MÄLARDALENS HÖGSKOLA

HoopStats
Requirements Specification

Version 1.0

Revision History

Date	Version	Description	Author
2012-11-05	0.1	Initial Draft	Andreas Köhle, Predrag Filipovikj
2013-01-12	0.2	Improvements of the initial draft	Predrag Filipovikj
2013-01-20	1.0	Final version	Predrag Filipovikj

Table of Contents

1. Introduction

1.1 Purpose of this document

The purpose of this document is to deliver the requirements laid out by the customer and the project team to successfully implement the HoopStats application. The document is created in the initial requirements engineering phase of the project and is revised and updated after every iteration. The document is the basis for the system design document and all implementation tasks for the team members.

1.2 Intended Audience

The document is intended for the following audience as project reference:

- Project team members
- Supervisor
- Customer
- External project stakeholders
- Future Developers

1.3 Scope

The scope of this document is to give a broad and detailed explanation about the requirements of the HoopStats project. It explains the main requirements defined by the customer as well as the underlying requirements defined by the project team to support the main requirements. It contains the explanation of each requirement as well as their deadlines.

1.4 Definitions and acronyms

1.4.1 Acronyms and abbreviations

Acronym or abbreviation	Definitions
XML	*Extensible Markup Language*
NFR	*Non-functional Requirements*
FQ	*Flexible Query*
VR	*Visual Representation*

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
- Design Definition
  - http://www.fer.unizg.hr/_download/repository/Design_Description%5B6%5D.pdf
- DatabaseBasketball Home
  - http://www.databasebasketball.com/
- Project Web application home
  - http://hoopstats.tk/

2. Requirements Description

To give a general idea about the functionality the application will provide, all of the functional requirements of the application have been exposed in this chapter. These general requirements are the basic building blocks for the definition of more detailed functionality and requirements.

2.1 Introduction of Application

The HoopStats project is envisioned as a reimaging of the web application http://databasebasketball.com/ . This application provides basketball statistics which are free to use by anybody. Unfortunately the application does not provide modern visualization features for the statistical data other than tables and plain text. Furthermore finding specific data in the current application is difficult task, as users have to search for the data in the tables by themselves. Figures 2.1 and 2.1 show examples from the web application:

Mahmo Abdul-rauf

Mahmoud Abdul-Rauf ((Chris Wayne Jackson))
Position: G
Height: 6' 1" **Weight:** 162
Born: 3/9/1969, in Gulfport, MS, USA
High School: Gulfport, in Gulfport, MS
College: Louisiana State University

About bio info

Draft History
1990 NBA - Round 1 by DEN

**Player News

Regular Season Stats
Click on **column header** to sort

Figure 2.1 Player data from databasebasketball.com

Regular Season Stats
Click on **column header** to sort

Year	Age	Team	Lg	G	Min	Pts	PPG	FGM
1972-73	3	GSW	NBA	46	629	208	4.5	82
1972-73	3	BUF	NBA	9	134	53	5.9	25
1990-91	21	DEN	NBA	67	1505	942	14.1	417
1991-92	22	DEN	NBA	81	1538	837	10.3	356
1992-93	23	DEN	NBA	81	2710	1553	19.2	633
1993-94	24	DEN	NBA	80	2617	1437	18.0	588
1994-95	25	DEN	NBA	73	2082	1165	16.0	472
1995-96	26	DEN	NBA	57	2029	1095	19.2	414
1996-97	27	SAC	NBA	75	2131	1031	13.7	411
1997-98	28	SAC	NBA	31	535	227	7.3	103
2000-01	31	VAN	NBA	41	486	266	6.5	120
9 Season Totals				586	15633	8553	14.6	3514

M - MVP
N - All NBA First Team
A - All Star

Figure 2.2 Table season statistics from databasebasketball.com

The goal of the HoopStats project is to take the free-to-use data from the web and create a new system, which would provide more dynamic and visually appealing approach. The application provides mechanism for users to create flexible queries and filter data. This is achieved by providing a mechanism with which users can choose from different templates for querying the application. Here the users can choose e.g. a player or team and select the data that they are interested in. After that they can combine this template with conditions to filter out the data that is not of interest.

