Requirements Document for Sojabohne - Application

Paul Fink Jens Henninger Florian Jennewein
Daniel Maier

November 12, 2015

Glossary

administrator (Admin) Human individual, who monitors the system and controls user activities. 19, 30

anonymous user User, who isn't registered in the system. 22, 30

Apache Tomcat Open source servlet container and http web server. 31

arff Attribute-Relation File Format - ASCII format used for machine learning. 18, 31, 35

Browser Web-Browser - Software application for presenting/traversing informations in web; common browsers: Internet Explorer(IE), Firefox, Chrome, Safari, Opera. 17, 27

csv Comma-separated values(csv): file format, which stores tabular data in plain text. 18, 31, 35

database organized collection of data. 19

EULA An end-user license agreement (EULA) or software license agreement is the contract between the licensor and purchaser, establishing the purchaser's right to use the software.. 33

framework A software framework is an abstraction in which software providing generic functionality can be selectively changed by additional user-written code, thus providing application-specific software.. 13

Hypertext Transfer Protocoll Secure Hypertext Transfer Protocol Secure (HTTPS) is a communications protocol for secure communication over a computer network, with especially wide deployment on the Internet. Technically, it is not a protocol in and of itself; rather, it is the result of simply layering the Hypertext Transfer Protocol (HTTP) on top of the SSL/TLS protocol, thus adding the security capabilities of SSL/TLS to standard HTTP communications. (source: en.wikipedia.org: 11.11.2013). 23

IEEE IEEE stands for the "Institute of Electrical and Electronics Engineers".. 33

J48 J48 is an open source Java implementation of the C4.5 algorithm in the weka data mining tool.. 21

jar Protocol and RDF Query Language(SPARQL): RDF query language for databases, able to retrieve and manipulate data stored in RDF format. 19, 31, 36

jar Java Archive (JAR) is a package file format typically used to aggregate many Java class files and associated metadata and resources (text, images, etc.) into one file to distribute application software or libraries on the Java platform.. 20

Java concurrent, object-oriented programming language. 23, 32

Linux Linux is a Unix-like computer operating system (OS) assembled under the model of free and open-source software development and distribution.. 18

Linux A computer operating system (OS). 18

Linux Random-access memory (RAM) is a form of computer data storage. 18

Linux A computer operating system (OS). 18

model . 17, 19, 35, 38

Random Forest Random forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random forests correct for decision trees' habit of overfitting to their training set.. 21

RDF Resource Description Framework. 19, 31

REST Representational State Transfer. 19, 32

SMO A Data-Mining algorithm. 21

test-data . 38

User a human individual, whoвЃЄs using the system. 17

web-app A web application (web-app) is a client-server software application in which the client (or user interface) runs in a web browser.. 13, 18

WEKA . 13, 18, 30

Contents

Part I

Preface

1.1 Readership

This Requirement Document is designed for the Data-Science StartUp.

1.2 Document Version

Version Sojabohne 2.1,
Date: November 12, 2015

Part II

Introduction

2.1 Purpose of the system

Until now there does not exists a web-application web-app for the data-mining programm WEKA.

2.2 Description of the Program

The goal is to design a framework for data analysis. The framework should provide basic data mining and machine learning algorithms and should provide possibilities to analyse arbitrary data sets.

Part III

User Requirements

3.1 UR0100

UR0100.0

Statement *A web-application for Users, which is able to apply algorithms from data-mining and machine-learning to the user's data and gives the solution, is needed.*

Priority *A*

UR0101.0

Statement *Not only the solution itself, but also the solution process, the processing length and the algorithms name and source as an evidence of his correctness shall be given to a user*

Priority *A*

UR0102.0

Statement *The application shall have options to choose how the solution is presented (e.g. visualization in a tree or graph)*

Priority *A*

UR0103.0

Statement *The solution shall be downloadable*

Priority *A*

UR0104.0

Statement *There shall be options to choose, which algorithm is want to be used; the application shall make a pre-selection of algorithms based on the uploaded data*

Priority *A*

UR0105.0

Statement *The user's data, all created models and solutions shall be stored in downloadable packages with access management*

