Software Requirements Specification
for

Ideal Team Builder
(STORM)

Version 1.1.12.1
Prepared by COS 750

1 October 2013

NBNBNB Some things to think about:

There are currently some inconsistencies in the document - they need to be cleaned up:

- **Are we going to talk about "the user" or "the Administrator"?**
  - **That depends on if we are talking about a user with elevated privileges (administrator) or a normal user of the system. (Richard)**
- **Are we going to talk about "the system" or just refer to it by name? (STORM)**
  - **Good point, we as a group never decided on this. Needed to be specified in the document conventions. (Richard)**
- **I also suggest removing all the colons (:) from the end of the REQ-xx and Scenario xx headings, as it clogs the table of contents with a million colons.**
  - **Most of the sections are clean**
- **I updated the default styles of the document. Please use them! Heading 1 for section headings, Heading 2 for sub headings etc etc. This makes changing the formatting of the document later on much much easier and it makes generating a correct TOC very easy!**
  - **Changed all the headings for 4.11 to 4.14 and section 5.4.1 to section 5.4.3 to use the same format**

Please discuss.

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Linda Marshall	1/10/2013	Document creation from template from: www.csc.villanova.edu/~tway/courses/csc4700/s2008/.../srs_**template**.doc	1.0
Richard Oellermann	3/10/2013	Initial purpose formulation	1.0.1
Richard Oellermann	16/10/2013	Project title, REQ1, REQ2, REQ3, REQ4, Section 4.2, Use case for section 4.2	1.0.2
Richard Oellermann	19/10/2013	REQ5, REQ6, Section 4.3, use case for section 4.3	1.0.3
Richard Oellermann	26/10/2013	Sections 1.3, 1.4, 2.3, 2.4	1.0.4
Richard Oellermann	28/10/2013	Section 2.1, 2.2, 2.5	1.0.5
Mercia Malan	30/10/2013	Skeleton for Section 4.4. REQ11,12, Section 4.5.1, 4.5.2 and 4.5.3. Updated document version above.	1.0.6
Sandra, Anthony, Chris	30/10/2013	Added reporting's non-functional requirements to section 5	1.0.7
Nico van Vuuren	30/10/2013	Added REQ 7-9. Changed numbering of REQ 11-12 to 10-11 due to omittance of initial REQ 10	1.0.8
Linda Marshall	31/10/2013	Included page numbers	1.0.8.1
Richard Oellermann	31/10/2013	Section 3.3, 3.4	1.0.9
Richard Oellermann,	31/10/2013	Added criteria's non-functional requirements to section 5	1.1.0

Nico van Vuuren, Mercia Malan			
Anthony Burgess, Chris Kirkwood, Sandra Passetti	31/10/2013	Added section 4.6 and REQ 12-16, started sections 4.7-4.8 and REQ 17-23.	1.1.1
Anthony Burgess, Chris Kirkwood, Sandra Passetti	1/11/2013	Completed sections 4.7-4.8 and REQ 17-23; Added use case diagram for reporting (i.e. sections 4.6-4.8).	1.1.2
Ernest Mashele, Thabang Petje	03/11/2013	Added section 6.1; Completed SEC-1 and SEC-2	1.1.3
Daniel Louw	03/11/2013	Added section 4.9 and 4.10	1.1.4
Daniel Louw	03/11/2013	Created a styles format that must be adhered to, see discussion on page 2. Updated the TOC	1.1.4.1
Christoff Kok and Philip Venter	03/11/2013	Added section 4.11 to 4.14 and section 5.4.1 to section 5.4.3	1.1.4.2
Christoff Kok and Philip Venter	04/11/2013	Update section 4.11 to 4.14 and section 5.4.1 to section 5.4.3	1.1.4.3
Daniel Louw	04/11/2013	Added section 1.2	1.1.5
	04/11/2013	Added Section 1.3	1.1.6
Richard Oellermann	04/11/2013	Reformulated sections 2.4 and 2.5	1.1.7
Ernest Mashele, Thabang Petje	04/11/2013	Added Section 6.2	1.1.8
Werner Scheffer	04/11/2013	Added Sections 4.x.4 to sections 4.1-4.14. Added additional information to section 4.9	1.1.9
Werner Scheffer, Daniel Louw	04/11/2013	Added UI descriptions to all the feature requirements in section 4	1.1.10

Anthony Burgess, Chris Kirkwood, Sandra Passetti	11/11/2013	Updated requirement numbers for reporting in the non-functional requirements section (5.1 and 5.3)	1.1.11
Richard Oellermann	11/11/2013	Updated requirement numbers for criteria in the non-functional requirements section (5.1 and 5.3)	1.1.12
Mercia Malan	11/11/2013	Updated some of the sections headings to be actual headings so that the table of content works.	1.1.12.1

1. Introduction

1.1 Purpose

Lecturers of the department of Computer Science, which resides at the University of Pretoria, approached students of the module “Educational Software Development” to develop a team shuffling system.

The system is going to be used by the lecturers of the module “Software Engineering” to determine teams for the “Rocking the boat” exercise of the Software Engineering module using a set of lecturer-defined criteria to select the teams with.

This document specifies all known requirements to date for the first version of the whole team shuffling system.

1.2 Document Conventions

This document does not make use of any specific styles or conventions that carry meaning. The only one that can be taken note of is the numbering scheme used when listing requirements. Any requirement noted in this document, regardless of the location of it, is titled with REQ-xx. This makes indexing and searching for requirements throughout the document easier.

1.3 Intended Audience and Reading Suggestions

This document targets the all the users, designers, developers and testers of the end-product, STORM.

This document provides the reader with a high-level overview of the system, which is found in chapter 2. Chapter 3 highlights the external interface requirements of the system. Chapter 4 guides the reader through the features of the system, after which in Chapter 5 the reader can view the non-functional requirements of the system.

Chapter 6 makes the reader aware of any other requirements of the system that is not listed in previous chapters.

1.4 Product Scope

The complete system should enable users to *build* teams, from a list of students, by selecting a set of criteria. This will aid the users in such a way that the users do not have to build the teams manually, which may require a lot of time. The users can spend their time rather on analysing the results of each “Rocking the Boat” round to change the criteria for the next round more effectively.

1.5 References

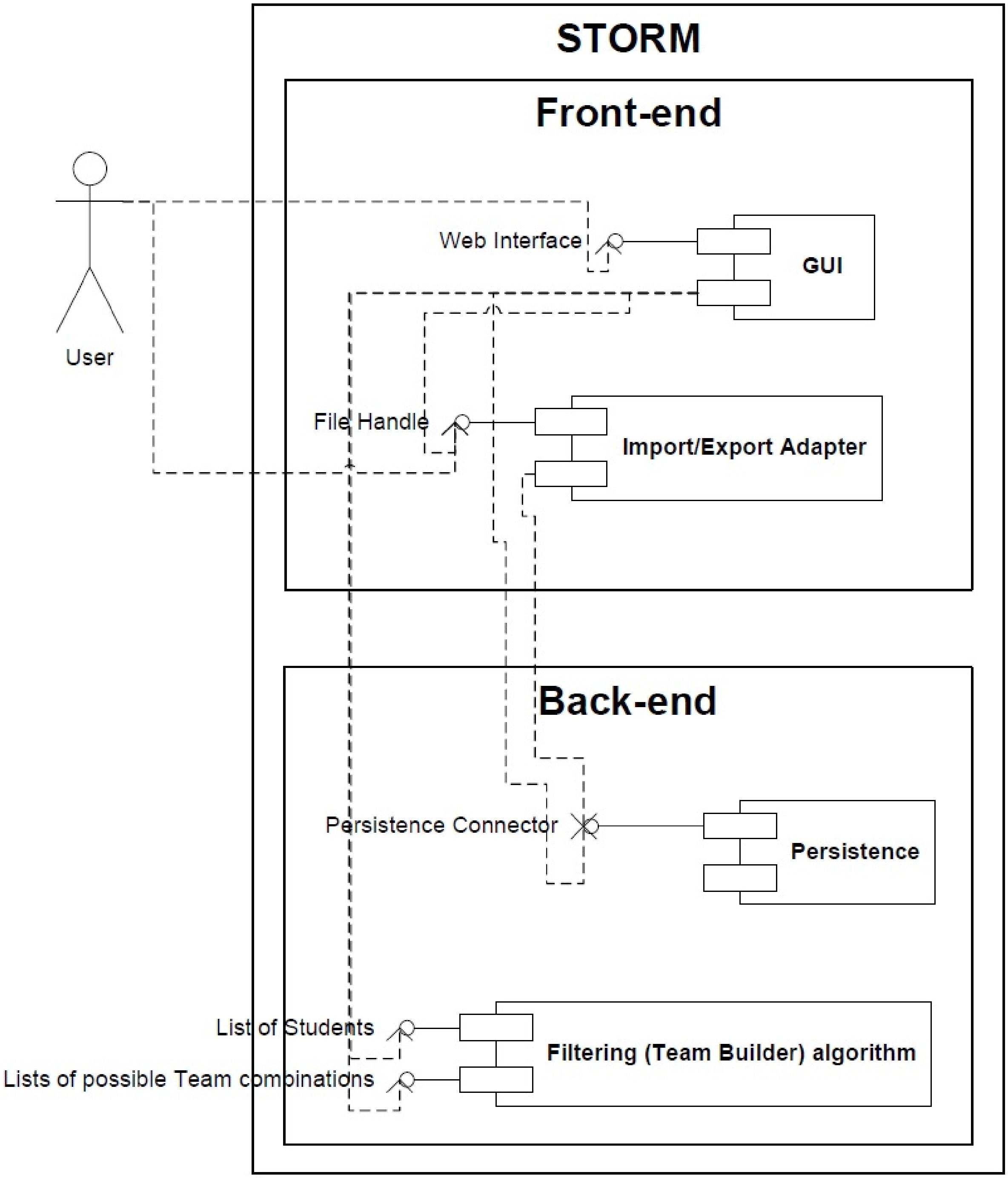
<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>

2. Overall Description

2.1 Product Perspective

STORM is a new self-contained project, that consists out of the following components:

- Front-end
  - GUI
  - Import/Export adapters
- Back-end
  - Persistence
  - The filtering (team builder) algorithm



The above figure visualizes the high-level components of STORM.