Furthermore the data output shown in Figure 2.1 and 2.2 will be additionally visualized with a more appealing visual forms. This style will be more user-friendly and will allow users to more intuitively interpret what has been presented to them. Figure 2.3 and 2.4 represent examples of how the visual data output might look like:

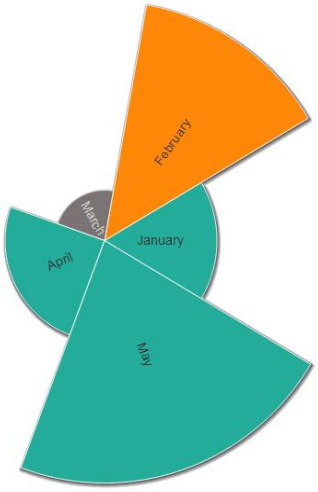


Figure 2.3 Pie chart for data visualization

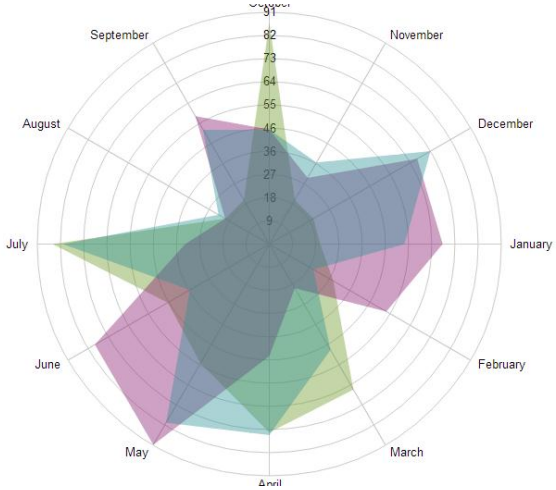


Figure 2.4 Net chart for data visualization

In addition to the visual changes a mobile application for Android phones will be developed to satisfy the needs of a mobile phone users.

The following list summarizes the main features of HoopStats again:

- Users can make their own flexible basketball related queries for data filtering
- Visuals representation of data in the application are improved by using modern visual output
- Mobile application to use HoopStats on-the-go

2.2 General requirements

The HoopStats project is divided into three main requirements which represent the basic concepts and functionality which are provided by the application.

2.2.1 FQ: Flexible creation of queries

The users of the application should be able to create their own statistics at runtime. This is done by giving them the possibility to combine predefined template blocks to create queries. This feature enables users to filter data and to receive output tailored specially according to their needs. To ensure that the users cannot combine templates which don't allow for useful statistics, constraints have to be put in place.

