Introduction to Internet and Web technologies – How to Structure the Course

Branislava Šandrih, Vladimir Filipović
Faculty of Mathematics, University of Belgrade

Jahorina, Bosnia and Herzegovina, 2016

Outline

- Informatics module
- “Introduction to Internet and Web Technologies” course
  - Staff and goals
  - Content of the course
  - Quick overview of the technologies
  - Methodical issues
- Strategy for reusing already familiar technologies
  - Node.js instead of PHP
- Conclusion

Informatics module

- Existing modules: Informatics, Mathematics and Astronomy
- New accreditation, duration 4 years (8 semesters), 240 ECTS
- Previous accreditation, duration 3 years (6 semesters), 180 ECTS
- According to official statistics, https://webserver.rcub.bg.ac.rs/sr/kandidati-zbirno currently 2nd most desired module in the Belgrade University

“Introduction to Internet and Web Technologies” course

- Mandatory course held during the first semester of the second year of undergraduate studies
- Classes, 3 + 2
- 6 ECTS
- Students should be able to understand basic concepts of
  - World Wide Web
  - Internet
  - Web technologies
- By the end of the course, student is able develop dynamical, interactive web application

“Introduction to Internet and Web Technologies” course (2)

- Outline of the course
  - Basic concepts of Networking, Internet and WWW
  - Internet services and Internet protocols
  - Markup languages
    - meta languages
    - language for creating web pages/applications
    - language for describing presentation of a document
    - adding interactivity to web pages
      - introducing DOM
      - client-side scripting language

“Introduction to Internet and Web Technologies” course (3)

- Web servers
  - installation and configuration
- Dynamical Web applications, server-side scripting
- Using DB in web environment
  - introducing DBMS
  - accessing DB from server-side script
- Asynchronous web applications
- Basics of web security and protection
- Browsers' optimization

“Introduction to Internet and Web Technologies” staff

- Two professors, three teaching assistants
- On average, 150 students
- The course is held only to students of Informatics
  - therefore brings big responsibility to just a few teachers
- Every change in concept of the course requires all teachers' agreements
- Teamwork is of crucial importance!

Content of the course

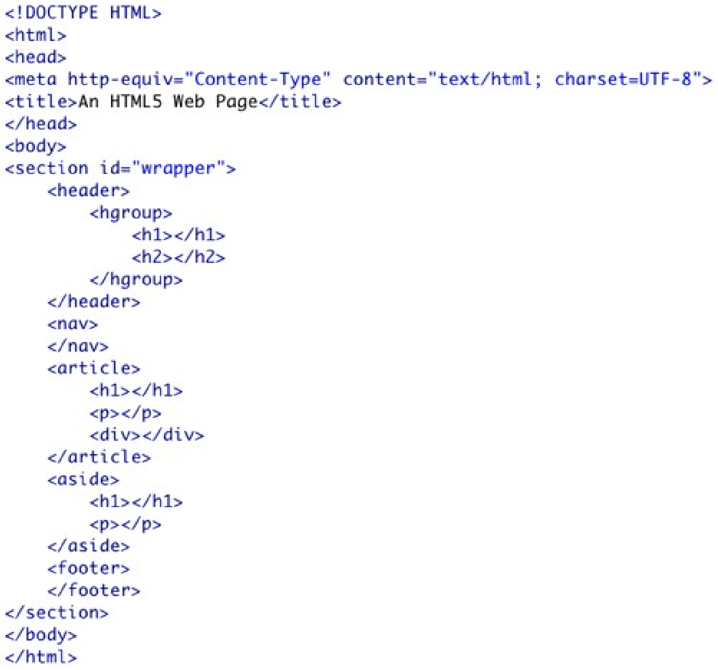
- (1.5 w) Introduction to Networking and Internet
  - Retrospective overview
  - Internet services and Internet protocols
    - HTTP, POP3, IMAP, SMTP, FTP
- (0.5 w) Markup languages, SGML and XML
- (2 w) HTML5
- (1.5 w) CSS3
- (1 w) DOM and client-side scripting, JavaScript

Content of the course (2)

- (1 w) Advanced JavaScript concepts
  - object/class, closures, lambdas
- (0.5 w) Web servers, Apache
- (0.5w) Server-side scripting language, PHP
- (1 w) Introduction to relational databases, MySQL
- (1 w) PHP MySQL API
- (1 w) AJAX
- (1 w) Miscellaneous
  - Web pages' design, data protection and security, SEO etc.

