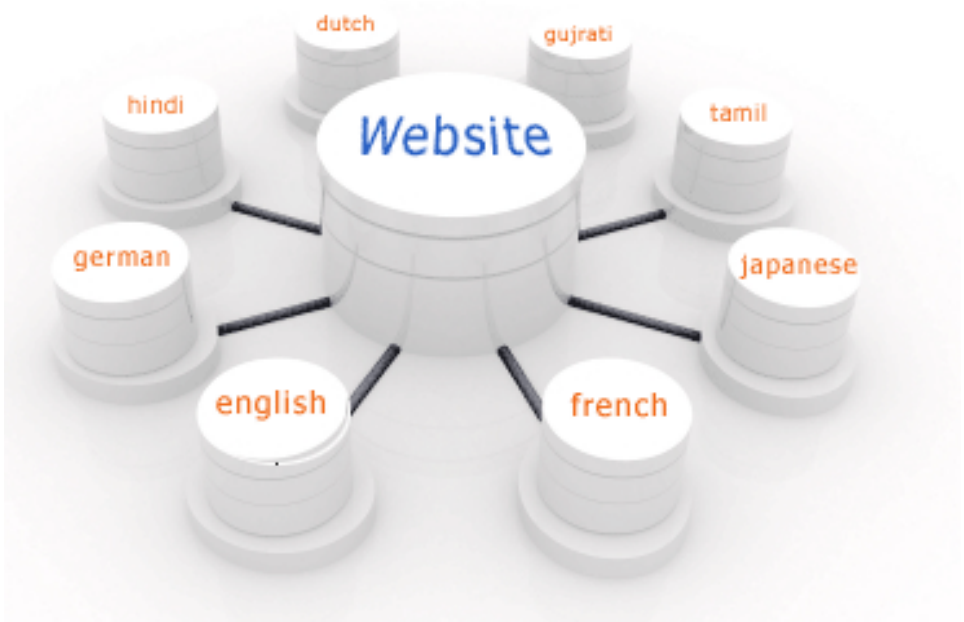
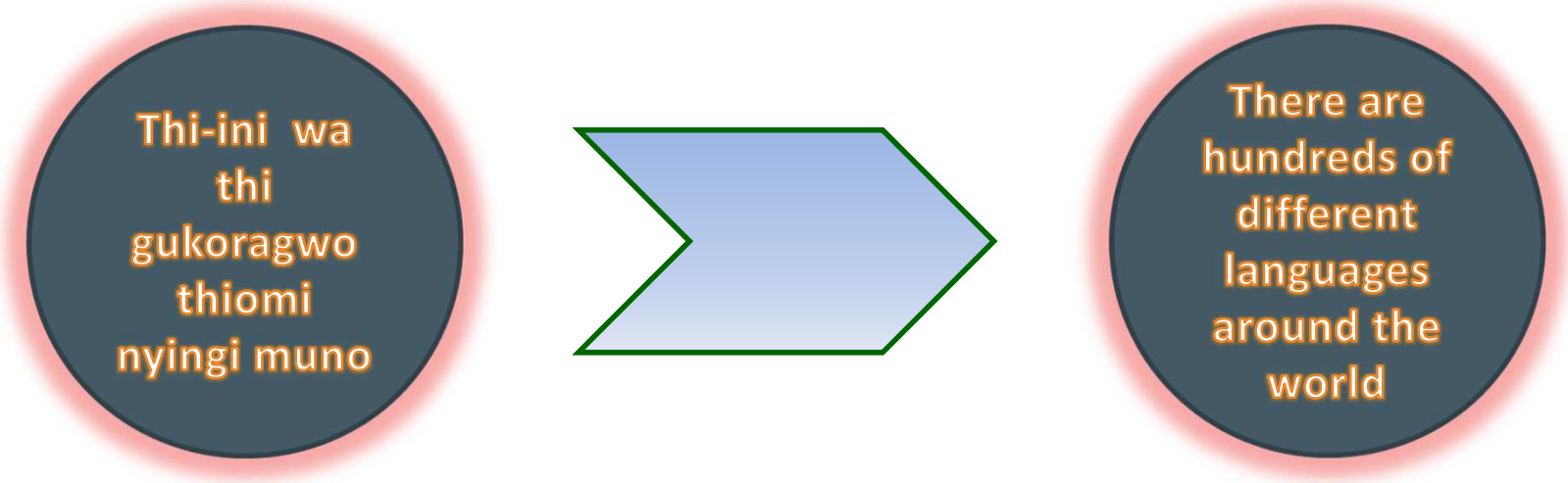
MIS6070