2.2.2 VR: Visual representation of data

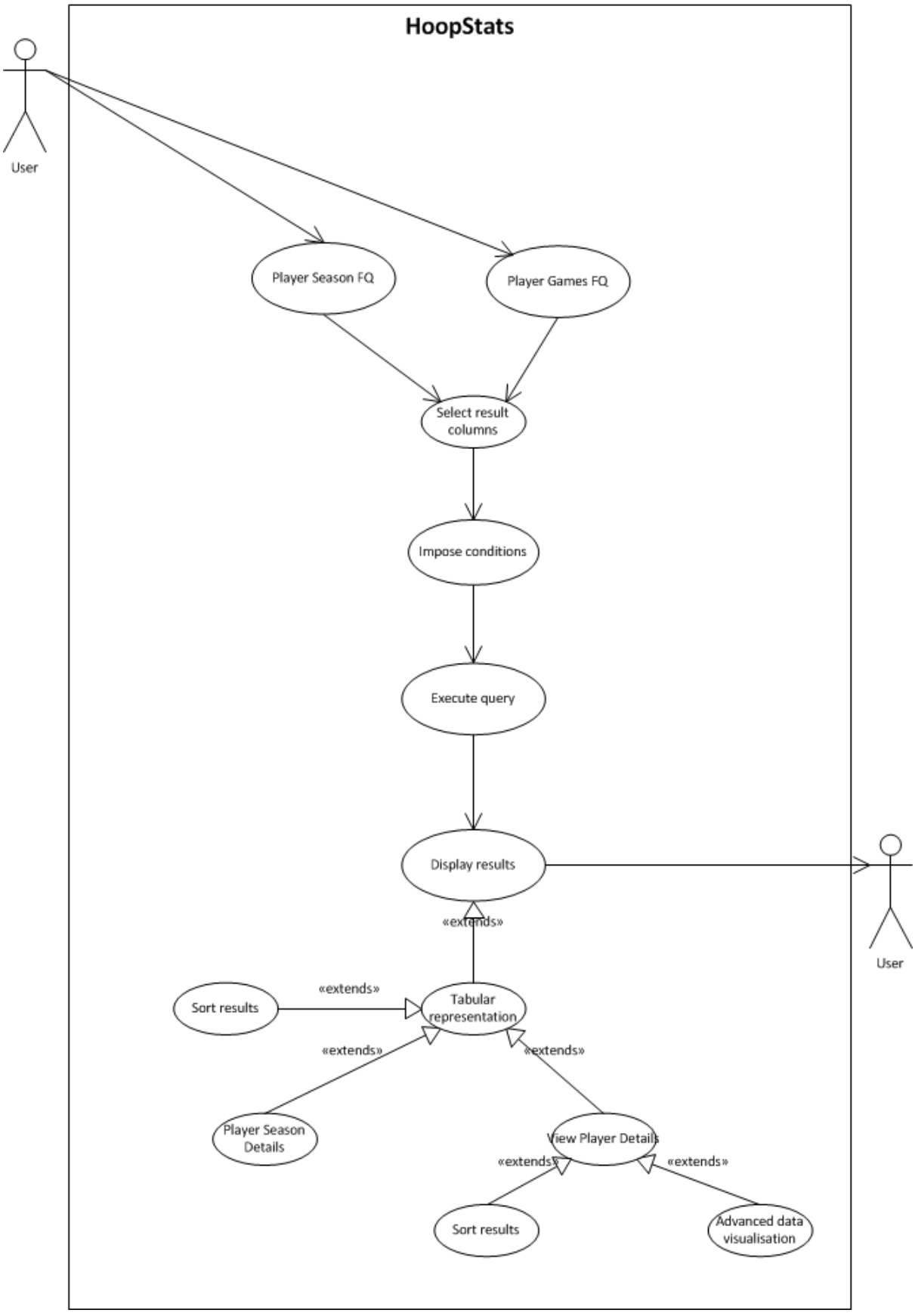
The second general requirement is to provide a visual representation of the data returned as a result from the query execution. For this purpose a visual output system has been developed, which allows for different means of graphical out e.g. pie, line, column charts or tables. The system allows the users to choose their own visual representation of the data obtained from the previously executed query.

3. Use Cases

In this chapter the main use case models for the HoopStats application will be explained and the individual use cases will be described. The whole application will be explained through three main use cases for advanced flexible querying. Additional actions after this are represented as extensions to the main use cases.

3.1 Use case model 1: Player data advanced querying

Use case model for flexible querying players' data is the most complex use case in the application. This is due to the fact that players' data has two main aspects for querying: players' statistical data for seasons and players' statistical data for individual games. Since this use cases are very similar to each other, they are represented in the same use case diagram.


HoopStats
User
Player Season FQ
Player Games FQ
Select result columns
Impose conditions
Execute query
Display results
User
«extends»
Sort results
«extends»
Tabular representation
«extends»
«extends»
Player Season Details
View Player Details
«extends»
«extends»
Sort results
Advanced data visualisation

3.1.1 Use case UC1: Player season FQ

Use case ID	UC1
Name	*Player season data FQ*
Goal	*Select template for showing data for a player season*
Participating actors	User
Precondition	
Main scenario	• *Select user season data from the application main menu.*
Exceptions	
Extensions	
Dependent UC	

3.1.2 Use case UC2: Player games FQ

Use case ID	UC2
Name	
Goal	*Select template for showing data for a player games*
Participating actors	User
Precondition	
Main scenario	• *Select user games data from the application main menu.*
Exceptions	
Extensions	
Dependent UC	