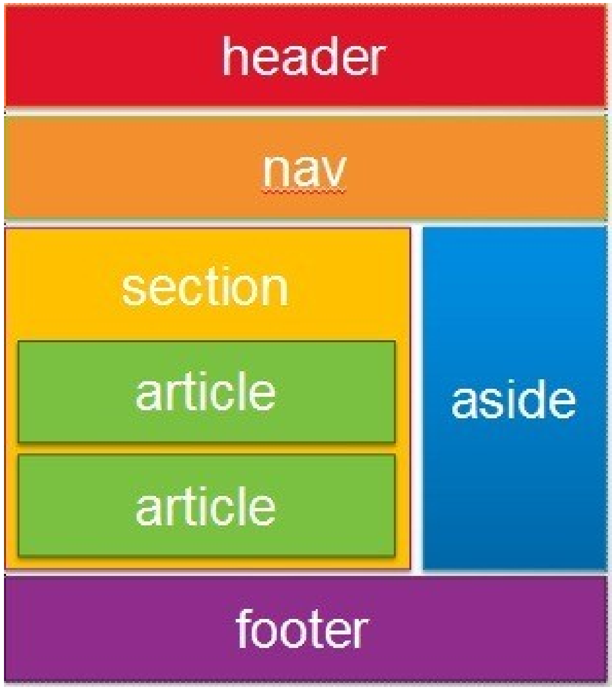
Quick overview of the technologies

- Hypertext Markup Language (HTML5)
  - markup language used for structuring and presenting content on the World Wide Web



```
<!DOCTYPE HTML>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>An HTML5 Web Page</title>
</head>
<body>
<section id="wrapper">
    <header>
        <hgroup>
            <h1></h1>
            <h2></h2>
        </hgroup>
    </header>
    <nav>
    </nav>
    <article>
        <h1></h1>
        <p></p>
        <div></div>
    </article>
    <aside>
        <h1></h1>
        <p></p>
    </aside>
    <footer>
    </footer>
</section>
</body>
</html>
```

Simple HTML document structure



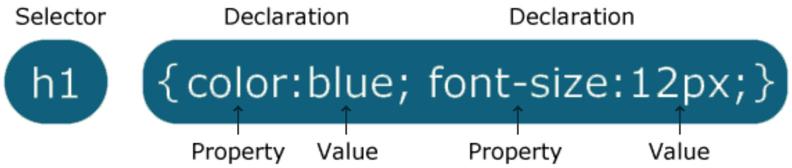
Visual document structure

Quick overview of the technologies (2)

- HTML5 elements covered within course
  - html, head, body
  - meta, title, script, style, link
  - header, main, footer, aside, article, section, nav
  - p, h1-h6, em, strong, small, b, i, u, sub, sup
  - address, blockquote, q, cite, pre, code
  - ul, ol, li, dl, dd, dt
  - img, audio, video, iframe
  - table, tr, td, th, caption
  - input, select, option, textarea, label
  - div, span

Quick overview of the technologies (3)

- Cascading Style Sheets
  - style sheet language used for describing the presentation of a document written in a markup
  - simple example