Priority *A*

UR0106.0

Statement *Usable with every common Browser*

Priority *A*

UR0107.0

Statement *Data-Input via .arff and .csv files*

Priority *A*

UR0108.0

Statement *Only the web-app is needed*

Priority *A*

UR0109.0

Statement *Should be readable via smartphone / tablet*

Priority *B*

UR0110.0

Statement *The whole application shall be build modular and able to interact with other data-mining/machine learning applications*

Priority *A*

UR0111.0

Statement *The application shall work with the WEKA libraries*

Priority *A*

UR0112.0

Statement *The application shall work on Linux*

Priority *A*

UR0113.0

Statement *The application shall work on Linux and Linux*

Priority *B*

UR0114.0

Statement *The application shall work on every device with min. 32gb Linux and eight-core-processor*

Priority *A*

UR0115.0

Statement *The application shall be able to test models. Models can be uploaded or taken from the database*

Priority *A*

UR0116.0

Statement *The user shall be able to run algorithms simultaneous, if an administrator allows*

Priority *A*

3.2 UR0200

UR0200.0

Statement *All data should be stored in an RDF-database*

Priority *A*

UR0201.0

Statement *The user's data shall be reusable*

Priority *A*

UR0202.0

Statement *Access via jar*

Priority *A*

UR0203.0

Statement *The database shall be updatable and always kept up to date*

Priority *A*

3.3 UR0300

UR0300.0

Statement *Access to the framework should be provided via a REST Interface*

Priority *A*

3.4 UR0400

UR0400.0

Statement *The application shall be as user friendly as possible, with an intuitive interface, for users with previous knowledge in data-mining and machine learning*

Priority *A*

UR0401.0

Statement *The application is accessible by the general public*

Priority *A*

3.5 UR0500 - Algorithms

UR0501.0

Statement *It should be possible to add algorithms to the application in a .jar file*

Priority *B*

UR0501.1

Statement *The admin has to approve all uploaded algorithms before they are released to the system*

Priority *B*

UR0502.0

Statement *The implemented algorithms should be unalterable.*

Priority *B*

UR0503.0

Statement *No time limit for registered users, algorithms can run forever; possible timeout for anonymous users.*

Priority *A*

UR0504.0

Statement *The user shall have the possibility to interrupt the running algorithm*

Priority *A*

UR0505.0

Statement *The standard and pre-installed algorithms are the following classifying algorithms: SMO, J48, Random Forest*

Priority *A*

UR0506.0

Statement *The application should support all data-mining algorithms (e.g. clustering-algorithms)*

Priority *B*

UR0507.0

Statement *If the number of requests exceed the constraint given by an admin then new requests are entered into a request-queue (See also UR0601.4)*

Priority *B*

3.6 UR0600

UR0600.0

Statement *The application should have an user administration*
There shall be the following three user views:

Priority *A*

3.6.1 UR0601.0 Registered - User view

UR0601.1

Statement *Registered user data is private, meaning it cannot be accessed by others.*

Priority *A*

UR0601.2

Statement *Registered user should be able to share informations with other users.*

Priority *A*

UR0601.3

Statement *It shall be possible to create user groups.*

Priority *A*

UR0601.4

Statement *Registered User are prioritized in the request-queue*

Priority *A*

3.6.2 UR0602.0 Anonymous - User view

UR0602.1

Statement *The anonymous user may use the same features as the registered users.*

Priority *A*

UR0602.2

BUT they have the following constraints:

UR0602.2.1 Time-constraints:

Priority *A*

UR0602.2.1.1

Statement *After 30 days their saved data is deleted*

Priority *A*

UR0602.2.1.2

Statement *The admin can set a runtime-timeout for algorithm*

Priority *A*

UR0602.2.2 Space-constraints:

Priority *A*

UR0602.2.2.1

Statement *The admin can set a maximum upload filesize*

Priority *A*

3.6.3 UR0603.0 Administrator view

UR0603.1

Statement *There shall be a group of administrators for the organisation of the users with the possibility to ban users from the web-application*