3.1.3 Use case UC3 Select result columns

Use case ID	UC3
Name	*Select result columns*
Goal	*Select set of columns from the displayed options to be shown as columns in result table.*
Participating actors	User
Precondition	
Main scenario	• *List of all available columns for particular entity.* • *User selects one or more columns to be shown as result from query execution.*
Exceptions	*User does not select any columns.*
Extensions	
Dependent UC	UC1, UC2

3.1.4 Use case UC4: Impose condition

Use case ID	UC4
Name	*Impose condition*
Goal	*User selects columns from the predefined set and inserts values for each of them to filter the data according to some criteria. The goal is to reduce the result set to data of interest.*
Participating actors	User
Precondition	User must have selected a template.
Main scenario	• *List of all available columns for particular entity.* • *User selects one or more columns to be shown as result from query execution.* • *User inputs values for each of the selected columns.*
Exceptions	
Extensions	
Dependent UC	UC1, UC2

3.1.5 Use case UC5: Execute query

Use case ID	UC5
Name	*Execute query*
Goal	*Submit the form and pass input to the parsing engine.*
Participating actors	User
Precondition	User must have selected at least one column to be displayed in the result table.
Main scenario	• *User selects some of the predefined conditions to apply to the data set*.
Exceptions	
Extensions	
Dependent UC	UC3

3.1.6 Use case UC6: Display results

Use case ID	UC6
Name	*Display results*
Goal	*Display the results yielded from the query to the user.*
Participating actors	User
Precondition	
Main scenario	• *Data yielded from the query is presented to the user.*
Exceptions	*Query yielded no results.*
Extensions	
Dependent UC	UC5

3.1.7 Use case UC7: Tabular representation of data

Use case ID	UC7
Name	*Tabular representation of data*
Goal	*Display results yielded from query in tabular form. Default data visualization.*
Participating actors	User
Precondition	UC1-3, UC4, UC5 and UC6 must be completed
Main scenario	• *Data is displayed*. • *Start the process again or go to some of the previous states.*
Exceptions	
Extensions	UC7, UC8, UC9
Dependent UC	UC6

3.1.8 Use case UC8: Sort result table

Use case ID	UC8
Name	*Sort result table*
Goal	*Sort table by some of the columns.*
Participating actors	User
Precondition	UC7
Main scenario	• *Data is displayed in table*. • *Click on column header to sort ASC or DESC order*.
Exceptions	
Extensions	
Dependent UC	UC7

3.1.9 Use case UC9: Player season details

Use case ID	UC9
Name	*Player season details*
Goal	*View player details for a particular season in tabular form.*
Participating actors	User
Precondition	UC7
Main scenario	• *Click on the season in the year column.*
Exceptions	
Extensions	
Dependent UC	UC7