2.2 Product Functions

The following major functions, which compose STORM, have been identified in the requirements elicitation discussion:

- Creation/deletion/editing of criteria

- Creation of teams by criteria
- Persistence of students, history and teams
- Reporting
- Importing of student lists
- Authorization mechanism
- Manual manipulation during team building process
- Error handling
- Feedback for each “rocking the boat” round
- User Interface

2.3 User Classes and Characteristics

The following user classes have been identified as possible users of the system:

- Administrators, which govern the overall system, users, database, projects and criteria.
- Senior Lecturers, which govern the the projects that they have set-up, add/edit/remove criteria, select criteria, import students and build teams.
- Lecturers, which can create add/edit/remove criteria, select criteria, import students and build teams.
- Students, which will see the build teams which they are allocated to. But this user class has no privileges on the system. They only see the end result of the system, which is the list of build teams.

2.4 Operating Environment

The following operating environment requirements have been suggested, but can be negotiated further with the client:

- The system should operate on an average desktop computer, with a specified 4 Gigabyte memory, Intel i3 2.4 Ghz processor and 500GB hard-disk.
- To keep the cost down, a Linux flavor operating system should be installed that is easily maintained and can support the persistence backend, which will be the newest stable MySQL database release. The system will run as a server in a tomcat web application container, which will be also installed on the server. The tomcat service should also be the newest stable release. Java 1.7 JVM should also be installed to support the system.

2.5 Design and Implementation Constraints

The following items may constrain the design of STORM:

- The Operating System.
    - This includes the database or persistence component.
- The system should be able to deploy on the mentioned specifications in chapter 2.4.
- The programming language used for the GUI.
- The filter or project builder algorithm component is not constrained to a particular language, but C++ is suggested to increase performance time.

2.6 User Documentation

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>

2.7 Assumptions and Dependencies

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>

3. External Interface Requirements

3.1 User Interfaces

STORM is to be viewed and used through a web browser. So the interface must be designed to reflect this. The UI is to make extensive use of AJAX (or similar) calls to minimize web traffic and create the impression of an actual application being used rather than just a website with dynamic content. At this stage only an interface for a desktop browser is necessary, but the nature of STORM's design makes other interfaces (mobile etc) possible during future expansion.

3.2 Hardware Interfaces

STORM is to run on commodity PC hardware, so no special hardware or hardware interfaces are required.

3.3 Software Interfaces

The newest stable release of MySql should be used as the persistence component. A Java Database Connector should be used to interact with the persistence component. STORM should function on a Linux flavour which is easy to install and maintain.

3.4 Communications Interfaces

Users of STORM are required to use web browsers to view the contents of the front-end. HTTP

will be used as the communication standard for the browsers. All common browsers (Chrome, Firefox, Internet Explorer, Safari, Opera) should be supported.
TCP/IP will be used as the communication standard for inter-machine communication.
All inter-machine communication must be secured using SSL encryption.

4. System Features

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

4.1 System Feature 1

<Don't really say "System Feature 1." State the feature name in just a few words.>

4.1.1 Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

4.1.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

4.1.3 Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use "TBD" as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1:
REQ-2:

4.2 Administrator should be able to set-up the list of criteria

4.2.1 Description and Priority

The administrator or stakeholder responsible for the new project should be able to create a set of criteria that the system should use when determining teams. This feature is part of the core system and is high risk.

4.2.2 Stimulus/Response Sequences

Scenario 1

Stimulus: Administrator creates a new project
Reponse: The project creation is halted and the administrator must first create selection criteria for the project or select from existing selection criteria. If all the criteria have been selected/created, then the project is created.

Scenario 2

Stimulus: Administrator wants to add additional criteria to an existing project.
Response: A form is displayed in which the Administrator can select currently unused criteria for the project, or the Administrator can create new selection criteria.

4.2.3 Functional Requirements

REQ1

The system should only create a new project when at least one criteria is defined for that project.

Error conditions:

- No criteria set-up or selected yet for a new project.
    - The system should direct administrator to create and/or select at least one selection criteria for the project.

REQ2

Only the authorised persons are allowed to change the selection criteria and settings for the project.

Error conditions:

- Unauthorized user accesses project settings to change it.
    - The system should not allow editing or even the viewing of the project settings.
- Unauthorized user attempts to access STORM system.
    - The system should respond with “Invalid username or password”.

REQ3

The system should dynamically add and remove selection criteria to a project, but up to a set level.

Error conditions:

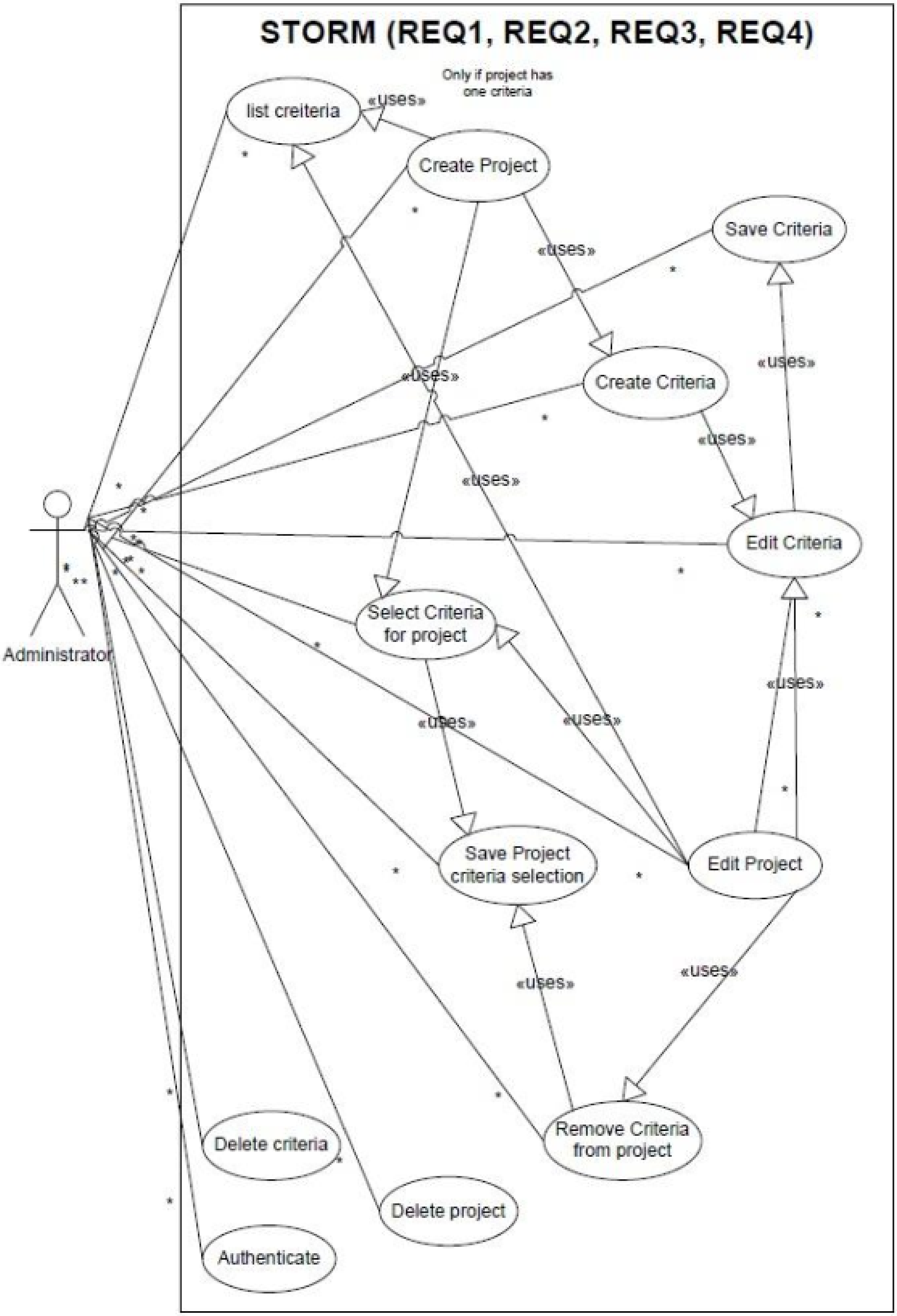
- Administrator attempts to remove all of the selection criteria for a project.
  - The system should allow all remove action up until the last criteria, then a error message appears instructing them on further actions and/or cause of the error.
- Administrator attempts to add another selection criteria which passes the above the global limit of selection criteria amount per project.
  - The system should not allow this to happen and displays an error message.

REQ4

The criteria are persisted and are re-usable.

