Server-side programming and data base interaction
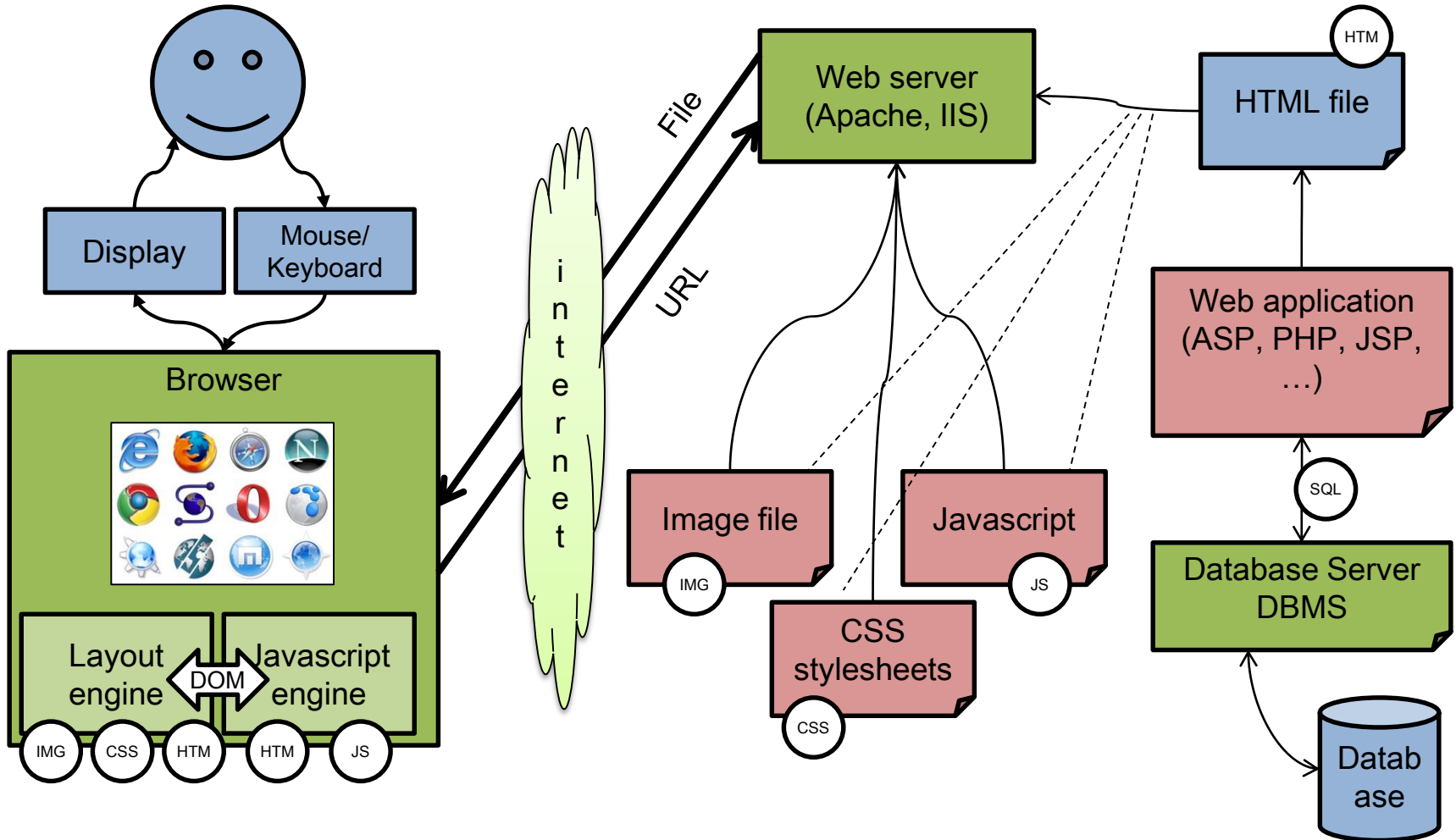
