**A9 – Main Accesses to the database and transactions**

This artefact shows the main accesses to the database in the Project Management application. All the main accesses are described with the corresponding SQL code and web resource reference. It also includes all the required transactions for this application.

1. **Main Accesses**

M01 Create new user

|  |  |
| --- | --- |
| SQL101 | Creates new user in the platform |
| Web resource | R105 |
| INSERT INTO User (e\_mail, password, URL, username)  VALUES ($e\_mail, $password, $URL, $username) | |

M02 Projects

|  |  |
| --- | --- |
| SQL201 | Search project |
| Web resource | R603 |
| SELECT id, name, description, start\_date, ts\_rank\_cd(textsearch, query) AS rank  FROM Project, to\_tsquery($search) AS query, to\_tsvector(name || ‘ ‘ || description) AS textsearch  WHERE query @@ textsearch\\ ORDER BY rank DESC; | |

M03 Boards

|  |  |
| --- | --- |
| SQL301 | Search board |
| Web resource | R215 |
| SELECT id, name, description ts\_rank\_cd(textsearch, query) AS rank  FROM Board, to\_tsquery($search) AS query, to\_tsvector(name || ‘ ‘ || description) AS textsearch  WHERE query @@ textsearch\\ ORDER BY rank DESC; | |

M04 Tasks

|  |  |
| --- | --- |
| SQL401 | Search task |
| Web resource | R307 |
| SELECT id, name, description ts\_rank\_cd(textsearch, query) AS rank  FROM Task, to\_tsquery($search) AS query, to\_tsvector(name || ‘ ‘ || description) AS textsearch  WHERE query @@ textsearch\\ ORDER BY rank DESC; | |

M05 Users

|  |  |
| --- | --- |
| SQL501 | Retrieve user's current projects |
| Web resource | R503 |
| SELECT username, full\_name, e\_mail FROM User  WHERE username LIKE %$search% OR full\_name LIKE %$search%;  ORDER BY username; | |

M06 User’s projects

|  |  |
| --- | --- |
| SQL601 | Retrieve user's current projects |
| Web resource | R113 |
| SELECT \*  FROM Project  INNER JOIN Project\_team ON Project.id=Project\_team.id\_project  WHERE Project.id\_coordinator=$user OR Project\_team.id\_user=$user | |

1. **Transactions**

|  |  |
| --- | --- |
| T01 | Update task progress |
| Isolation level |  |
| Justification | The task progress must be updated in two different tables, which means that if information is retrieved in the middle of the transaction, the information will not be coherent. |
| BEGIN TRANSACTION  SET TRANSACTION ISOLATION LEVEL READ COMITTED  -- Insert new progress\_update  INSERT INTO Progress\_update (new\_value, id\_user, id\_task)  VALUES ($new\_value, $id\_user, $id\_task)  -- Update column progress in table Task  UPDATE Task  SET progress = $new\_value  WHERE id = $id\_task  COMMIT | |

|  |  |
| --- | --- |
| T02 | Send message to forum |
| Isolation level |  |
| Justification | When a new message is sent (new instance in Message table), the last messages sent to the forum must be re-selected, or the user will not have access to updated information. |
| BEGIN TRANSACTION  SET TRANSACTION ISOLATION LEVEL READ COMITTED  -- Insert new message  INSERT INTO Message (message, id\_user, id\_project)  VALUES ($message, $id\_user, $id\_project)  -- Select last messages  SELECT \*  FROM Message  WHERE Message.id\_project=$projectId AND Message.date >= DATEADD(day, -7, GETDATE())  COMMIT | |

|  |  |
| --- | --- |
| T03 | Create task |
| Isolation level |  |
| Justification | The board’s tasks must be re-read after the addition of each new task, or the user will not have access to updated information. |
| BEGIN TRANSACTION  SET TRANSACTION ISOLATION LEVEL READ COMITTED  -- Insert new task  INSERT INTO Task (description, name, id\_creator, id\_board, deadline, budget)  VALUES ($description, $name, $id\_creator, $id\_project, $deadline, $budget)  -- Select tasks from board  SELECT Task.id  FROM Task  WHERE Task.id\_board=$boardId  COMMIT | |

|  |  |
| --- | --- |
| T04 | Create board |
| Isolation level |  |
| Justification | The project’s boards must be re-read after the addition of each new board, or the user will not have access to updated information. |
| BEGIN TRANSACTION  SET TRANSACTION ISOLATION LEVEL READ COMITTED  -- Insert new board  INSERT INTO Board (description, name, id\_creator, id\_project)  VALUES ($description, $name, $id\_creator, $id\_project)  -- Select boards from project and user  SELECT \* FROM Board  INNER JOIN Board\_team ON Board.id=Board\_team.id\_board  WHERE Board.id\_project=$projectId AND Board\_team.id\_user=$id\_user  COMMIT | |

GROUP1734, 21/04/2018:

* Maria Inês Gonçalves, up201402784@fe.up.pt
* Maria Teresa Valério, up201405655@fe.up.pt
* Sara Gomes, up201405085@fe.up.pt