Project Management

A9: Main Accesses to the database and transactions

1. Main Accesses

Main accesses to the database.

1.1 M01 : Authentication and Profile Page

SQL101	Check if User exists and if password is correct
Web Resource	R103
`SELECT * FROM "user" WHERE username = $username AND password = $password;`	

SQL102	Creates new User
Web Resource	R106
`INSERT INTO "user" (name, username, email, image, password) VALUES ($name, $username, $email, $image, $password);`	

SQL103	Updates User information
Web Resource	R109

`UPDATE "user"`
`SET name = $name, username = $username, email = $email,`
`password = $password, image = $image`
`WHERE id = $user_id;`	

SQL104	Creates new Project
Web Resource	R110

```
INSERT INTO project (name, description, isPublic)
VALUES ($name, $description, $isPublic);
```

SQL105	Accepts invite to join the Project
Web Resource	R111

```
INSERT INTO project_members (user_id, date, project_id,
isCoordenator)
VALUES ($user_id, $date, $project_id, $isCoordenator);
```

SQL106	Reject Invite to Join Project
Web Resource	R112

```
DELETE FROM invite
    WHERE user_invited_id = $user_invited_id AND project_id =
$project_id;
```

SQL107	Unsign user form Project
Web Resource	R113

```
DELETE FROM project_members
    WHERE user_id = $user_id AND project_id = $project_id;
```

SQL108	Search Projects of a specific User
Web Resource	R114

```
SELECT project.name, project.description,
project_members.iscoordinator
FROM "user", project_members, project
WHERE "user".id = $user_id AND project_members.user_id =
"user".id
AND project_members.project_id = project.id
ORDER BY ts_rank(
 setweight(to_tsvector('english', project.name),’A’) ||
 setweight(to_tsvector('english', project.description),’B’),
 plainto_tsquery('english', $search)) DESC, name
LIMIT 5 OFFSET $n;
```

SQL109	Searches through the system a public project in which its name or description matches the input make by the user
Web Resource	R115

```
SELECT name, description FROM project WHERE
isPublic = TRUE ORDER BY
ts_rank(
 setweight(to_tsvector('english', project.name),’A’) ||
 setweight(to_tsvector('english', project.description),’B’),
 plainto_tsquery('english', $search)) DESC, name LIMIT 10
OFFSET $n;
```

SQL110	Searches Projects of a specific User, by role
Web Resource	R114

```
SELECT project.name, project.description,
project_members.isCoordinator
FROM "user", project_members, project
WHERE "user".id = $user_id AND project_members.user_id =
"user".id
AND project_members.project_id = project.id
AND project_members.isCoordinator = $isCoordinator
LIMIT 5 OFFSET $n;
```

SQL111	List all projects of a specific user, with their info
Web Resource	R107

```
SELECT project.name, project.description,
project_members.isCoordinator
FROM "user", project_members, project
WHERE "user".username = $username AND project_members.user_id =
"user".id
AND project_members.project_id = project.id
LIMIT 5 OFFSET $n;
```

SQL112	List all notifications of User
Web Resource	R107
`SELECT * from notification WHERE user_id = $user_id;`	

SQL113	Get Information about the user
Web Resource	R107
`SELECT * FROM "user" WHERE username = $username;`	

SQL114	Get the number of tasks completed by a user in the current week
Web Resource	R107

```
SELECT COUNT(*) FROM task_state_record
WHERE user_completed_id = $user_id AND state = 'Completed'
AND (SELECT extract('week' FROM task_state_record.date)) =
(select extract('week' from current_date));
```

SQL115	Get the number of tasks completed by a user in the current month
Web Resource	R107

```
SELECT COUNT(*) FROM task_state_record
WHERE user_completed_id = $user_id AND state = 'Completed'
```

```
AND (SELECT extract('month' FROM task_state_record.date)) =
(select extract('month' from current_date));
```

SQL116	Get the number of sprints the user contributed to (by being assigned or completed them)
Web Resource	R107

```
SELECT COUNT(task.sprint_id) FROM task_state_record, task
WHERE task_state_record.user_completed_id = $user_id
AND task_state_record.state != 'Created'
AND task_state_record.task_id = task.id;
```

1.2 M02 : Project

SQL201	list members of one project
Web Resource	R201/R202

```
SELECT "user".username FROM project_members
WHERE project_members.project_id = $project_id AND user.id =
project_members.user_id
LIMIT 10 OFFSET $n;
```

