Project and Seminar

2024/2025 - 2nd Semester

Bachelor in Computer Engineering and Informatics

Database Documentation

Ângelo Azevedo, n.º 50565, e-mail: a50565@alunos.isel.pt
António Alves, n.º 50539, e-mail: a50539@alunos.isel.pt

Advisor: Pedro Matutino, e-mail: pedro.miguens@isel.pt

March 2025

Introduction

This document provides a overview of the database, entities and attributes and relationships. It also includes implementation decisions.

Database overview

The database has been modeled using an Entity-Relationship (ER) approach. This approach allows for a better understading of the relationships between entities. The following section shows a figure with the ER model.

The database is implemented with PostgreSQL and tested using fake information in a postgres docker container.

The following section will address the ER model.

Entity-Relationship Model

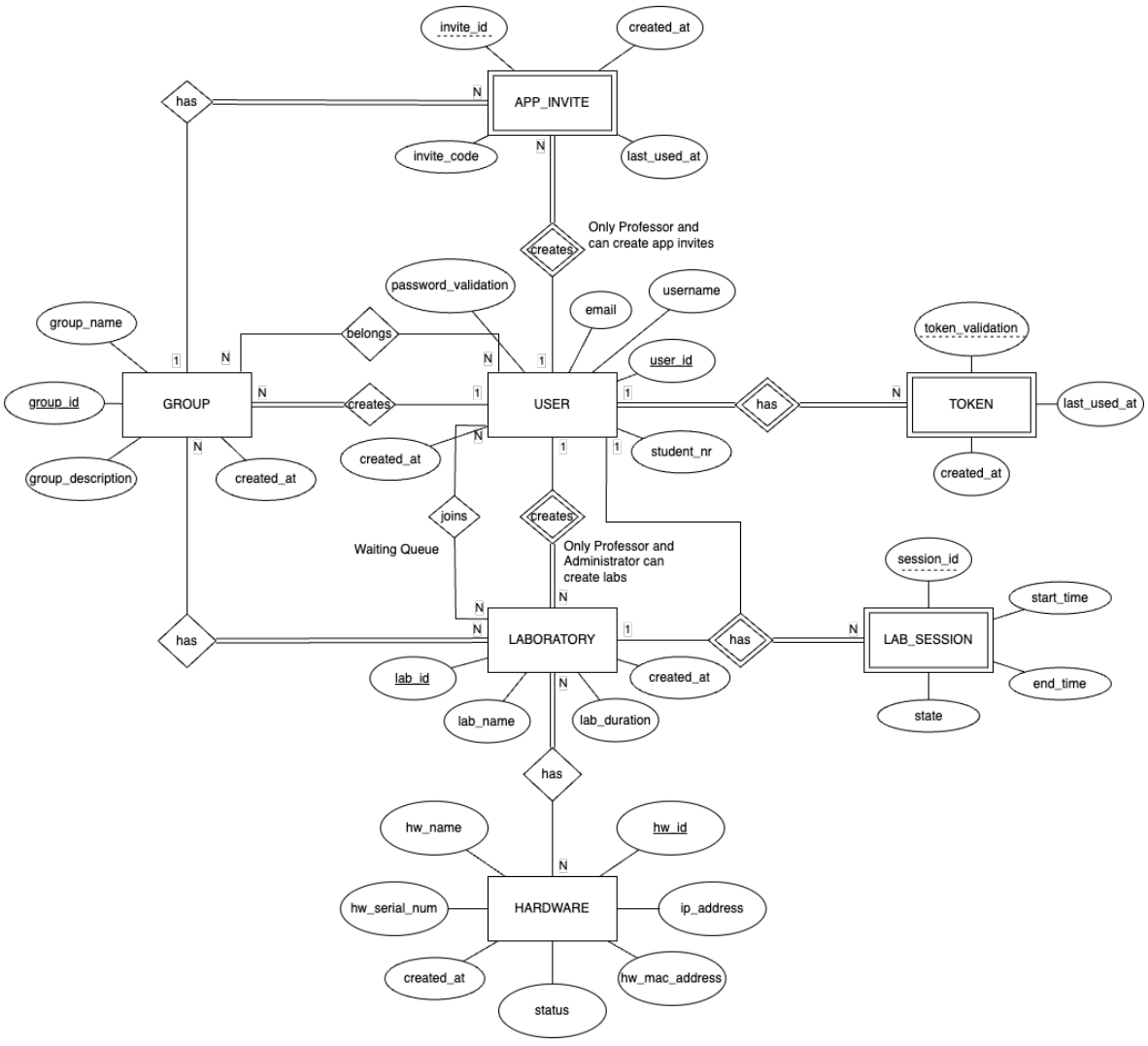

Figure 1: Entity-Relationship Model

Entities and Attributes

This section provides a comprehensive description of each entity and their attributes.

