text field on one or more conditions (numeric values for the numeric conditions or team name on the team name condition). User applies the conditions by dragging them into "Conditions" area or by clicking the pin on the condition.</li> <li>4. User clicks the "Execute" button to start the search.</li> </ol>	
<i>Output definition</i>	<ol style="list-style-type: none"> <li>1. The search is not started. The text fields on the conditions turn red, indicating a missing value.</li> </ol>	
<i>Remarks:</i>	The WATFQ006 test has no known bugs and delivers results as expected.	
<i>Execution Status:</i>	Pass.	

<i>Test Name</i>	<b>Team name autocomplete</b>	<b>WATFQ007</b>
<i>Description</i>	Test the functionality of autocomplete plugin.	
<i>Test type</i>	Positive.	
<i>Preconditions</i>	<ol style="list-style-type: none"> <li>1. User is connected to the internet.</li> <li>2. Team seasons table data is available in the database.</li> <li>3. The web application is up and running.</li> <li>4. JavaScript must be enabled.</li> </ol>	
<i>Input definition</i>	<ol style="list-style-type: none"> <li>1. User clicks the "Team" tab on the home page to open the Team Flexible Querying.</li> <li>2. User selects the properties (desired columns) by dragging them into "Selected columns" area or by clicking the pin on the property.</li> <li>3. User selects the team name as a condition to the search.</li> <li>4. User starts typing team name into the autocomplete text box</li> <li>5. User select one offered value</li> <li>6. User clicks on mouse or enter to select the desired value</li> <li>7. The value has been transferred into the text box</li> </ol>	
<i>Output definition</i>	The player name input has been given the right value.	
<i>Remarks:</i>	No remarks.	
<i>Execution Status:</i>	Pass.	

<i>Test Name</i>	<b>Team advanced search results paging plugin</b>	<b>WATFQ008</b>
<i>Description</i>	Test the functionality of paging custom plugin.	
<i>Test type</i>	Positive.	
<i>Preconditions</i>	<ol style="list-style-type: none"> <li>1. User is connected to the internet.</li> <li>2. Team seasons table data is available in the database.</li> <li>3. The web application is up and running.</li> </ol>	

HoopStats	Version: 1.00
Acceptance Test Report	Date: 2013-01-21

<i>Input definition</i>	<ol style="list-style-type: none"> <li>4. JavaScript must be enabled.</li> <li>1. User clicks the "Team" tab on the home page to open the Team Flexible Querying.</li> <li>2. User selects the properties (desired columns) by dragging them into "Selected columns" area or by clicking the pin on the property.</li> <li>3. User selects the conditions for the search, but doesn't input any of the condition values in the text field on one or more conditions (numeric values for the numeric conditions or team name on the team name condition). User applies the conditions by dragging them into "Conditions" area or by clicking the pin on the condition.</li> <li>4. User clicks the "Execute" button to start the search.</li> <li>5. User is redirected to the result table page</li> <li>6. User clicks on the plugin links</li> </ol>
<i>Output definition</i>	The result tables for the regular seasons and playoffs have been displayed along with the plugin for paging.
<i>Remarks:</i>	No remarks.
<i>Execution Status:</i>	Pass.

#### 4.1.1.3 Test group Web Application Coach Flexible Query (WACFQ)

<i>Test Name</i>	<b>Coach Flexible Query</b>	<b>WACFQ-001</b>
<i>Description</i>	This test case is used to search the Coach information.	
<i>Test type</i>	Positive	
<i>Preconditions</i>	<ol style="list-style-type: none"> <li>5. User is connected to the internet.</li> <li>6. Coach seasons table data is available in the database.</li> <li>7. The web application is up and running.</li> <li>8. JavaScript must be enabled.</li> </ol>	
<i>Input definition</i>	<ol style="list-style-type: none"> <li>1. User select the Coaches tab</li> <li>2. User select at least one column to maximum all column(using select all button) from select result property</li> <li>3. User select column to impose the condition from selection condition for search</li> <li>4. User select the one condition (LT,GT,ET)</li> <li>5. User provide value for selected column</li> <li>6. User clicks the Execute button</li> </ol>	
<i>Output definition</i>	System displays" <b>Coach Seasons</b> " information with selected columns which satisfied the imposed condition.	
<i>Remarks:</i>	This test has no known bugs and delivers results as expected.	
<i>Execution Status:</i>	Pass.	