Selector

Declaration

Declaration

h1 { color:blue; font-size:12px;}

Property

Value

Property

Value

  - 1/20 part of youtube's css file

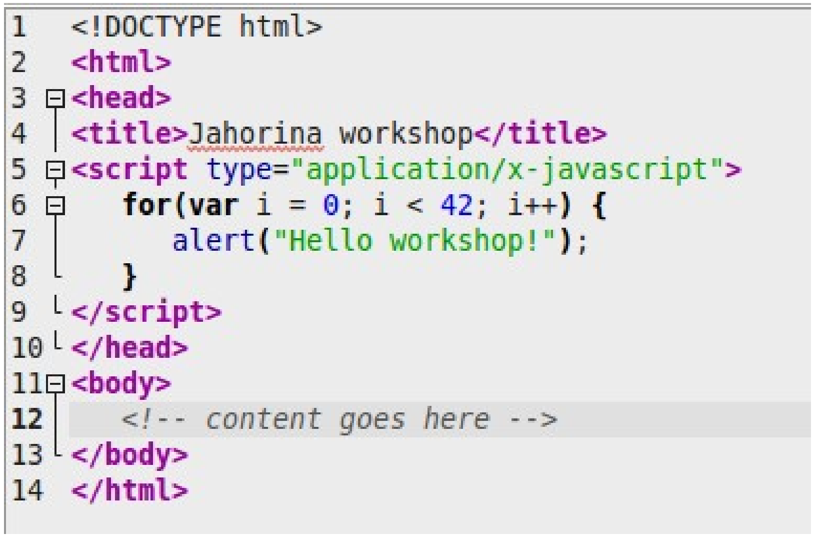
```
.yt-help-icon{opacity:.5;cursor:pointer;background:no-repeat url(//s.ytimg.com/yts/imgbin/www-hitchhiker-vflhdhwjt.png) -556px -190px;background-size:auto;width:15px;height:16px}.yt-help-icon:hover,.yt-help-icon.active,.yt-uix-helpcard-active .yt-help-icon{opacity:1}.add-to-widget .lightweight-message{padding:0 10px}.add-to-widget.yt-uix-clickcard-card-content{width:auto;padding:0}.add-to-widget .yt-scrollbar ::-webkit-scrollbar{width:9px}.add-to-widget .yt-scrollbar ::-webkit-scrollbar-thumb,.add-to-widget .yt-scrollbar ::-webkit-scrollbar-track{border-left-width:0}.add-to-widget .menu-panel{display:none}.add-to-widget .menu-panel.active-panel{display:block}.add-to-widget .playlists{width:260px;max-height:245px;padding-bottom:10px;overflow-x:hidden;overflow-y:auto;border-bottom:1px solid #ccc}.add-to-widget .addto-playlist-item{position:relative;display:block;padding:0 10px;line-height:25px;color:#333;white-space:nowrap;cursor:pointer}.add-to-widget .addto-playlist-item-subtitle{color:#adadad;margin-left:25px;width:190px;font-size:11px;overflow:hidden;white-space:nowrap;word-wrap:normal;-o-text-overflow:ellipsis;text-overflow:ellipsis}.add-to-widget .create-playlist-item{display:none;width:100%;text-align:left;padding:5px 0 5px 16px;margin:10px 0;color:#333;white-space:nowrap;cursor:pointer}.addto-playlist-item.icon-hidden{padding-left:16px}.add-to-widget .addto-playlist-item.hid{display:none}.add-to-widget .addto-playlist-item:hover,.add-to-widget .create-playlist-
```

Quick overview of the technologies (4)

- CSS elements covered within course
  - including CSS declarations within web pages
  - selectors
    - names, identifiers, classes, pseudo-classes, attributes, nested elements, ancestors, descendants etc.
  - CSS3 Properties:
    - font: font-family, font-size, font-weight: bold, font-style: italic, text-decoration: none, text-decoration: underline, text-align, text-indent, text-transform, vertical-align, line-height, color, width, height, box-sizing: border-box, box-sizing: content-box, min-width, max-width, min-height, max-height, margin, padding, border, border-radius, background-color, background-image, background-repeat, list-style-type: square, list-style-type: lower-alpha, list-style-type: upper-roman, list-style-image. display: block, display: inline, display: inline-block, display: table-cell, visibility: hidden, position: static, position: relative, position: absolute, position: fixed, float, clear

