WEB APPLICATION PROGRAMMING

Instructor: Yusuf Uğur SOYSAL
@n11.com

COURSE AIMS

This course will discuss the fundamental concepts of web-based systems, and design and programming techniques of these systems. Specifically, it will discuss the backend systems and application servers in great details.

SYLLABUS

» WEEK 1

- Introduction
- History of Web
- How Internet Works
- What is HTTP
- Web Application Design Patterns

» WEEK 2

- Java Recap
- Introduction to Spring
- Spring Modules
- Spring Boot
- JUnit

SYLLABUS

» WEEK 3

- Presentation Layer
  - HTML
  - CSS
  - Javascript
- Spring Views
- Template Engines

» WEEK 4

- More on HTTP
- Cookies and Sessions
- Application Servers
- Filters and Interceptors

SYLLABUS

» WEEK 5

- Database Integration
- What is ORM
- RDBMS or NOSQL
- Caching

» WEEK 6

- APIs
    - SOAP
    - REST
    - RSS

SYLLABUS

» WEEK 7

- User Accounts and Security

» WEEK 8

- Midterm

SYLLABUS

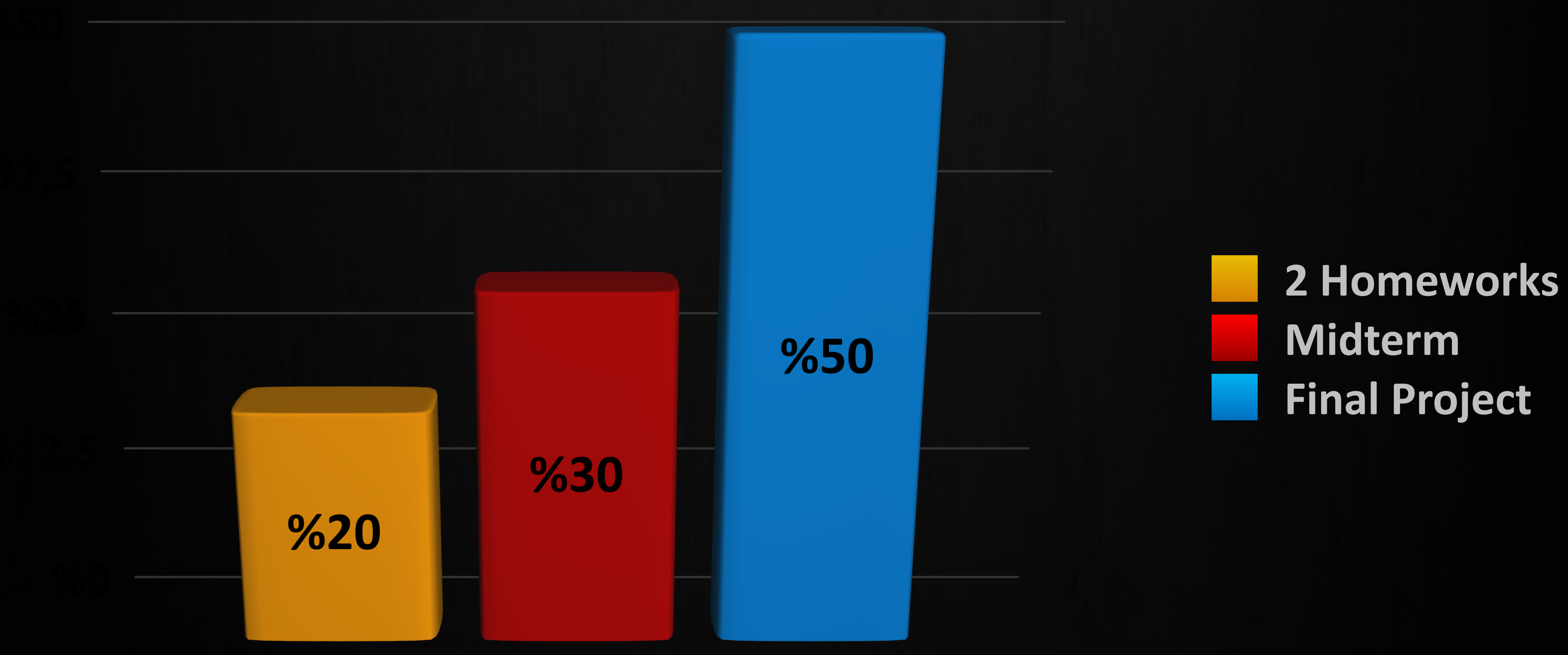
» WEEK 9-12

- Sample Project - Blog Engine

» WEEK 13-14

- Final Project Presentations

GRADING



Source Codes

https://github.com/n11com/WebApplicationDevelopment

HISTORY OF WEB

Why Web is invented?