<i>Test Name</i>	<b>Coach Flexible Query</b>	<b>WACFQ-002</b>
<i>Description</i>	This test case is used to search the Coach information without selecting the column from select result property.	
<i>Test type</i>	negative	
<i>Preconditions</i>	<ol style="list-style-type: none"> <li>1. User is connected to the internet.</li> <li>2. Coach seasons table data is available in the database.</li> <li>3. The web application is up and running.</li> <li>4. JavaScript must be enabled.</li> </ol>	
<i>Input definition</i>	<ol style="list-style-type: none"> <li>1. User select the Coaches tab</li> </ol>	

HoopStats	Version: 1.00
Acceptance Test Report	Date: 2013-01-21

	<ol style="list-style-type: none"> <li>2. User forget to select the column</li> <li>3. User select column to impose the condition from selection condition for search</li> <li>4. User select the one condition (LT,GT,ET)</li> <li>5. User provide value for selected column</li> <li>6. User clicks the Execute button</li> </ol>
<i>Output definition</i>	System will generate error message <b>“You must select at least one column”</b>
<i>Remarks:</i>	This test has no known bugs and delivers results as expected.
<i>Execution Status:</i>	Pass.

<i>Test Name</i>	<b>Coach Flexible Query</b>	<b>WACFQ -003</b>
<i>Description</i>	This test case is used to search the Coach information without selecting the column from selection condition for search.	
<i>Test type</i>	Positive	
<i>Preconditions</i>	<ol style="list-style-type: none"> <li>1. User is connected to the internet.</li> <li>2. Coach seasons table data is available in the database.</li> <li>3. The web application is up and running.</li> <li>4. JavaScript must be enabled.</li> </ol>	
<i>Input definition</i>	<ol style="list-style-type: none"> <li>1. User select the Coaches tab</li> <li>2. User select at least one column to maximum all column(using select all button) from select result property</li> <li>3. User forget to select column to impose the condition from selection condition for search</li> <li>4. User clicks the Execute button</li> </ol>	
<i>Output definition</i>	System displays” <b>Coach Seasons”</b> information with selected column without comparing any conditions.	
<i>Remarks:</i>	This test has no known bugs and delivers results as expected.	
<i>Execution Status:</i>	Pass.	

<i>Test Name</i>	<b>Coach Details</b>	<b>WACD-004</b>
<i>Description</i>	This test case is used to view the detail information of the Coach.	
<i>Test type</i>	Positive	
<i>Preconditions</i>	<ol style="list-style-type: none"> <li>1. User is connected to the internet.</li> <li>2. Coach seasons table data is available in the database.</li> <li>3. The web application is up and running.</li> <li>4. JavaScript must be enabled.</li> <li>5. User must be at “CoachAdvancedSearchResults” page.</li> </ol>	
<i>Input definition</i>	1. User click on either first name or last name of coach	
<i>Output definition</i>	System displays <b>“Coach Details”</b> and <b>“Coach Season (all columns)”</b> table.	
<i>Remarks:</i>	This test has no known bugs and delivers results as expected.	
<i>Execution Status:</i>	Pass.	

<i>Test Name</i>	<b>Coach Details Sort</b>	<b>WACDS-005</b>
<i>Description</i>	This test case is used to sort the detail information of the coach seasons data.	
<i>Test type</i>	Positive	
<i>Preconditions</i>	<ol style="list-style-type: none"> <li>1. User is connected to the internet.</li> <li>2. Coach seasons table data is available in the database.</li> <li>3. The web application is up and running.</li> </ol>	

HoopStats	Version: 1.00
Acceptance Test Report	Date: 2013-01-21

	4. JavaScript must be enabled. 5. User must be at "CoachAdvancedSearchResults" page.
<i>Input definition</i>	1. User click on any column name of coach season table to sort the coach data according to their need.
<i>Output definition</i>	System sorts coach season table according to selected column data order.
<i>Remarks:</i>	This test has no known bugs and delivers results as expected.
<i>Execution Status:</i>	Pass.