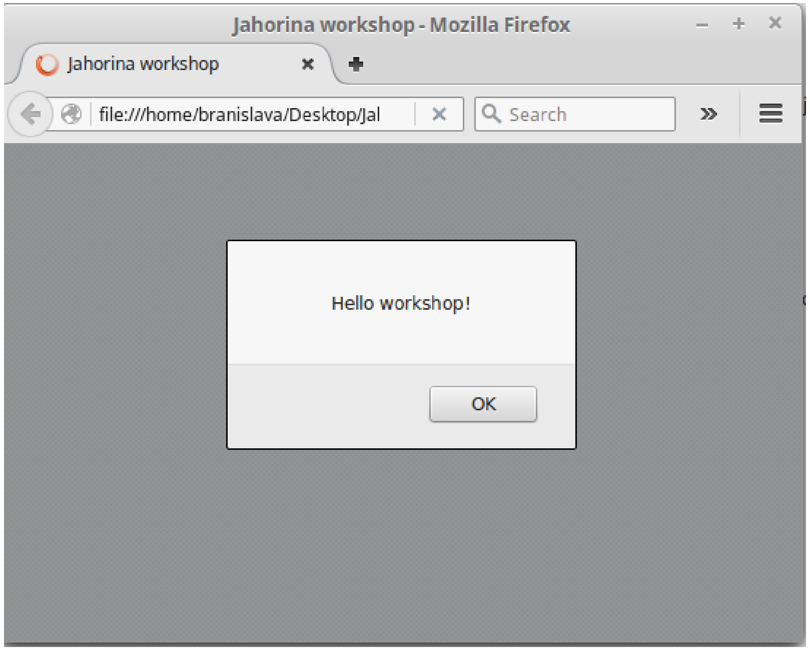
Quick overview of the technologies (5)

- JavaScript
  - scripting language, most often run in web browser applications to create dynamic content



```
<!DOCTYPE html>
<html>
<head>
<title>Jahorina workshop</title>
<script type="application/x-javascript">
    for(var i = 0; i < 42; i++) {
        alert("Hello workshop!");
    }
</script>
</head>
<body>
    <!-- content goes here -->
</body>
</html>
```

Simple JavaScript code written directly inside HTML document


Jahorina workshop - Mozilla Firefox
Jahorina workshop
file:///home/branislava/Desktop/Jal
Search
OK

Hello workshop!

Result (times 42)

Quick overview of the technologies (6)

- client-side scripting language
- from 2011 can be used for server-side scripting
  - Node.js
- C-style language (syntacticly, but “under the hood” vastly different from C and C++!)
  - object oriented: classes and instances, behavior encapsulation, using constructors and the new keyword for creating instances of objects
  - creating elements: inserting elements into the DOM
  - canvas: painting and animating on the HTML5 canvas element
  - local storage: saving things across refreshes
  - regular expressions: e.g. form validation
  - errors and exceptions: deep control during the runtime
  - closures: function that returns a function (analogue to object with a single method)
  - libraries: JQuery, AngularJS

Quick overview of the technologies (7)

- PHP, MySQL, Apache
  - Hypertext Preprocessor: server-side scripting language designed for web development, but also used as a general-purpose programming language
  - My Structured Query Language: relational database management system
  - Apache: web server software

```
function login() {
    $username = trim($_POST['username']);
    $password = trim($_POST['password']);

    if(!($this->connection = mysql_connect($this→db_host, $this->username,$this-
>pwd))             return false;

    $pwdmd5 = md5($password);
    $qry = "select name, email from login where username='$username' and
               password='$pwdmd5' ";

    $result = mysql_query($qry, $this->connection);

    if(!$result || mysql_num_rows($result) <= 0) return false;

    session_start();
    return true;
}
```

Simple PHP code for user login

Summary

- Covered both
  - client-side programming, server-side programming
- Very dense structure
  - yet, limited amount of time
- Every single topic is abundant
  - every technology itself could be the focus of the single course
  - choosing quantity over quality?
- Consequence: “a little about everything”
- How bad it really is?
  - e.g. not enough time for important and advanced concepts of JavaScript
    - 1 week is not nearly enough to thoroughly cover all the related topics

Possible solution: reuse JavaScript

- One possible way to overcome this issue would be using JavaScript for both, client-side and server-side programming
- Re-using JavaScript on the server could be helpful
  - since client-side is firstly covered, syntax is already familiar
  - switching between languages for browser and the server slows learning process, since that much new information cannot be properly processed in such a short amount of time
    - fresh amount of new information is served from week to week
    - this reduces amount of depth a course can cover a language in

What is Node.js?