Error conditions:

- The criteria cannot persist because of some persistence or persistence service error.
  - The system should respond with an error instructing them with further action.


STORM (REQ1, REQ2, REQ3, REQ4)
Only if project has one criteria
list creiteria
«uses»
Create Project
Save Criteria
«uses»
«uses»
Create Criteria
«uses»
«uses»
«uses»
Edit Criteria
Administrator
Select Criteria for project
«uses»
«uses»
«uses»
Save Project criteria selection
Edit Project
«uses»
«uses»
Remove Criteria from project
Delete criteria
Delete project
Authenticate
*
**

4.2.4 User Interface

Administrator should be provided with list of existing Projects, the Administrator can then open an existing project from the list or click on a “Create New” button to create a new project or click on the “Delete” button to delete an existing Project. When creating a new project error providers need to be set up for required fields and the Administrator should not be able to save unless above mentioned requirements are met.

4.3 The system should be able to filter persons according to a set of criteria

4.3.1 Description and Priority

The system should allow per person filtering according to a set of criteria. Using the results, the user may select teams manually. This feature is part of the core of the system, but with a medium priority.

4.3.2 Stimulus/Response Sequences

Scenario 1

Stimulus: The user wants to find students with similar criteria.
Reponse: The system takes the list of persons, their relevant information and the list of user prioritized selection criteria as input and returns the list of matching students.

Scenario 2

Stimulus: The user wants to find students with similar criteria and add each student individually to a team..
Reponse: The system filters the list of students according to the list of criteria and displays them. The system should give the user the option to form teams.

4.3.3 Functional Requirements

REQ5

The system should be able to provide per person filtering according to a set of selection criteria and the history of each person.

Error conditions:

- No students were found matching the criteria.
  - The system shows an empty list to the user with an explanation that no results were found.
- No criteria selected.
  - The system should not start the filtering process and indicates to the user that no

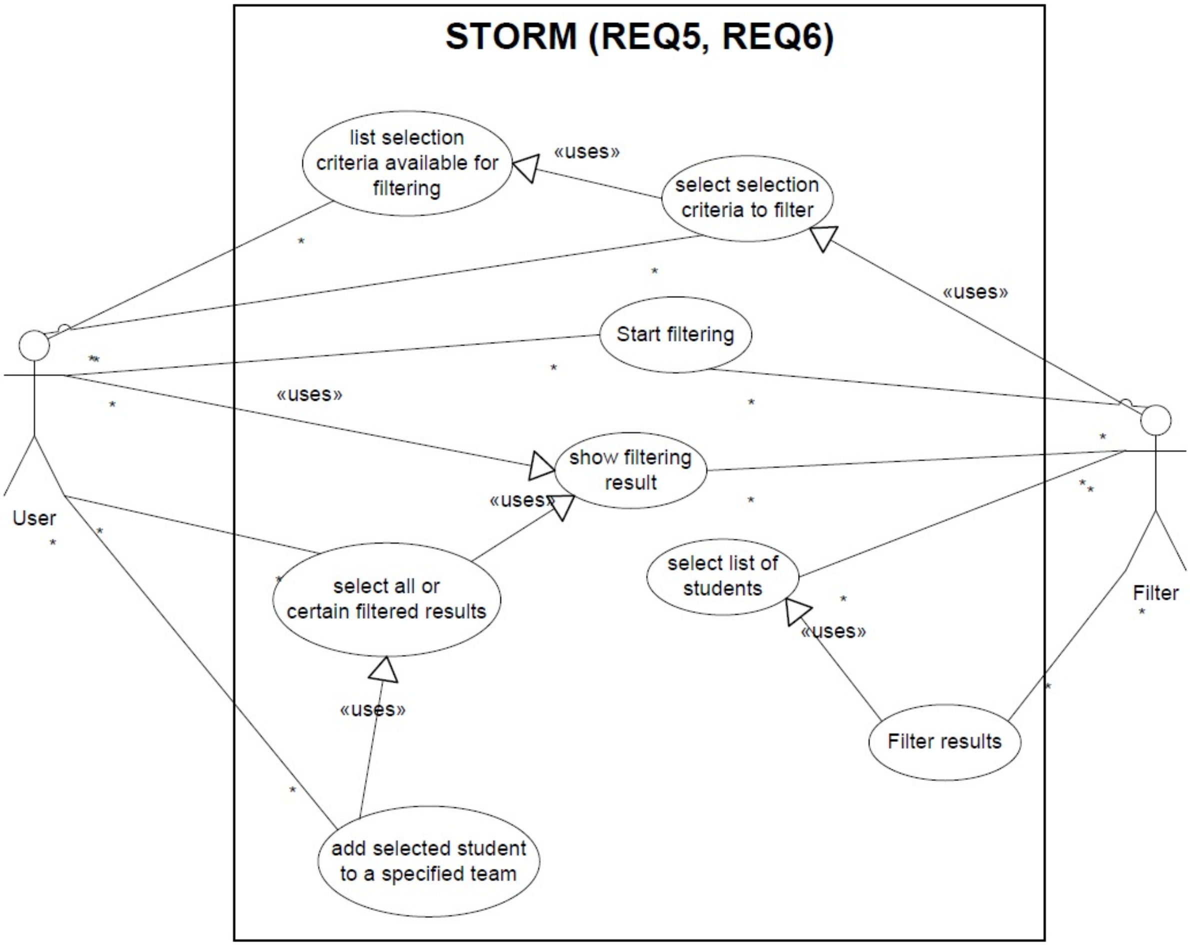
selection criteria was selected.

REQ6

The system should allow for manual team creation based on the user filtered individuals.

Error conditions:

- The target team is already at its limit of capacity.
  - The system should not allow adding of students to the target team and should indicate this to the user via an error message.
- The selected student is already in a team and cannot be added to more than one team.
  - The system should not allow this to happen and displays an error message.


STORM (REQ5, REQ6)
list selection criteria available for filtering
«uses»
select selection criteria to filter
«uses»
Start filtering
«uses»
show filtering result
«uses»
select all or certain filtered results
select list of students
«uses»
«uses»
Filter results
add selected student to a specified team
User
Filter

4.3.4 User Interface

Administrator is provided with list of existing students, the Administrator should then be able to add criteria to a multi select filter field. The Administrator can then click on the “Filter” button to retrieve the list of students that fall under the provided filter criteria.

4.4 The system must allow the user to select the set of criteria for building teams.

4.4.1 Description and Priority

The system should allow the user to select a combination of criteria as/into a set. This is part of the core system with a medium priority.

4.4.2 Stimulus/Response Sequences

Scenario 1

Stimulus: User wants to create a project and add criteria for the project
Response: A selection oriented form opens up from which the user can select different criteria.

Scenario 2

A user wants to edit the criteria for an existing project:
Response: Authentication for user, if user is allowed the selection form opens up with existing criteria set preloaded.

4.4.3 Functional Requirements

REQ7

Minimum amount of criteria for any given round.

Selecting no Criteria for a project is infeasible. The system should provide a minimum amount of criteria to be chosen. This number should be provided by a user setting.

Error conditions:

- No Criteria Selected/Minimum not fulfilled
  - User must be unable to continue with setup/project creation until the minimum has been provided.

REQ8

User should be able to edit criteria at any time. Authentication permettent.

The user should be able to edit the criteria for a project at any time and be able to save his/her preferences. Authentication should be kept in mind, especially once roles are considered for this system. Only users authorised to edit criteria should be allowed to do so.

Error conditions:

- Selected Criteria not validated/has constraints (based on REQ 9)
  - System should prevent saving of faulty criteria.
- User not authorised to change
  - System should prevent user from making changes

REQ9

Feedback on Selection constraints. Prevent selection of criteria dependant on others Ex. If criteria C can't be selected due to Criteria A, user must be unable to select C

If at any stage there are criteria which rely on the outcome of another criteria, the system should prevent the user from selecting/adding this constraint to the selection. For example: An "Electives" criteria can't be selected if a "Degree" criteria has also not been specified. In this case, the "Electives" criteria relies on the chosen "Degree", as each degree has its own electives.

Error conditions:

- Criteria dependent on another selected
  - System should prevent the selection of the dependee.
  - The user should be informed that the criteria depends on another.
  - The user should also be informed which *dependants* there are.
- Criteria dependant on another selected, but the criteria on which it depends was lost/cannot be found
  - System should prevent the selection of the dependee.
  - System should inform the user of a lost *dependency*.
  - Administration rights will be needed to remove/edit criteria.

4.4.4 User Interface

Administrator is provided with list of existing Projects, the Administrator should then be provided with a grid where the Administrator can then add criteria or remove criteria for the project. Before the Administrator saves the changes the above mentioned requirements must be checked.

4.5 The system must have a variety of prioritised criteria according to which teams can be built.

4.5.1 Description and Priority

The system should have a variety of criteria according to which teams can be built, be able to combine these criteria together and to prioritise the criteria from most important to least important.

4.5.2 Stimulus/Response Sequences

Scenario 1

Stimulus: The user wants to select one or more criteria according to which the teams should be build
Reponse: A form is displayed where the user can select multiple criteria and the system then uses this criteria in its team building algorithm

Scenario 2

Stimulus: The user wants to prioritise the combination of criteria, from most important to least important
Reponse: A form is displayed where the user can prioritise the selected criteria. The system then adjusts the team building algorithm to take these priorities into account.