SQL202	Remove Member from Project
Web Resource	R203

```
DELETE FROM project_members
    WHERE user_id = (SELECT id FROM "user" WHERE username =
$username);
```

SQL203	List all requests to join the project
Web Resource	R204

```
SELECT "user".username FROM invite, "user"
```

```sql
WHERE project_id = $project_id AND invite.user_who_invited_id
IS NULL
AND invite.user_invited_id = "user".id
LIMIT 10 OFFSET $n;
```

SQL204	Accept request to join the project
Web Resource	R205

```sql
INSERT INTO project_members (user_id, date, project_id,
isCoordenator)
VALUES ($user_id, $date, $project_id, $isCoordenator);
```

SQL205	Reject request to join the project
Web Resource	R206

```sql
DELETE FROM invite
    WHERE user_inited_id = $user_inited_id AND project_id =
$project_id;
```

SQL206	Edit Project Information
Web Resource	R207

```sql
UPDATE Project
SET name = $name, description = $description, isPublic =
$isPublic
WHERE id = $project_id;
```

SQL207	Top 3 contributors in a project (statistics)
Web Resource	R208

```sql
SELECT "user".username, "user".image, COUNT(*) AS num
FROM "user", task_state_record, task
WHERE task.project_id = $project_id
```

```sql
AND task_state_record.task_id = task.id
AND "user".id = task_state_record.user_completed_id
AND task_state_record.state = 'Completed'
GROUP BY "user".username, "user".image
ORDER BY num DESC LIMIT 3;
```

SQL208	Number of tasks completed this month in a project, by each day (statistics)
Web Resource	R208

```sql
SELECT COUNT(*), date_part('day',date) AS day
FROM task_state_record, task
WHERE task.project_id = $project_id AND
task_state_record.task_id = task.id
AND task_state_record.state = 'Completed' AND
date_part('month',date) = date_part('month',now())
GROUP BY day;
```

SQL209	Number of sprints completed this year in a project, by each month (statistics)
Web Resource	R208

```sql
SELECT COUNT(*), date_part('month',date) AS month
FROM sprint_state_record, sprint
WHERE sprint.project_id = $project_id AND
sprint_state_record.sprint_id = sprint.id
AND sprint_state_record.state = 'Completed' AND
date_part('year',date) = date_part('year',now())
GROUP BY month;
```

SQL210	Number of tasks completed in a project (statistics)
Web Resource	R208

```sql
SELECT COUNT(*)
FROM task, task_state_record
```

```
WHERE task.project_id = $project_id
AND task_state_record.task_id = task.id
AND task_state_record.state = 'Completed';
```

SQL211	Number of sprints completed in a project (statistics)
Web Resource	R208

```
SELECT COUNT(*)
FROM sprint, sprint_state_record
WHERE sprint.project_id = $project_id
AND sprint_state_record.sprint_id = sprint.id
AND sprint_state_record.state = 'Completed';
```

SQL212	Search a team member or coordinator of a specific Project
Web Resource	R209/R210

```
SELECT "user".username, "user".image,
project_members.isCoordinator
FROM project_members, "user"
WHERE project_members.project_id = $project_id AND
project_members.user_id = "user".id
AND "user".username LIKE '%$search%'
LIMIT 10 OFFSET $n;
```

1.3 M03 : Tasks

SQL301	List all tasks of one specific project (name and state), and not related to any sprint
Web Resource	R301

```
SELECT task.name, task_state_record.state FROM task,
task_state_record
WHERE task.project_id = $project_id AND
task_state_record.task_id = task.id
```

```
AND task_state_record.state =
(SELECT "state" FROM task_state_record WHERE task_id = task.id
GROUP BY "state", date ORDER BY date DESC LIMIT 1)
GROUP BY task.name, task_state_record.state;
```

SQL302	List all comments of one specific task
Web Resource	R301

```
SELECT comment.content, comment.date, "user".username,
"user".image
FROM comment, task, "user"
WHERE task.id = $task_id AND comment.task_id = task.id AND
comment.user_id = "user".id;
```

SQL303	Shows the detailed information of Task
Web Resource	R302

```
SELECT name, description, effort
FROM task
WHERE task.id = $task_id;
```