User

The first important entity is the **User**. This entity represents a user in the database.

It has a **user_id** as primary key and identity column. An identity column is a special column that is generated automatically from an implicit sequence. So, whenever a user is inserted in the database it will generate a id. The user_id is a int data type.

It also has, as char sequence, **username**, **email** and **password_validation**. All of them are not null and email is unique. Password_validation is the hash of the user's password.

Finally has a **student_nr** (student number) as Int type and unique and **created_at** as a timestamp type and not null. The student number can be null. This allows users, like professor or other, to be a user entity in the DB. Is worth mentioning that there is not a descriminator attribute because this descrimination will be implemented in the RBAC system.

Token

Token is a weak entity because it cannot be uniquely identified by its attributes alone. This means that a token needs a user_id to be identified. This way, token requires the user to exist.

Since it is a weak entity, it must have a partial key. This partial key is **token_validation**. Token_validation is a randomly generated hash. It is a char sequence and not null.

The last attributes are **created_at** and **last_used_at**. Both are timestamp types and not null.

App Invite

App Invite is a weak entity. For the same reason as the token, this entity requires the user to exist and cannot be uniquely identified. For example an app invite should not exist if the user is deleted.

As partial key, the **invite_id** with user_id uniquely identify an app invite. This invite_id is generated always as identity and is a int type.

It has an **invite_code** attribute as char sequence and not null. This invite code can take up to 255 bytes but, the true dimension is decided by the application domain.

It also has a **created_at** and **last_used_at** like the token entity.

Group

The **Group** entity represents a group. This group can be a class of students, a work group, professors group, etc...

It has a **group_id** attribute as primary key and uniquely identifies a group. It is generated always as identity and is a int type as well.

Also has a **group_name** as a char sequence and not null. This attribute represents the group name choose by the user.

It has a **group_description** as text type. This is a attribute where a user can write something about the group that he likes. It can be null and is a text type. A text type has a unlimited length, so that the user does not have a limit.

At last, it has a **created_at**. This attribute has the same characteristics like token's created at.

Laboratory

Laboratory, as the name indicates, represents a laboratory.

This entity has a **lab_id** as primary key. This is the unique identifier and indicates the laboratory id. It is generated always as identity and is a int type.

It has a **lab_name** attribute to respresent the laboratory name. It is a char sequence and not null.

Finally has a **lab_duration** attribute to represent the duration of each lab session. It is a int type and is not null. And it has a **created_at** attribute like the group's created at.

Lab Session

The **Lab Session** is a weak entity. This session should only exist depending on the user and a laboratory. It represents a lab session, that is, a session where a user can manipulate something in the laboratory.

As attributes has **session_id** as partial key. With user id and lab id a lab session can be uniquely identified. It is generated always as identity and is int type.

It has a **start_time** and **end_time** attributes to represent the date and hour a session start and ends. Both are timestamps and not null.

At last it has a **state** attribute to indicate the session state, that is, if it is active, inactive or scheduled. This attribute is a char sequence and is not null. If necessary this attribute can have more attributes. If it added more states this document should be updated.

Hardware

The last entity is **Hardware**. This entity can represent any hardware. For example it can represent a computer or a FPGA.

As primary key has a **lab_id**. It is generated always as identity and is a int type. This attribute uniquely identifies the hardware.

It has a **hw_name** attribute for the hardware name. It is a char sequence and not null.

It has a **hw_serial_num** attribute that represents the hardware serial number. It is a char sequence and not null.

It also has a **ip_address** and **hw_mac_address**. Both are char sequence and depending on the type of hardware they can be null.

Has a **hw_status** attribute as a char sequence and not null. This attribute indicates the status of the hardware. For example if it is available, occupied, etc...

Finnally, it has a **created_at** attribute with the same characteristics as laboratory's created at.

References

[1] PostgreSQL Global Development Group. Character types. `https://www.postgresql.org/docs/current/datatype-character.html`.

[2] PostgreSQL Global Development Group. Identity columns. `https://www.postgresql.org/docs/current/ddl-identity-columns.html`.