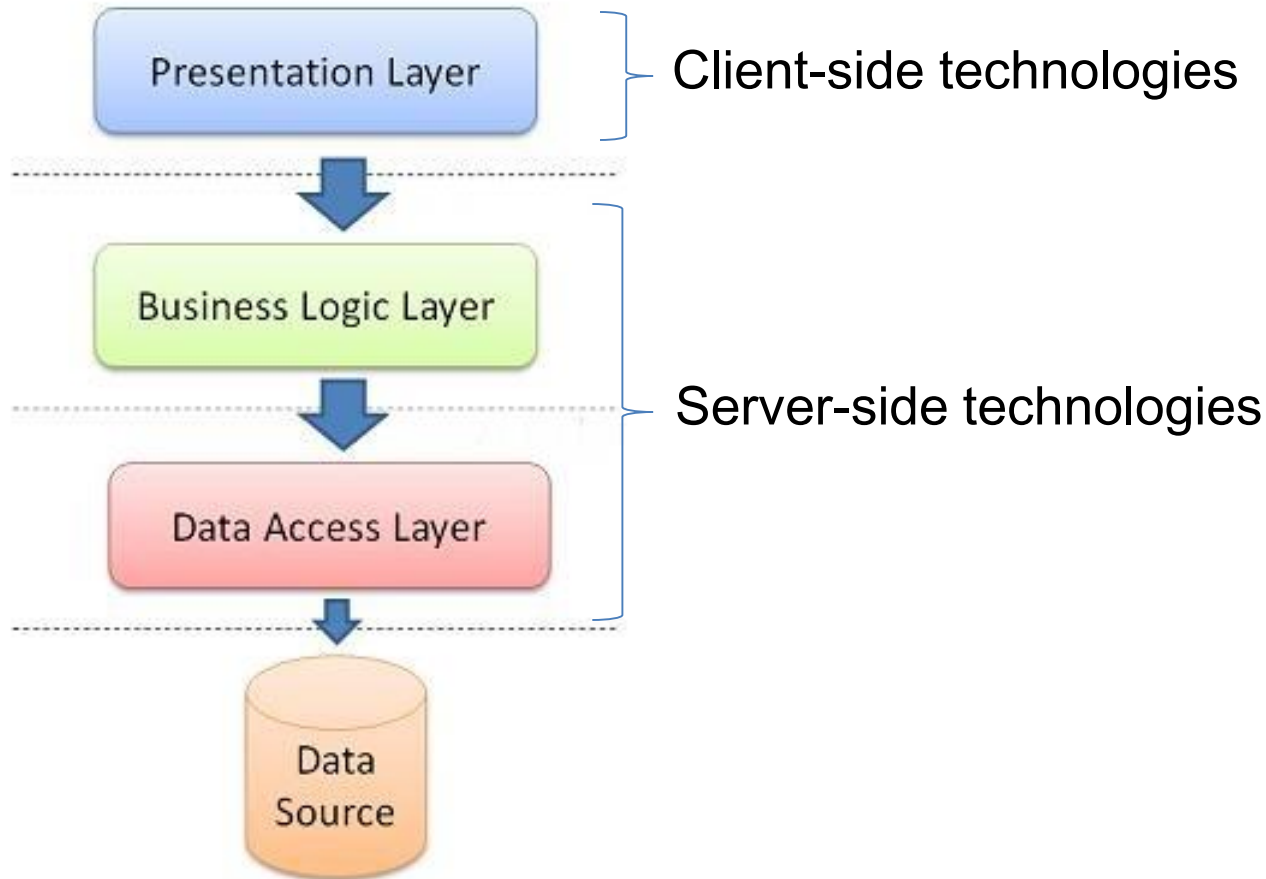# Rich-client asynchronous transactions