WEB BASED INFORMATION SYSTEMS



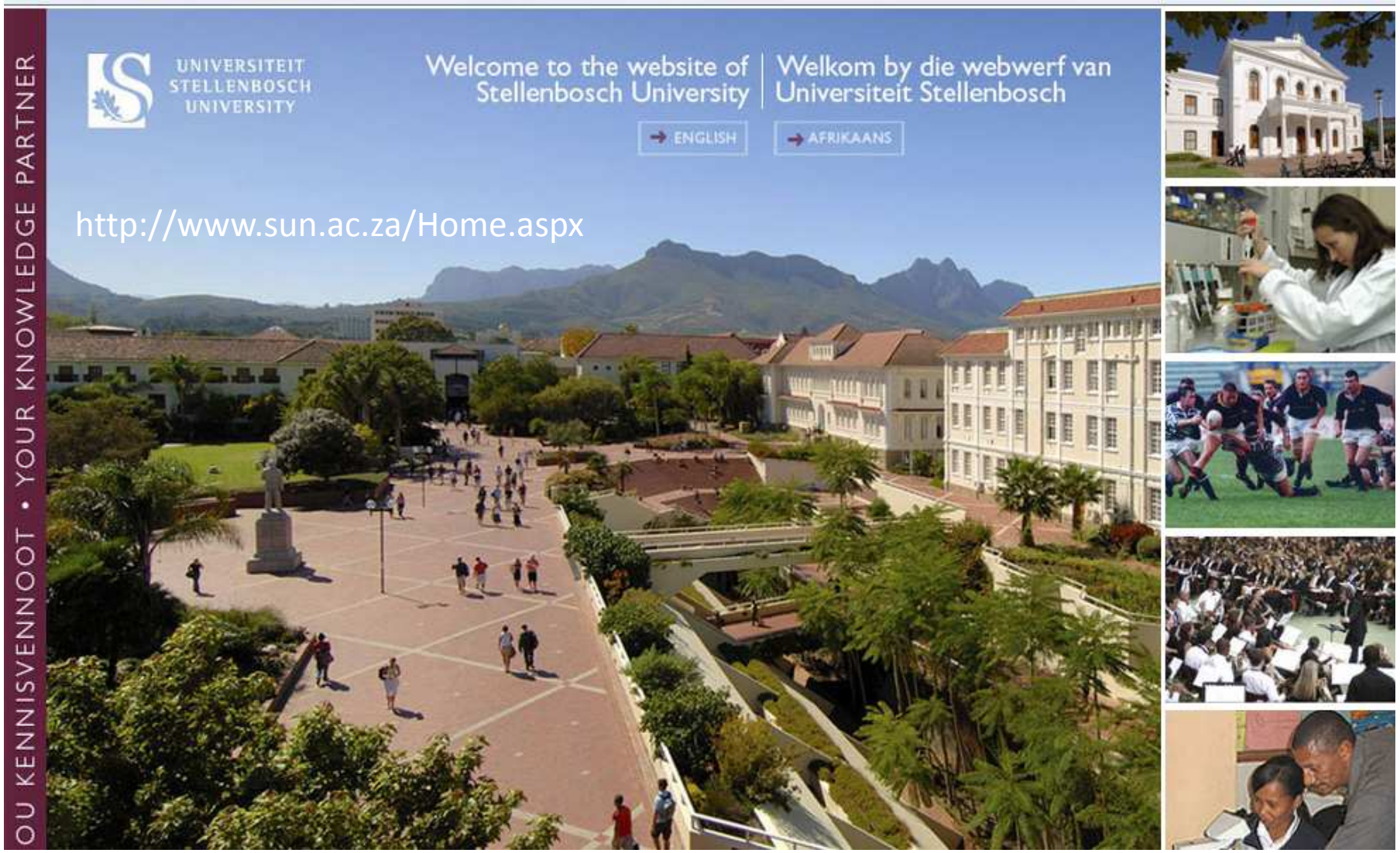
Lesson 4

Internationalization and Accessibility of Web Applications



1. For many people in the world who use a less widely spoken language, though, the Web is a foreign Place.
2. Yet Web is expected to be a global communication Medium.

Internationalization and Accessibility of Web Applications



Barriers to Web Access

- Not all web pages can be accessed by everyone

Disability

Culture, Language

Barriers to Web Access

- Being able to immediately share information around the world has opened new modes of operations for business, research, education, and vitually all the other human endeavours.
- It has not automatically **overcome the cultutural barries that make interpersonal communication difficulty**.
- We can easily access websites, but unless the information there is designed for **our culture and language, it has no value**

Barriers to Web Access

- Web users with disabilities also find a sort of cultural barrier
  - Many websites incorporates textual, visual and audio information and assume that users will be able to interact with web applications in standard ways such as using the keyboard and mouse.
  - Without these abilities is like trying to access a web written in a foreign language
  - Accessibility is the issue of how to make web content available to disabled users.
    - See Web Content Accessibility Guidelines http://www.w3.org/TR/WCAG

Barriers to Web Access

- To maximize availability and fairness, web sites can be easily enhanced for
  - **International access by people of different cultures and languages**
  - **Accessibility to people with various disabilities**

Internationalization and Localization

- Internationalization refers to **the process of designing a software application so that it can be adapted to various languages and regions without engineering changes.**
  - Activities included in internationalization include: use of character encoding that supports multiple languages, separation of localizable content from source code and providing support for localization by using languages, libraries, and style sheets that are designed for localization.
  - For Web applications, this is of particular importance because the **potential users may be from all over the world.**
- The process of preparing an application to SUPPORT MORE THAN ONE LANGUAGE and DATA FORMAT is called **INTERNATIONALIZATION**
- Process of ADAPTING an INTERNATIONALIZED application to support a SPECIFIC REGION or LOCALE is known as **LOCALIZATION**

Internationalization and Localization

- Many web sites need to be adapted to a local or global audience.
- This can be accomplished by internationalization and localization of the web site
- **Localization** is the process of adapting an application for a particular locale (to meet the needs of users in a specific locale
  - The following activities are included in localization: calendar, color, date and time, keyboard layout, language, measurements, number format, postal address, sorting sequence, symbols, and telephone numbers.
  - For example websites like www.hp.com, www.google.com, **provide options to the user to switch to a different locale** (**country + language combination**) **at runtime**.