For Emojis of course!   - emojitracker.com

www.internetlivestats.com

Brief history of web

1957 First remote connections to computers

1958 DARPA (Defense Advanced Research Project Agency) is founded

1966 ARPANET development began for easy knowledge transfer

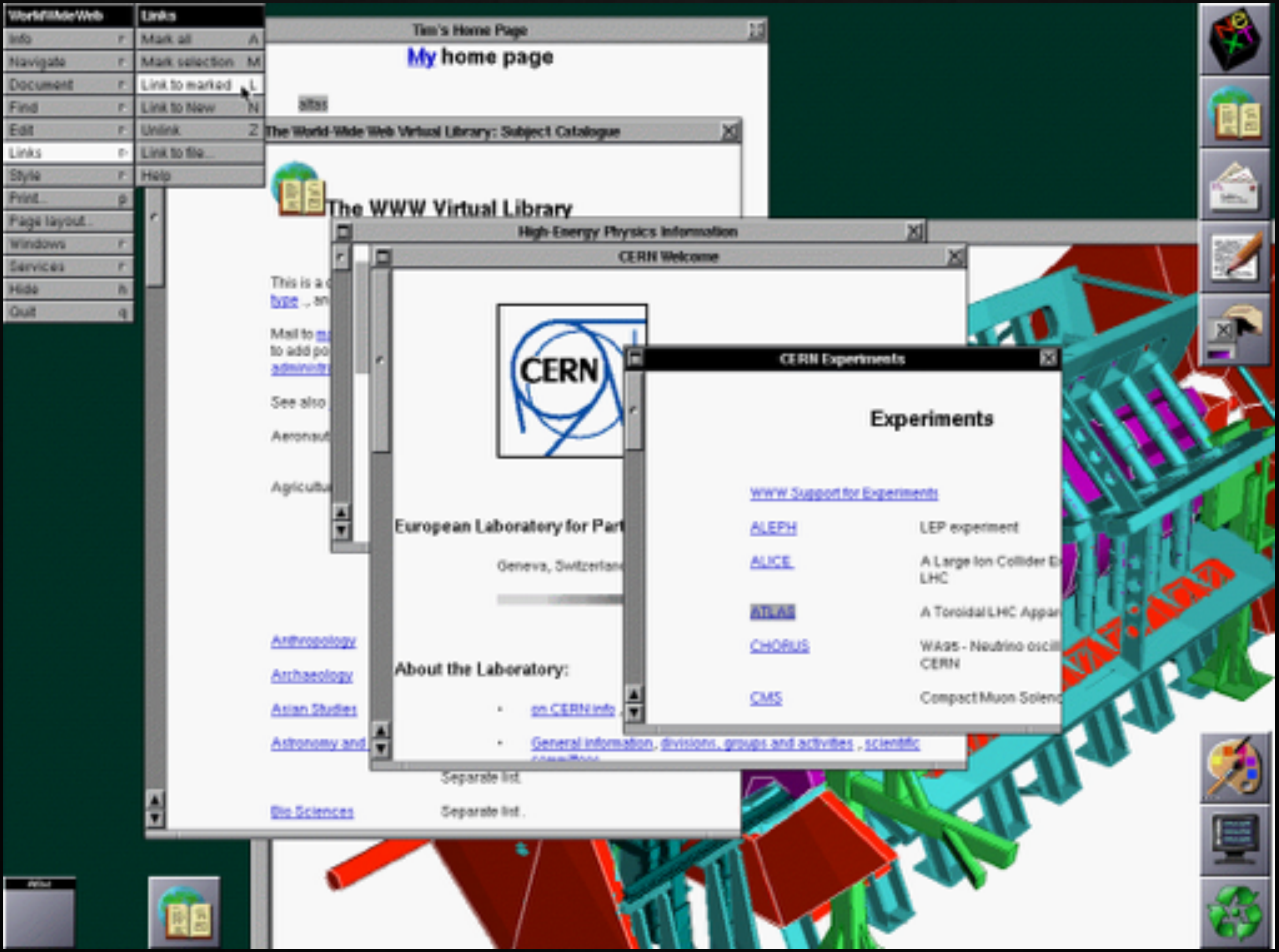
1969 ARPANET's first message from Los Angeles to Stanford