Application server

- A server program in distributed network that provides the business logic for an application program
- Dynamic page generation and content generation
- It's the middle tier between the client browser and the data residing on a database
- Implements the session mechanisms
- Different technologies and architectures are available
  - Java servlets, ASP (active server pages), JSP, PHP, PERL, Python, Ruby, …
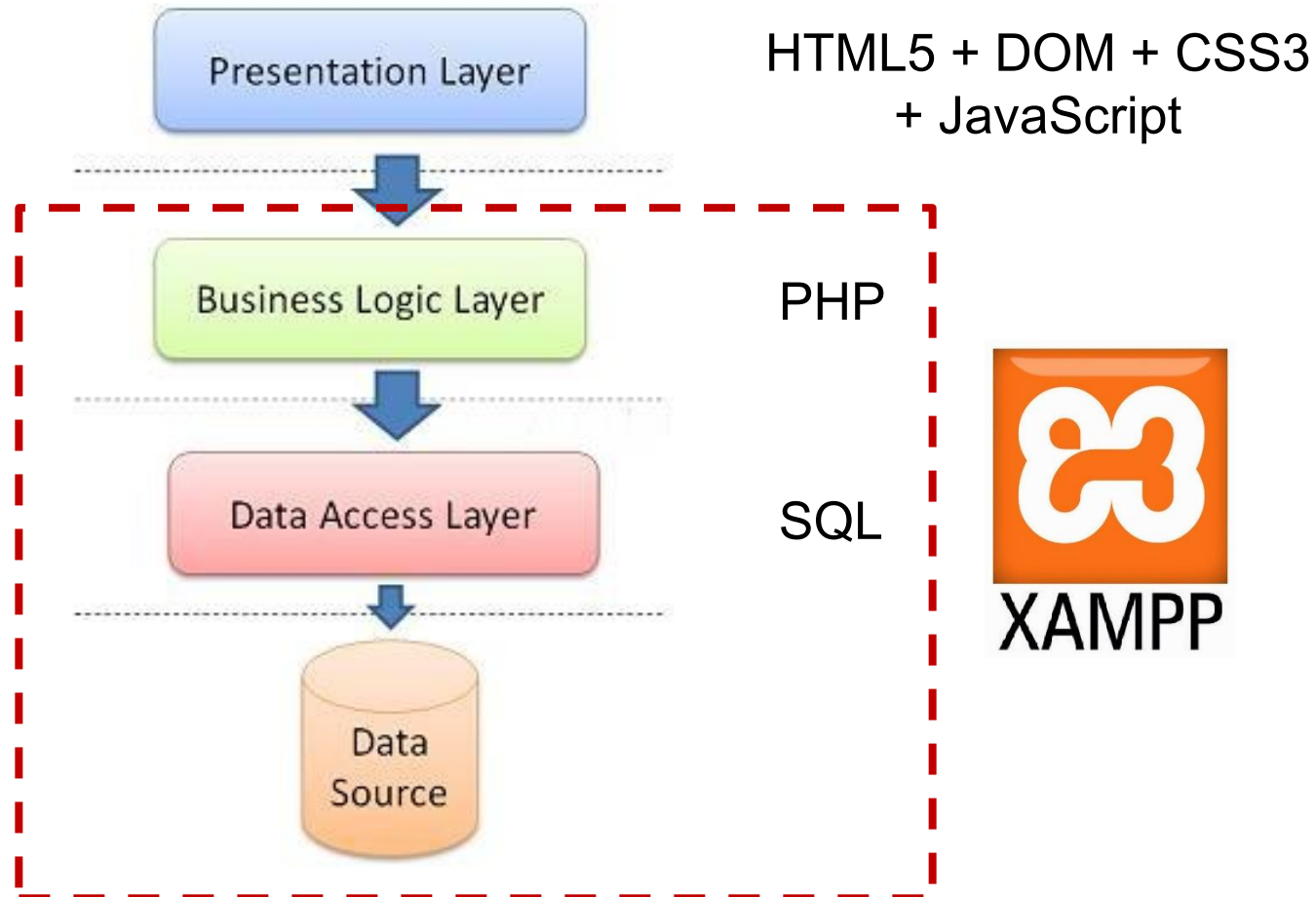
Database server