Internationalization and Localization

- When internationalizing a web site, a language and country are usually used as a combination that is referred to as the **region**, **locale** or **culture**.
- It is removing barriers to global use of a web based application and hence made more accessible
- **Internalization** is accomplished during **design** while **localization** is accomplished during **development**, deployment and operation

Internationalization and Localization

- A **locale** is a set of parameters that define preferences based on
  - natural language
  - culture (often associated with a country)

for how information is presented to the users

- Preferences include spoken language, text presentation, data formats, etc.
- Locales are specified by combining two-letter codes
  - language: en, fr, sp, etc.
  - country: US, CA, FR, etc.
- For example
  - en_US English speaker, US
  - sp_US Spanish speaker, US
  - fr_CA French speaker, Canadian
  - fr_FR French speaker, France

Localization (L10n)

- **Localization** is the process of adapting an existing system for a new locale
- Adopting a software application or component to meet the needs of users in a specific locale.
- Localization is usually abbreviated as ***L10n***.
  - ***L10n***, in which 10 represents the number of omitted characters
  - **Example:** a website designed for use in the US by English speakers – can be expanded to the target market to include more of N. America's major populations
  - add:
    - Spanish language content for locale sp_MX
    - French language content for locale fr_CA

Elements of Localization

- Localization of an application can entail adapting any new number of locale-sensitive items including the following.
- **Calendar:**
  - The Gregorian calendar is used widely used in the world, but many people, especially in Asia and the Middle East, also use lunar calendars that have cultural or religious significance

Elements of Localization

- **Date Formats**
  - Gregorian date formats vary considerably around the world
    - MM/DD/YY, common in US or DD/MM/YY common in Europe or even YYYYMMMDD, etc., causes considerable confusion.
    - Month names may be used instead of numbers to remove positional ambiguity, for example 20th July 2012
    - However month names are also locale-specific and require more space to express
  - Month Names: January, Janvier, Enero, etc.
  - Era: BC/AD, BCE/CE
- **Time Formats**
  - Time may be represented using 12-hour or 24-hour clock
  - In the case of 12-hour format, the AM/PM, πμ/μμ, designators are important and also locale-dependent.

Elements of Localization Colors

- The cultural significance of colors is not universal.
  - Colors have different significance in different cultures
- For example:
  - Red: May signify danger, luck, purity, passion depending on locale
  - Green: associated with religion in some cultures, and is also becoming the symbol of environment awareness
  - White: may signify purity or death and mourning

Elements of Localization Language

- Preferred language varies with locale
- Preferred language is designated as part of the locale code.
  - This designation can be used within an application to choose appropriate content and to present appropriate messages and prompts.
- Language choice is dependent upon the character set being used
  - Unicode is the universal set
- Language also dictates character flow (left-right, up-down) to be used in output for the user

Elements of Localization
Numbers and Measurements

- **Numbers**
  - Format Variations include the decimal symbol, digit group (thousands) separator, and percentage symbol
    - Decimal format variations: 12,345.67 12.345,67 12 345.67
    - Currency symbols: $, £, ¥, €, etc.
    - Telephone number format: (123)456-7890, 12-34-56-78-90.
- **Measurements**
  - Common measurement systems include the imperial system : pound/gallon/foot/acre,
  - Metric System: kilogram/liter/meter/hectare
  - Each culture has its own preference

Elements of Localization
Postal Address

- Postal formats vary by
  - Placement of street number
  - Postal code size and placement
  - Spelling of country and city names

Mr. Henry Smith
Alpo Automotive, Inc.
447 Main St.
Yorktown, VA 55512
USA

Herrn Hans Schmidt
Alpo Auto GmbH
Humboldt Straße 337
48147 Münster DEUTSCHLAND

Wanjiru Wanjiri,
Kayole mwisho,
Nyuma ya Paradise bar,
Plot ya Othis hapo
opposite kwa chief

Elements of Localization
Sorting Sequence

- Sorting character strings is done with respect to the lexicographic ordering of characters
- Ordering of alphabets can vary by culture, but however it is dependent on locale.
  - In German, the letter *Ä* is between A and B while in Swedish, *Ä* follows Z

Germany	Sweden
Adams	Adams
Ångstrom	Wegner
Äthiopien	Voelker
Voelker	vonNeumann
vonNeumann	Ångstrom
Wegner	Äthiopien

Elements of Localization
Keyboard

- Keyboard layout:
  - Various keyboard layouts are available to support character sets for different languages.
- Software that directly handles keyboard input (Typically found in an operating system) must be adapted to the specific keyboard in use.

Internationalization

- Internationalization is the process of designing and developing a web application so that it can be easily localized.
- Ideally, localization should be achievable simply by adding appropriate locale parameters to an existing application without changing its design or coding.
- Internationalization is usually abbreviated as ***I18n.***
  - Where 18 represents the number of omitted characters

Internationalization

- Some of the aspects from application internationalization perspective
  - Platform Independence:
    - An internationalized software product should be able to operate in any locale on any operating system (Windows, Linux, Solaris etc) and should deliver consistent results.
  - Multiple Locales on same installation:
    - An internationalized product should be able to support multiple locales in a single installation by providing options to the user to switch to a different locale (country + language combination) at runtime

Internationalization

- **Different timezones:**
  - An internationalized product should be able to deal with different timezones in a consistent manner.
  - It should be able to store and retrieve the time in multiple timezones and display the correct time as per the selected locale.
- **Help and Documentation:**
  - The product should be able to display the Help documentation as per the language in the selected user locale.
- **Images/Audios/Videos:**
  - The product should be capable enough to pick the correct image/audio/video clips based on the user locale

Internationalization

- **Bi-directional Support:**
  - Product should also support bi-directional languages.
  - Most of the languages across the globe are written left to right (LTR), while there are some scripts which are written from right to left (RTL) (Urdu, Arabic, Hebrew for example).
- **Ease of adding new Language:**
  - Product design should be such that support for any new language can be added without any code/design changes and recompilation.