3.1.10 Use case UC10: Player details

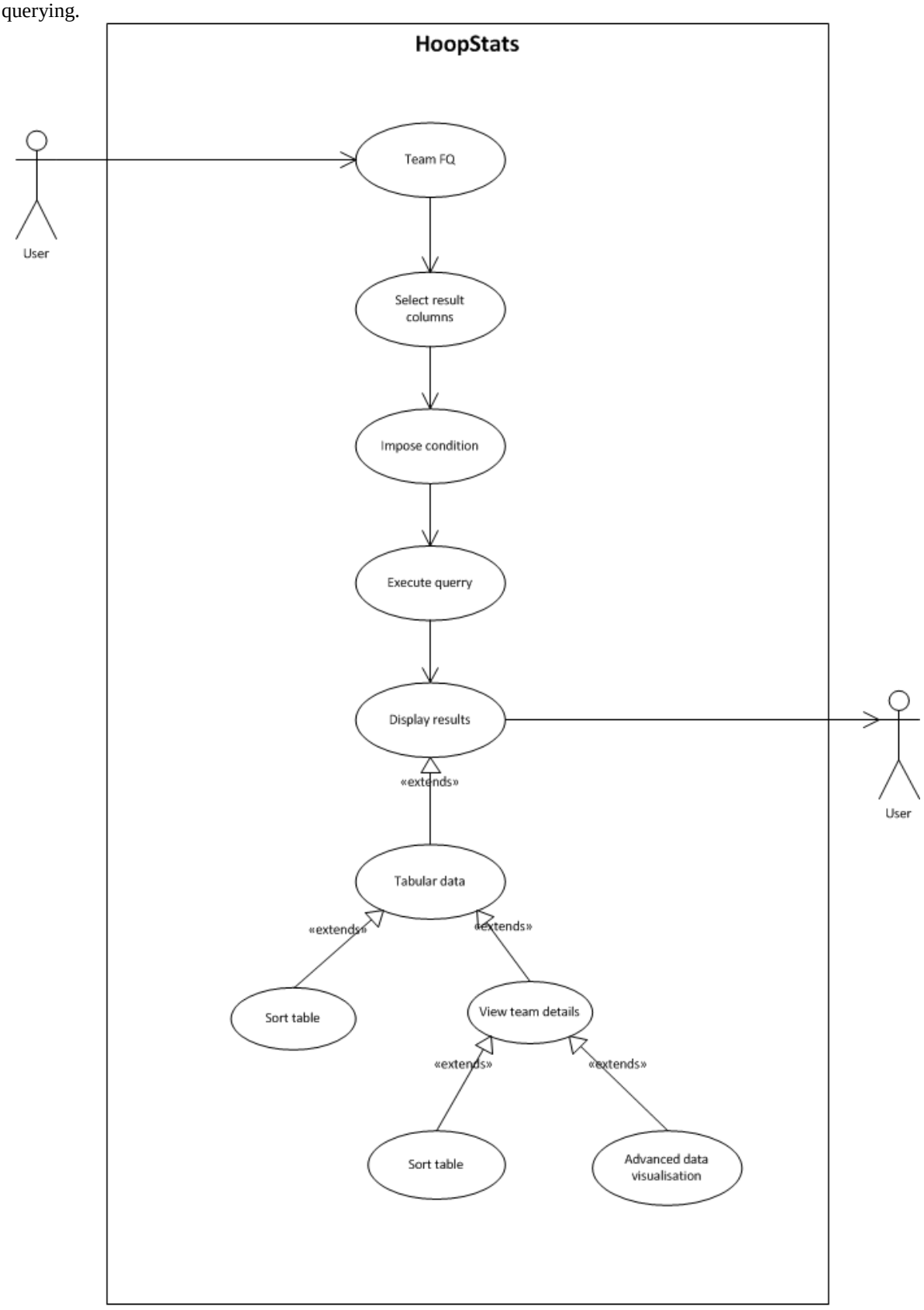
Use case ID	UC10
Name	*Player details*
Goal	*View player details in tabular form.*
Participating actors	User
Precondition	UC8
Main scenario	• *Click on the player name column entry*
Exceptions	
Extensions	UC11
Dependent UC	UC7

3.1.11 Use case UC11: Advanced data visualization

Use case ID	UC11
Name	*Advanced data visualization*
Goal	*Visualize player data using the pie, line or column chart*
Participating actors	User
Precondition	UC10
Main scenario	• *Select column to visualize* • *Select type of visualization (line, pie, column)* • *Execute visualization function*
Exceptions	
Extensions	
Dependent UC	UC10

3.2 Use case model 2: Team data advanced querying

This is the use case diagram that shows how teams' data can be flexible queried using the system for advanced querying.



3.2.1 Use case UC20: Team FQ (Flexible querying)

Use case ID	UC20
Name	*Team FQ*
Goal	*Select template for showing data for a teams' data*
Participating actors	User
Precondition	
Main scenario	• *User selects Team FQ from the main application menu.*
Exceptions	
Extensions	
Dependent UC	

3.2.2 Use case UC21: Select result columns

Use case ID	UC21
Name	*Select result columns*
Goal	*Select columns to be displayed as result from the query execution.*
Participating actors	User
Precondition	UC20
Main scenario	• *User selects columns that want to be shown in the result table.*
Exceptions	
Extensions	
Dependent UC	

3.2.3 Use case UC22: Impose condition

Use case ID	UC22
Name	*Impose condition*
Goal	*User selects columns from the predefined set and inserts values for each of them to filter the data according to some criteria. The goal is to reduce the result set to data of interest.*
Participating actors	User
Precondition	User must be on team advanced querying page.
Main scenario	• *User selects one or more columns to be shown as result from query execution.* • *User inputs values for each of the selected columns.*
Exceptions	
Extensions	
Dependent UC	

