



# HoopStats Acceptance Test Plan

Version 0.1



HoopStats	Version: 0.1
Acceptance Test Plan	Date: 2012-12-30

## Revision History

Date	Version	Description	Author
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

HoopStats	Version: 0.1
Acceptance Test Plan	Date: 2012-12-30

## 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	5
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	6
3.	Approach	6
3.1	Approach to configuration and installation	6
3.2	Item pass/fail criteria	7
3.3	Documentation problems	7
3.4	Suspension criteria and resumption requirements	7
3.5	Responsibilities	7
3.5.1	Developers	7
3.5.2	Customer	7
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)	18
5.	Approvals	22

HoopStats	Version: 0.1
Acceptance Test Plan	Date: 2012-12-30

## 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 applications in the HoopStats project. It is used to determine if the defined project requirements have been fulfilled 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 get an overview about functionalities to be tested and the expected outcome
- Customer: To verify that the specified requirements have been met
- Supervisor: Responsible for monitoring the status of the project, its direction and outcomes
- Future Developers: To get an overview about the expected application behavior

### 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 take place.

### 1.4 Acronyms

#### 1.4.1 Acronyms and abbreviations

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

### 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%5B3%5D.pdf](http://www.fer.unizg.hr/download/repository/Project_Plan%5B3%5D.pdf)
- ❖ Requirements Definition
  - [http://www.fer.unizg.hr/download/repository/Requirements\\_Definition%5B4%5D.pdf](http://www.fer.unizg.hr/download/repository/Requirements_Definition%5B4%5D.pdf)
- ❖ Project Design
  - [http://www.fer.unizg.hr/download/repository/Design\\_Description%5B3%5D.pdf](http://www.fer.unizg.hr/download/repository/Design_Description%5B3%5D.pdf)
- ❖ Web application home
  - <http://www.hoopstats.tk/>

HoopStats	Version: 0.1
Acceptance Test Plan	Date: 2012-12-30

## 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 applications which will be the major test items for the acceptance test and will be shortly described in the following paragraphs:

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

### 2.2 Features to be tested

The test cases defined in this document aim to test if the features implemented in both applications fulfill the defined requirements. Therefore they are divided in two sections:

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

### 2.3 Features not to be tested

Under the constraints of the project as an university course the following features will not be tested:

1. Performance: e.g. Test cases to ensure that the application performs under certain amount of parallel users.
2. Security: Test cases to ensure that the web application is secured from unauthorized access to the database.

### 2.4 Test environment

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

#### Hardware:

- Personal computer, Laptop
- Android mobile phone

#### Software:

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

#### Other:

- Working internet connection

## 3. Approach

The test approach is straight forward. The test cases provides in this document are conducted by the development team as well as the customer and supervisor.

### 3.1 Approach to configuration and installation

The web application is available via the internet. The test cases for this application will be conducted on server provided by the development team. Therefore no efforts have to be undertaken by the customer and supervisor to install and configure the web application for testing. Both persons are provided with the hyperlink to the application and can then start to test.

HoopStats	Version: 0.1
Acceptance Test Plan	Date: 2012-12-30

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 can be done by the customer and supervisor themselves, as the APK automates it.

In terms of the android application it has to be ensured, that the android version matches the defined version of section 2.4. With older mobile devices the application won't start. Furthermore the internet connection has to be enabled and internet access for the HoopStats has to be granted. Without this the application is not testable on a mobile device.

### 3.2 Item pass/fail criteria

In the following table the test types are described which will be conducted for each test case. Overall, each test case provided 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 artifacts are written before this document, no major problems should occur which influences the other 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.

### 3.4 Suspension criteria and resumption requirements

Testing can always be paused and continued. The developed test cases have a short runtime and can be performed rather quickly. Therefore no specific process for pausing and resuming testing have to be developed. If bugs are discovered during the testing phase, the test cases are still all done. 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.

#### 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.

#### Customer

The customer has to inform the project team in case of occurring bugs during the test of the application.

HoopStats	Version: 0.1
Acceptance Test Plan	Date: 2012-12-30

## 4. Test procedure

In the following section each of the test cases is described and grouped together for the different items and functionalities. The cases are divided in the two main items, the android application and the web application.

### 4.1 Test case specifications

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

In the section the test cases for the web application are described.