SQL304	Edit Task
Web Resource	R304

```
UPDATE Task
SET name = $name, description = $description, effort = $effort
WHERE id = $task_id;
```

SQL305	Delete Task
Web Resource	R305

```
DELETE FROM task
    WHERE id = $task_id;
```

SQL306	Mark Task as completed
Web Resource	R306
`INSERT INTO task_state_record (date, state, user_completed_id, task_id) VALUES ($date, 'Completed', $user_completed_id, $task_id) ;`	

SQL307	Mark Task as assigned
Web Resource	R307
`INSERT INTO task_state_record (date, state, user_completed_id, task_id) VALUES ($date, 'Assigned', $user_completed_id, $task_id) ;`	

SQL308	Mark Task as unassigned
Web Resource	R308
`INSERT INTO task_state_record (date, state, user_completed_id, task_id) VALUES ($date, 'Unassigned', $user_completed_id, $task_id) ;`	

SQL309	Create Task of Project
Web Resource	R309
`INSERT INTO Task (name, description, effort, project_id) VALUES ($name, $description, $effort, $project_id);`	

SQL310	New Comment
Web Resource	R310

```
INSERT INTO comment (content, date, user_id, task_id,
thread_id) VALUES ($content, $date, $user_id, $project_id,
NULL);
```

SQL311	Edit Comment
Web Resource	R311

```
UPDATE comment
  SET content = $content, date = $date
    WHERE id = $comment_id;
```

SQL312	Delete Comment
Web Resource	R312

```
DELETE FROM comment
    WHERE id = $comment_id;
```

1.4 M04 : Sprints

SQL401	List all sprints of one specific project and their state
Web Resource	R401

```
SELECT sprint.id, sprint.name, sprint.deadline,
sprint_state_record.state FROM sprint, sprint_state_record
WHERE sprint.project_id = $project_id AND
sprint_state_record.sprint_id = sprint.id
AND sprint_state_record.state =
(SELECT "state" FROM sprint_state_record WHERE sprint_id =
sprint.id
      GROUP BY "state", date ORDER BY date DESC LIMIT 1)
GROUP BY sprint.id, sprint.name, sprint.deadline,
sprint_state_record.state
ORDER BY deadline ASC;
```

SQL402	List all tasks of one specific sprint (name and state)
Web Resource	R401

```sql
SELECT task.name, task_state_record.state FROM task,
task_state_record
WHERE task.project_id = $project_id AND
task_state_record.task_id = task.id AND task.sprint_id =
$sprint_id
AND task_state_record.state =
(SELECT "state" FROM task_state_record WHERE task_id = task.id
GROUP BY "state", date ORDER BY date DESC LIMIT 1)
GROUP BY task.name, task_state_record.state;
```

SQL403	List all comments of one specific task
Web Resource	R401

```sql
SELECT comment.content, comment.date, "user".username,
"user".image
FROM comment, task, "user"
WHERE task.id = $task_id AND comment.task_id = task.id AND
comment.user_id = "user".id;
```

SQL404	Edit Sprint Information
Web Resource	R403

```sql
UPDATE sprint
SET name = $name, deadline = $deadline, effort = $effort
WHERE id = $sprint_id;
```

SQL405	Create Sprint
Web Resource	R405

```
INSERT INTO sprint (name, deadline, effort, project_id, user_creator_id)
VALUES ($name, $deadline, $effort, $project_id, $user_creator_id);
```

SQL406	Delete Sprint
Web Resource	R406
`DELETE sprint WHERE id = $sprint_id;`	

1.5 M05 : Project Forum

SQL501	Lists all Threads form the project
Web Resource	R501

```
SELECT thread.name, "user".username, thread.date FROM thread,
"user"
WHERE thread.project_id = $project_id AND "user".id =
thread.user_creator_id
LIMIT 20 OFFSET $n;
```

SQL502	Create Thread
Web Resource	R503

```
INSERT INTO thread (name, description, date, project_id,
user_creator_id)
VALUES ($name, $description, $date, $project_id,
$user_creator_id);
```

SQL503	Show Thread Information
Web Resource	R504

```
SELECT thread.name, thread.description, "user".username,
"user".image, thread.date FROM thread, "user"
WHERE thread.id = $thread_id AND "user".id =
thread.user_creator_id;
```