<i>Test Name</i>	<b>Coach-Team Detail</b>	<b>WACTD-006</b>
<i>Description</i>	This test case is used to view the detail information of team of the coach.	
<i>Test type</i>	Positive	
<i>Preconditions</i>	<ol style="list-style-type: none"> <li>1. User is connected to the internet.</li> <li>2. Coach seasons table data is available in the database.</li> <li>3. The web application is up and running.</li> <li>4. JavaScript must be enabled.</li> <li>5. User must be at "CoachAdvancedSearchResults" page.</li> </ol>	
<i>Input definition</i>	1. User must click on Column name " <b>team</b> ".	
<i>Output definition</i>	System displays " <b>Team Details</b> " and " <b>Team Season (all columns)</b> " table.	
<i>Remarks:</i>	This test has no known bugs and delivers results as expected.	
<i>Execution Status:</i>	Pass.	

<i>Test Name</i>	Coach-Team Detail Sort	WACTDS-007
<i>Description</i>	This test case is used to sort the detail information of team of the coach.	
<i>Test type</i>	Positive	
<i>Preconditions</i>	<ol style="list-style-type: none"> <li>1. User is connected to the internet.</li> <li>2. Coach seasons table data is available in the database.</li> <li>3. The web application is up and running.</li> <li>4. JavaScript must be enabled.</li> <li>5. User must be at "CoachAdvancedSearchResults" page.</li> </ol>	
<i>Input definition</i>	1. User click on any column name of team season table to sort the team data according to their need.	
<i>Output definition</i>	System sorts team season table according to selected column data order.	
<i>Remarks:</i>	This test has no known bugs and delivers results as expected.	
<i>Execution Status:</i>	Pass.	

HoopStats	Version: 1.00
Acceptance Test Report	Date: 2013-01-21

#### 4.1.2 Test Group Android Application (AA)

This section contains all test cases for the android application and the main functions.

##### 4.1.2.1 Test Group Android Application Player Search (AAPS)

<i>Test Name</i>	<b>Player Search</b>	<b>AAPS001</b>
<i>Description</i>	Test case describes searching for players and displaying results of the search.	
<i>Test type</i>	Positive	
<i>Preconditions</i>	<ol style="list-style-type: none"> <li>1. HoopStats application installed on the device</li> <li>2. Internet connection available</li> </ol>	
<i>Input definition</i>	<ol style="list-style-type: none"> <li>1. User opens HoopStats application</li> <li>2. In the home screen the Players button is tapped</li> <li>3. Desired player's name is entered in the textual field</li> <li>4. Search button is tapped</li> </ol>	
<i>Output definition</i>	User can see the results of a search query. If there are no players that satisfy the condition the list is empty. If there are more than 10 results, only first 10 will appear in the list.	
<i>Remarks:</i>	No remarks	
<i>Execution Status:</i>	Pass.	

##### 4.1.2.2 Test Group Android Application Player Details (AAPD)

<i>Test Name</i>	<b>Player Details</b>	<b>AAPD001</b>
<i>Description</i>	Test case describes displaying details of a desired player.	
<i>Test type</i>	Positive	
<i>Preconditions</i>	<ol style="list-style-type: none"> <li>1. HoopStats application installed on the device</li> <li>2. Internet connection available</li> <li>3. The search results are displayed and contain at least one result</li> </ol>	
<i>Input definition</i>	<ol style="list-style-type: none"> <li>1. User taps on the player in the result list</li> </ol>	
<i>Output definition</i>	Player details are shown. There are 2 tabs: Player Details, and Season Stats. Player Details tab is initially selected.	
<i>Remarks:</i>	No remarks	
<i>Execution Status:</i>	Pass.	