##### 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 used 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. The 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>Test Name</i>	<b>Player regular seasons and playoff search with conditions</b>	<b>WAPRSPOFQ002</b>
<i>Description</i>	Test case is used 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>	
<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 or 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	

HoopStats	Version: 0.1
Acceptance Test Plan	Date: 2012-12-30

<i>Test Name</i>	<b>Player regular seasons and playoff search without selecting projections.</b>	<b>WAPRSPOFQ003</b>
<i>Description</i>	Test to show error message 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. 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>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 base. 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: 0.1
Acceptance Test Plan	Date: 2012-12-30

<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>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 very likely 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.	

HoopStats	Version: 0.1
Acceptance Test Plan	Date: 2012-12-30

<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>	1. User clicks the Select All button	
<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>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>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 the Clear Selection button.	
<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>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>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 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.	

HoopStats	Version: 0.1
Acceptance Test Plan	Date: 2012-12-30

<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.	

#### 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.	

HoopStats	Version: 0.1
Acceptance Test Plan	Date: 2012-12-30

<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>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.	

HoopStats	Version: 0.1
Acceptance Test Plan	Date: 2012-12-30

<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>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: 0.1
Acceptance Test Plan	Date: 2012-12-30

<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.	

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

<i>Test Name</i>	Coach Flexible Query	WACFQ-001
<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.	

HoopStats	Version: 0.1
Acceptance Test Plan	Date: 2012-12-30

<i>Test Name</i>	Coach Flexible Query	WACFQ -002
<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> <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>Test Name</i>	Coach Flexible Query	WACFQ -003
<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>Test Name</i>	Coach Details	WACD-004
<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.	

HoopStats	Version: 0.1
Acceptance Test Plan	Date: 2012-12-30

<i>Test Name</i>	Coach Details Sort	WACDS-005
<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> <li>4. JavaScript must be enabled.</li> <li>5. User must be at "CoachAdvancedSearchResults" page.</li> </ol>	
<i>Input definition</i>	<ol style="list-style-type: none"> <li>1. User click on any column name of coach season table to sort the coach data according to their need.</li> </ol>	
<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>Test Name</i>	Coach-Team Detail	WACTD-006
<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>	<ol style="list-style-type: none"> <li>1. User must click on Column name "team".</li> </ol>	
<i>Output definition</i>	System displays "Team Details" and "Team Season (all columns)" table.	
<i>Remarks:</i>	This test has no known bugs and delivers results as expected.	

<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>	<ol style="list-style-type: none"> <li>1. User click on any column name of team season table to sort the team data according to their need.</li> </ol>	
<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.	

HoopStats	Version: 0.1
Acceptance Test Plan	Date: 2012-12-30

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

This section contains all test cases for the android application and the respective 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	

##### 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 3 tabs: Info, Stats and Graph. Info tab is initially selected.	
<i>Remarks:</i>	No remarks	

<i>Test Name</i>	<b>Player Details - Info</b>	<b>AAPD002</b>
<i>Description</i>	Test case describes displaying Info 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 Info 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	

HoopStats	Version: 0.1
Acceptance Test Plan	Date: 2012-12-30

<i>Test Name</i>	<b>Player Details - Stats</b>	<b>AAPD003</b>
<i>Description</i>	Test case describes displaying 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>	1. User taps on the Stats tab in the player details view	
<i>Output definition</i>	Player's statistics are shown in the format of a table. In the portrait position of the device, only basic statistics are shown in the table and in the landscape mode, all available statistics are shown.	
<i>Remarks:</i>	No remarks	

<i>Test Name</i>	<b>Player Details - Graph</b>	<b>AAPD003</b>
<i>Description</i>	Test case describes displaying Graph 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>	1. User taps on the Graph tab in the player 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	

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

<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>3. HoopStats application installed on the device</li> <li>4. Internet connection available</li> </ol>	
<i>Input definition</i>	<ol style="list-style-type: none"> <li>5. User opens HoopStats application</li> <li>6. In the home screen the Coaches button is tapped</li> <li>7. Desired coach's name is entered in the textual field</li> <li>8. 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	

HoopStats	Version: 0.1
Acceptance Test Plan	Date: 2012-12-30

#### 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>	<ul style="list-style-type: none"> <li>4. HoopStats application installed on the device</li> <li>5. Internet connection available</li> <li>6. The search results are displayed and contain at least one result</li> </ul>	
<i>Input definition</i>	2. User taps on the coach in the result list	
<i>Output definition</i>	Coach details are shown. There are 3 tabs: Info, Stats and Graph. Info tab is initially selected.	
<i>Remarks:</i>	No remarks	