1972 Email is introduced

Brief history of web

1974	TCP is developed
1989	Tim Berners-Lee published “Information Management: A Proposal” and invents WWW in CERN
1990	TBL developed HTTP and created HTML
1990	TBL developed the first browser (WorldWideWeb.app) and HTTP server (httpd)
1990	Internet went public
1994	TBL founded W3C

First Browser



Tim's Home Page

My home page

The World-Wide Web Virtual Library: Subject Catalogue

The WWW Virtual Library

This is a
type ... an

Mail to
to add po
administr

See also

Aeronaut

Agricultu

Anthropology

Archaeology

Asian Studies

Astronomy and

Separate list.

Bio Sciences Separate list.

High-Energy Physics Information

CERN Welcome

European Laboratory for Part

Geneva, Switzerlan

About the Laboratory:

- on CERN info
- General information, divisions, groups and activities, scientific committees

CERN Experiments

Experiments

WWW Support for Experiments	
ALEPH	LEP experiment
ALICE	A Large Ion Collider E LHC
ATLAS	A Toroidal LHC Appar
CHORUS	WA95 - Neutrino oscil CERN
CMS	Compact Muon Soleno

Evolution of the web

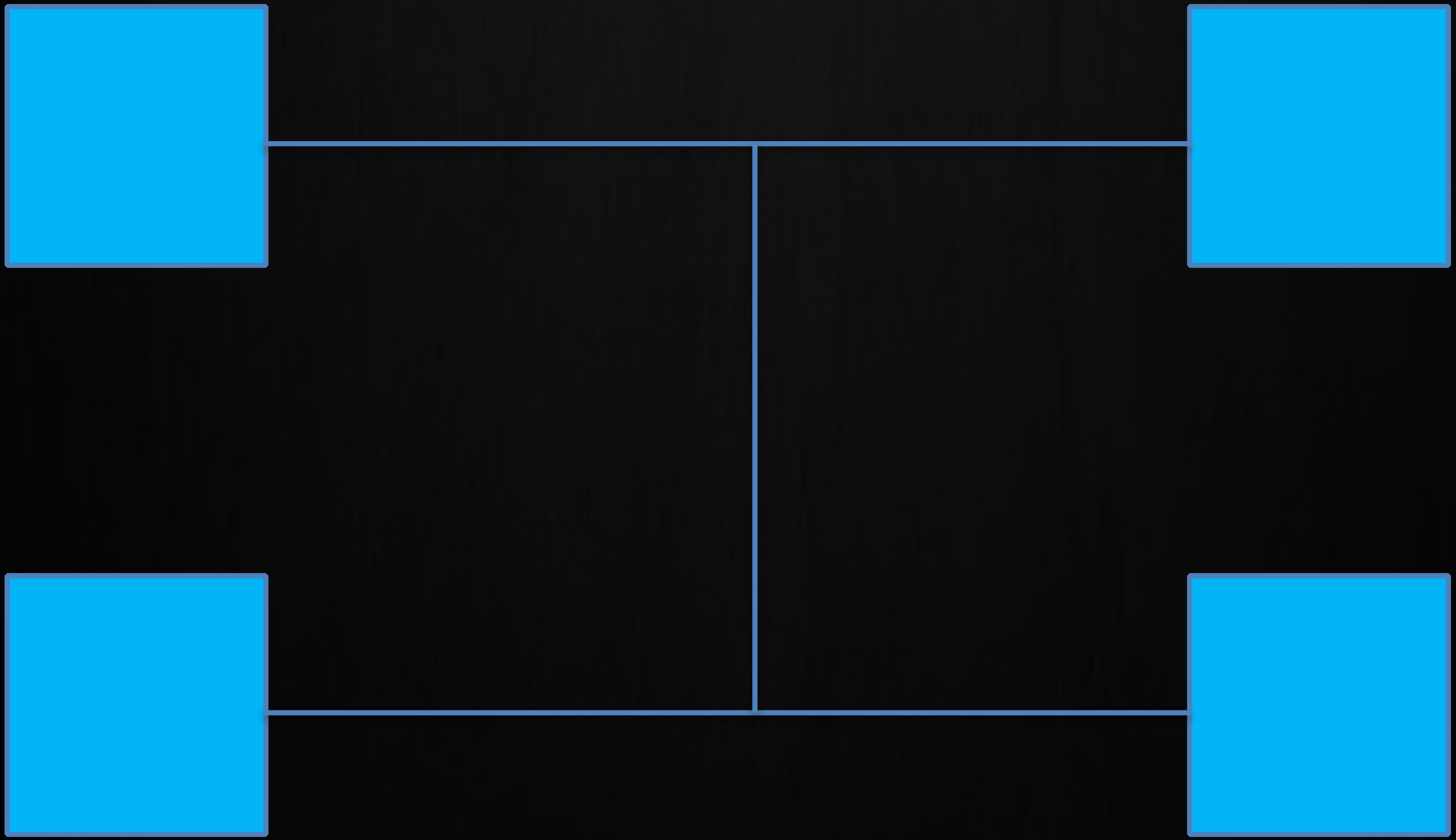
http://www.evolutionoftheweb.com/

HOW INTERNET WORKS?

How does the Internet works?

INTERNET => Interconnected Networks

Internet is a wire!

