

# Changelog

Project: *TRIMM - Case visualizing runner data*

Group number: **42**

Period: 16 - 26 June

Please describe all changes in your project from 16 to 26 June in this changelog document. Export as PDF and add it to a 'documents' folder in your git project.

Keep it short and concise, but be specific about what changed. Feel free to resize the feedback text areas.

## DESIGN

### Project Description (project manual 4.1)

**23/06** - Rewrote *project.pdf* so that now it explains what the project is about and shows the outline

### HTML/CSS Mockup (project manual 4.2)

'unchanged'

### Use Case Diagrams (project manual 4.3)

**17/06** - Fixed use case diagram with added two extension use cases about verifying the email and changing passwords. - **use\_case\_diagram\_fixed.jpg**

### Database Design: Class Diagram and schema.sql (project manual 4.3 and 4.4)

**17/06** - Fixed class diagram with added class Account, merged User and HTTPSessions into one class in the diagram and added emailverify class - **class\_diagram\_fixed.jpg**

21/06- Fixed db schema so that now it shows primary and foreign keys and changed it to sql extension-**schema.sql**

## FUNCTIONAL REQUIREMENTS

### Added or revised functionality

21/6 - can export graphs from **Run1.html** as .png

19/6 - register is changed after you type your info a token is send to your email and you have to copy this token to the field on the register page and then send this to the server

18/6 - Added account functionality. Now the user can access his/her account and change some of the information that can be changed(e.g the user may change the height because he/she has gained up some centimeters)

19/6 - Added changing passwords functionality if the user forget its password to change it in a secure way

19/6 - Removed getStepData method which fetches all the runs on a request and created new method which fetches only one run now because of performance issues. So now it does not fetch all the runs for every request but only the run requested.

22/6- Created a graph where runs can be compared by variables in **comparepage.html**

24/6- Created a pie chart in **Run1.html** to show the surfaces of a run.

25/6- Added a button with window in **Run1.html** near the pie chart to show the shoes information of the run

#### After 26th June to 1st July:

26/6 - implemented url lock in all the pages that the user can see after being logged in. Now the user can't see those pages by just typing them in the url

27/6 - Added dynamic change of the title in the comparepage.html graphs. The change is in comparepage.js

27/6 - added checks in CreateAccount.js so the user can't give a password less than 8 symbols and implemented scoring of the strength of the password shown by colors and words in the design.

27/6 - changed MyRuns.html/js/css so it looks with different design and now it shows the date of every run by hovering.

27/06- Added Delete functionality for the account with a nice warning message- Account.html, AccountJS.js, UserResource.java, UserDao.java

## HTML/CSS/JS Frontend

16/6 – updated the design for the [comparepage.html](#) – made the graphs bigger and only one graph on each section, design for the buttons was fixed, updated the design for the specific run page – the page is now scrollable with an automatic scroll and additional graph is added to it

16/6 – created a graph for comparing variables through the runs in [comparepage.html](#).

16/6 – Changes in the button design and the layout of the [comparepage.html](#) page.

16/6 – Changed design of [Run1.html](#) page and design of graphs

16/6 – added more variables from the database in the second [Run1.html](#) graph

17/6 – the design for the [comparepage.html](#) was made responsive for mobile devices

18/6 – the design for the [comparepage.html](#) was updated – some of the buttons were fixed to appear on mobile devices

18/6 – programmed the functionality of the [comparepage.html](#). The buttons are moving and you can add and remove runs to the both graphs.

18/6 – added new html page [forgetpass.html](#) with design as the [CreateAccount.html-webWorkDesign/forgetpass.html](#)

18/6 – added javascript functionality to the page for popping messages dynamically with javascript in red/Invalid email

green/Please check your email-[forgetpass.html](#)

18/6 – added js functionality for changing the button onclick event to refer to [sendNewPass](#) function

19/6 – added pie chart in [Run1.html](#) for comparing surfaces

19/6 – fixed [comparepage.html](#) by adding what you have selected to the graph made the second graph scalable based on the browser window and fixed bugs that occurred because of project changes.

19/6 – removed an unnecessary link from the menu which purpose was only for testing the second graph.

19/6 – changed the roadmap of the website by changing the links in all the menus and created a new page [mainPage.html](#) which is going to be the first page the user sees after log in.

19/6 – fixed the links in [mainPage.html](#) and also fixed the z-index because of an error on the page

22/6 – changed the design of the [help.html](#). Also created a [helpMain.html](#) which is with different menu when the user has logged in.

22/6 – added logout request to every page after the user has logged in

22/6 – fixed an error in [comparepage.html](#), [removeDataset](#), because of errors in the page. Fixed the remove all button for the bar graph as well.

23/6 – change design of [help.html](#) and [helpMain.html](#)

23/6 – Every single page is responsive on various screen sizes and resolutions.

23/6 – Changes were made on [help.html](#) and [helpMain.html](#) and their readability was drastically improved.