<i>Test Name</i>	<b>Coach Details - Info</b>	<b>AACD002</b>
<i>Description</i>	Test case describes displaying Info tab in the details of a desired coach.	
<i>Test type</i>	Positive	
<i>Preconditions</i>	<ul style="list-style-type: none"> <li>4. HoopStats application installed on the device</li> <li>5. Internet connection available</li> <li>6. Details of a coach on the screen</li> </ul>	
<i>Input definition</i>	2. 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>Test Name</i>	<b>Coach Details - Stats</b>	<b>AACD003</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>	<ul style="list-style-type: none"> <li>4. HoopStats application installed on the device</li> <li>5. Internet connection available</li> <li>6. Details of a coach on the screen</li> </ul>	
<i>Input definition</i>	2. User taps on the Stats tab in the coach details view	
<i>Output definition</i>	Coach's statistics are shown in the format of a table by year. In the portrait position of the device, only basic statistics are shown in the table and in the landscape mode, all available statistics are shown.	
<i>Remarks:</i>	No remarks	

<i>Test Name</i>	<b>Coach Details - Graph</b>	<b>AACD003</b>
<i>Description</i>	Test case describes displaying Graph tab in the details of a desired coach.	
<i>Test type</i>	Positive	
<i>Preconditions</i>	<ul style="list-style-type: none"> <li>4. HoopStats application installed on the device</li> <li>5. Internet connection available</li> <li>6. Details of a coach on the screen</li> </ul>	
<i>Input definition</i>	2. 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	

HoopStats	Version: 0.1
Acceptance Test Plan	Date: 2012-12-30

#### 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 work in the android application	
<i>Test type</i>	Positive	
<i>Preconditions</i>	<ol style="list-style-type: none"> <li>5. Android Phone is connected to the internet</li> <li>6. Team "New York Knicks" must be available in the database</li> </ol>	
<i>Input definition</i>	<ol style="list-style-type: none"> <li>1. The web application is up and running</li> <li>2. HoopStats Android is running</li> <li>3. Open team search</li> <li>4. Input team "New York Knicks" in the search field</li> <li>5. Press the search button</li> </ol>	
<i>Output definition</i>	"New York Knicks" team should be displayed as search result.	
<i>Remarks:</i>	No remarks	

<i>Test Name</i>	<b>Change team season statistics</b>	<b>AATS002</b>
<i>Description</i>	Test cases aims to check if the functionality to change the season for a team works respectively	
<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. Team "New York Knicks" must be available in the database</li> <li>3. Team season "2006" for "New York Knicks" must be available in the database</li> </ol>	
<i>Input definition</i>	<ol style="list-style-type: none"> <li>1. The web application is up and running</li> <li>2. HoopStats Android is running</li> <li>3. open team search</li> <li>4. Input team "New York Knicks" in the search field</li> <li>5. Go to tab seasons</li> <li>6. Klick on season spinner</li> <li>7. change season to "2006"</li> </ol>	
<i>Output definition</i>	Season data should change and display the selected season.	
<i>Remarks:</i>	No remarks	

HoopStats	Version: 0.1
Acceptance Test Plan	Date: 2012-12-30

<i>Test Name</i>	<b>Change graphical season statistics</b>	<b>AATS003</b>
<i>Description</i>	Test cases aims to check if the functionality to change the visual graph for a team season to the given statistic	
<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. Team "New York Knicks" must be available in the database</li> <li>3. Team season "2006" for "New York Knicks" must be available in the database</li> </ol>	
<i>Input definition</i>	<ol style="list-style-type: none"> <li>1. The web application is up and running</li> <li>2. HoopStats Android is running</li> <li>3. open team search</li> <li>4. Input team "New York Knicks" in the search field</li> <li>5. Go to tab Graph</li> <li>6. Klick on statistic spinner</li> <li>7. change statistic to rebounds</li> </ol>	
<i>Output definition</i>	Visual output should change according to the selected statistic value	
<i>Remarks:</i>	No remarks	

## 5. Approvals

Name	Title	Date yyyy-mm-dd	Signature