A9: Main accesses to the database and transactions

This artefact shows the main accesses to the database, including the transactions.

Module 1: Authentication and Personal Overview

Register and create a new project

We have to insure that when a user creates a project upon registry, the process is fully completed. The isolation level, READ UNCOMMITTED, avoids ending up with a new project without users or a user without project.

```
BEGIN;

SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;

INSERT INTO user_table (name, email, username, password )
VALUES($name, $email, $username, $password);

INSERT INTO user_project(id_user,id_project, is_coordinator)
VALUES ($id_user, $id_project,$coordinator);
COMMIT;
```

Accept invitation, register and enter an existing project

If a user registers after getting an invitation, we need to insure the whole process is completed. The isolation level, READ UNCOMMITTED, guarantees that if anything fails we don't lose the invitation.

```
BEGIN;

SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;

DELETE FROM invited_user WHERE email = $email;

INSERT INTO user_table (name, email, username, password )
VALUES($name, $email, $username, $password);

INSERT INTO user_project(id_user,id_project, is_coordinator)
VALUES ($id_user, $id_project,$coordinator);

COMMIT;
```

Module 2: Dashboard

Get project members

```
SELECT * FROM user_table, user_project
WHERE user_project.id_user = user_table.id AND user_project.id_project =
$project_id;
```

Get project tasks

```
SELECT name, deadline FROM task WHERE deadline > CURRENT_DATE AND project_id
= $project_id
ORDER BY deadline ASC LIMIT 3;
```

Get project forum posts

```
SELECT * FROM forum_post WHERE id_project = $project_id ORDER BY
date_modified DESC  LIMIT 3;
```

Get the project meeting that the user is attending

```
SELECT * FROM meeting, user_meeting WHERE meeting.date > CURRENT_DATE AND
meeting.id_project = $project_id AND meeting.id=user_meeting.meeting_id AND
user_meeting.user_id = $user_id;
```

Get project files

```
SELECT * FROM file, user_table WHERE user_table.id = file.uploader_id  AND
project_id = $project_id ORDER BY upload_date  DESC LIMIT 3;
```

Get project description

```
SELECT description FROM project WHERE id = $project_id;
```

Get project name

```
SELECT name FROM project WHERE id = $project_id;
```

Module 3: Team Management

Get project members and their personal info

```
SELECT id, name, username, email, phone_number, photo_path, birth_date,
country_id, city, is_coordinator
FROM user_table, user_project
WHERE user_project.id_user = user_table.id AND user_project.id_project =
$project_id;
```

Invite a user to the project

```
INSERT INTO invited_users (id_project, email) VALUES ($project_id,
$user_email);
```

Remove a user from a project

```
DELETE FROM user_project WHERE id_user = $user_id AND id_project =
$project_id;
```

Set a user as Coordinator

```
UPDATE user_project
SET is_coordinator = TRUE
WHERE id_user = $user_id AND id_project = $project_id;
```

Remove Coordinator privileges from a user

```
UPDATE user_project
SET is_coordinator = FALSE
WHERE id_user = $user_id AND id_project = $project_id;
```

Module 4: Project Forum

Get project posts

```
SELECT * FROM forum_post WHERE id_project = $project_id ORDER BY
date_modified DESC;
```

Get post replies

```
SELECT * FROM forum_reply WHERE forum_reply.post_ID = $id_post ORDER BY
creation_date ASC;
```

Submit a new post

```
INSERT INTO forum_post
title,creation_date,content,id_project,date_modified,id_creator)
VALUES  ($title,$date,$content,$project_id,$date,$creator_id);
```

Submit a reply to a post

```
INSERT INTO forum_reply (creation_date, content, post_id, creator_id)
VALUES ($date,$content,$id_post,$id_creator);
```

Update post content

```
UPDATE forum_post SET content = $content WHERE id = $post_id;
```

Module 5: Project Tasks

Get project tasks

```
SELECT * FROM task WHERE project_id = $project_id ORDER BY deadline ASC;
```

Get task completer name

```
SELECT name FROM user_table, task WHERE task.id = $task_id AND
task.completer_id = user_table.id;
```

Add a task

```
INSERT INTO task (id, name, description, deadline, creator_id, assigned_id,
completer_id,
 project_id) VALUES (DEFAULT, $name, NULL, NULL, $creator_id, NULL, NULL,
$project_id)
 RETURNING id;
```

Delete a task

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

Complete task

```
UPDATE task SET completer_id = $completer_id WHERE id=$task_id;
```

Assign task

```
UPDATE task SET assigned_id = $assinged_id WHERE id=$task_id;
```

Change task name

```
UPDATE task SET name = $task_name WHERE id=$task_id;
```

Add comment to task

```
INSERT INTO comment (id, creation_date, content, id_user, id_task)
VALUES (DEFAULT, LOCALTIMESTAMP, $content, $user_id, $task_id) RETURNING id;
```

Set task deadline

```
UPDATE task SET deadline = $deadline WHERE id=$task_id;
```

Set task description

```
UPDATE task SET description = $description WHERE id=$task_id;
```

Module 6: Project Meetings

Schedule a Meeting

To create a meeting we have to insert the user that creates the meeting on the user_meeting table. READ UNCOMMITTED is the necessary isolation level, in order to guarantee that both action always happen together (atomicity).

```
BEGIN;
```

```
SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;

INSERT INTO meeting(name, duration, description, id_project, DATE ) VALUES
($name,$duration,$description,$id_creator,$id_project,$date);

INSERT INTO user_meeting (meeting_id, user_id, is_creator)
VALUES ($id, $user_id, $is_creator);

COMMIT;
```

Invite member to a meeting

```
INSERT INTO user_meeting (meeting_id, user_id, is_creator) VALUES ($id,
$user_id, $is_creator)
```

See meeting details

```
SELECT name,description,duration,id_creator,DATE FROM meeting WHERE
meeting.id = $meeting_id
```

Delete Meeting

When a meeting is deleted, all entries in user_meeting table and the entry in meeting table must be deleted. Because table meeting has a foreign key to user_meeting with cascade deletion it is not necessary to start a transaction.

```
DELETE FROM meeting WHERE id = $id_meeting;
```

Get Future Meetings

```
SELECT id,name,description,id_creator,DATE FROM meeting
WHERE meeting.date > CURRENT_DATE AND meeting.id_project = $meeting_id;
```

Get Invited Users Photos

```
SELECT photo_path
FROM user_table
INNER JOIN user_meeting ON
meeting_id = $meeting_id AND user_table.id = user_meeting.user_id;
```

Module 7: Project Files

Upload File

```
INSERT INTO file(upload_date, uploader_id, project_id, name, path)
VALUES ($date, $user_id, $project_id, $name, $file_path);
```

Download File

```
SELECT path FROM file WHERE id = $file_id;
```

See File Details

```
SELECT upload_date, uploader_id, name, path FROM file WHERE id = $file_id;
```