4.5.3 Functional Requirements

REQ10

The system should provide the ability to have a variety of criteria and the user should be able to select multiple criteria from this variety. Criteria includes, but is not exclusive to: Learning Style, Personalities, Expertise, Academic Strength, Degree choice - including electives, team composition History, Size, Location and Peer Review Outcome.

Error conditions:

- There is no data for the specific criteria to use to create teams
  - The system should not allow users to select a criteria if there is no information available for that criteria and should be displayed to the user with a message.
- A criteria cannot be selected more than once
  - The system should not allow users to select the same criteria more than once.

REQ11

The system should provide the ability to prioritise the selected criteria. A default prioritisation should also be in place.

Error conditions:

- Each criteria must have a unique priority
    - The system should notify the user with an error message if the same prioritisation is given twice. Alternatively a drag-drop system should be in place so that criteria can be ordered in only one way thus eliminating the chance of this error.
- Each criteria must have a priority, whether default or specified
    - The system should notify the user with an error message if there is a selected criteria without priority.

4.5.4 User Interface

When creating a project the Administrator is provided with a grid where he can then added and remove criteria. The Administrator should also be able to arrange the criteria by use of “Up” and “Down” buttons or be able to drag the criteria up and down in order to set the priority.Before the Administrator saves the changes the above mentioned requirements must be checked.

4.6 Administrator should be able to generate reports.

4.6.1 Description and Priority

The system should allow a user to specify which constraints they wish to generate a report with. Once these constraints have been chosen the data should be shown. The user can choose whether to view the data in an incremental display or only the final result set. This requirement has high priority.

4.6.2 Stimulus/Response Sequences

Scenario 1

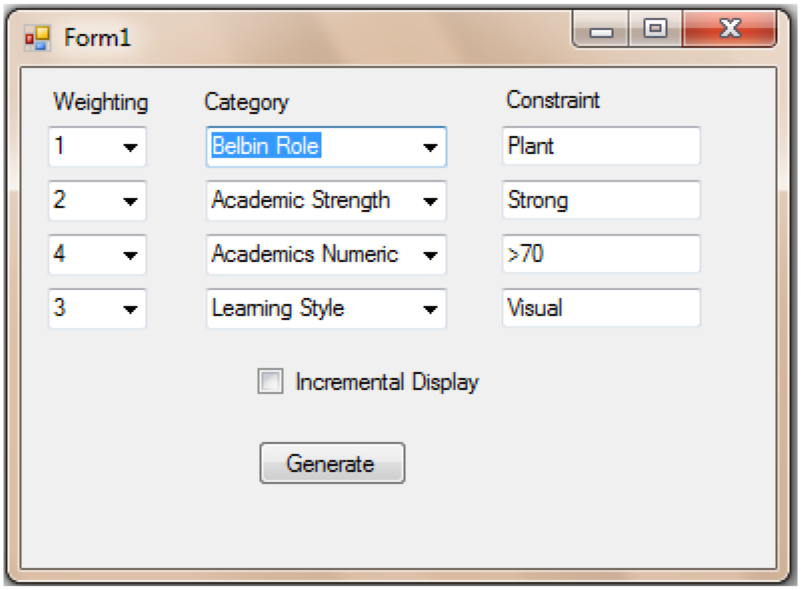
Stimulus: User wishes to generate report showing only strong academic students and students with the Belbin role of Plant.

Reponse: The user must choose Academics as the first constraint from a dropdown menu. The numeric constraint can be entered. Depending on what is stored under Academics it can be “Strong” or >70.

The user then specifies the second constraint which is Belbin Role. The constraint value would be Plant. The order of the constraints should also be specified, particularly when using the incremental display.

Once this is entered, the user can choose to have an incremental display of the data which shows the effect each constraint has the data. For example, the first instance of data would show only with the strong academics constraints whilst the second instance would show both the strong academics and Belbin role constraint. This allows the user to see the affect a constraint has on the resulting data set.

Here is a simple example (The Academic Strength and Academics Numeric are together to shown the different type of variation that can occur):



Weighting	Category	Constraint
1	Belbin Role	Plant
2	Academic Strength	Strong
4	Academics Numeric	>70
3	Learning Style	Visual

□ Incremental Display

4.6.3 Functional Requirements

REQ12:

The system must be able to display the data with all of the applied constraints (not in incremental display).

Error conditions:

- The constraint is not valid.
  - The system should show an error message informing the user the constraint was not corrected entered. (For example: Entering a non-numeric value for a numeric constraint)

REQ13:

The system must be able to display the data segmented with incremental constraints applied (incremental display).

Error conditions:

- The constraint is not valid.
  - The system should show an error message informing the user the constraint was not corrected entered. (For example: Entering a non-numeric value for a numeric constraint)
- Failure to apply constraints properly.
  - When the final incremental step shows more data than the previous steps a system error should be displayed informing the user that data is not valid and the report should be regenerated.

REQ14:
The system should be able to accept the given criteria (e.g. Personalities, Expertise, Academic Strength, Degree Choice, Team composition History, Size of teams, Learning Styles, Location, Peer review outcome) as valid criteria.

REQ15:
The system must be able to correctly order the constraints based on the chosen weighting.

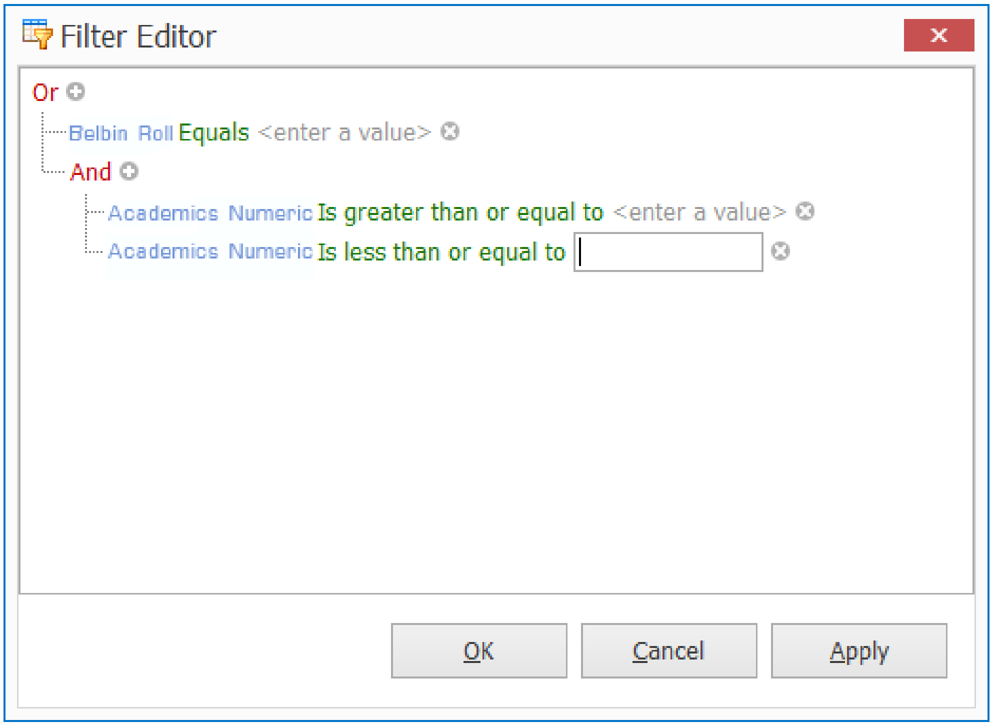
REQ16:
The system must be able to allow the user to create new criteria and constraints.

Error conditions:

- Invalid constraint.
    - Each piece of data must be able to have the constraint applied to it. If there is no comparable piece of data then the error condition should be flagged.

4.6.4 User Interface

The Administrator should be provided with a dialogue where he can set up the constraints for the desired Report the Administrator should be able to set up “and”, “or” and “not” conditions. The Administrator should then press the “Submit” button and be displayed a report the the desired parameters.



```
Or
├─ Belbin Roll Equals <enter a value>
└─ And
   ├─ Academics Numeric Is greater than or equal to <enter a value>
   └─ Academics Numeric Is less than or equal to
```

4.7 Administrator should be able to view reports

4.7.1 Description and Priority

The system should provide functionality for users to view reports once they have been generated, as well as to view reports that have been saved. This requirement has high priority.

4.7.2 Stimulus/Response Sequences

Scenario 1

Stimulus: User generates a report and wants to view it.
Response: The system uses the query data acquired from generating the report and displays it in a well-formatted, human-readable report.

Scenario 2

Stimulus: User wants to view a past (i.e. previously generated and saved) report.
Response: The system asks the user to specify the report that they want to view from a list of reports saved in the system in a generic format. The system then loads the selected report data from the file and displays it in a well-formatted, human-readable report.

4.7.3 Functional Requirements

REQ17:

The system must be able to display report data on a monitor in a well-formatted, human-readable representation.

Error conditions:

- The current user is not authorized to view reports.
  - The system should show an error message informing the user that they do not have the necessary privileges to perform the task.

REQ18:

The system must be able to display a list of previously created reports saved internally in a generic format so that a user can select a specific report.

Error conditions:

- The current user is not authorized to view saved reports.
  - The system should show an error message informing the user that they do not have the necessary privileges to perform the task.
- No reports have been generated yet.
  - The system should display a message to inform the user that no reports have

been generated yet.

REQ19:

The system must be able to load report data from a report saved internally in a generic format.