- Open-source, cross-platform runtime environment for developing server-side Web applications
- Many of its basic modules are written in JavaScript, and it is extensible with new modules written in JavaScript
- Good for real-time communication programs and browser games
- Uses a document (not relational) database (such as MongoDB or CouchDB) and JSON data representation

Node.js features

- Asynchronous and Event Driven: all APIs of Node.js library are asynchronous that is, non-blocking. It essentially means a Node.js based server never waits for an API to return data. The server moves to the next API after calling it and a notification mechanism of Events of Node.js helps the server to get a response from the previous API call
- Very Fast: being built on Google Chrome's V8 JavaScript Engine, Node.js library is very fast in code execution
- Single Threaded but Highly Scalable: it uses a single threaded model with event looping. Event mechanism helps the server to respond in a non-blocking way and makes the server highly scalable as opposed to traditional servers which create limited threads to handle requests
- No Buffering : Node.js applications never buffer any data.
  These applications simply output the data in chunks
- License: Node.js is released under the MIT license

Node.js analogy

Balls (requests) are being thrown without any order (asynchronously);
Tennis player knows about the ball, as he sees one approaching (event driven);
Tennis player (Node.js web server) hits the ball (process request) immediately after receiving it;
Tennis player is alone and can hit one ball at a time (single threaded);
Tennis player does not wait the ball to return and has no knowledge if the shot was correct (does not wait for an answer);
Tennis player does not save any balls (no buffering);

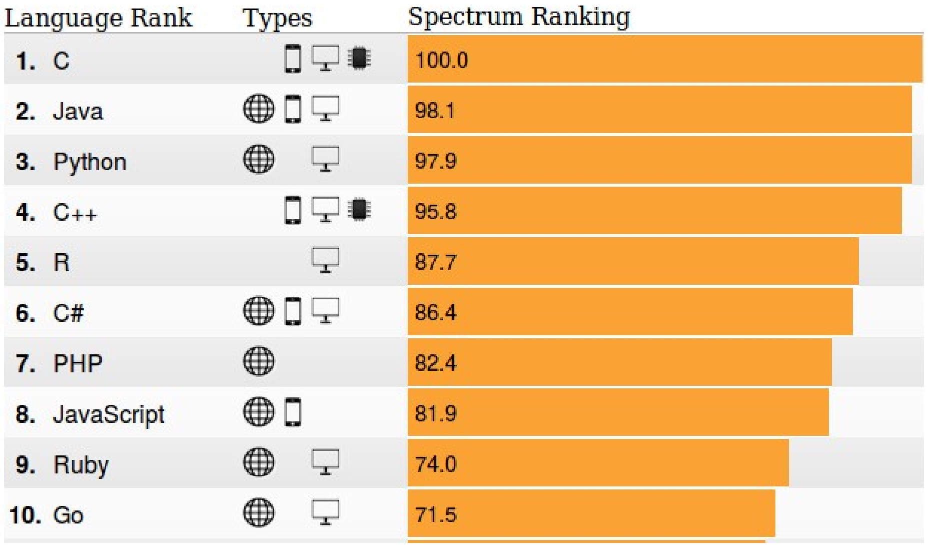
Strategy for replacing PHP with Node.js

➔ Pros:

- language competency (syntactical familiarity)
- smooth learning curve (continuing to rely on the gained JavaScript skills while making steps toward greater complexity)
- building better client-side skills on the way
- functional programming and closures
- document databases instead of relational
- better understanding of the concepts of asynchronous I/O and events, instead of classical procedural programming

Strategy for replacing PHP with Node.js (2)

