

Lecture 12

Implementing and testing web applications

Mr. Mubashir Ali

Lecturer (Dept. of Computer Science)

dr.mubashirali1@gmail.com

Summary of the Previous lecture

- **Software system architecture**
- **Specifics of web application architecture**
- **Layered web architecture**
 - **2-layered architecture**
 - **3-layered architecture**
 - **N-layered architecture**

Outline

- **Technologies for web applications**
- **Testing web applications**

1. Technologies for web applications

- **When we have decided the ‘What’ of the web application i.e.**
 - requirements are defined
 - system architecture is decided
 - system model and design is ready
- **We are ready for ‘how’ i.e. to implementation phase**

1. Technologies for web applications...

- The implementation phase begins with **deciding** the technologies for development
- Technologies for web application development **concerns** within three 'views'
 - request (client)
 - response (server)
 - rules for communication between them(protocols)

1.1 client/server communication on the web

- Client/server **paradigm** forms the backbone between the user and the application
- This communication model is based on **two-layer architecture**
- How ever the web server **integrates** additional systems i.e. database server, application server etc.
- Several **protocols** play an important role to guide this communication

1.1 client/server communication on the web

SMTP- simple mail transfer protocol:

- **SMTP** along with **POP3**(post office protocol) or **IMAP** (internet message access protocol) allows us to send email

RTSP- real-time streaming protocol:

- **Designed to facilitate delivery of multimedia data in real time**
 - allows transmission in timely manner instead of whole

1.1 client/server communication on the web

HTTP- hyper text transfer protocol:

- Most popular **transport** protocol for web contents
 - a text based **stateless** protocol
 - controls how resources are **accessed**
 - resources are addressed by **URL**
 - URL is used with **domain name system** to find the server where the resource is located

1.1 client/server communication on the web

Session tracking:

- Web applications must be able to **distinguish** requests by multiple simultaneous users
 - also need to **identify** request from the same user
- The term **session** is used to define a sequence of HTTP requests between a specific user and the server
- Whenever a user **sends** a request to the server, it identify itself with session id

1.2 Client-side technologies

Helper program and plug-in:

- Applications that can **add** functionality to browsers
- When the browser **receives** a media type included in the helper program or plugin list, the media file is **forwarded** to external program
- Installed by the user

1.2 Client-side technologies

Java applets:

- Java applets are programs written in **Java** that are loaded **dynamically** into the browser
 - have controlled access to system resources after checking security policies
- Applets are loaded by server and executed in browser within JVM
- Can run on all platforms with a JVM

1.2 Client-side technologies

Client side scripting:

- **Refers** to the class of computer programs on the web that are executed at client-side, by the user's web browser
- Usually **embedded** in HTML code
- Browser **interpret** several client side scripting
- Used to add **dynamic affects** in HTML page

1.3 Document specific technologies

HTML- hypertext markup language

- **HTML** describes the element
 - to mark contents
 - Hypertext
- Defines a large number of tags to denote different semantics

1.4 Server side technologies

URL handlers:

- special **applications** used to **process** HTTP requests and to **deliver** a requested resource
- Client request for a resource by **URL**
 - takes the request and forwards it for execution
 - result of this execution is then returned to the web server

1.4 Server side technologies...

Server side scripting:

- Are executed by the **web server** when the user requests a document
- Usually **embedded** in HTML code
- Server-side scripts require that their language's **interpreter** be installed on the server

2. Testing web applications

- Testing is an **activity** conducted to evaluate the quality of a product and to improve it by **identifying** defects and problems
- If we run a program with the **intent** to find errors, then we talk about testing
- By testing we determine the **quality state** of the system
 - which provides a basis for improvement

2. Testing web applications...

- We say that an **error** is present if the **actual** result from a test run does not comply with the **expected** result
 - each **deviation** from the requirements definition is an error

2. Testing web applications...

Objectives:

- Finding error instead of showing their absence (**defect testing**)
 - if no error is found it does not mean that there is no error
 - a test run is **successful** if errors are detected
- To demonstrate to the developer and the customer that the software meets its requirements (**validation testing**)

2. Testing web applications...

Testing Levels:

- **Unit tests:** test the smallest testable units (Web pages, etc.), independently of one another
- **Unit testing is done by the developer during implementation**

2. Testing web applications...

Testing Levels:

- **Integration tests:** evaluate the **interaction** between distinct and separately tested units once they have been integrated
- Integration tests are performed by a tester, a developer, or both jointly

2. Testing web applications...

Testing Levels:

- **System tests:** test the complete, integrated system
- System tests are typically performed by a specialized test team

2. Testing web applications...

Testing Levels:

- **Acceptance tests:** evaluate the system in cooperation with the client
- Acceptance tests use real conditions and real data
- **Beta tests:** let users work with early versions of a product with the goal to provide early feedback

2. Testing web applications...

Web application testing:

- **Link testing**
- **Browser testing**
- **Usability testing**
- **Load, stress and continuous testing**
- **Security testing**
- **Content testing**

2. Testing web applications...

Link testing

Goals:

- **broken links** (linked document does not exist)
- **orphan pages** (page does not link any other page)

Strategy:

- All links are systematically visited

2. Testing web applications...

Browser testing

Goals:

- Try to find **errors** in web application due to **incompatibilities** between different Web browsers

Strategy:

- Test application on all popular combinations (browser, version, operating system)

2. Testing web applications...

Usability testing

Goals:

- Evaluate ease-of-use, lay-out and navigation structure

Strategy:

- By a set of representative users
- By one or more HCI specialists

2. Testing web applications...

Load testing

Goals:

- system meets **response time** requirements

Strategy:

- Identify load profile
- Identify response time
- Perform the test

2. Testing web applications...

Stress testing

Goals:

- **system reaches the required response times and the required throughput under stress**

Continuous testing

Goals:

- **Testing system behavior over a period of time**

2. Testing web applications...

Security testing

Goals:

- **Regulate access to information, to verify user identities, and to encrypt confidential information**

Strategy:

- **A systematic test scheme**

2. Testing web applications...

Content testing

Goals:

- Test the quality of contents

Strategy:

- Proofreading

2. Testing web applications...

Challenges in web testing

- **Content testing requires costly manual measures**
- **Usability is difficult to measure**
- **Divers platforms (devices, operating environment)**
- **Globality (understanding cultural differences)**
- **Dominance of change makes is more challenging**

Summary

- **Technologies for web development**
- **Protocol**
 - client-side technologies
 - server-side technologies
- **Testing web applications**
 - Objectives
 - Levels
 - Web application specifics
 - challenges

References

- **Chapter 6,7**, Kappel, G., Proll, B. Reich, S. & Retschitzegger, W. (2006). **Web Engineering**, Hoboken, NJ: Wiley & Son