Error conditions:

- Some file I/O error occurs or the saved report file has been corrupted.
  - The system should show an error message informing the user that the file could not be read.
- The saved report file has been corrupted or is not a valid saved report file.
  - The system should show an error message informing the user that the report could not be loaded, due to the file being corrupted or in the wrong format.

4.7.4 User Interface

The Administrator is to be provided with a list of generated reports he can then open an existing report as to view it or click a “Create New” button to generate a new report.

4.8 Administrator should be able to export reports

4.8.1 Description and Priority

The system should provide a way for users to export reports to a reusable format such as a CSV file that could easily be read into a spreadsheet. This requirement has a medium priority.

4.8.2 Stimulus/Response Sequences

Scenario 1

Stimulus: User wants to export a report to use in a different context.
Reponse: The system asks the user to specify the report they want to export from a list of reports saved in the system in a generic format. The system then asks the user to specify the format they want the report to be exported to.

4.8.3 Functional Requirements

REQ20:

The system must be able to display a list of previously created reports saved internally in a generic format so that a user can select a specific report.

Error conditions:

- The current user is not authorized to view reports.
    - The system should show an error message informing the user that they do not have the necessary privileges to perform the task.
- No reports have been generated yet.
    - The system should display a message to inform the user that no reports have been generated yet.

REQ21:

The system is able to transform a report from the internal generic format into different external formats. These formats include CSV and PDF files.

REQ22:

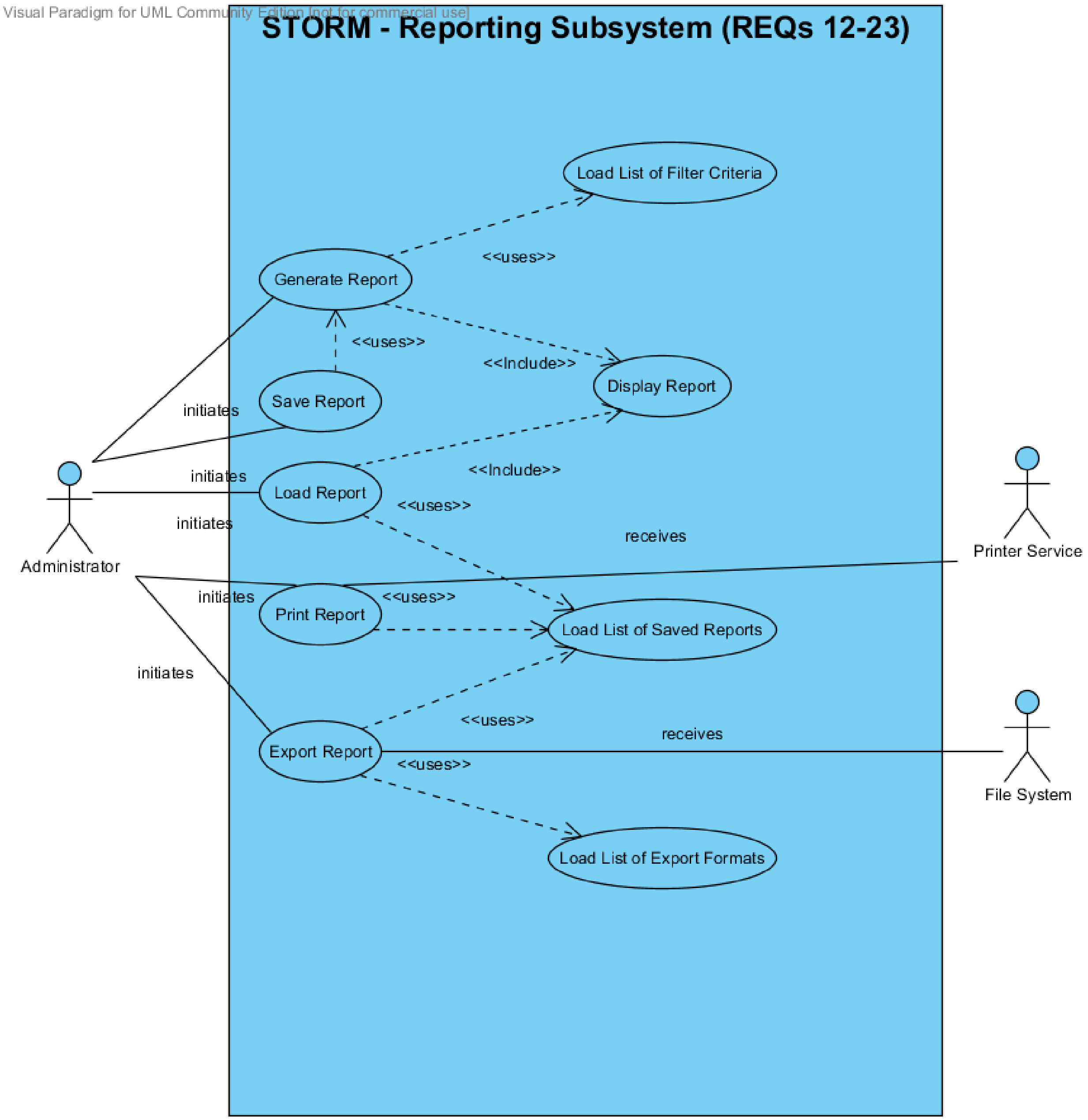
The system must be able to show the user the list of possible export formats to choose from so that the user can select a specific format.

REQ23:

The system is able to transform the selected report from the internal generic format into the specific external format the user chose.

Error conditions:

- No report has been selected.
    - The system should show an error message prompting the user to choose a report first.
- No export format has been selected.
    - The system should show an error message prompting the user to choose an export format first.
- The current user is not authorized to export reports.
    - The system should show an error message informing the user that they do not have the necessary privileges to perform the task.



4.8.4 User Interface

When the Administrator view a report he should have a “Export” button which displays a choice of set Export formats, the Administrator can then select a format he want the report to export to.

4.9 The System must have file import functionality

The administrator must be able to import class lists in the form of a comma separated value (.csv) file or similar.

4.9.1 Description and Priority

In most cases the class list will be located on another system that is not part of STORM. For STORM to function it needs a complete class list of all the students currently enrolled in the course. To manually enter all the information of the students is time consuming and error prone. The system should be able to accept a .csv (or similar) file that has been exported from another course management system that the course/university already uses. This feature is of Medium priority, as the system is still usable without it, although it will become cumbersome.

4.9.2 Stimulus/Response Sequences

Scenario 1

Stimulus: The Administrator wants to import a class list that has been exported from another system.
Response: The system provides an interface where the Administrator can define the structure of the data being imported and all the other options as discussed.

4.9.3 Functional Requirements

REQ-24

The format and structure of .csv files can differ depending on the system where it was exported from. Some systems is possibly unable to alter the structure of an exported .csv. STORM should be able to compensate for this: It must enable the user to define the structure and format of the file prior to importing. STORM should indicate the minimum required fields that is needed to store a student in it's system. The Administrator should be able to indicate which fields in the .csv file corresponds to what fields internal to STORM's database. The Administrator should also be able to identify which fields should be used to update existing information. This process should be a flexible and fluid process, with the Administrator being able to easily change field mappings between imports.

REQ-25

Some .csv exporters include field names in the first row of the file. Obviously it is undesirable for this row to be imported. So the Administrator should be able to instruct STORM to ignore the first row if it contains field headings.

REQ-26

Some fields present in the exported .csv might be irrelevant to STORM It will thus be desirable to ignore some fields completely. The Administrator must be able to instruct STORM which fields to ignore if any. These fields must then not be imported into the system.

4.9.4 User Interface

The Administrator should have a easy to use screen where he can open a comma separated value (.csv) file or similar. The column headers for the file is displayed in a grid then the user can then link the column header to the fields in the system, the Administrator should also be able to identify the columns that need to be used to update existing data and if not found new entries will be created. Once done the Administrator can click on and “Import” button to start the process. The Administrator should also be notified once the import has been completed.

4.10 The System must be able to manage team selection criteria <work in progress>

During each iteration teams will be selected based on some set of criteria that has been entered by the Administrator. The criteria might change between iterations or they might not. STORM must be able to keep track of these sets of criteria and provide the Administrator a means to manage it.

4.10.1 Description and Priority

The system must keep track of criteria used to create teams in previous rounds so that the user can review this information when building new teams. The Administrator might spend a considerable amount of time to assemble the criteria for a team selection and he might wish to reuse this same set of criteria during another team selection iteration.STORM should provide management capabilities to the Administrator to make the reuse, modification and saving of criteria sets easier.

4.10.2 Stimulus/Response Sequences

Scenario 1

Stimulus: The Administrator wishes to apply a criteria set that was used in a previous iteration.
Response: STORM provides a list of all available sets to the Administrator. He can then select a specific set and it is then applied for the current iteration.

Scenario 2

Stimulus: The Administrator wishes to modify an already used set of criteria before applying it to the selection.
Response: STORM provides an interface where the Administrator can select a set from the list. All the criteria are then displayed and the Administrator is able to modify the criteria and save it out to the database.

4.10.3 Functional Requirements

REQ-27

All criteria sets must be automatically saved when used in a selection. It must be identified by a date-stamp by default if a name is not supplied. If a name is supplied then it must be saved under that name instead.

REQ-28

The Administrator must be able to select a saved criteria set to apply to a current iteration.

REQ-29

If the Administrator modifies a selected criteria before it is applied to the iteration then it must be re saved under the same name but a different version. Ancestry of the set must remain intact. If the Administrator chooses to rename the set then it must be saved under this new name with the ancestry still remaining intact.