Internationalization

- How can web application designers and developers internationalize a web application. **3 simple ways**
  - Use a character encoding that supports multiple languages.
    - The ISO Universal Character Set (UCS) is a standard character encoding that supports almost all of the world's living languages.
    - UCS should be incorporated into Unicode which also includes protocols for displaying text properly

Internationalization

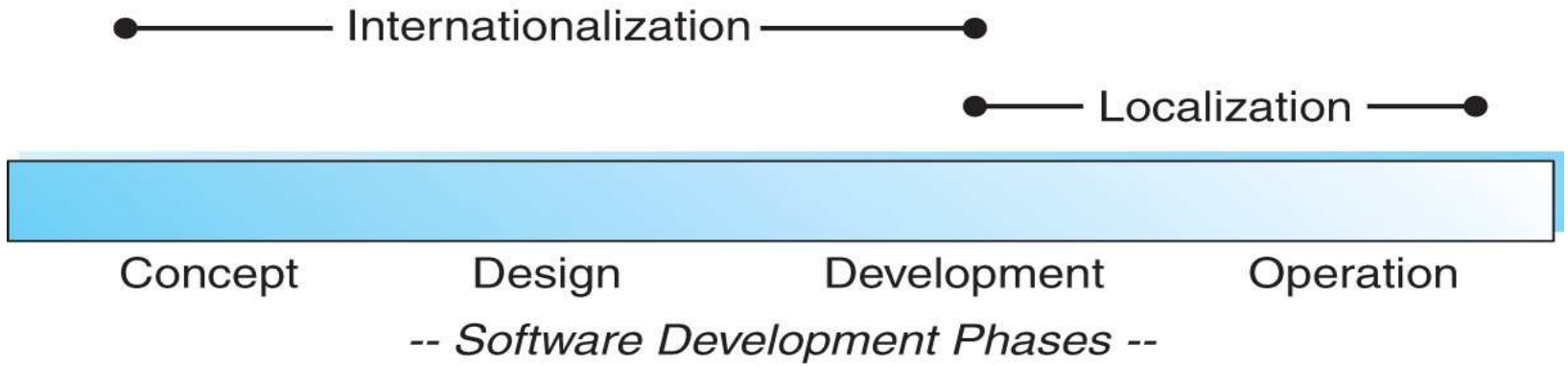
- Separate localizable content from source code.
  - Content, prompts, error messages, and dates that are localizable should be obtained by the software from the appropriate files or databases and not hard-coded into the software
- Provide support for localization by using languages, libraries, and style sheets that are designed for localization.

Internationalization

- Internationalization and Localization occur at distinct points in the Web Application software development life cycle.
  - Internationalization is accomplished primarily during software design, whereas localization is accomplished during development, deployment, and operation.
  - I18n therefore proceeds and enables L10n.

Internationalization

- Internationalization is the responsibility of the designers and the initial developers.
- Localization is the responsibility of developers and administrators, and may be performed throughout the application's lifetime.
- The figure below illustrates the relationship between the two.



Internationalization

- HTTP includes two headers for establishing the locale used in an HTTP transaction.
- First, the client can request the locale to be used as the bases for the transaction with an ***Accept-Language*** request header.

Accept-Language: en-us, en; q=0.7, de-de; q=0.3

- The list following the header name specifies the requested locales and numerical preference for each
  - Locales *en-US* or *en* are the first priority, with a preference of 70% and the locale *de-DE* is second with a preference of 30%. The preference levels do not have direct impact on the server's response.

Internationalization

- They simply provide an indication of the user's preference with respect to language, which the server can interpret in any way.
- The server can then respond with a ***Content-Language*** header specifying the actual language used in the response.

Content-Language: en

Internationalization with Java

- Resources are available in Java for localization
  - The *Locale* object in Java encapsulates the language and culture of a user.
  - *Locale resource bundles* are used to generate locale-specific dynamic content.
  - NumberFormat and DateFormat classes allow numbers and dates to be represented according to locale.

Internationalization with Java

- Locale
  - Localization in Java depends upon the use of objects of the *Locale* data type, which define locale features;
    - primarily language and country code
  - Locale objects can be used with other Java Classes (such as the *date*) to obtain localized presentations, and can be used to obtain application-specific localized content as well.

Internationalization with Java

- Each JVM has a default locale, which can be obtained by

  Locale myLocale = Locale.getDefault();

- Locale objects for specific locales can also be created using a locale constructor.

  Locale myLocale = new Locale(“fr”, “CA”);

  Locale myLocale = new Locale (“de”);

- The first constructor creates a Locale object appropriate for Francophones in Canada, the second creates a Locale object for German speakers, regardless of their location.