Priority *A*

UR0603.2

Statement *Set maximum number of simultaneously running algorithms*

Priority *A*

UR0603.3

Statement *Set the number of algorithms, a user can run simultaneous*

Priority *A*

3.7 UR0700

UR0700

Statement *Programming language shall be Java 8*

Priority *A*

3.8 UR0800

UR0800

Statement *The interface and system language shall be english*

Priority *A*

3.9 UR0900

UR0900

Statement *The application shall use Hypertext Transfer Protocoll Secure*

Priority *A*

3.10 UR1000

UR1000

Statement *The application shall work on multiple servers, which are able to interact with each other through the application.*

Priority *A*

Part IV

System Requirements

Non-Functional Requirements

4.1 Product Requirements

4.1.1 Usability Requirements

NFR001.0

Statement *The application shall be usable with every common Browser, especially:*
-Internet Explorer 10 or greater
-Firefox 31 or greater
-Safari 8 or greater
-Chrome 40 or greater

Priority *A*

User Requirement *UR0106.0*

NFR002.0

Statement *The application shall be usable on Opera 27 or greater*

Priority *B*

User Requirement *UR0106.0*

NFR003.0

Statement *It shall be possible to learn the main functionalities of the application within 10 hours for a B.Sc. Informatik.*

Priority *A*

User Requirement *UR0400.0*

NFR004.0

Statement *The administration of the application shall be easy to learn, which means that a person with deeper knowledge in system administration can learn it in less than 50 hours.*

Priority *A*

User Requirement *UR0400.0*

NFR005.0

Statement *There shall be a server application of the system*

Priority *A*

User Requirement *UR0100.0*

NFR006.0

Statement *The server shall work with Linux Ubuntu 14.1 or higher.*

Priority *A*

User Requirement *UR0112.0*

NFR007.0

Statement *The application shall work with Windows 8 or higher and Mac OS X 10.9 or higher*

Priority *B*

User Requirement *UR0113.0*

4.1.2 Efficiency Requirements

Performance Requirements

NFR008.0

Statement *The running-time of an algorithm can take as long as it takes for registered user; and less than the limit set by the administrators for anonymous user.*

Priority *A*

User Requirement *UR0602.2.1.2*

Space Requirements

NFR009.0

Statement *The uploaded data can be as big as it is for registered user and smaller than the limit set by the administrators for anonymous user*

Priority *A*

User Requirement *UR0602.2.2.1*

NFR010.0

Statement *The uploaded data of anonymous user will be deleted from the database after 30 days*

Priority *A*

User Requirement *UR0602.2.1.1*

4.1.3 Dependability Requirements

NFR011.0

Statement *A server shall not stop working for more than 4 hours at a stretch*

Priority *A*

NFR012.0

Statement *The correctness and traceability of the computed solutions is only given in the limits of science and depends on the correctness of the user's data*

Priority *A*

User Requirement *UR0100.0*

4.1.4 Security Requirements

NFR013.0

Statement *The system shall be protected against the common forms of vandalism*

Priority *A*

NFR014.0

Statement *The responsibility for the protection of every kind of malpractice lies with the administrators*

Priority *B*

User Requirement *UR0601.3*

4.2 Organisational Requirements

4.2.1 Environmental Requirements

NFR015.0

Statement *The computed solutions shall have qualified quotability(and traceability)*

Priority *A*

NFR016.0

Statement *Each user belongs to one of the following user groups:*
-administrator
-registered user
-anonymous user

Priority *A*

User Requirement *UR0600.0*

NFR017.0

Statement *The web application shall be accessible by everyone, which means the use of the application arenвΓ́Єt subject of conditions(e.g. to be enrolled in a college), although basic knowledge of Data Mining/ Machine Learning is recommend*

Priority *A*

User Requirement *UR0401.1*

NFR018.0

Statement *The application shall work with the WEKA 3.6 libraries*

Priority *A*

User Requirement *UR0111.0*

NFR019.0

Statement *The application and system language shall be american english*

Priority *A*

User Requirement *UR0800.0*

NFR020.0

Statement *The application shall work on multiple servers at the same time. Every server shall be able to interact with another (e.g. access to the other database)*