REQ-30

An interface must be available for the administrator to view, edit and save sets of criteria. This interface must be separate from the team selection iteration interface, as the Administrator might wish to build a set of criteria before any team selection iterations are to take place.

4.10.4 User Interface

The System should provide the Administrator with screen where he can view the teams with their criteria for each iteration.

4.11 Create\Edit\Delete member profiles (students)

4.11.1 Description and Priority

The system should be able to give the user the ability to create team member profiles in a specific format to capture the attributes of the individual. This is a high-priority requirement, because the rest of the system is dependent on the data to generate the teams.

4.11.2 Stimulus/Response Sequences

Scenario 1

Stimulus:

The administrator wants to capture a new profile.

Reponse:

The system provides a UI for capturing member\student profiles:

- Personal Information (e.g. Title, Name, Surname)
- Demographics (e.g. Gender)
- Attributes required (based on the criteria)
    - Learning Styles
    - Belbin Type
    - Academic Record
    - Personality Information

Scenario 2

Stimulus:

The administrator wants to edit a selected profile.

Reponse:

The system retrieves the selected profile's information and gives the user the ability to change the profile or to add\remove new attributes to the profiles.

Scenario 3

Stimulus:

The administrator wants to delete a selected profile.

Reponse:

The system retrieves the selected profile's information and asks for confirmation to delete the user profile. The system should only do a "soft" delete for auditing and retrieval purposes.

4.11.3 Functional Requirements

REQ31

The system must be able to create a new profile and save it to the database.

Error conditions:

- The current user does not have enough privileges\permissions to create a new profile.
- Cannot create duplicate profiles
    - The system should check if the profile does not exist in the database, and if it does, display appropriate message to the user.
- Invalid input fields.
    - The system should show an error message and indicate the required or invalid fields.

REQ32

The system must be able to edit an existing profile and save it to the database.

Error conditions:

- The current user does not have enough privileges\permissions to edit a profile.
- Invalid input fields.
  - The system should show an error message and indicate the required or invalid fields.

REQ33

The system must be able to delete a new profile and make the necessary changes to the record to indicate that the profile is deleted..

Error conditions:

- The current user does not have enough privileges\permissions to delete the profile.

REQ34

The system should provide pre-defined attributes to allow the user to pick the attribute and add it to the profile.

Error conditions:

- Cannot add duplicate attributes to the profile. The user must be notified that the attribute already exists.

4.11.4 User Interface

The Administrator should be provided with a list of profiles he can then open an existing profile and edit or the Administrator can click on a “Create New” button to create a new profile.Before the Administrator saves the changes the above mentioned requirements must be checked.

4.12 Configure\Select the criteria and rules for team creation

4.12.1 Description and Priority

The system should provide a way to setup the criteria and rules for the current project in preparation to generate suggested teams. This is a high-priority requirement.

4.12.2 Stimulus/Response Sequences

Scenario 1

Stimulus:

The user wants to setup the criteria and rules for the current project before team generation.

Reponse:

The system needs to provide a UI that allows the user to select from the existing criteria and rules.

4.12.3 Functional Requirements

REQ35

The system should allow the user to select the criteria from a list of predefined criteria.

Error conditions:

- Cannot add duplicate criteria to the project. The user must be notified that the criteria already exists.

REQ36

The system should add the default rules that are required to the project.

Example Default Rules:

- All students should be in a team
- Students shouldn't be with the same people more than once

REQ37

The system should provide functionality to set the team size.

Error conditions:

- Cannot have a team size smaller than the amount of students available.

REQ38

The user should be able to mark the rules that are allowed to be overridden by the manual manipulation phase after teams are generated.

4.12.4 User Interface

All the available rules and criteria should be displayed on the screen in the form of check boxes or drop down menus. The Administrator should be able to easily select and/or change the set of criteria before performing the team selection. A menu listing save criteria sets should be available that will autofill the current set of criteria. A “Run” button must be available that instructs the system to start performing the team selections.

4.13 Generate suggested teams

4.13.1 Description and Priority

The system should generate suggested teams based on all the information provided, the captured student data and the configuration of the criteria and rules. The system should then display the results in a way that the user can easily evaluate the results visually to distinguish between the teams. Figure 1 below depicts this requirement. This requirement has a high-priority.

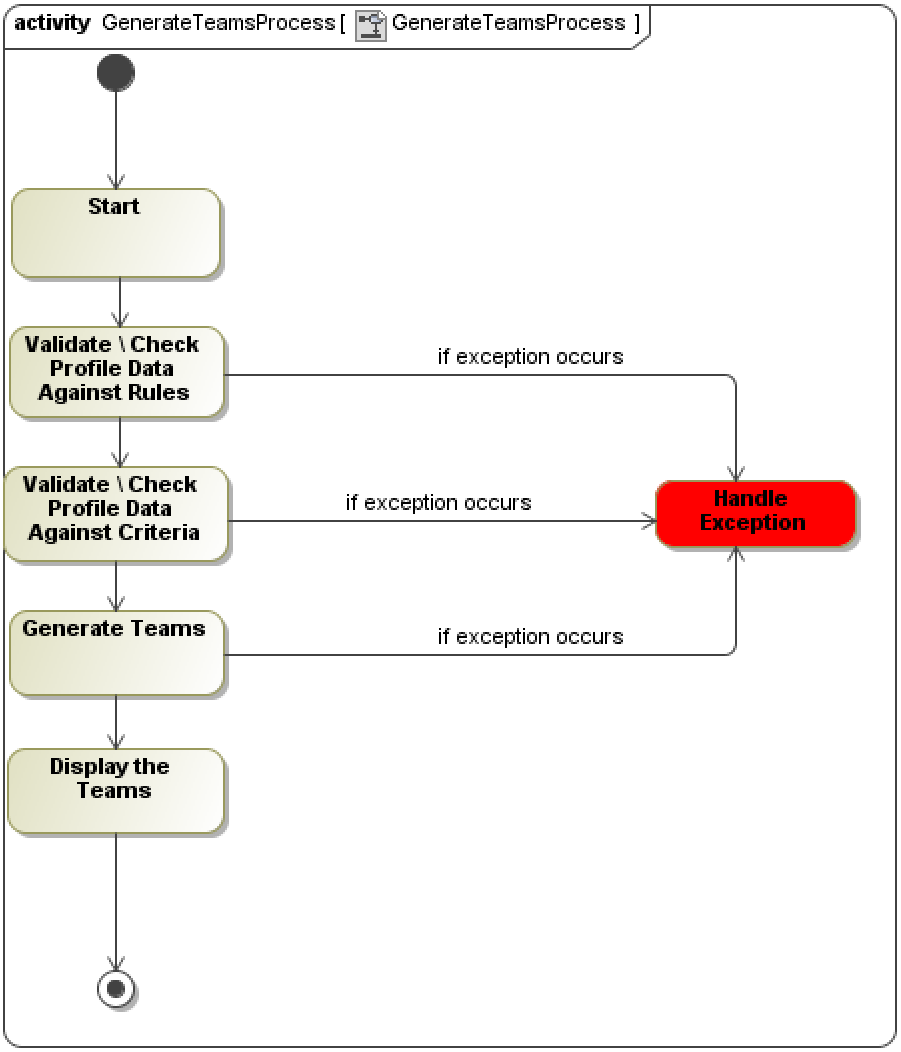

activity GeneratelTeamsProcess [GenerateTeamsProcess]
Start
Validate \ Check Profile Data Against Rules
if exception occurs
Validate \ Check Profile Data Against Criteria
if exception occurs
Handle Exception
Generate Teams
if exception occurs
Display the Teams

Figure 1: Generate suggest teams

4.13.2 Stimulus/Response Sequences

Scenario 1

Stimulus:

User wants to generate and view the suggested teams generated by the system after configuring the criteria and rules.

Reponse:

The system should generate the teams and display the results.

4.13.3 Functional Requirements

REQ39

Validate\Check the information using the list of candidate students against the rules and criteria defined for the project.

Error conditions:

- Validation failed, because it did not meet the default rules or criteria specified for the project.
  - (e.g. The list of students has already been allocated once in every team and violates the default rule)

REQ40

Generate the list of teams that fit the criteria and rules using an algorithm.

Error conditions:

- Cannot generate the teams
  - Display an appropriate message to the user indicating the cause of the error and why the generation failed.

REQ41

Display the suggested teams in a visually or readable format.

4.13.4 User Interface

This is a back-end requirement, so it does not require an interface.

4.14 Manual manipulation of teams

4.14.1 Description and Priority

The system should allow a lecturer to manually manipulate teams to cater for deregistrations, new registrations and teacher preferences. Figure 2 below depicts this requirement. Use cases in red depicts exceptions that should be handled by the system's exception handling. Use cases in blue depicts exceptions that should be handled by the system's feedback handling. See the non-functional requirements for exception and feedback handling for more information. This requirement has a high-priority as the team generation output could lose it's usability should this not be possible.