Internationalization with Java

- Lists of currently supported languages and country codes can be obtained from the Locale class with

```
String[] languages=locale.getISOLanguages();
String[] countries.getISOCountries();
```

Internationalization with Java

- Resource Bundles
  - Program elements such as prompts, labels, images, and so on, that are locale-specific can be localized as Java Resource Bundles.
  - A bundle of prompts, labels, and messages in the form of stings can be most easily defined as an instance of the Java Class ***PropertyResourceBundle***, via a text file.
  - Resource Bundle: a collection of messages, images, etc. that are particular to a locale or set of locales
    - **DateFormat:** formats date and time for a locale
    - **NumberFormat:** formats numbers for a locale
    - **Collator:** supports locale-sensitive sorting

Internationalization with Java

- Example of Resource Bundle

<h3>Please Login:<h3>

<form action="/serveletlonginController" method=''post">

User Id: <input type="text" name="userid" />

 Password: <Input type="password" name="passwd" />

<input type="submit" name="login"
Value="login"

- In order to internationalize this form, the **heading and labels** (User Id, Password, and Login) must be set up so that they can **be localized** while User-id, password, login, **need not be localized**, as they are internal to the program and are never seen by the user

Locale Class

- Locale locale = Locale.getDefault();
- Resourcebandle labels=resourcebundle.getbundle("loginlabels", locale)
  - gets the default locale for the JVM (server)
- Locale loc = request.getLocale()
  gets the locale of the client submitting an HTTP request
- Labels.getstring
  - Returns the appropriate string

Date / Time Formats

- Three classes are necessary, the ***calendar***, ***date*** and ***date format***
- Localized date display

```
Locale loc = request.getLocale();
Date day = Calendar.getInstance.getTime();
DateFormat df = DateFormat.getDateInstance(style);
String dateOut = df.format(day, loc);
```

- Localized time display

```
DateFormat tf = DateFormat.getTimeInstance(style);
String timeOut = df.format(day, loc);
```

style = DateFormat. [SHORT | MEDIUM | LONG | FULL]

Number Formats

- The class ***NumberFormat*** can format ordinary numbers, monetary values, and percentages according to the customs of a locale.
- Localized number / currency formats

```
double value;
Locale loc = request.getLocale();
NumberFormat nf = NumberFormat.getInstance(loc);
String number = nf.format(value);
```

- Currency Formatting

```
NumberFormat cf =
   NumberFormat.getCurrencyInstance();
```

Resource Bundle

- A resource bundle is a collection of name/value pairs that can define labels, prompts, messages, file names, etc., that are specific to a locale
- Resource bundle contents are used to localize document elements

Resource Bundle Definition

- One way to define a resource bundle is as a properties file.
- File name format:
  <bundle name>_<locale name>.properties
  e.g., MessageBundle_de_DE.properties
- File contents:
  - comments: # this is a comment
  - Name/value pairs: greeting = Hello!

Resource Bundle Location

- Given that the current locale is LL_CC, the loader will search for the best match:

1. MessageBundle_LL_CC.properties
2. MessageBundle_LL.properties
3. MessageBundle.properties

Character Sets

- A character set is a set of scripts (letters) that can be used to write text
- Some sets (e.g., ASCII, EBCDIC) support a few languages only
- Unicode is the character set that supports all of the world's written languages

Character Encodings

- A character encoding provides an efficient numerical representation for the characters in a character set
  - IS0-8859 is an encoding that supports scripts used in European languages
  - UTF-8 is an encoding for Unicode
    - UTF-8 stands for **U**nicode**T**ransformation**F**ormat-**8**. It is an octet (8-bit) lossless encoding of Unicode characters.
- Encoding declaration:

  <?xml version="1.0" encoding="UTF-8"?>
  - Tells the client how to interpret the encoding

Sorting (Ordering)

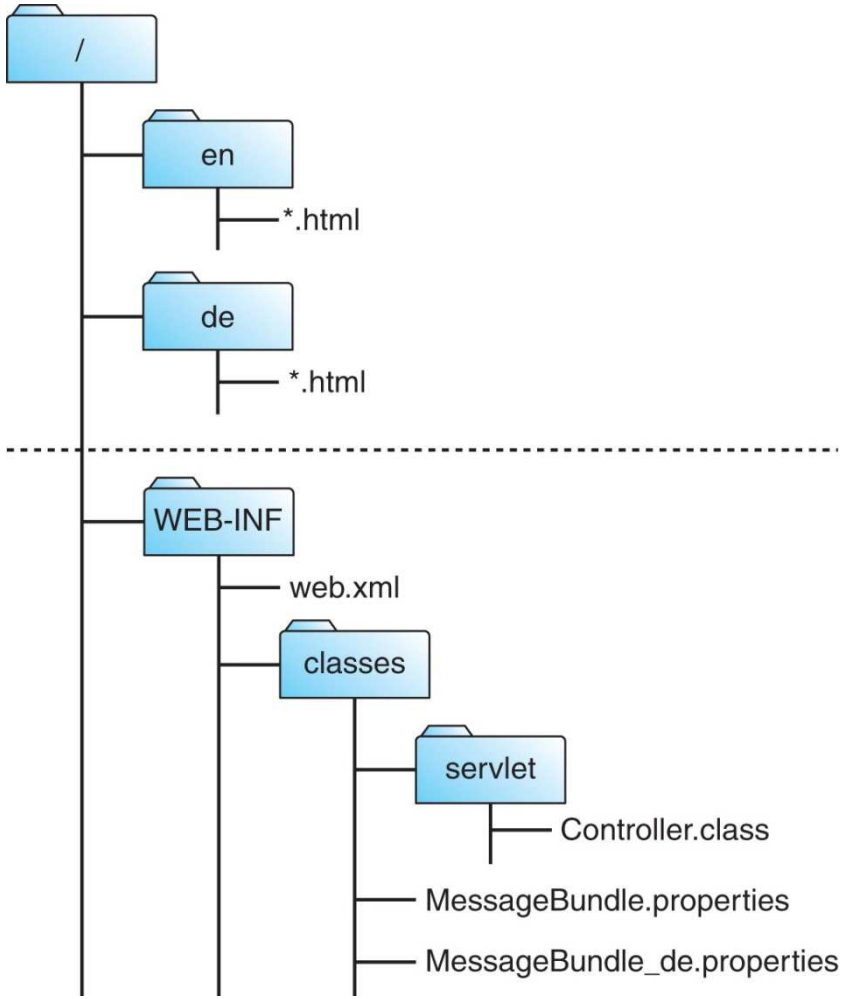
- To obtain a locale-specific ordering of strings, use the Collator class

```
Locale loc = request.getLocale();
Collator col = Collator.getInstance(loc);
…
if (col.compare(string1, string2) < 0) …
```