3.2.4 Use case UC23: Execute query

Use case ID	UC23
Name	*Execute query*
Goal	Pass the selected columns as results and conditions to the parsing engine
Participating actors	User
Precondition	User must have selected at least one result column
Main scenario	• User press the "Execute" button • Form has been submitted.
Exceptions	*User has not selected any result column.*
Extensions	
Dependent UC	UC21, UC22

3.2.5 Use case: UC24: Display results

Use case ID	UC24
Name	*Display results*
Goal	*Display the results yielded from the query to the user.*
Participating actors	User
Precondition	
Main scenario	• *Data yielded from the query is presented to the user.*
Exceptions	*Query yielded no results.*
Extensions	
Dependent UC	UC23

3.2.6 Use case UC25: Tabular data representation

Use case ID	UC25
Name	*Tabular representation of data*
Goal	*Display results yielded from query in tabular form.* *Default data visualization.*
Participating actors	User
Precondition	U23
Main scenario	• *Data is displayed to the user.*
Exceptions	
Extensions	
Dependent UC	UC24

3.2.7 Use case UC26: Sort table

Use case ID	UC26
Name	*Sort result table*
Goal	*Sort table by some of the columns.*
Participating actors	User
Precondition	
Main scenario	• *Data is displayed in table.* • *Click on column header to sort ASC or DESC order.*
Exceptions	
Extensions	
Dependent UC	UC25

3.2.8 Use case UC27: View team details

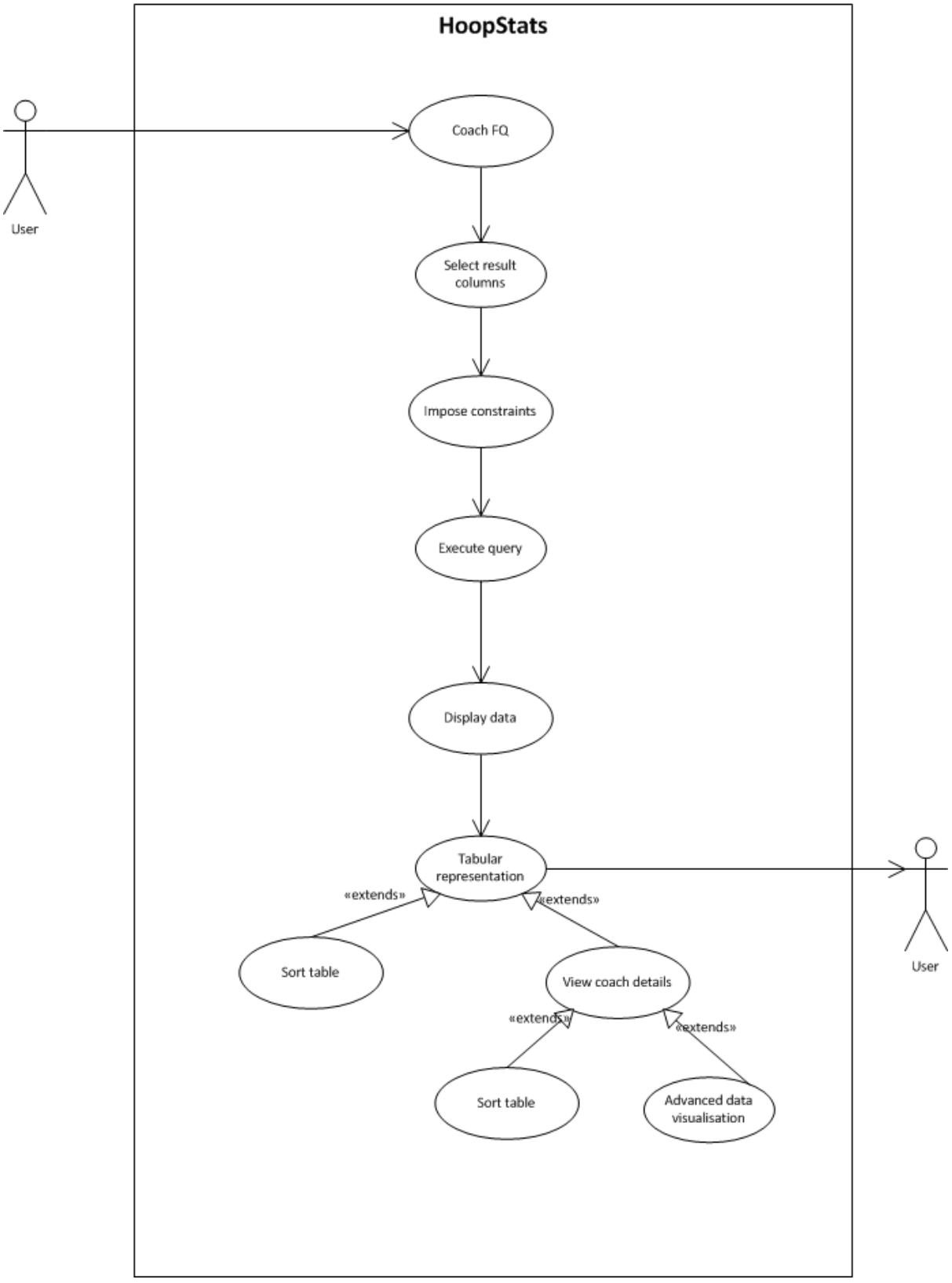
Use case ID	UC27
Name	*Team details*
Goal	*View team details in tabular form.*
Participating actors	User
Precondition	UC25
Main scenario	• *Click on the team name column entry*
Exceptions	
Extensions	UC28
Dependent UC	