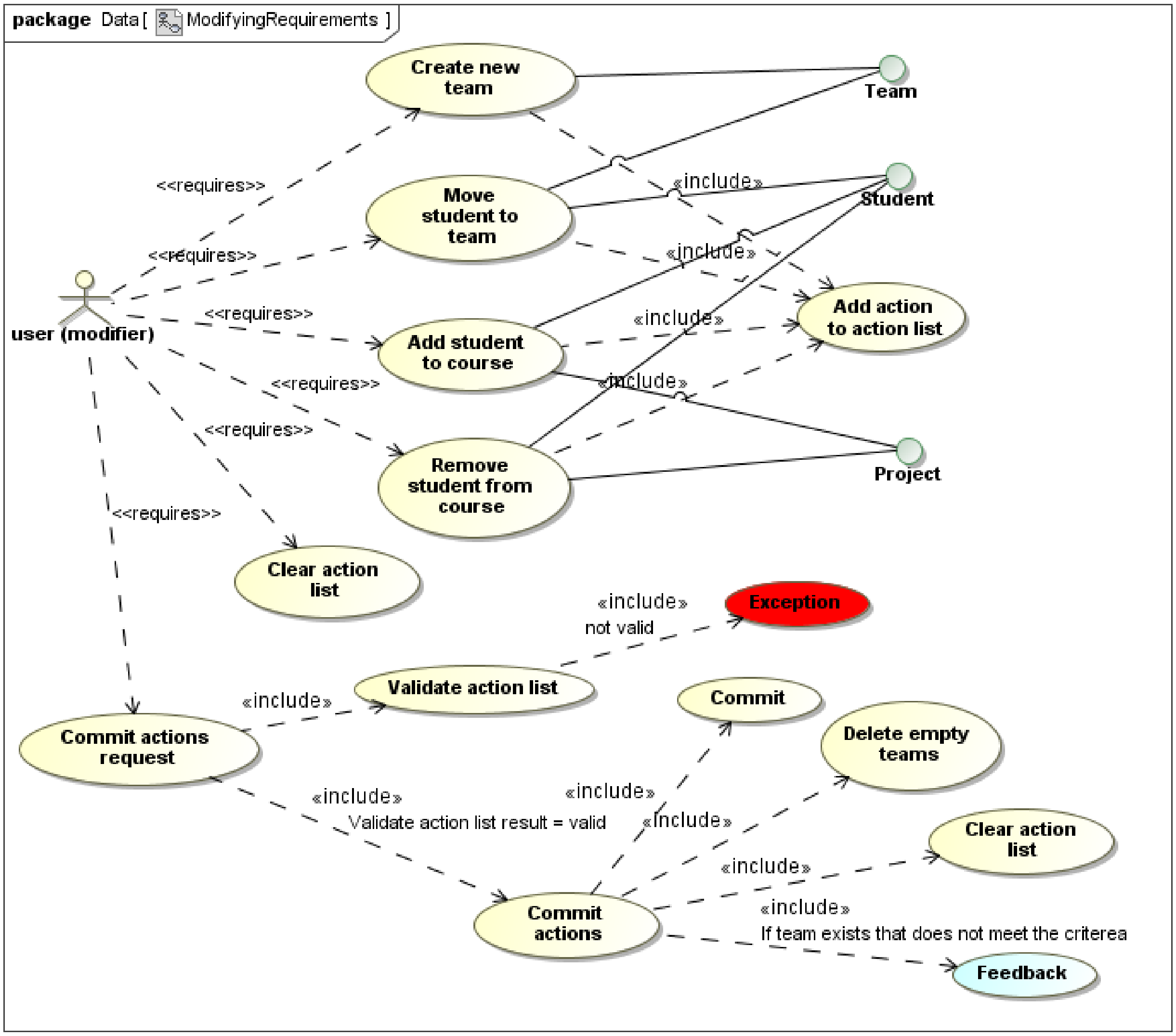


Figure 2: Manual Manipulation

4.14.2 Stimulus/Response Sequences

Scenario 1

Stimulus:

The teams was already generated and afterwards a new student registered for the course. The lecturer now wants to add the student to the course.

Reponse:

The lecturer, through the the 'add student to course' option, adds the student to the course.

Scenario 2

Stimulus:

The teams was already generated and afterwards a student deregistered for the course. The lecturer now wants to remove the student from the course and his team.

Reponse:

The lecturer, through the 'remove student from course' option, removes the student from the course. This action automatically removes the student from his team as well as.

Scenario 3

Stimulus:

The teams was already generated and the lecturer wish to move a student to a different team.

Reponse:

The lecturer, through the 'move student to team' option, moves the student from one team, to another existing team.

Scenario 4

Stimulus:

The teams was already generated and afterwards a new student registered for the course. The lecturer wish to move this student to a team.

Reponse:

The lecturer, through the 'move student to team' option, moves the student to a team.

Scenario 5

Stimulus:

The teams was already generated and afterwards a couple of new student registered for the course. The lecturer wish to create a new team to add the students to.

Reponse:

The lecturer, through the ‘create new team’ option, creates a new team.

Scenario 6

Stimulus:

The lecturer wish to move all of the students in a team to different teams. However, the minimum team size rule will be violated during this process.

Reponse:

The system builds a list of requested manipulation actions waiting to be committed by the user. A commit on these actions will cause the actions to be validated before allowing the actions to be committed.

Scenario 7

Stimulus:

The lecturer wish to commit the actions he added to the manipulate action list.

Reponse:

The system validates the actions, should no rules be violated, then the system performs the actions. Otherwise, an exception is thrown informing the user of the rule that was violated.

Scenario 8

Stimulus:

The lecturer removed all the members from a team, an unwanted empty team is left behind. This will violate the minimum team size rule.

Reponse:

The system, after a commit, deletes all the empty teams in the project at hand.

Scenario 9

Stimulus:

The lecturer wishes to start afresh with the manipulation action list, discarding the actions currently in the list.

Reponse:

The lecturer, through the system, chooses the clear manipulation action list option which results in the system discarding all the manipulation actions in the manipulation action list.

Scenario 10

Stimulus:

The lecturer commits the actions in the manipulation action list and wishes to start afresh with new manipulation actions.

Reponse:

After a commit, the system clears manipulation action list.

4.14.3 Functional Requirements

REQ42

The lecturer's action requests to manipulate the teams should be added to a manipulation action list.

Error conditions:

- The current user does not have enough privileges\permissions to add a manipulation action. An exception is thrown.

REQ43

The lecturer should be able to clear the manipulation action list on request.

Error conditions:

- The current user does not have enough privileges\permissions to clear the manipulation action list (which, in this case should not be able to have any actions in it). An exception is thrown.

REQ44

The lecturer should be able to add students to the course after teams has been generated. This action is added to the manipulation action list.

REQ45

The lecturer should be able to remove students from the course after teams has been generated. This action is added to the manipulation action list.

REQ46

The system should be able to commit the users requested manipulation actions.

Error conditions:

- The current user does not have enough privileges\permissions to perform this action. An exception is thrown.
- If validation fails for the manipulated actions, an exception is thrown. The commit is not allowed to be performed.

REQ47

After a successful commit, the system should provide feedback to the user should a team not match the requested criteria.

REQ48

After a successful commit, the system should delete all the empty teams in the project.

REQ49

After a successful commit, the system should clear the manipulate action list.

4.14.4 User Interface

An interface should be available that displays the current teams as selected by the system according to the criteria sets. The data should be displayed as an editable spreadsheet that the Administrator can easily modify. A “Save” button should then save the modified team selections out to the database.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

<Requirement 4.6.> Reports should be generated in a reasonable amount of time (less than 1 minute per report).

<Requirement number for criteria>: The team generation algorithm should generate combination of teams in a reasonable amount of time. The list of criteria should also be displayed in a reasonable amount of time.

5.2 Safety Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety

issues that affect the product’s design or use. Define any safety certifications that must be satisfied.>

5.3 Security Requirements

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

<Requirement 4.7 and 4.8> Only authorized users should be able to view and export reports. Users should be authenticated using usernames and passwords.

<Requirement number for criteria>: Only authorised users should be able to create/select criteria to generate teams.

SEC-1: Users shall be required to log on to STORM for all operations.
SEC-2: STORM shall permit Users to modify only projects that they created.
SEC-3: STORM shall encrypt user’s password using SHA1 salted with username.

5.4 Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

5.4.1 Exception Handling

5.4.1.1 Description and Priority

The system should handle exceptions thrown by its components in a meaningful way. Unhandled exceptions should be avoided to avoid causing system failure. This requirement has a high-priority.

5.4.1.2 Stimulus/Response Sequences

Scenario 1
Stimulus:

A component caters for an invalid action performed by throwing an exception.

Reponse:

The system catch this exception and provide the appropriate feedback as defined by the exception.

Scenario 2

Stimulus:

A component, or the system throws an unknown exception.

Reponse:

The system catch this exception, logs it to the user audibility log (this requirement is defined in section 5.4.3) as well as to inform the user that a fatal error has occured. If the hosting OS has an event log, then the system should log the exceptions details to this event log. If possible, continue without crashing the software.

5.4.1.3 Non-Functional Requirements

REQ1

The system should be able to handle exceptions catered for and thrown by the systems components. The feedback as defined by the exception should be given to the user.

REQ2

The system should be able to handle unknown exception thrown by the system. Unknown exceptions should be logged to the hosting OS's event log as well to the user audibility log. If possible, continue without crashing the software.

5.4.2 Feedback Handling

5.4.2.1 Description and Priority

Components in the system should be able to provide feedback to the user through a feedback service. Having it as a non-functional requirement enabling every component in the system to provide feedback as well as allowing easy expansion to the feedback mechanism. E.g. Integrated with user audibility, help to add traceability as to where the error occurred which could be displayed to the user and functionality such as feedback type could be added to give the user more clarity on the importance, e.g. Type equals Warning or Information. Should the System be decoupled from the UI, just displaying a message box from the component's location will not work. Figure 3 depicts this requirement. This requirement has a high-priority.