Priority *A*

User Requirement *UR1000.0*

4.2.2 Operational Requirements

NFR021.0

Statement *The application provides .arff, .csv and serialized Java objects format for the uploaded data (from user)*

Priority *A*

User Requirement *UR0107.0*

NFR022.0

Statement *Apache Tomcat v.8.0.15 will be used as the web server*

Priority *A*

NFR023.0

Statement *A server can be every device with 32 GB RAM or more*

Priority *A*

User Requirement *UR0114.0*

NFR024.0

Statement *For storing user data, a RDF-database is used, with access via jar*

Priority *A*

User Requirement *UR0200.0*

4.2.3 Development Requirements

NFR025.0

Statement *An already existing database shall be portable to the system and shall be updatable*

Priority *A*

User Requirement *UR0203.0*

NFR026.0

Statement *For a maximum of modifiability/updatability the system shall have a modular construction*

Priority *A*

User Requirement *UR0110.0*

NFR027.0

Statement *The software shall be programmed in Java*

Priority *A*

User Requirement *UR0700.0*

NFR028.0

Statement *The framework shall be provided with a REST-Interface*

Priority *A*

User Requirement *UR0300.0*

4.3 External Requirements

4.3.1 Regulatory Requirements

NFR029.0

Statement *1.3.1.1 The whole application and itвЃЄs developement is subject to restrictions of the german IT-law*

Priority *A*

4.3.2 Ethical Requirements

NFR030.0

Statement *The development process stands under IEEE standards of ethical development*

Priority *A*

4.3.3 Legislative Requirements

Accounting Requirements

NFR031.0

Statement *User license corresponding to the legal right (EULA)*

Priority *A*

NFR032.0

Statement *Disclaimer in case of wrong information*

Priority *A*

Safety / Security Requirements

Functional Requirements

5.1 The application uses an RDF-Database to store all data

FR001

Statement *Data that is stored:*
-All uploaded Files (test and trainings data (.arff, .csv))
-All options chosen by the user (algorithm, input parameters, output format)
-Output of the algorithm (model)
-Package glsID
-Starting time of the algorithm
-Ending time of the algorithm
-Output from running test on the model
-User Id of creator
-Users with reading rights
-Users with writing rights
-License
-Description

Priority *A*

User Requirement *UR0105.0*

FR002

Statement *All this data makes up a data package more specifically a data package may consists of:*
-Package id (but only needs a package id to exist)
-An unlimited number of test files
-One trainings data file
-The chosen algorithm id
-The chosen parameters
-The output (model)
-The starting time of the algorithm

-The ending time of the algorithm
-The user Id of the creator of this package
-Unlimited number of user ids with reading rights
-Unlimited number of user ids with writing rights
-The license
-The Description

Priority *A*

User Requirement *UR0105.0*

FR002

Statement *All this data makes up a data package more specifically a data package may consists of:*
-Package id (but only needs a package id to exist)
-An unlimited number of test files
-One trainings data file
-The chosen algorithm id
-The chosen parameters
-The output (model)
-The starting time of the algorithm
-The ending time of the algorithm
-The user Id of the creator of this package
-Unlimited number of user ids with reading rights
-Unlimited number of user ids with writing rights
-The license
-The Description

Priority *A*

FR003

Statement *Database may be accessed via jar*
Users can only access data to which they have reading rights
Users may only change data to which they have writing rights

Priority *A*

User Requirement *UR0202.0*

FR004

Statement *All data is saved and updated into the database as soon as it appears/changes*

Priority *A*

5.2 The application has a user administration. There are three types of users:

5.2.1 Registered Users:

User Requirement *UR0601.0*

FR005

Statement *People can registered themselves to the application. Therefore they need three things:*
-Username: Usernames can only contain letters (a-z), numbers (0-9), dashes (), underscores (_), apostrophes ()́, and periods ()̇ (and must be unique?).
-Password: Passwords can contain any combination of ASCII characters and must contain a minimum of 8 characters
-Email-address: must be valid and unique.