23/6 – Navigation bar added on pages: [login.html](#), [CreateAccount.html](#) and [forgetpass.html](#)

24/6 – Javascript and CSS separated from HTML

24/6 – .js and .css files moved to separate folders (js and css, respectively)

24/6 – added shoe data as a dropdown in [Run1.html](#)

24/06 Added new design of the [loadingPage.html](#) when you log in

25/06 Changed the design for the dropdown in [Run1.html](#)  
25/06 Changed the sort in [comparepage.html](#) so it can sort dynamically

After 26th June to 1st July:

26/6 - created a new function `checkIfLoggedIn()` in all the pages that the user sees after being logged in  
27/6 - changed the design of [MyRuns.html](#)  
27/6 - changed [MyRuns.js](#) to work with the new design and show the date of the run  
27/6 - added update graph to the graph titles in [comparepage.js](#) so it changes the styles dynamically based on the buttons clicked.  
27/6 - added checks for the strength of the password and the length in [CreateAccount.js](#) and [forgetpass.js](#)  
28/6 - changed the photo for mobile devices in [index.html](#)  
01/7 - made the checks for strength to also detect if there is no input and re-check when parts of the password are deleted  
01/7 - fixed the menu sidenav on the [MyRuns](#) page  
01/7 - made the landing page after log in to be [MyRuns.html](#) as requested by Trimm

### Transactions (project manual 5.9)

23/6 -Added transaction isolation levels(serializable) and rollback for the update and insert queries  
23/6 Added transaction lower isolation levels(read committed) for reading run data for performance and rollback for the cases when the transaction is not successful.  
24/6 Added transaction isolation levels and rollback to all the dao classes and especially to those with two queries

After 26th June to 1st July

27/6-Added transaction isolation levels and rollback to the new method `deleteUser` in [UserdDao.java](#)

### JDBC / Prepared Statements (project manual 5.7)

16/6 Added new dao for fetching the acc information- [UserDao.java](#)  
17/6 Added new method for updating existing user with UPDATE query with prepared statements-[UserDao.java/updateUser](#)  
17/6 Added new method getting the total steps of a user with stored procedure TotalSteps [UserdDao.java/getTotalSteps](#)  
17/6 Changed [ShoesDao](#) to work with `sid` as parameter and return the shoe object

accordingly-ShoesDao.java

18/6 Added method deletetoken which deletes a session token -SessionTokenDao.java

19/6 Added method getRunWithoutMeta because of performance issues on the front-end-RunnerDataDao.java

20/6 Added new method getRun which fetches all the data for one run only in order to improve performance -RunnerDataDao.java

21/6 Finished ShoesDao.java which fetches information about the shoe

Prepared statements used in all dao classes

After 26th June to 1st July

27/6-Added new method deleteUser to UserdDao.java

28/6-Added new method RunDate to BodyPackDao.java

## XML/JSON (project manual 5.5)

17/6- Added @PUT and @Consumes (MediaType.APPLICATION\_JSON) above update user method in the UserResource.java

18/6-Added @DELETE annotation above the method logout in the AuthenticationResource.java

All of the resource methods use MediaType.APPLICATION\_JSON and MediaType.APPLICATION\_XML

19/6-Added @XMLRootElement above all model classes in order to use JAXB serialization properly you can check models package

After 26th June to 1st July

27/6- Added @DELETE annotation above method deleteUser in UserResource.java

## Usage of JSP/REST (project manual 5.5)

17/6 Added new REST point GET app /rest/user/{sessiontoken} for requesting the account information and PUT app /rest/user/{sessiontoken} AND user object as json in the body for updating the account-UserResource.java

17/6 Added new REST point POST app /rest/authentication/forgetpassword for requesting change of passwords- AuthenticationResource.java

17/6 Added new REST endpoint GET app/rest/user/{sessiontoken}-UserResource.java

17/6 Added new REST endpoint PUT app/rest/user/{sessiontoken}-UserResource.java

18/6 Added new REST point POST /app/rest/authentication/forgetpassword/{token} for

posting the new password to the server and checking if the email and the token are valid one-**AuthenticationResource.java**

17/6 Added new REST point GET app/rest/data/run which returns a whole a object of a run with its data according to the query params-**RunDataResource.java**

18/06 Changed REST point authentication/reg to two REST endpoint POST app/rest/authentication/reg and POST app/rest/authentication/reg/{verifytoken} so that we can verify the email before registering the user-**AuthenticationResource.java**

After 26th June to 1st July

27/6- Added REST point DELETE app/rest/user/{sessiontoken} for requesting deletion of the user in **UserResource.java**

28/6-Added REST point GET app/data/dates **RunDataResource.java** for requesting the dates of user's runs

## Testing (project manual 5.2 and 6.2)

17/6 Added account junit test to test the fetching of account information and updating account information-**AccountTest.java**

18/6 Added **RunDataTest.java** which test the fetching and calculating of run data

17/6 Fixed Authentication test so that it tests the new encryption method-**AuthFuncTest.java**

21/6 Added new junit test **ShoeTest.java** which tests the shoe functionality

24/6 Added Selenium automated test cases **SeleniumTest.java**

After 26th June to 1st July

27/6- Changed URL of the **SeleniumTest.java** to the deployment server host

## Security (project manual 6.3)

17/6 Implemented better security using **PBKDF2withHmacSha256** with **60K** iterations and random salting with SecureRandom-**EncryptionPassStronger.java** (passwords are first hashed then stored)

18/6 Implemented secure way of sending email for forget password opening **TLS** channel to gmail server - **SendEmail.java**

18/6 Implemented token generation using SecureRandom which is used for verifying the user which requests change of passwords by its email in the database. You cannot abuse it by writing your own email and stealing the account-**SendEmail.java/generateSafeToken**

19/6 Implemented session management control with session tokens generated with Secure Random to uniquely recognize the users- **SessionToken.java**

**20/6** Added library **DOM.purify** to sanitize user input on the front end to prevent Cross-Site scripting and command injections.

**Prepared statements are used all the way in the project to prevent SQL injection attacks**

**USED ONLY CPRNG- Cryptographically secure pseudorandom number generator**

**Security follows strictly the OWASP guidelines**

## OTHER

### Other changes

*Thorough testing of the application with junit, Selenium, black box and system testing.*

*Better security following OWASP guidelines.*

*Increased performance.*

*Used all HTTP verbs available in the course and used many different Jersey annotations.*

*Original custom build css and html interfaces*