< 0	string1 < string2
= 0	string1 = string2
> 0	string 1 > string2

Webapp Design for I18n

- A convenient way to organize localized content is by separate packages (en, de, etc.)
- The Controller decides which package to use based upon the client's locale



Accessibility Problems and Solutions

- Many web users have disability problems that make access difficult
  - vision disability
  - hearing disability
  - physical limitations
- Both social responsibility and good business practices require that these users be accommodated
- The W3C Web Accessibility Initiative (WAI) has developed Web Content Accessibility Guidelines (WCAG) for developing accessible websites

Barriers to Web Access

- **Visual Disabilities**
  - Visual disabilities can range in intensity from colour blindness (difficulty in distinguishing colour) to impaired vision or total blindness.
- **Auditory Disabilities**
  - There are a number of auditory disabilities ranging from mild hearing problems to partial or total deafness.
- **Language-related Disabilities**
  - Speech impediments indicate difficulties in producing understandable language.
    - Language-related disabilities usually do not cause serious web access problems because access to web content via voice is rather uncommon.

Barriers to Web Access

- **Cognitive and Neuro-sensorial Disabilities**
  - Cognitive and sensorial disabilities include a range of limitations such as dyslexia, attention span disorder, limited intelligence, limited memory, mental health problems, epilepsy, and others.
- **The concept of assistive technology**
  - Assistive technologies are tools which enable a person with a disability to carry out the same tasks as a person without any type of disability.

Barriers to Web Access

- All **assistive technologies** should have the following characteristics:
  - Simplicity: they should be easy to use;
  - Efficacy: they should offer a suitable solution to the problem;
  - Propriety: they should be used when there is no other reasonable means by which to solve the problem.

Barriers to Web Access

- **Assistive Technologies for Web access**
- **Screen readers**
  - Screen readers are specialised software which enable blind users or those with severe vision impairment to use a computer. The software is used in combination with hardware (speakers, voice synthesizers, etc.) to interpret text and images appearing on the screen and transform them into voice.
    - **JAWS** is one of the most popular screen readers on the market today, providing access to software applications as well as the Internet. It also provides support for Braille devices and comes with an integrated voice synthesiser.

Barriers to Web Access

- **Assistive Technologies for Web access**
- **Speaking Browsers**
  - A speaking browser is software enabling blind or visually impaired users to browse through web content expressed in voice form.
  - Speaking browsers are somewhat limited compared to screen readers as they do not guide the user in the use of the operating system and they lack advanced accessibility options such as the reading of complex tables, location of headers, lists, etc.
    - IBM Home Page Reader is a speaking browser which reads the information on a webpage aloud, including conventional text, tables, graphic descriptions, fields and forms for the input of data, etc.

Barriers to Web Access

- **Assistive Technologies for Web access**
- **Screen magnifiers**
  - Screen magnifiers are software tools which considerably enlarge the size of information appearing on the screen thus facilitating access for visually impaired users.
  - Additionally, they have the ability to adjust contrast, colour, font types and sizes.
    - Magic enlarges screen content up to 16 times its original size and includes an optional screen reader.
    - It is also equipped with a voice synthesiser making it possible to read any text appearing on the screen.

Barriers to Web Access

- **Assistive Technologies for Web access**
- **Text-only browsers**
  - A text-only browser enables the user to navigate through web content, displaying all information in text format (e.g. through disabling style sheets and displaying text alternatives instead of images).
    - Lynx is the most popular text-only browser worldwide.
      - http://lynx.browser.org/
    - It is freely distributed and can be downloaded automatically.

Barriers to Web Access

- **Assistive Technologies for Web access**
- **Voice recognisers**
  - Voice recognisers are information input devices used by persons who have difficulties using a keyboard or mouse. The voice recogniser converts the user's voice into commands, which are sent to the operating system via a microphone, taking the place of mouse movements or keyboard strokes.
  - The supply of voice recognisers is limited. Below is a description of the most frequently used:
    - Dragon Naturally Speaking is a voice recogniser with a high voice to text conversion precision rate. Complex tasks can be conducted (activation of a link, sending of forms, etc.) with a simple word or phrase.

Barriers to Web Access

- **Assistive Technologies for Web access**
- **Braille display**
  - A Braille display is a hardware device which, in combination with specific software, displays screen content in Braille characters, enabling blind and deafblind users to access information on the computer screen.
  - The system interprets the words or graphics on the screen (similar to screen readers) and translates them in Braille.
  - Braille displays offer some screen movement options and even command activation, although the input device is the conventional keyboard.