Priority *A*

FR006

Statement *A unique user id is assigned to each new registered user.*

Prioriy *A*

FR007

Statement *A unique user id is assigned to each new registered user.*

Prioriy *A*

FR008

Statement *After registering, users can login by using their email-address and password.*

Prioriy *A*

FR009

Statement *User that are logged in can do the following things:*
1.) Profile settings:
o Change their password (requires confirmation?)
o Change their email-address (requires confirmation?)

2.) Manage data: Overview of all data packages they have access to. List of model Ids and their corresponding description.

o Search bar, which finds a package via package Id
o Selecting a package will reveal all its data
Вљ The user can change permissions (See UR0601.2)
Вљ Run test on the model by using existing test data or uploading new data. (See UR0115.0)
ВљВљ Format of test data: .arff, .csv
Вљ Change the description (needs writing rights)
Вљ Delete uploaded data or whole package (must be creator)
Вљ Download model/test-data (See UR0103.0)

3.) Create new model:
o Upload data (.arff, .csv format) (See UR0107.0)
o Choose split factor of data into training and test data (in %)
o Optionally upload test data (.arff, .csv format)
o Choose an algorithm (See UR0104.0)
o Set algorithms parameters (See UR0104.0)
o Choose an output mode (See UR0102.0)
o Run the algorithm (User can cancel the algorithm at any time) (See UR0504.0)
o Application returns created model and results of test
o Optionally set description
o Optionally change permissions

4.) Upload new algorithms: (See UR0501.0)
o Must be a .jar file which implements the algorithms interface
o Uploaded algorithms must be approved by an administrator before the may be used. (See UR0501.1)
o Registered users only

5.) Upload existing model: (See UR0115.0)
o Must be a serialized java object (See UR0107.0)

Prioriy *A*

FR010

Statement *Registered users have full access to the application. They may upload any number of files and run any number of algorithms.*

Prioriy *A*

User Requirement *UR0503.0*

FR011

Statement *All data packages of registered users are set to private by default, meaning only the creator has reading and writing rights.*

Prioriy *A*

User Requirement *UR0600.0*

FR012

Statement *Registered users can change writing and reading permissions on each data package they created.*

Prioriy *A*

User Requirement *UR0601.2*

FR013

Statement *Righting rights to not entail change permissions of data packages. Only the creator may change permissions to his or her data packages.*

Priority *A*

User Requirement *UR0600.0*

5.2.2 Anonymous Users:

User Requirement *UR0900.0*

FR014

Statement *People may also use the application without registering. The anonymous user is essentially treated as a registered user, who automatically gives everyone reading and writing rights to his or her data packages.*

Priority *A*

User Requirement *UR0602.1*

FR015

Statement *The anonymous user may also have extra constraints given by the administration*

Prioriy *A*

User Requirement *UR0602.2*

5.2.3 Administrator

User Requirement *UR0603.0*

FR016

Statement *Administrators also have to be registered to the application and can then login with their email-address and password.*

Prioriy *A*

FR017

Statement *Administrators automatically have reading and writing rights to all data packages.*

Prioriy *A*

FR018

Statement *Administrators can do everything a registered user can.*

Prioriy *A*

FR019

Statement *Additionally administrators have an extra view with the following options:*

1.) Administrators can set constraints to the anonymous users:

o Time constraints: Administrators can set (See UR0602.2.1)

Bљ Time limits (in minutes) for algorithms (individually ?) (See UR0602.2.1.2)

Bљ An expiration date for data packages (in days), after which data packages are deleted from the database (See UR0602.2.1.1)

o Space constraints: Administrators can set a maximum upload file size (in GB) (See UR0602.2.2)

2.) Administrators can set the maximum number of simultaneously running algorithms for the whole application AND every single user (See UR0603.2 /UR0603.3)

3.) Administrators can ban user from the application. Banned users are deleted from the system. (This entails deleting all data packages created by the banned user) (See UR0603.1)

4.) Administrators have an overview over all work request and can cancel any number of them and change the order of the request in the workflow-queue

5.)Administrators have a view over all request for new algorithms. They may download them to check the code and then approve them.