- Cons:
  - it is bad not to learn PHP
  - Web programming course
    - master studies
    - elective course
    - 9 ECTS
    - covers PHP



Language Rank	Types	Spectrum Ranking
1. C	📱 🖥 ▣	100.0
2. Java	🌐 📱 🖥	98.1
3. Python	🌐 🖥	97.9
4. C++	📱 🖥 ▣	95.8
5. R	🖥	87.7
6. C#	🌐 📱 🖥	86.4
7. PHP	🌐	82.4
8. JavaScript	🌐 📱	81.9
9. Ruby	🌐 🖥	74.0
10. Go	🌐 🖥	71.5

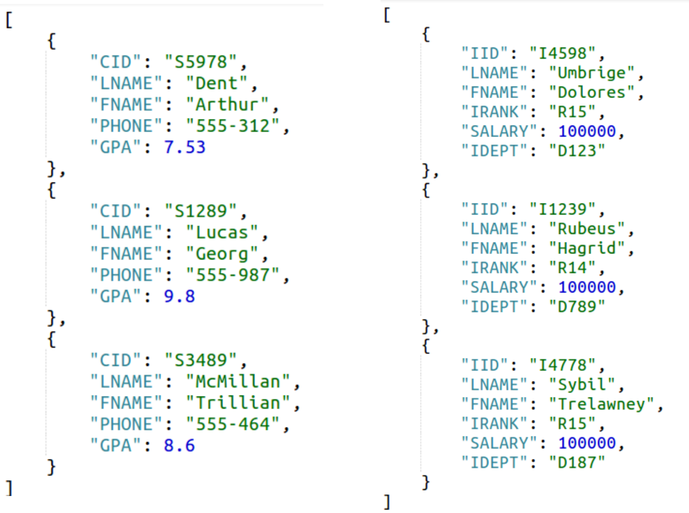
http://spectrum.ieee.org/static/interactive-the-top-programming-languages-2016

Strategy for replacing PHP with Node.js (2)

- As long as students learn about PHP while taking “Web programming” course, this is not an issue!
- Uncertainties:
  - not all students take the course
  - is learning about PHP at master studies too late?
  - what about databases?
    - Node.js natively communicates with NoSQL databases, but has support for MySQL too

Relational vs. NoSQL databases

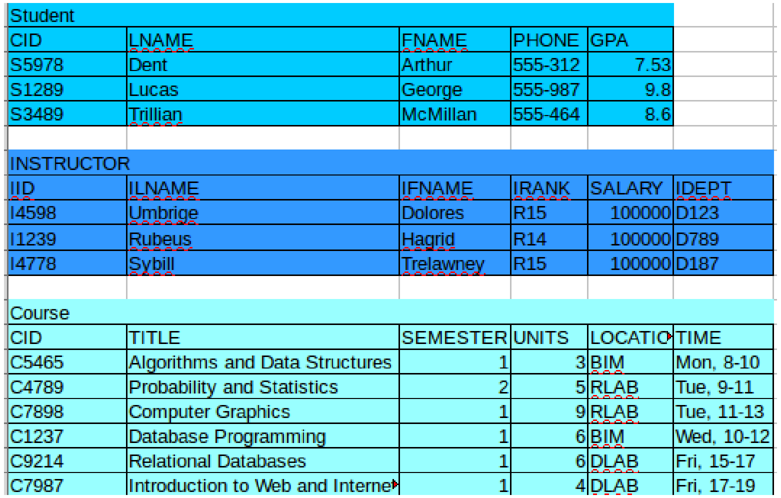
- Relational databases:
  - data are stored in tables
  - instance per row
- None SQL databases:
  - Instead of row, instance is represented as JSON (JavaScript object)



```
[
    {
        "CID": "S5978",
        "LNAME": "Dent",
        "FNAME": "Arthur",
        "PHONE": "555-312",
        "GPA": 7.53
    },
    {
        "CID": "S1289",
        "LNAME": "Lucas",
        "FNAME": "Georg",
        "PHONE": "555-987",
        "GPA": 9.8
    },
    {
        "CID": "S3489",
        "LNAME": "McMillan",
        "FNAME": "Trillian",
        "PHONE": "555-464",
        "GPA": 8.6
    }
]
```

```
[
    {
        "IID": "I4598",
        "LNAME": "Umbrige",
        "FNAME": "Dolores",
        "IRANK": "R15",
        "SALARY": 100000,
        "IDEPT": "D123"
    },
    {
        "IID": "I1239",
        "LNAME": "Rubeus",
        "FNAME": "Hagrid",
        "IRANK": "R14",
        "SALARY": 100000,
        "IDEPT": "D789"
    },
    {
        "IID": "I4778",
        "LNAME": "Sybil",
        "FNAME": "Trelawney",
        "IRANK": "R15",
        "SALARY": 100000,
        "IDEPT": "D187"
    }
]
```