3.2.9 Use case UC28: Advanced data visualization

Use case ID	UC28
Name	*Advanced data visualization*
Goal	*Visualize team data using the pie, line or column chart*
Participating actors	User
Precondition	UC27
Main scenario	• *Select column to visualize* • *Select type of visualization (line, pie, column)* • *Execute visualization function*
Exceptions	
Extensions	
Dependent UC	UC27

3.3 Use case model 3: Coach data advanced querying

Use case diagram that shows how coaches' data can be flexible queried using the system for advanced querying.



3.3.1 Use case UC30: Coach flexible querying

Use case ID	UC30
Name	*Coach FQ*
Goal	*Select template for showing data for a coaches' data*
Participating actors	User
Precondition	
Main scenario	• *User selects Coach FQ from the main application menu.*
Exceptions	
Extensions	
Dependent UC	

3.3.2 Use case UC31: Select result columns

Use case ID	UC31
Name	*Select result columns*
Goal	Select columns to be displayed as result from the query execution.
Participating actors	User
Precondition	UC30
Main scenario	• *User selects columns that want to be shown in the result table.*
Exceptions	
Extensions	
Dependent UC	

3.3.3 Use case UC32: Impose conditions

Use case ID	UC32
Name	*Impose condition*
Goal	*User selects columns from the predefined set and inserts values for each of them to filter the data according to some criteria. The goal is to reduce the result set to data of interest.*
Participating actors	User
Precondition	User must be on team advanced querying page.
Main scenario	• *User selects one or more columns to be shown as result from query execution.* • *User inputs values for each of the selected columns.*
Exceptions	
Extensions	
Dependent UC	

3.3.4 Use case UC33: Execute query

Use case ID	UC33
Name	*Execute query*
Goal	Pass the selected columns as results and conditions to the parsing engine
Participating actors	User
Precondition	User must have selected at least one result column
Main scenario	• *User press the "Execute" button* • *Form has been submitted.*
Exceptions	*User has not selected any result column.*
Extensions	
Dependent UC	UC31, UC32

3.3.5 Use case UC34: Display data

Use case ID	UC44
Name	*Display results*
Goal	*Display the results yielded from the query to the user.*
Participating actors	User
Precondition	
Main scenario	• *Data yielded from the query is presented to the user.*
Exceptions	*Query yielded no results.*
Extensions	
Dependent UC	UC33

3.3.6 Use case UC35: Tabular representation

Use case ID	UC35
Name	*Tabular representation of data*
Goal	*Display results yielded from query in tabular form.* *Default data visualization.*
Participating actors	User
Precondition	U33
Main scenario	• *Data is displayed to the user.*
Exceptions	
Extensions	
Dependent UC	UC34