o Approved algorithms are considered active and can be run by all users

o Administrators can also deactivate algorithms

Prioriy *A*

5.3 Server/Algorithms

FR020

Statement *It is possible to run multiple algorithms at the same time*

Prioriy *A*

User Requirement *UR0507.0*

FR021

Statement *If the number of simultaneously running algorithms reaches the limit given by the (See UR0507.0) administration any new request are entered into a workflow-queue.*
Bљ Registered users are prioritized over anonymous users in the queue. (See UR0601.4)

Prioriy *A*

User Requirement *UR0507.0, UR0601.4*

FR022

Statement *Based on the users trainings data input the choice of algorithms is limited. (e.g.: if input data contains floating point numbers, then only algorithms that can handle those numbers will be available)*

Prioriy *A*

User Requirement *UR0104.0*

Scenarios

6.1 Registered User - Scenario for creating a new model

INITIAL ASSUMPTION: *A registered user wants to create a new model*

NORMAL: *The user enters the homepage and logs in. Next he chooses the tap "create new model". Then he uploads his input data. All uploaded data and chosen options are save to the database as soon as they are available. Next he chooses the split between trainings data and test data in percent (It is not possible to choose 0% for trainings data). Afterwards he may upload any number of additional test data. Next he chooses an algorithm for his input (a preselection has been made by the server based on the input data. Therefore only algorithm that can handle the input may be selected). After that the user may set the algorithms parameters and choose an output option. He can also stick with default options. Next the user can start the algorithm by clicking on the button "run algorithm". Until the algorithm is finished the user has the option the cancel it at any time. Upon completion the output appears and is saved to the database.*

WHAT CAN GO WRONG:

1.) Server not available
2.) Server crashes while the user is uploading data
3.) Server crashes while saving to the database
4.) Server crashes while running the algorithm
5.) The user uploads file with the wrong format - Error

OTHER ACTIVITIES: *none*

SYSTEM STATE ON COMPLETION: *User is logged in. Database contains a new package with all information given by the user and the result(output) of the algorithm.*

6.2 Registered User - Scenario for uploading an already existing model

INITIAL ASSUMPTION: *A registered user wants to upload an already existing model into the database*

NORMAL: *The user enters the homepage and logs in. Next he chooses the tab "upload existing model". Then he uploads his model, which is saved to the database.*

WHAT CAN GO WRONG:
1.) Server not available
2.) Server crashes while the user is uploading data
3.) Server crashes while running the algorithm
4.) The user uploads file with the wrong format - Error

OTHER ACTIVITIES: *none*

SYSTEM STATE ON COMPLETION: *User is logged in. Database contains a new package with the model uploaded by the user.*

6.3 Registered User - Scenario for running tests on a model from the database

INITIAL ASSUMPTION: *a registered user wants to run a test on a model from the database*

NORMAL: *The user enters the homepage and logs in. Next he chooses the tab "browse packages", which brings him to a page listing all existing packages. Then he picks the package with the model he wants to run test on. They are identifiable by a unique package id. There is also a search bar, where he may enter the package id. After finding his package and clicking on it the option "run test on model" appears. The user may now choose to upload own test data (which will be saved to the database) or pick already existing (if any) test data from the package. Afterwards he starts the test via the "start test"-button. Upon completing the test the server return the result.*

WHAT CAN GO WRONG:
1.) Server not available
2.) Server crashes while the user is uploading data
3.) Server crashes while saving to the database
4.) Server crashes while running the test
5.) The user uploads file with the wrong format - Error
6.) The user tries to access a package that he has no reading rights to - Error

OTHER ACTIVITIES: *none*

SYSTEM STATE ON COMPLETION: *User is logged in. Database contains uploaded data of the user (if he uploaded any)*

6.4 Registered User - Scenario for change permissions on a data package

INITIAL ASSUMPTION: *A registered user wants to share a data package with a friend*

NORMAL: *The user enters the homepage and logs in. Next he or she chooses the tab "my packages", which brings him or her to a page that shows all packages he created. Now he or she clicks on the package he wants to share (Packages are identifiable by a unique package id). The option "change permission" is now available. After clicking on it, the user can now give reading and/or writing rights to any registered user by entering their username or email address into the specified field. By entering everyone into the field, he or she makes is package public, meaning that anyone can access the package even unregistered users. After hitting the save button, the new permissions are saved into the database and the package is handled accordingly from then on.*