- Stores the data on which the application server works
- Executes the queries issued by the application server
  - Updates the stored data
  - Inserts new data
  - Provides query results back
- The most frequent/complex queries can be implemented internally as stored procedures (pre-compiled queries with parameters)
- Standards
  - SQL (structured query language), ODBC (open database connectivity) to access data bases

Web technologies and languages

Presentation Layer — Client-side technologies
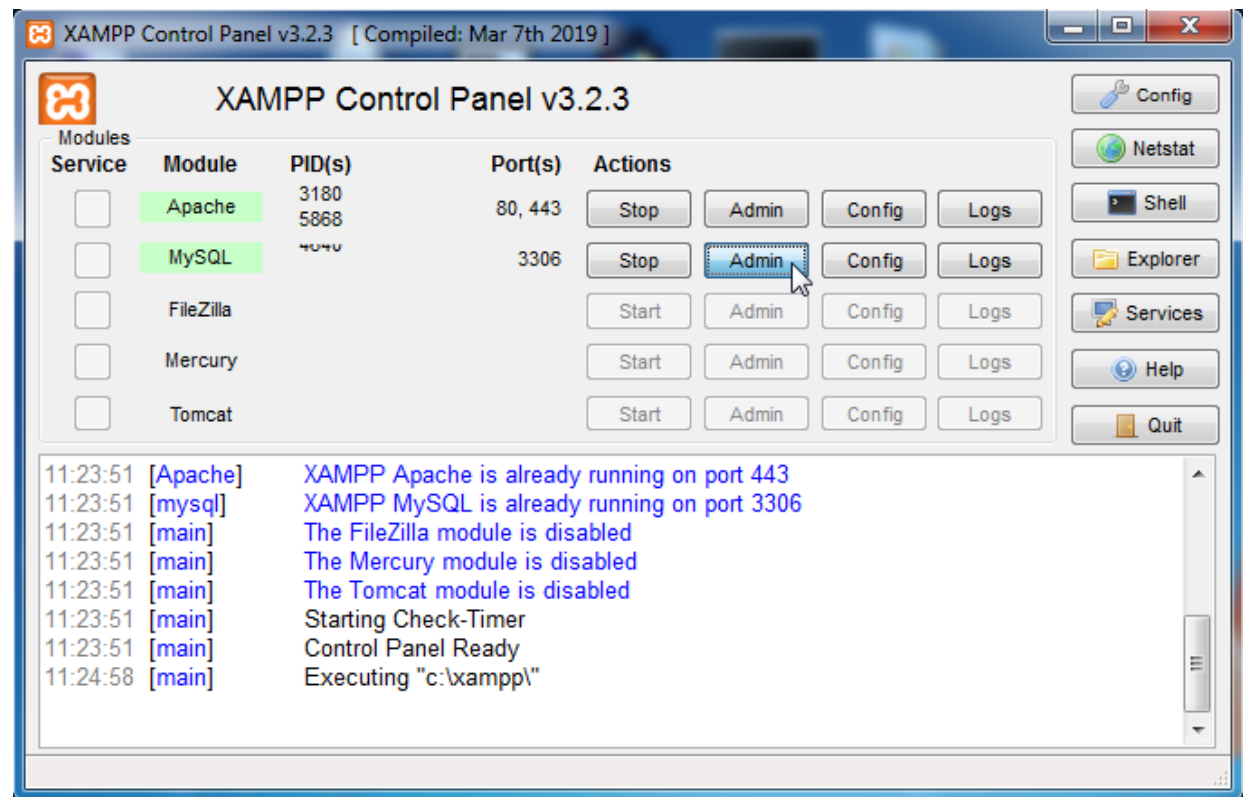
Business Logic Layer

