Changelog

Project: *KickIn*

Group number: *26*

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)** |
| *‘unchanged’* |

|  |
| --- |
| **HTML/CSS Mockup (project manual 4.2)** |
| * *Using a <table> element from HTML instead of the “cards” to show the list of events.* * *mapEdit.html has a different layout than the mockup.* * *mapView.html has a different layout than the mockup.* |

|  |
| --- |
| **Use Case Diagrams (project manual 4.3)** |
| * *Added ‘date’ attribute on the Event* |

|  |
| --- |
| **Database Design: Class Diagram and schema.sql (project manual 4.3 and 4.4)** |
| *-Added date field in events table.*  *- Added nickname field in users table.*  *-Added the active material list into the database.*  *-Added stored procedures as below:*   1. *select getAllEvents();* 2. *select getAllMaps();* 3. *select getAllResources();* 4. *select getEvent(eid bigint);* 5. *select getMap(mid bigint);* 6. *select getresource(rid bigint);* 7. *select getresource(rid bigint);* 8. *select generateMapReport(mid bigint);* 9. *select generateEventReport(eid migint);* 10. *select \* from terminateIdle();*   *To speed up some queries.*   * *The lastEditedBy, CreatedBy are converted to Foreign Key of userId in Users on both the relation Events, and Map to only allow users who are in the database, previously it was something that the user had to enter and were string, now they are bigint.*   *Added stored procedures for Users, Session table.*   1. *getAllUsers(),* 2. *getUser(uid bigint),* 3. *getAllSessions(),* 4. *getSession(sid bigint)*   *Refactored the JDBC so that there was less repeated code. A method called executeUpdate was added, since most creations have a similar pattern.*  *The method databaseUpdate was changed since again, a lot of repeated pattern.* |
|  |

FUNCTIONAL REQUIREMENTS

|  |
| --- |
| **Added or revised functionality** |
| * *ADDED event - map relationship (many to many)*   + *User can relate an event to different maps, and a map to different events.* * *ADDED export event as .pdf functionality*   + *User can download the report of the event in pdf form in the event page.* * *ADDED export map as .png functionality*   + *User can download the image of the map in png form in the map view page.* * *ADDED full functionality of map editing*   + *User can add objects to the map, scale it, move around etc.* * *ADDED adding/editing/deleting resource functionality* * *ADDED adding/editing/delete map functionality* * *ADDED adding/editing/deleting user functionality (only if admin)* |

|  |
| --- |
| **HTML/CSS/JS Frontend** |
| *Focus on the used software / frameworks, added pages, functionality you are proud of etc.*   * *Added search functionality on MapEdit.js to search for specific items* * *Fixed a few design issues on every page* * *Adopted good JavaScript coding conventions on the MapEdit and requests JavaScript files* * *Arrow button to go to previous pages works* * *Added deleting functionalities to mapView.html* * *Modified nav bar to now have access to events, maps, resources and users.* * *New users and resources can be added.* * *Information on users and resources can now be edited.* * *Information on users and resources can be accessed.* * *The image of a resource can be viewed.* * *Only .png files are allow for resource images.* * *The report of the items required for an event can now be downloaded.* * *The map image can be exported as .png.* |

|  |
| --- |
| **Transactions (project manual 5.9)** |
| *Transactions were added to all methods in the JDBC. All tests and the use is unaffected by additions, while increasing data integrity.* |

|  |
| --- |
| **JDBC / Prepared Statements (project manual 5.7)** |
| *-There were two methods which did not close properly, checkExists() and load() in SessionDao and UsersDao, which has been fixed.*  *- Since date was added in events, on creation, save of events a new parameter date is added which takes in a string and is casted to date.*  *-lastEditedBy, createdBy, were never meant to be typed by the user, but dynamically been resolved, this has been implemented along with the update of them being foriegn keys in the DB to Users table, so only the userId is needed not the name, the link is made automatically. This required changing the type of these fields from String to Long since that is the java equivalent of bigint in postgresql..*  *-For UsersDao and SessionDao the new methods created for the front end were added, no change made to the Prepared Statements themselves.* |

|  |
| --- |
| **XML/JSON (project manual 5.5)** |
| *-In the database the stored procedures as mentioned above return JSON objects, casted to string to be sent directly to the client side. This reduces the time it takes to serialize data for which there is no defined model, allowing for flexibility, fewer throw away classes and speed.* |

|  |
| --- |
| **Usage of JSP/REST (project manual 5.5)** |
| *For the mapEdit.html page, we need massUpdate, massPost, massPut for reducing overhead on the server if they have hundred objects, so the endpoints in mapObjects were changed to accept arrays instead of just the object instance.*  *Think of added or revised REST endpoints or JSPs and the way your front-end communicates with your REST endpoints.*  *Since the frontend, made the advancement, the new functionality of modifying users required the endpoints for users. This follows a very close pattern to other ones, including*   1. *getAll* 2. *get* 3. *save* 4. *delete* 5. *deleteAll*   *They are all secured appropriately, so as no Visitor can get the info on all other users, only editor can view and only the admin can modify.*  *Change to the materials endpoint.*  *The links are now changed without change of functionality but better logic and naming convention.*   1. *resources/materials* 2. *resources/drawing*   *are added in change of the previous ones which were inefficient enough and confusing that I have forgotten what they were.* |

|  |
| --- |
| **Testing (project manual 5.2 and 6.2)** |
| *New Tests added:*   * *The JUnit tests were on these resources:* * *Event* * *GET event* * *PUT event* * *DELETE event* * *Events* * *GET all events* * *POST an event* * *With valid Format* * *With invalid date format* * *DELETE all events* * *Map* * *GET map* * *PUT map* * *DELETE map* * *Maps* * *GET all maps* * *POST map* * *DELETE all maps* * *EventMap* * *POST eventmap* * *DELETE eventmap* * *DELETE all relations for the event (delete all eventmap for an event)* * *DELETE all eventmap* * *Resources* * *GET resource* * *DELETE resource* * *GET all resources* * *DELETE all resource* * *GET all materials* * *PUT material* * *DELETE material* * *GET all drawings* * *PUT drawing* * *DELETE drawing* * *User* * *GET user* * *PUT user* * *DELETE user* * *Users* * *GET all users* * *POST users* * *DELETE all users* * *MapObjects* * *GET all mapobjects for map* * *DELETE all objects on map* * *GET map report* * *Add new objects to map* * *DELETE objects from map* * *PUT objects to map* * *DELETE all objects from map* * *AuthenticationEndpoint (Login and Logout)* * *POST login details to login.* * *DELETE to logout.* * *Exception Testing* * *Testing multiple Exceptions in JUnit test like NOT\_FOUND errors, since those were not explicitly tested.* |
|  |

|  |
| --- |
| **Security (project manual 6.3)** |
| *-The process has been simplified, and made more efficient in terms of processing (we were looking at the cookies, decoding them and looking up the user every time (for authentication, authorization, and then the method which was invoked), but now it done once per request..*  *-On the front end a simple XXS input sanitization was added.*  *-Now storing hash passwords instead of plain text. With Argon.* |

OTHER

|  |
| --- |
| **Other changes** |
|  |