<i>Test Name</i>	<b>Player Details - Player Details</b>	<b>AAPD002</b>
<i>Description</i>	Test case describes displaying Player Details tab in the details of a desired player.	
<i>Test type</i>	Positive	
<i>Preconditions</i>	<ol style="list-style-type: none"> <li>1. HoopStats application installed on the device</li> <li>2. Internet connection available</li> <li>3. Details of a player on the screen</li> </ol>	
<i>Input definition</i>	<ol style="list-style-type: none"> <li>1. User taps on the Player Details tab in the player details view</li> </ol>	
<i>Output definition</i>	Player's basic info is shown: his name, height, weight, position, etc.	
<i>Remarks:</i>	No remarks	
<i>Execution Status:</i>	Pass.	

HoopStats	Version: 1.00
Acceptance Test Report	Date: 2013-01-21

<i>Test Name</i>	<b>Player Details – Season Stats</b>	<b>AAPD003</b>
<i>Description</i>	Test case describes displaying Season Stats tab in the details of a desired player.	
<i>Test type</i>	Positive	
<i>Preconditions</i>	<ol style="list-style-type: none"> <li>1. HoopStats application installed on the device</li> <li>2. Internet connection available</li> <li>3. Details of a player on the screen</li> </ol>	
<i>Input definition</i>	<ol style="list-style-type: none"> <li>1. User taps on the Season Stats tab in the player details view</li> </ol>	
<i>Output definition</i>	Tab appears in which there is a spinner from which user can choose a property to be displayed in a form of a graph where year is mapped to x-axis and value of a chosen property is mapped to the y-axis.	
<i>Remarks:</i>	No remarks	
<i>Execution Status:</i>	Pass.	

#### 4.1.2.3 Test Group Android Application Coach Search (AACS)

<i>Test Name</i>	<b>Coach Search</b>	<b>AACS001</b>
<i>Description</i>	Test case describes searching for coaches and displaying results of the search.	
<i>Test type</i>	Positive	
<i>Preconditions</i>	<ol style="list-style-type: none"> <li>1. HoopStats application installed on the device</li> <li>2. Internet connection available</li> </ol>	
<i>Input definition</i>	<ol style="list-style-type: none"> <li>1. User opens HoopStats application</li> <li>2. In the home screen the Coaches button is tapped</li> <li>3. Desired coach's name is entered in the textual field</li> <li>4. Search button is tapped</li> </ol>	
<i>Output definition</i>	User can see the results of a search query. If there are no coaches that satisfy the condition the list is empty. If there are more than 10 results, only first 10 will appear in the list.	
<i>Remarks:</i>	No remarks	
<i>Execution Status:</i>	Pass.	

#### 4.1.2.4 Test Group Android Application Coach Details (AACD)

<i>Test Name</i>	<b>Coach Details</b>	<b>AACD001</b>
<i>Description</i>	Test case describes displaying details of a desired coach.	
<i>Test type</i>	Positive	
<i>Preconditions</i>	<ol style="list-style-type: none"> <li>1. HoopStats application installed on the device</li> <li>2. Internet connection available</li> <li>3. The search results are displayed and contain at least one result</li> </ol>	
<i>Input definition</i>	<ol style="list-style-type: none"> <li>1. User taps on the coach in the result list</li> </ol>	
<i>Output definition</i>	Coach details are shown. There are 2 tabs: Details and Stats. Details tab is initially selected.	
<i>Remarks:</i>	No remarks	
<i>Execution Status:</i>	Pass.	

HoopStats	Version: 1.00
Acceptance Test Report	Date: 2013-01-21

<i>Test Name</i>	<b>Coach Details - Details</b>	<b>AACD002</b>
<i>Description</i>	Test case describes displaying Details tab in the details of a desired coach.	
<i>Test type</i>	Positive	
<i>Preconditions</i>	<ol style="list-style-type: none"> <li>1. HoopStats application installed on the device</li> <li>2. Internet connection available</li> <li>3. Details of a coach on the screen</li> </ol>	
<i>Input definition</i>	1. User taps on the Info tab in the coach details view	
<i>Output definition</i>	Coach's basic info is shown: his name, season wins/losses, playoff wins/losses.	
<i>Remarks:</i>	No remarks	
<i>Execution Status:</i>	Pass.	