Data Access Layer — Server-side technologies

Data Source

Web technologies and languages

HTML5 + DOM + CSS3
+ JavaScript

PHP

SQL

XAMPP

- Free, easy to install Apache distribution containing MariaDB, PHP, and Perl.
  - https://www.apachefriends.org/it/index.html
  - The user interface is the same as EasyPHP, but the tool is more stable
- Includes
  - A web server (Apache)
  - A database management system (MySQL)
  - A PHP script interpreter
  - A graphic DB administrator (phpMyAdmin)

phpMyAdmin

- Data Base management
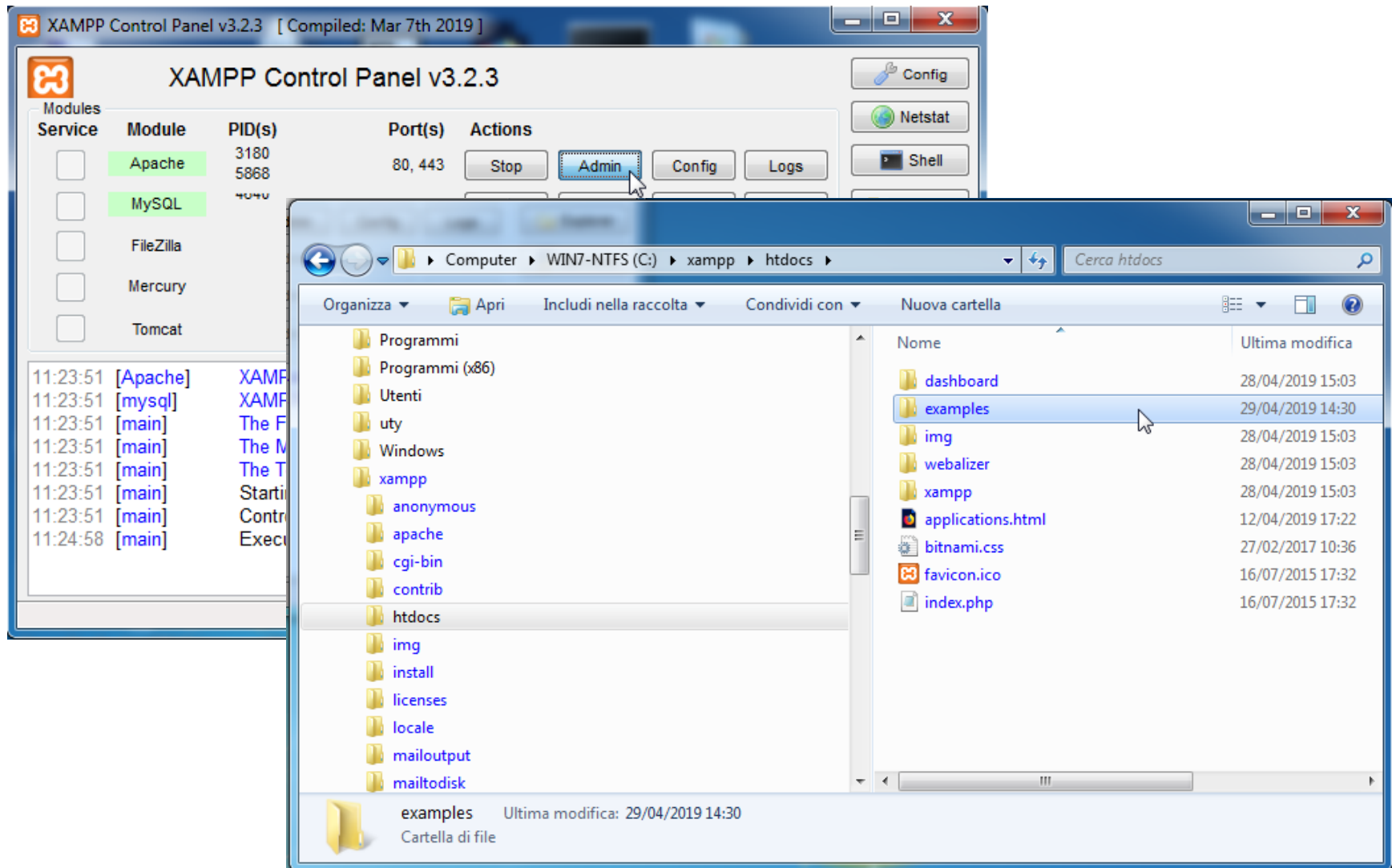 - Graphic user interface

phpMyAdmin

Local web server

Objective

- Create an application that query a database and return results

- Components
  - User interface (HTML5, CSS, JavaScript)
  - Server-side application (web server, PHP)
  - DBMS (MySQL)