Example of data stored in JSON notation



Student

CID	LNAME	FNAME	PHONE	GPA
S5978	Dent	Arthur	555-312	7.53
S1289	Lucas	George	555-987	9.8
S3489	Trillian	McMillan	555-464	8.6

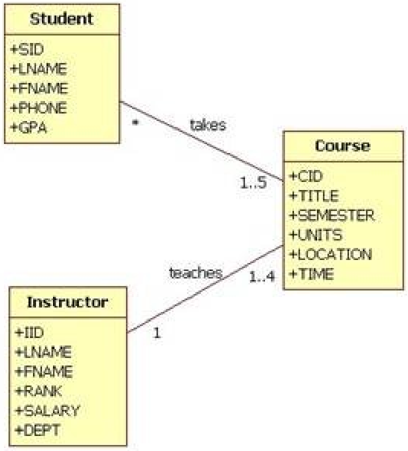
INSTRUCTOR

IID	ILNAME	IFNAME	IRANK	SALARY	IDEPT
I4598	Umbrige	Dolores	R15	100000	D123
I1239	Rubeus	Hagrid	R14	100000	D789
I4778	Sybill	Trelawney	R15	100000	D187

Course

CID	TITLE	SEMESTER	UNITS	LOCATIO	TIME
C5465	Algorithms and Data Structures	1	3	BIM	Mon, 8-10
C4789	Probability and Statistics	2	5	RLAB	Tue, 9-11
C7898	Computer Graphics	1	9	RLAB	Tue, 11-13
C1237	Database Programming	1	6	BIM	Wed, 10-12
C9214	Relational Databases	1	6	DLAB	Fri, 15-17
C7987	Introduction to Web and Interne	1	4	DLAB	Fri, 17-19

Example of a small database, visually



Relational database model

Relational vs. NoSQL databases (2)

- Relational databases are thoroughly covered within two more courses
  - Relational databases, 5th semester, 6 ECTS
  - Database programming, 6th semester, 6 ECTS
- Maybe replace MySQL with NoSQL database
  - e.g. Node.js compatible MongoDB

Relational vs. NoSQL databases (3)

- MongoDB mongoDB®
  - free and open-source cross-platform document-oriented database
  - JavaScript Object Notation-like documents with dynamic schemas
    - lightweight data-interchange format
    - language independent
    - "self-describing" and easy to understand
    - uses JavaScript syntax
      (but the JSON format is only a text, consisting of attribute–value pairs)
  - currently the most popular database for document stores

Conclusion

- Different methodical and technological aspects have been considered
  - outcome: no final answer so far
- The decision depends
  - of the content in other related courses
  - trend in Node.js/PHP popularity
  - trend in document databases popularity
  - needs of the market at the time being
  - teaching technology at the right time
  - students' desires

Conclusion (2)

- Democracy - “Two heads think better than one”
  - Students, who passed the exam, were asked about their opinion
  - they were offered
    - 6 client-side programming languages
    - 4 server-side programming languages
    - 5 databases
    - free options (to state some other languages)
  - still waiting to find out which skills related to Internet and Web technologies our students would like to gain!

Thank you!

- Contact
  - branislavas@matf.bg.ac.rs
  - vladaf@matf.bg.ac.rs
- Any questions?