<i>Test Name</i>	<b>Coach Details - Stats</b>	<b>AACD002</b>
<i>Description</i>	Test case describes displaying Stats tab in the details of a desired coach.	
<i>Test type</i>	Positive	
<i>Preconditions</i>	<ol style="list-style-type: none"> <li>1. HoopStats application installed on the device</li> <li>2. Internet connection available</li> <li>3. Details of a coach on the screen</li> </ol>	
<i>Input definition</i>	1. User taps on the Graph tab in the coach details view	
<i>Output definition</i>	Tab appears in which there is a spinner from which user can choose a property to be displayed in a form of a graph where year is mapped to x-axis and value of a chosen property is mapped to the y-axis.	
<i>Remarks:</i>	No remarks	
<i>Execution Status:</i>	Pass.	

#### 4.1.2.5 Test Group Android Application Team Search (AATS)

<i>Test Name</i>	<b>Team Search</b>	<b>AATS001</b>
<i>Description</i>	Test cases aims to check if the functionality to search for a specific team works in the android application.	
<i>Test type</i>	Positive	
<i>Preconditions</i>	<ol style="list-style-type: none"> <li>1. Android Phone is connected to the internet</li> <li>2. HoopStats application is installed on the device</li> </ol>	
<i>Input definition</i>	<ol style="list-style-type: none"> <li>1. User opens HoopStats application</li> <li>2. In the home screen the Teams button is tapped</li> <li>3. Desired teams name is entered in the textual field</li> <li>4. Search button is tapped</li> </ol>	
<i>Output definition</i>	List of teams with the desired name should be displayed as search result.	
<i>Remarks:</i>	No remarks	
<i>Execution Status:</i>	Pass.	

HoopStats	Version: 1.00
Acceptance Test Report	Date: 2013-01-21

#### 4.1.2.6 Test Group Android Application Team Details (AATD)

<i>Test Name</i>	<b>Team Details</b>	<b>AATD001</b>
<i>Description</i>	Test case describes displaying details of a desired coach.	
<i>Test type</i>	Positive	
<i>Preconditions</i>	<ol style="list-style-type: none"> <li>HoopStats application installed on the device</li> <li>Internet connection available</li> <li>The search results are displayed and contain at least one result</li> </ol>	
<i>Input definition</i>	<ol style="list-style-type: none"> <li>User taps on the team in the result list</li> </ol>	
<i>Output definition</i>	Team details are shown. There are 3 tabs: Details, Season and Stats. Details tab is initially selected.	
<i>Remarks:</i>	No remarks	
<i>Execution Status:</i>	Pass.	

<i>Test Name</i>	<b>Team Details - Details</b>	<b>AATD002</b>
<i>Description</i>	Test case describes displaying Details tab in the details of a desired coach.	
<i>Test type</i>	Positive	
<i>Preconditions</i>	<ol style="list-style-type: none"> <li>HoopStats application installed on the device</li> <li>Internet connection available</li> <li>Details of a team on the screen</li> </ol>	
<i>Input definition</i>	<ol style="list-style-type: none"> <li>User taps on the Details tab in the coach details view</li> </ol>	
<i>Output definition</i>	Team's logo and basic info is shown: league and whether the team is active.	
<i>Remarks:</i>	No remarks	
<i>Execution Status:</i>	Pass.	

<i>Test Name</i>	<b>Team Details - Season</b>	<b>AATD003</b>
<i>Description</i>	Test case describes displaying Season tab in the details of a desired team.	
<i>Test type</i>	Positive	
<i>Preconditions</i>	<ol style="list-style-type: none"> <li>HoopStats application installed on the device</li> <li>Internet connection available</li> <li>Details of a team on the screen</li> </ol>	
<i>Input definition</i>	<ol style="list-style-type: none"> <li>User taps on the Season tab in the team details view</li> </ol>	
<i>Output definition</i>	Tab appears in which there is a spinner from which user can choose a season to be displayed.	
<i>Remarks:</i>	No remarks	
<i>Execution Status:</i>	Pass.	