- Review
  - Slides on PHP and on PHP+MySQL (see course web site)

Steps

- Design the database
  - E-R diagram (conceptual design)
  - E-R to tables (logical design)
- Implement the database
  - phpMyAdmin user interface (or SQL)
- Populate the database
  - phpMyAdmin user interface (or SQL)
- Write and test the query
  - phpMyAdmin user interface and SQL
- Write the user interface
  - HTML5 form, CSS, JavaScript, PHP(?)
- Write the result page
  - HTML5, CSS, JavaScript, PHP

Example

- Design and implement a loan service for video equipment (customer: Visionary Lab)

- User interface
  - For loaners: register, show equipment, show available equipment on a given date, search for a specific equipment, ask for a loan, …
  - For owners: update equipment list, show list of owners, show owners that are late, …

E-R diagram
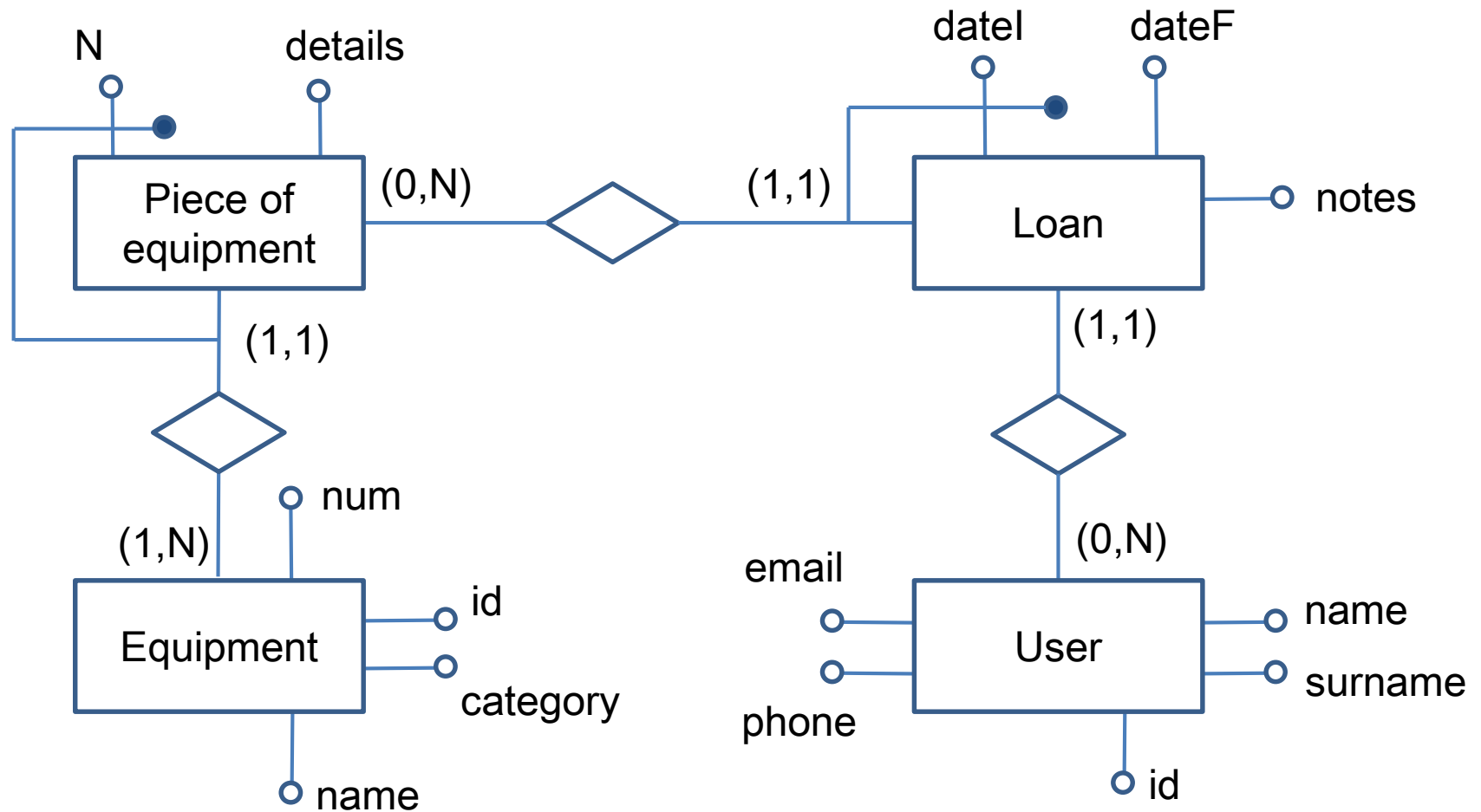
Tables
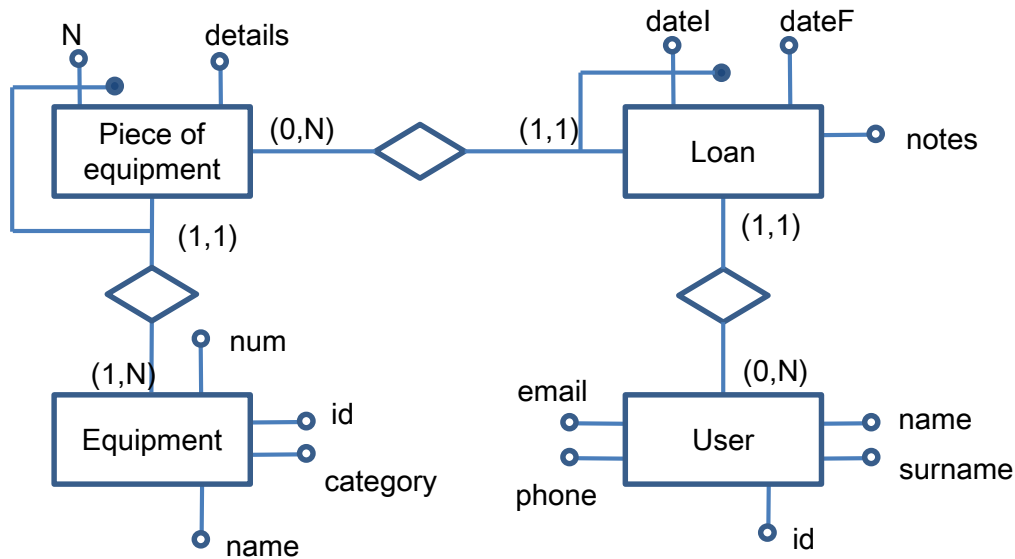
User (<u>idU</u>, name, surname, email, phone)
Equipment (<u>idEq</u>, name, category, num)
PieceOfEquipment (<u>N</u>, <u>idEq</u>, details)
Loan (<u>N</u>, <u>idEq</u>, <u>dateI</u>, idU, dateF, notes)

Tables

```
CREATE TABLE IF NOT EXISTS USER (
        idU VARCHAR(10) PRIMARY KEY,
        name VARCHAR(30) NOT NULL,
        surname VARCHAR(30) NOT NULL,
        email VARCHAR(50) NOT NULL,
        phone VARCHAR(30) NOT NULL
);

CREATE TABLE IF NOT EXISTS EQUIPMENT (
        idEq VARCHAR(10) PRIMARY KEY,
        name VARCHAR(255) NOT NULL,
        category VARCHAR(50),
        num INTEGER
);



CREATE TABLE IF NOT EXISTS PIECEOFEQUIPMENT (
        N INTEGER,
        idEq VARCHAR(10),
        details VARCHAR(255),
        PRIMARY KEY (N, idEQ),
        FOREIGN KEY (idEQ)
                REFERENCES EQUIPMENT (idEQ)
                ON UPDATE CASCADE
                ON DELETE CASCADE
);
```