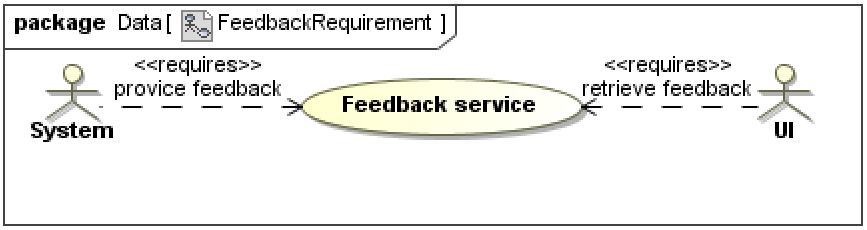


Figure 3 System feedback requirement.

5.4.2.2 Stimulus/Response Sequences

Scenario 1

Stimulus:

A component provides the user feedback regarding a certain action.

Reponse:

The system component sends its the feedback to the feedback service, the UI retrieves the feedback from the feedback service and displays it to the user.

5.4.2.3 Non-Functional Requirements

REQ3

The system components should be able to provide feedback to the user through a feedback service which the UI access to display the feedback to the user. See figure X for a depiction of this requirement.

5.4.3 System action audibility

5.4.3.1 Description and Priority

The system should provide the audibility on its actions together with the user that requested the actions so that it can be used to trace user actions as well as the path that led to an issue. This audibility should be persisted. This requirement has a low-priority.

5.4.3.2 Stimulus/Response Sequences

Scenario 1

Stimulus:

A lecturer wants to determine who deleted student X out of the team.

Reponse:

The lecturer should check the system audibility log which tracks the system’s actions.

5.4.3.3 Non-Functional Requirements

REQ4

User actions should be logged together with the user who initiated the actions to an log accessible by the administrators.

<Requirement 4.6, 4.7 and 4.8>

- Adaptability**:** the reporting system should be written in a way that makes it easy to add new types of reports and add the functionality to export reports to new formats.
- Usability: the reporting functionality must be easy and practical for the user to learn and use.

<Requirement 4.2, 4.3, 4.4 and 4.5>:

- Usability: The adding/removing/editing of criteria should be easy to learn and easy to use. It should be easy to add priorities to selected criteria. The team generation algorithm should be easy to use.
- Scalability: The system should handle an increasing number of scalability effectively. A huge number of criteria should not impact the performance of the system.
- Availability: The criteria that the users created (and not deleted yet) should always be available. The team generation algorithm should never be down.
- Extensibility: The system should easily allow for adding of new criteria.
- Maintainability: The system should be maintainable by any authorised person (not necessarily only the developers of STORM) with minimum knowledge.
- Reliability: The team generation algorithm should be stable, reliable and sound.
- Reusability: The criteria and the team generation algorithm should be re-used as much as possible.

5.5 Business Rules

<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>

6. Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>

6.1 Data Requirements

6.1.1 Logical Data Model

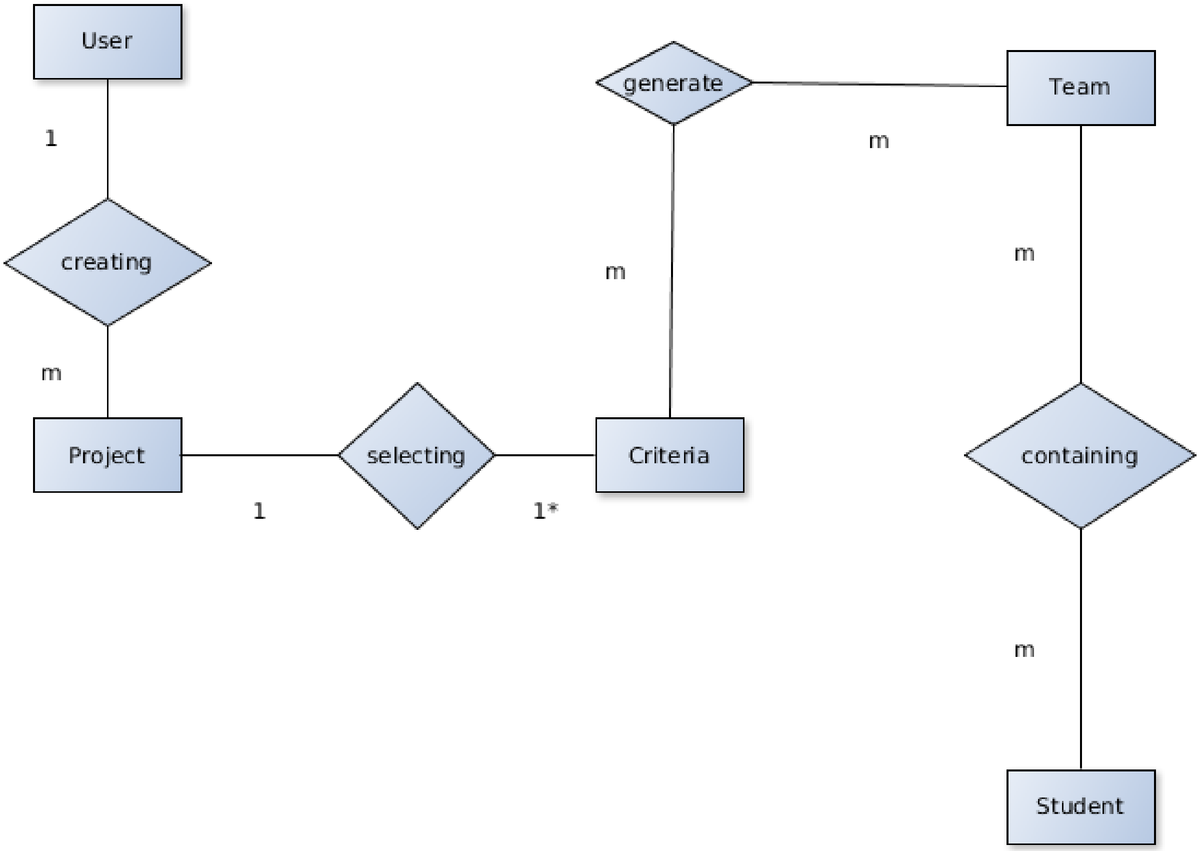


Figure 4: Partial data model

6.1.2 Data Dictionary

Data element	Description	Composition or data type	Length	Values
student ID	university issued student number	integer	8	7 digits for IDs starting with 0
student	an individual being grouped	student name + student surname + student ID + student contact number + student email address		
student email	email address of the student being grouped	alphanumeric	50	

student name	name of the student being grouped	alphabetic	30	
student surname	surname of the student being grouped	alphabetic	30	
student contact number	contact number of the student being grouped	integer	11	27 + (number without the leading 0)
team	consists students that are grouped together	team ID + round		
team ID	identifies the team at a specific round	integer		Initial value is 1 in each round
round	an iteration pass in the grouping process or project	integer		default=1 maximum= number of projects or rounds created for that year
criteria	a factor used for grouping students	criteria weight + criteria description		
criteria weight	factor in the whole project	decimal	2,5	
team override	information about manual team placement	team ID + student ID + round		
operation	information about an operation performed against STORM	user ID + operation string + operation date		

user ID	STORM issued ID for user carrying out operations in the system	integer	11	autogenerated
operation string	details data modifying operation	string	512	
operation date	date in which the operation has taken place	datetime		default=now

6.1.3 Data Integrity, Retention, and Disposal

DI-1: STORM shall retain individual Student personal information for a year following module's registration date.
DI-2: STORM shall purge Student information from groups after module completion.

6.2 Data Persistence

The database for the STORM project will use a server side implementation that holds information on of certain stakeholders. Those stakeholders and information will include:

- Users = lecturers
- Users = Administrators
- The criteria that will be used to measure by
- Transactions = History, teams, imported lists

6.2.1 Description

The system shall use a MySQL database that will ensure quick access, updating and viewing of information. The database will also help with security of the data though database passwords and usernames ensuring that only authorised personnel can directly access the database.

6.2.2 Input

The administrator and the lecturers will be responsible for input into the database. These will include Excel, CSV reports generated from each round of the "rocking the boat" iteration. Student details shall be inserted and updated by the administrator and they should make sure that the data is always correct and not corrupted. The following will be stored:

- Student details
- History of teams
- Project details
- Criteria to establish new teams

6.2.3 Display

The GUI will display information from the database. Lecturers, administrators will be able to retrieve information from the database which will also include being able to display information in a report manner.

6.2.4 System output

The database shall record a number of things, including:

- Log files on the server
- System events including errors
- Data access times and backup times

The system administrator shall have access to all of the output generated by the database and the lecturers will be able to export reports in either excel, CSV format or PDF format

6.2.5 Other

The database should be able to work on a Linux, UNIX or Windows environment. There needs to be at least 10GB of free hard disk space to begin with supported by at least 1GB of RAM and a properly configured MySQL.

6.2.6 Constraints

The amount of traffic the database can handle will depend on the speed of the server but for this project, traffic should not be such a factor as we do not expect that many users will be accessing the database at the same time and therefore, traffic is not so much an issue.
The amount of space needed, not allocated if we will be using the University Of Pretoria’s server.

6.2.7 Data Handling

All data is handled through a database as described above