Ten Quick Tips for Accessibility

1. Provide alt descriptions for images, for that visually disabled users

2. Client-side image maps (clickable maps) are more user friendly for disabled users than server-side maps. Also include text descriptions of "hot spots"

<map name="citymap">

<area shape="rect" coords="0,0,100,100"

href=“KCAneighbourhood.htm" target="_blank"

alt=“Mathare 4a" />

Ten Quick Tips for Accessibility

3. Provide captioning and transcripts of audio, and descriptions of video for users with vision or hearing disabilities.
4. Write hypertext links that make sense when read out of context. Instead of this:

   Click here **to learn more about the history of Europe.**

   create the link from descriptive text:
   Learn more about the *history of Europe* here.

Ten Quick Tips for Accessibility

5. Make appropriate use of page structure and layout elements.
   - Don't use tables for page structure
   - Don't use headings for highlighting
6. Include a summary description of graphs or charts in a document.

Ten Quick Tips for Accessibility

7. Browsers for disabled users might not support content in proprietary formats (flash, applets, PDF, etc.).
   - Provide alternative content or descriptions.
8. Documents that use frames for structure should include the <noframes> element, which provides an alternate structure for browsers that do not support frames.

Ten Quick Tips for Accessibility

9. Using table summaries and headings to provide better descriptions of HTML tables

```
<table summary="population growth" ...>
  <caption>Population Growth</caption>
  <tbody>
  <tr><th>Year</th><th>Population</th></tr>
  <tr><td>1900</td><td>580,000</td>
  ...
```

10. Use a validation checklist, inspections, and automated tools to ensure that content meets accessibility guidelines.

Performance and Reliability of Web-based Applications

Performance and Reliability

- What features of a web application are most likely to attract users and maintain their satisfaction level?
  - New and interesting content,
  - Lower prices, and
  - Better services.
- But maintaining customer satisfaction requires that the site also provide
  - a **reliable** and **fast** response

Performance and Reliability

- Typical users will wait no more that a few second for a website to respond before moving to something else
- Users also expect **100% availability** from websites and **won't tolerate frequent outages**
- Adequate capability and reliability in a web application depend to a larger extent on:
  - Proper design and
  - Correct implementation

Performance and Reliability

- However, it is impossible to predict exactly how well an application will perform in practice, and some adjustments may be necessary during implementation
- Ongoing improvements and maintenance are also required in order to maintain adequate performance levels.

Performance and Reliability

- **Performance measurement** is the process of collecting and analysing data that relate to how well a web application or web server provides service.
- Accurate performance measurement is critical to maintaining adequate levels of service and therefore fulfilling the business objectives of a web application
- Performance measurements may be used for other purposes as well
  - Setting marketing strategies
  - Billing clients of an internet service provider

Performance and Reliability

- There are many different measures of service level, and the usefulness of each depends upon the purpose of measurement.
- Definitions of service measures also vary and meanings can be ambiguous.
- The following measurements are the **most widely used** and are **relatively well-defined**
  - Throughput
  - Workload
  - Response time
  - Latency

Performance and Reliability

- **Throughput:** The measure of the number of transactions or data units *(depending on context)* processed per unit time.
  - HTTP requests handled per minute would be appropriate measure of webserver or web application throughput
  - Throughput can also be used as a measure of network traffic, such as bits per second or packets per second processed by a network component
- **Throughput:** the number of **transactions processed per unit of time**
  - e.g., 1000 transactions/minute. What is the M-Pesa throughput?

Performance and Reliability

- **Workload:** The measure of the number of transactions or data units that **NEED** to be processed per unit time.
  - Ideally, all available work will be processed.
- This will happen as long as a server is operating within its capabilities and throughput in that case will be equal to workload.
- When the server capacity is exceeded, however, throughput will be less than the workload which means that some transactions will not be processed.
  - This might happen for a brief instant, due to peak workload that exceeds capacity, or it might be a chronic problem due to server that is poorly configured or underpowered for its workload.

Performance and Reliability

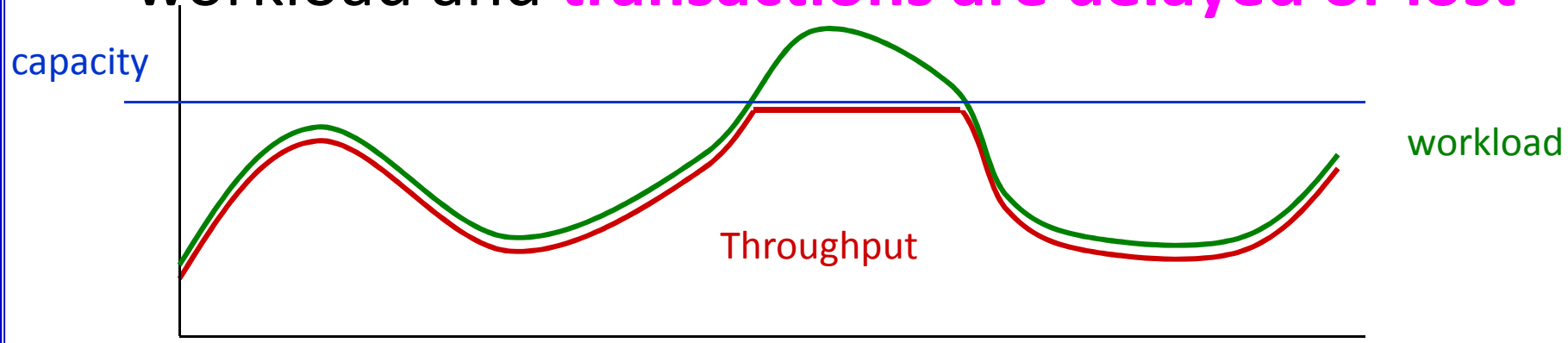
- Example:
  - A server receives 500 transactions / minute, but is able to process only 80% of that load
    - workload: 500 transactions / minute
    - throughput: 400 transactions / minute
- Which of these systems is/are failing to satisfy its workload?
    - Workload = 120,000 transactions/hour
    - Throughput =110,000 transactions/hour