<i>Test Name</i>	<b>Team Details - Stats</b>	<b>AATD004</b>
<i>Description</i>	Test case describes displaying Stats tab in the details of a desired team.	
<i>Test type</i>	Positive	
<i>Preconditions</i>	<ol style="list-style-type: none"> <li>HoopStats application installed on the device</li> <li>Internet connection available</li> <li>Details of a team on the screen</li> </ol>	
<i>Input definition</i>	<ol style="list-style-type: none"> <li>User taps on the Stats tab in the team details view</li> </ol>	
<i>Output definition</i>	A graph appears with year mapped to the x-axis and won games and lost games to	

HoopStats	Version: 1.00
Acceptance Test Report	Date: 2013-01-21

	the y-axis.
Remarks:	No remarks
Execution Status:	Pass.

#### 4.1.2.7 Test Group Android Application Advanced Search (AAAS)

Test Name	<b>Advanced Search</b>	AAAS001
Description	Test case describes navigation to the screen for advanced search.	
Test type	Positive	
Preconditions	<ol style="list-style-type: none"> <li>HoopStats application installed on the device</li> <li>Internet connection available</li> </ol>	
Input definition	<ol style="list-style-type: none"> <li>User opens HoopStats application</li> <li>In the home screen the Advanced Search button is tapped</li> </ol>	
Output definition	User can see three tabs: Players, Coaches and Teams. Each of those tabs corresponds to the kind of entity the advanced search should be performed on. Players tab is initially selected.	
Remarks:	No remarks	
Execution Status:	Pass.	

#### 4.1.2.8 Test Group Android Application Players Advanced Search (AAPAS)

Test Name	<b>Players Advanced Search</b>	AAPAS001
Description	Test case describes navigation to the screen for players advanced search.	
Test type	Positive	
Preconditions	<ol style="list-style-type: none"> <li>HoopStats application installed on the device</li> <li>Internet connection available</li> <li>The advanced search part of application is opened</li> </ol>	
Input definition	<ol style="list-style-type: none"> <li>User taps on the Players tab of Advanced search screen</li> </ol>	
Output definition	User can pick between season and games. User can see all possible columns (properties) upon which he can impose conditions for the search. Each column has a checkbox indicating whether the corresponding condition is used in a search query. Numeric properties have a spinner with LT, ET, GT values (less than, equal to and greater than), and a text field where the desired number can be defined. Non-numeric properties (e.g. name) don't have a spinner. They only have a text field which is exactly matched to corresponding column's value. There is also an execute button.	
Remarks:	No remarks	
Execution Status:	Pass.	

#### 4.1.2.9 Test Group Android Application Players Advanced Search (AAPAS)

Test Name	<b>Players Advanced Search - execution</b>	AAPAS002
Description	Test case describes the execution of the players advanced search.	
Test type	Positive	
Preconditions	<ol style="list-style-type: none"> <li>HoopStats application installed on the device</li> <li>Internet connection available</li> <li>The players advanced search part of application is opened</li> </ol>	
Input definition	<ol style="list-style-type: none"> <li>User inputs desired conditions of the query</li> <li>User taps Execute button</li> </ol>	
Output definition	If there are results that satisfy the imposed conditions results are show in the form of a scrollable table. Maximum number of rows is 20.	
Remarks:	No remarks	