WHAT CAN GO WRONG:
1.) Server not available
2.) Server crashes while saving to the database
3.) User exits the page without saving

OTHER ACTIVITIES: *none*

SYSTEM STATE ON COMPLETION: *User is logged in. Package permission settings are saved. The package is now available for reading and writing to all added users.*

6.5 Unregistered User - The normal work process

Unregistered User *See the Scenarios for the registered User from above, with the following differences:*
-If the unregistered user tries to upload data, which is bigger than the admin-set limitation the user gets an error message and will be redirected to the previous page
-If the unregistered user tries to start multiple algorithms, with a total number bigger than the admin-set limitation the user gets an error message and will be redirected to the previous page
-If an algorithm of an unregistered user runs longer than the admin-set limitation, he gets a timeout error message and will be redirected to the

previous page
-He has no possibility to create a group and set access rights and his data and results will be deleted after 30 days
-On the main page he will be referred that his uploaded data and results are deleted after 30 days, the missing possibility to create groups with access-right-system, and that there are the admin-set constraints; below a button for registration
-If he clicks on the registration button he will be directed to a registration page, where he has to enter his email, a username (he will be noticed, if a username is already in use) and a password; after confirmation his account is created

6.6 Administrator - Control functions and crowd control

INITIAL ASSUMPTION: *The administrator has to do his job, in this scenario the checking of Algorithms.*

NORMAL: *The administrator enters the homepage and logs into his account. He enters the tap "Uploaded User Algorithms". There he has a list of the new uploaded user created algorithms. He chooses and clicks on one of the list. After a short overview over the source code to verify that thereвЃЃs no obvious junk, he clicks on "Test" to test the algorithm on the data the user uploaded in a secured suitcase. If the administrator would find something destructive or illegal he could click on the username of the user, who uploaded the code and would enter the "User settings" tap (this tap he can access every time when he clicks on "Registered User" on his main page and then on a username of the displayed list; so heвЃЃs able to ban user or to see the user data and activities whenever he wants). There he could click on the "Ban this User" button to ban the user from the site, delete his account and saved data. While the test is running, heвЃЃs able to check other algorithms or to do whatever he has to do. If the test finishes with success the administrator clicks on "Unlock algorithm", so the user gets the result of the algorithm and the possibility to choose it for the next time(so the algorithm is stored in the database). If the algorithm doesnвЃЃt work, the file would be marked as fail and the user who uploaded it would get an error message. The admin can continue working.*

WHAT CAN GO WRONG:
1.) Junk code stays undetected
2.) A user could be banned for no reason
3.) The algorithm could run endless

OTHER ACTIVITIES: *none*

SYSTEM STATE ON COMPLETION: *Upon clicking the return button the admin is redirected to the list of algorithms*

6.7 Administrator - Setting constraints

INITIAL ASSUMPTION: *The administrator has to do his job, in this scenario the setting of constraints.*

NORMAL: *The administrator enters the homepage and logs into his account. Then he clicks on the tap "Main Settings" and on the displayed page on the tap "Constraints". Now he has following choices:*
- Under the point "Runtime constraint", he can enter a time in hours; thatâĂŹs from the moment of confirmation the maximum amount of time, an algorithm may run for anonymous user
- Under the point "Upload constraint", he can enter the space in GB; thatâĂŹs from the moment of confirmation the maximum amount of space of uploaded data may run for an anonymous user
- Under the point "Algorithms per user", he has two textboxes, in which he can enter the allowed number of parallel running algorithms for registered user(in the first box) and anonymous user (in the second box)
- If he clicks on the tap "Running Algorithms", he finds a list of all currently running algorithms, all algorithms in the queue and the current server workload; here he can enter the allowed number of total running algorithms, manipulate the queue order and interrupt running algorithms
After he made his changes, he can continue working.

WHAT CAN GO WRONG:
1.) System overload could be overseen
2.) Settings could be set unuseful

OTHER ACTIVITIES: *none*

SYSTEM STATE ON COMPLETION: *Upon clicking the return button the admin is redirected to the main page.*