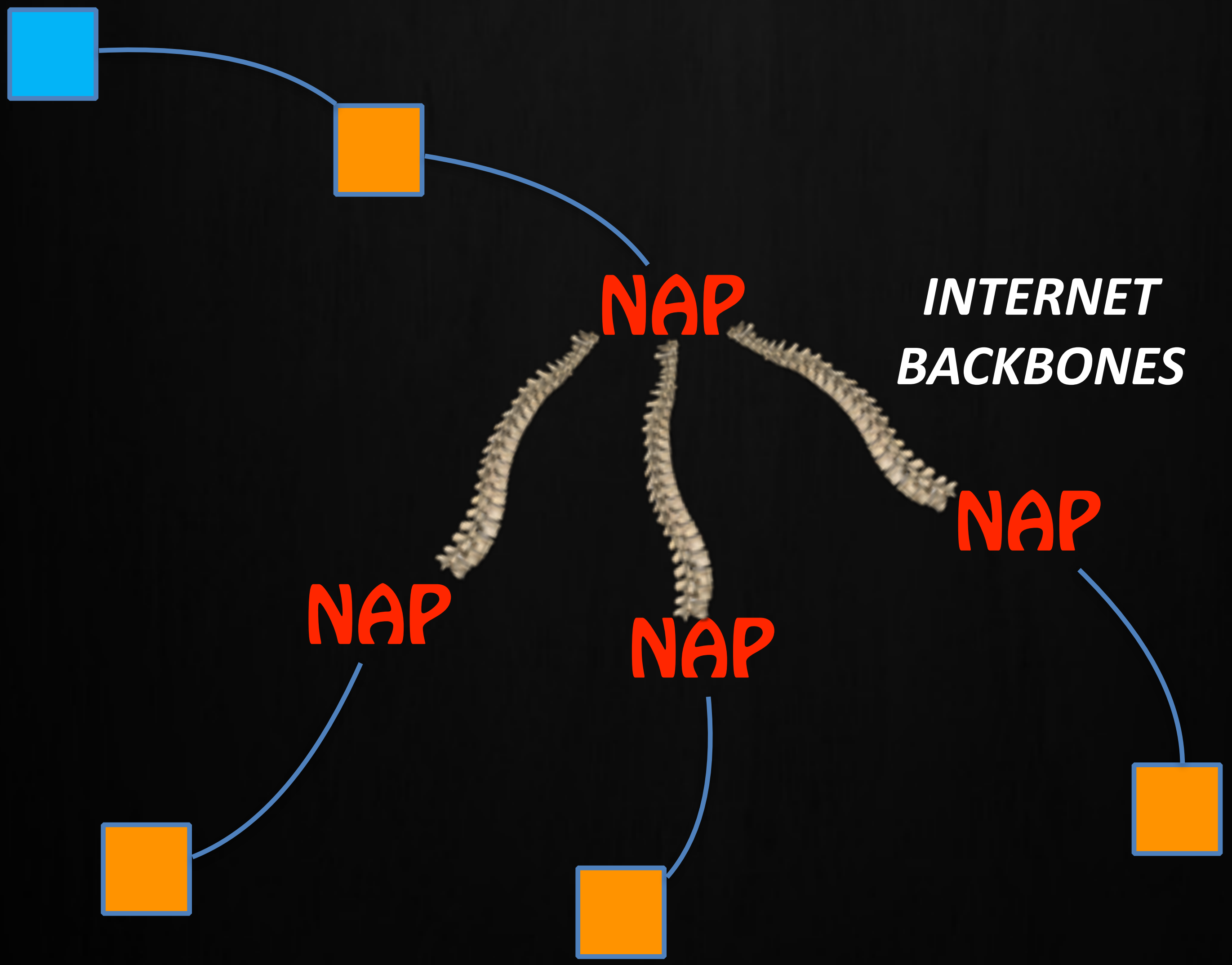


How does the Internet works?

For Internet to work, we need some key things

1. Rules for Packaging Data - HTTP
2. Interconnected Network
3. Route for Data

How does the Internet works?



How does the Internet works?

Data sent in short packets

Data is chopped along with assembly instructions

The packet’s destination address is an IP address

Packets are routed by “Routers”

HTTP

The Hypertext Transfer Protocol is an application protocol for distributed, collaborative, hypermedia information systems

Foundation of data communication for WWW

Functions as a request-response protocol

Presumes that the underlying and reliable transport layer protocol - TCP - though works with unreliable protocols as well - UDP.

Resources are identified and located by Uniform Resource Identifiers (URI)

HTTP is connectionless
HTTP is media independent
HTTP is stateless

HTTP - URI

Uniform Resource Identifiers are case insensitive

URI = "http:" "//" host [":" port] [abs_path ["?" query]]

HTTP - Message

- A Start line
- Zero or more header fields followed by CRLF
- An empty line (i.e., a line with nothing preceding the CRLF) indicating the end of the header fields
- Optionally a message-body

HTTP Methods

Defines methods to indicate the desired action to be performed

GET

Requests a representation of the specified resource

HEAD

The HEAD method asks for a response identical to that of a GET request, but without the response body. Useful for retrieving meta-information written in response headers