HoopStats	Version: 1.00
Acceptance Test Report	Date: 2013-01-21

<i>Execution Status:</i>	Pass.
--------------------------	-------

#### 4.1.2.10 Test Group Android Application Coaches Advanced Search (AACAS)

<i>Test Name</i>	<b>Coaches Advanced Search</b>	<b>AACAS001</b>
<i>Description</i>	Test case describes navigation to the screen for coaches advanced search.	
<i>Test type</i>	Positive	
<i>Preconditions</i>	<ol style="list-style-type: none"> <li>1. HoopStats application installed on the device</li> <li>2. Internet connection available</li> <li>3. The advanced search part of application is opened</li> </ol>	
<i>Input definition</i>	<ol style="list-style-type: none"> <li>1. User taps on the Coaches tab of Advanced search screen</li> </ol>	
<i>Output definition</i>	User can see all possible columns (properties) upon which he can impose conditions for the search. Each column has a checkbox indicating whether the corresponding condition is used in a search query. Numeric properties have a spinner with LT, ET, GT values (less than, equal to and greater than), and a text field where the desired number can be defined. Non-numeric properties (e.g. name) don't have a spinner. They only have a text field which is exactly matched to corresponding column's value. There is also an execute button.	
<i>Remarks:</i>	No remarks	
<i>Execution Status:</i>	Pass.	

#### 4.1.2.11 Test Group Android Application Coaches Advanced Search (AACAS)

<i>Test Name</i>	<b>Coaches Advanced Search - execution</b>	<b>AACAS002</b>
<i>Description</i>	Test case describes the execution of the coaches advanced search.	
<i>Test type</i>	Positive	
<i>Preconditions</i>	<ol style="list-style-type: none"> <li>1. HoopStats application installed on the device</li> <li>2. Internet connection available</li> <li>3. The coaches advanced search part of application is opened</li> </ol>	
<i>Input definition</i>	<ol style="list-style-type: none"> <li>1. User inputs desired conditions of the query</li> <li>2. User taps Execute button</li> </ol>	
<i>Output definition</i>	If there are results that satisfy the imposed conditions results are show in the form of a scrollable table. Maximum number of rows is 20.	
<i>Remarks:</i>	No remarks	
<i>Execution Status:</i>	Pass.	

#### 4.1.2.12 Test Group Android Application Teams Advanced Search (AATAS)

<i>Test Name</i>	<b>Teams Advanced Search</b>	<b>AATAS001</b>
<i>Description</i>	Test case describes navigation to the screen for teams advanced search.	
<i>Test type</i>	Positive	
<i>Preconditions</i>	<ol style="list-style-type: none"> <li>1. HoopStats application installed on the device</li> <li>2. Internet connection available</li> <li>3. The advanced search part of application is opened</li> </ol>	
<i>Input definition</i>	<ol style="list-style-type: none"> <li>1. User taps on the Teams tab of Advanced search screen</li> </ol>	
<i>Output definition</i>	User can see all possible columns (properties) upon which he can impose conditions for the search. Each column has a checkbox indicating whether the corresponding condition is used in a search query. Numeric properties have a spinner with LT, ET, GT values (less than, equal to and greater than), and a text field where the desired	

HoopStats	Version: 1.00
Acceptance Test Report	Date: 2013-01-21

	number can be defined. Non-numeric properties (e.g. name) don't have a spinner. They only have a text field which is exactly matched to corresponding column's value. There is also an execute button.
<i>Remarks:</i>	No remarks
<i>Execution Status:</i>	Pass.

4.1.2.13 Test Group Android Application Teams Advanced Search (AAPAS)

<i>Test Name</i>	<b>Teams Advanced Search - execution</b>	<b>AATAS002</b>
<i>Description</i>	Test case describes the execution of the teams advanced search.	
<i>Test type</i>	Positive	
<i>Preconditions</i>	<ol style="list-style-type: none"> <li>1. HoopStats application installed on the device</li> <li>2. Internet connection available</li> <li>3. The teams advanced search part of application is opened</li> </ol>	
<i>Input definition</i>	<ol style="list-style-type: none"> <li>1. User inputs desired conditions of the query</li> <li>2. User taps Execute button</li> </ol>	
<i>Output definition</i>	If there are results that satisfy the imposed conditions results are show in the form of a scrollable table. Maximum number of rows is 20.	
<i>Remarks:</i>	No remarks	
<i>Execution Status:</i>	Pass.	

**5. Approvals**

Name	Title	Date yyyy-mm-dd	Signature