SQL504	Edit Thread Information
Web Resource	R506

```
UPDATE thread
   SET name = $name, description = $description
    WHERE id = $thread_id;
```

SQL505	New Comment in the Thread
Web Resource	R507

```
INSERT INTO comment (content, date, user_id, task_id,
thread_id) VALUES ($content, $date, $user_id, NULL,
$thread_id);
```

SQL506	Edit comment in the thread
Web Resource	R509

```
UPDATE comment
  SET content = $content, date = $date
    WHERE id = $comment_id;
```

SQL507	Delete Thread
Web Resource	R510

```
DELETE FROM thread
   WHERE id = $thread_id;
```

SQL508	Delete Comment in the Thread
Web Resource	R511

```
DELETE FROM comment
    WHERE id = $comment_id;
```

1.6 M06 : Admin Administration, Report and Static Pages

SQL601	Lists Comment Reports
Web Resource	R601
`SELECT * FROM report WHERE reportType = 'CommentReported';`	

SQL602	Lists User Reports
Web Resource	R602
`SELECT * FROM report WHERE reportType = 'UserReported';`	

SQL603	Shows the detailed Comment Report information
Web Resource	R603
`SELECT report.date, report.summary, report.user_reported_id FROM report WHERE reportType = 'CommentReported' AND comment_reported_id = $comment_reported_id;`	

SQL604	Shows the detailed User Report information
Web Resource	R604
`SELECT report.date, report.summary, report.user_reported_id FROM report WHERE reportType = 'UserReported' AND user_reported_id = $user_reported_id;`	

SQL605	Dismiss report
Web Resource	R605
`DELETE FROM report WHERE id = $report_id`	

SQL606	Disable User from the platform
Web Resource	R606

```
UPDATE "user"
SET disable = "TRUE"
WHERE id = $user_id;
```

SQL607	Delete Comment
Web Resource	R607

```
DELETE FROM comment
    WHERE comment_id = $comment_id;
```

SQL608	Delete Project
Web Resource	R609

```
DELETE FROM project
    WHERE id = $project_id;
```

SQL609	Create Comment Report
Web Resource	R611

```
INSERT INTO report (date, summary, user_id, type,
comment_reported_id, user_reported_id) VALUES ($date, $summary,
$user_id, 'commentReported', $comment_reported_id, NULL);
```

SQL610	Create User Report
Web Resource	R612

```
INSERT INTO report (date, summary, user_id, type,
comment_reported_id, user_reported_id) VALUES ($date, $summary,
$user_id, 'userReported', NULL, $user_reported_id);
```

2. Transactions

Transactions needed to assure the integrity of the data, with a proper justification.

T01	Once a sprint is deleted, all tasks of the sprint are deleted (if the users chooses)
Isolation Level	SERIALIZABLE
Justification	When the sprint is deleted, it's tasks will also be deleted. Therefore there can be no change made to these tasks in between their deletion.

```
BEGIN TRANSACTION
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE READ WRITE

DELETE FROM task WHERE sprint_id = $sprint_id;
DELETE FROM sprint WHERE id = $sprint_id;

COMMIT
```

T02	Once a sprint is deleted, the tasks of the sprint can become orphan, that is, it will only belong to the project and not related to any sprint.
Isolation Level	SERIALIZABLE
Justification	When a sprint is deleted, it's tasks will be "moved" to the project, meaning that they won't belong to any sprint of the project but only to the project. While the updating of the sprint_id field of tasks, there can be no change in the task.

```
BEGIN TRANSACTION
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE READ WRITE

UPDATE task SET sprint_id = NULL WHERE sprint_id = $sprint_id;
DELETE FROM sprint WHERE id = $sprint_id;

COMMIT
```

Note: One possible transaction would be the insertion of notifications once an action is done that transpires in a creation of one. This type of transaction isn't here because all notifications are created by triggers, that create notifications with the specific type, depending on the action that occurred.

Revision History

1. Addition of weights to the full text search in queries SQL108 and SQL109.

Grupo 1717, 29/4/2018

Ana Margarida Oliveira Pinheiro da Silva, up201505505@fe.up.pt
Luís Miguel Cardoso Lopes Correia, up201503342@fe.up.pt
Pedro Daniel dos Santos Reis, up201506046@fe.up.pt
Vicente Fernandes Ramada Caldeira Espinha, up201503764@fe.up.pt