POST

Requests that the server accept the entity enclosed in the request as a new subordinate of the web resource identified by the URI

PUT

Requests that the enclosed entity be stored under the supplied URI.

DELETE

Deletes the specified resource.

HTTP Methods

Defines methods to indicate the desired action to be performed

TRACE

Echoes the received request so that a client can see what (if any) changes or additions have been made by intermediate servers

OPTIONS

Returns the HTTP methods that the server supports for the specified URL. This can be used to check the functionality of a web server by requesting '*' instead of a specific resource.

CONNECT

Converts the request connection to a transparent TCP/IP tunnel, usually to facilitate SSL-encrypted communication (HTTPS) through an unencrypted HTTP proxy

PATCH

Applies partial modifications to a resource

HTTP - Sample Request

```
GET /hello.htm HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE5.01; Windows NT)
Host: www.n11.com
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Connection: Keep-Alive


HTTP/1.1 200 OK
Date: Mon, 27 Aug 2015 12:28:53 GMT
Server: Apache/2.2.14 (Win32)
Last-Modified: Mon, 27 Aug 2015 12:05:56 GMT
Accept-Ranges: bytes
Content-Length: 88
Content-Type: text/html
Connection: Closed

<html><body><h1>Hello, World!</h1></body></html>
```

HTTP - Status Code

1XX: Informational
It means the request has been received and the process is continuing