3.3.7 Use case UC36: Sort table

Use case ID	UC36
Name	*Sort table*
Goal	*Sort table by some of the columns.*
Participating actors	User
Precondition	
Main scenario	• *Data is displayed in table.* • *Click on column header to sort ASC or DESC order.*
Exceptions	
Extensions	
Dependent UC	UC35

3.3.8 Use case UC37: View coach details

Use case ID	UC37
Name	*Coach details*
Goal	*View coach details in tabular form.*
Participating actors	User
Precondition	UC35
Main scenario	• *Click on the coach name column entry*
Exceptions	
Extensions	UC38
Dependent UC	

3.3.9 Use case UC38: Advanced data visualization

Use case ID	UC38
Name	*Advanced data visualization*
Goal	*Visualize coach data using the pie, line or column chart*
Participating actors	User
Precondition	UC37
Main scenario	• *Select column to visualize* • *Select type of visualization (line, pie, column)* • *Execute visualization function*
Exceptions	
Extensions	
Dependent UC	UC37

4. Requirements Definition

4.1 Requirement Group Definitions

Identification	Description	Rem.
VR	Requirements for the visual data representation	
FQ	Requirements for the flexible data querying	
DV	Requirements for details view	
DI	Requirements for the integration of external data sources	

4.2 Requirement Sources

Source	Description	Rem.
Ctm	Requirements from customer for the HoopStats application	
Tm	Internal Requirements	

4.3 Requirement definitions

Identity	Status	Priority	Description	Source
VR	I	1	Visual representation of data	Ctm
VR 1	I	1	Provide static graphical data output	Ctm
VR 1.1	I	1	Provide tabular representation of data.	TM
VR 1.2	I	1	Provide pie chart visual representation.	TM
VR 1.3	I	1	Provide graph visual representation.	TM
VR 2	I	2	Provide dynamic graphical data output	Ctm
VR 2.1	I	2	Provide table output	TM
VR 2.2	I	2	Provide pie chart visual representation.	TM
VR 2.3	I	2	Provide graph visual representation.	TM
VR 3	I	1	Provide webpage to display graphical output	Ctm
FQ	I	1	Flexible creation of queries	Ctm
FQ 1	I	1	Deliver templates to create queries	Ctm
FQ 1.1	I	1	Player table template	Ctm
FQ 1.2	I	1	Team table template	Ctm
FQ 1.3	D	1	Player comparison template	Ctm
FQ 1.4	I	1	Templates for properties as projections	TM
FQ 1.5	I	1	Templates for conditions	TM
FQ 2	I	1	Workflow to structure the template selection of the user	Ctm
FQ 2.1	D	1	Selection of table or comparison template	Ctm
FQ 2.2	I	1	Selection of property template	TM
FQ 2.3	I	1	Selection of condition template	Tm
FQ 2.4	I	1	Selection for different graphical output	Ctm
FQ 3	I	1	Constraints to cut out unreasonable queries	Ctm
FQ 3.1	I	1	Constraint to cut out unreasonable properties	Tm
FQ 3.2	D	1	Constraint to cut out unreasonable comparisons	Tm
FQ 3.3	A	3	Modify previously executed query	Ctm
DV	I	1	Details view for entities	Tm
DV1	I	1	Details view for players	Tm
DV2	I	1	Details view for teams	Tm

DV3	I	1	Details view for coaches	Tm
DV4	I	2	Season details for player	Tm
Dv5	I	2	Games details for player	Tm
DI	I	1	Integration of basketball information	Tm
DI 1	I	1	Integration of CSV files from Databasebasketball.com	Tm
DI 1	I	1	Integration of team images from Wikipedia.com	Tm
DI 1	D	3	Integration of player images from Wikipedia.com	TM
DI 1	A	1	Scrapping game logs data from http://databasebasketball.com/	Tm

Requirement status:

I = initial (this requirement has been identified at the beginning of the project),
D = dropped (this requirement has been deleted from the requirement definitions),
H = on hold (decision to be implemented or dropped will be made later),
A = additional (this requirement was introduced during the project course).

Requirement priority:

1 = First iteration of development
2 = Second iteration of development
3 = Third iteration of development