Instances

```
SET autocommit=0;

START TRANSACTION;

INSERT INTO USER (idU, name, surname, email, phone)
VALUES ('98143','Mario','Rossi','mario.rossi@me.com','+39 333
3457865');

INSERT INTO USER (idU, name, surname, email, phone)
VALUES ('91432','Giada','Verdi','giada.verdi@hotmail.com','+39 323
4554673');

...

INSERT INTO EQUIPMENT (idEq, name, category, num)
VALUES ('AV01','Casse audio','Audio/video',3);
...

INSERT INTO PIECEOFEQUIPMENT (N, idEq, details)
VALUES (1, 'AV01','No cables');

INSERT INTO PIECEOFEQUIPMENT (N, idEq, details)
VALUES (2, 'AV01','Cables included');
...

INSERT INTO LOAN (N, idEq, dateI, dateF, idU, notes)
VALUES (2, 'AD01','2019-04-10','2019-04-14','91432', '');
...

COMMIT;
```

Web pages

- Input (dynamic select menu content)

Equipment Loan Request

Equipment: Visore_GEAR_VR_con_controller ▾
From: 04 / 03 / 2019 ⊗
To: 07 / 03 / 2019 ⊗

Reset Ask for Loan

Equipment Loan Request

Equipment: Visore_GEAR_VR_con_controller ▾
From: Casse_audio
To: Matrox_TripleHead2Go_SE
      Visore_GEAR_VR_con_controller

Reset Ask for Loan

Web pages

- Result

Loan Request Result

Matrox_TripleHead2Go_SE is available for loan on 2019-05-14

N	idEq	name
1	AD01	Matrox_TripleHead2Go_SE
2	AD01	Matrox_TripleHead2Go_SE

Web pages

- Result

Loan Request Result

Visore_GEAR_VR_con_controller is not available for loan on 2019-03-04.

Available equipment on 2019-03-04

N	idEq	name
1	AD01	Matrox_TripleHead2Go_SE
1	AV01	Casse_audio
2	AV01	Casse_audio
3	AV01	Casse_audio

Queries

- ## All equipment

```
SELECT name
FROM equipment
ORDER BY name
```

- ## Equipment on loan on a specific date

```
SELECT N, PE.idEq, name
FROM equipment AS E, pieceofequipment AS PE
WHERE E.idEq = PE.idEq
AND (N, PE.idEq) NOT IN (SELECT N, idEq
                              FROM loan
                              WHERE dateI="2019-03-24")
```

- ## Specific equipment on loan on a specific date

```
SELECT N, PE.idEq, name
FROM equipment AS E, pieceofequipment AS PE
WHERE E.idEq = PE.idEq
AND name="Casse_audio"
AND (N, PE.idEq) NOT IN (SELECT N, idEq
                              FROM loan
                              WHERE dateI="2019-03-24")
```

References

- PHP
  - https://www.w3schools.com/php/default.asp
- PHP + MySQL
  - https://www.w3schools.com/php/php_mysql_intro.asp
- XAMPP
  - https://www.apachefriends.org/it/index.html

License

- This work is licensed under the Creative Commons "Attribution-NonCommercial-ShareAlike Unported (CC BY-NC-SA 3,0)" License.
- You are free:
  - to Share - to copy, distribute and transmit the work
  - to Remix - to adapt the work
- Under the following conditions:
  - Attribution - You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).
  - Noncommercial - You may not use this work for commercial purposes.
  - Share Alike - If you alter, transform, or build upon this work, you may distribute the resulting work only under the same or similar license to this one.
- To view a copy of this license, visit http://creativecommons.org/license/by-nc-sa/3.0/