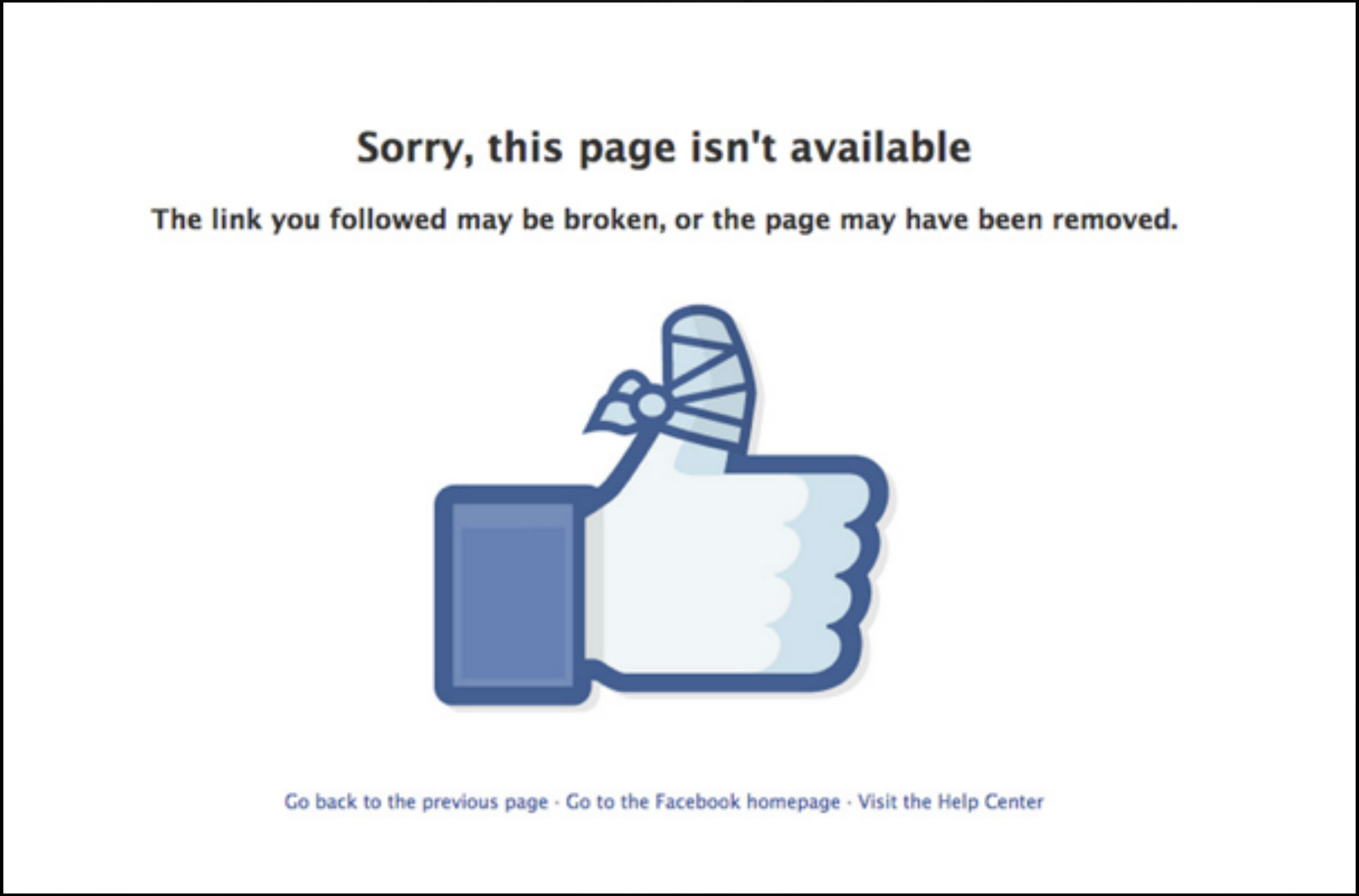
2XX: Success
It means the action was successfully received, understood, and accepted.

3XX: Redirection
It means further action must be taken in order to complete the request.

4XX: Client Error
It means the request contains incorrect syntax or cannot be fulfilled.

5XX: Server Error
It means the server failed to fulfill an apparently valid request.

HTTP - Status Code



Sorry, this page isn't available

The link you followed may be broken, or the page may have been removed.

Go back to the previous page · Go to the Facebook homepage · Visit the Help Center

WEB APP DESIGN PATTERNS

MVC

Model = what it is.
View = what it looks like.
Controller = what it does.

The view sits at the top of the architecture, the controller sits below the view. The model sits below the controller.

So the view knows about the controller, the controller knows the model. The view is notified when the model changes.

MVC

Request Based MVC: This is how Spring MVC and Struts works

Component Based MVC: This is how JSF and Pla! works

ANY QUESTIONS?
PING ME!

yusuf.soysal@n11.com