    - Workload = 27,000 transactions/hour
    - Throughput = 27,000 transactions/hour

    - Workload = 396,000 transactions/hour
    - Throughput = 390,000 transactions/hour

Throughput and Workload

- Ideally, throughput = workload
  - Throughput can be less than workload but normally, throughput will equal workload
- When workload exceeds capacity, throughput < workload and **transactions are delayed or lost**



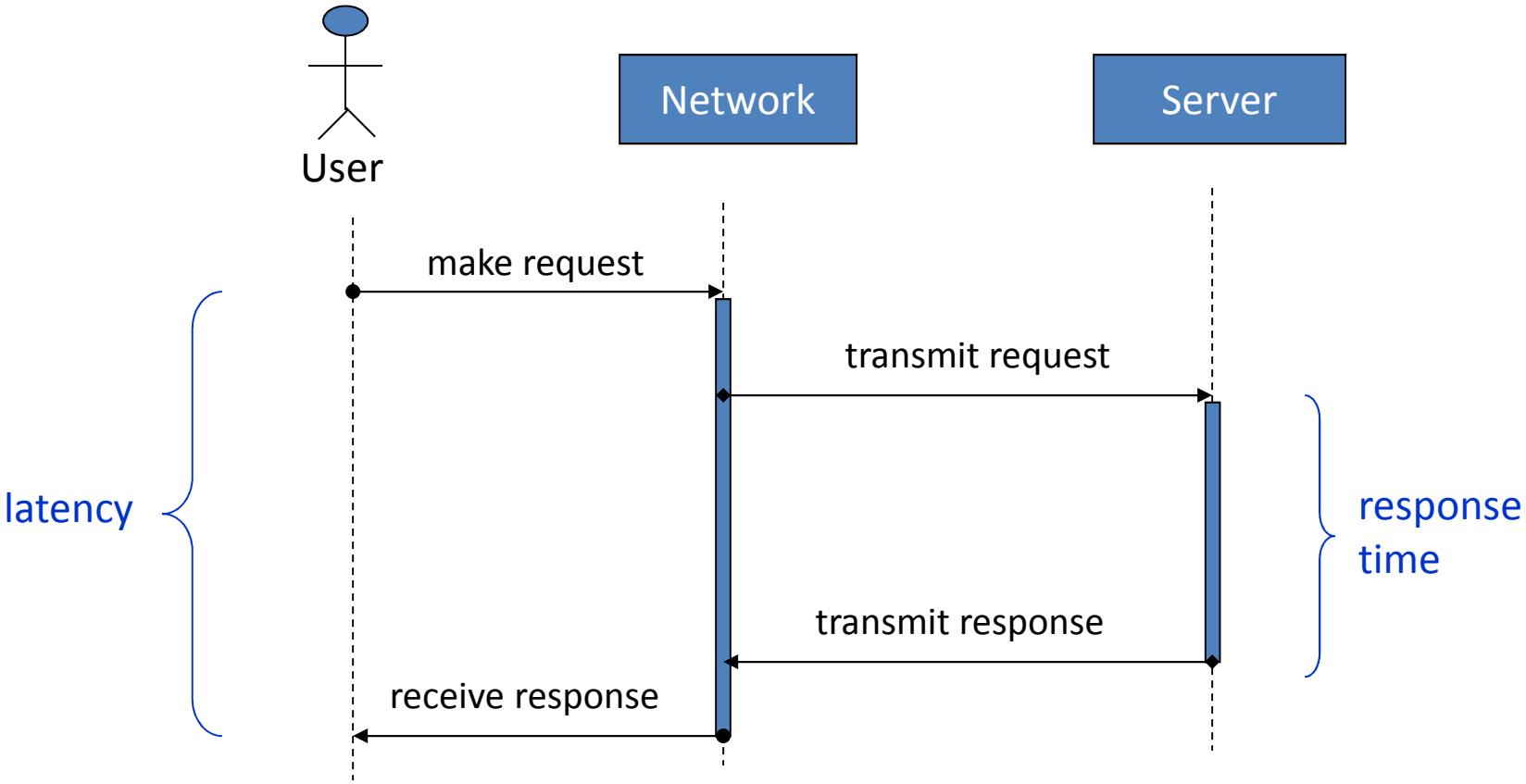
Performance and Reliability

- **Response time:** The time required to process a single transaction by webserver or application.
  - It is measured as the elapsed clock time between when a transaction is **received at a port** and when the response is **sent from the port**.
  - Usually aggregated over a series of transactions in a given time window and stated as the average, median or maximum time required to process and individual transaction over that interval

Performance and Reliability

- **Latency:** The time required to process a request, as **perceived by the user**.
  - Latency includes not only the **response time** but also **network transmission time** and the **time required to process the response by user's client software** (e.g., we browser).
  - Latency is affected by design decisions such as web page complexity, page caching, physical sever location, and geographical replication of servers.
- Latency includes network delay as well as response time

Response Time and Latency



Performance and Reliability

- Performance measurement can be used to support performance improvement, which is accomplished through
  - design,
  - configuration, and
  - enhancement of a web server or application.
- During design, the components of a webserver or application must be selected to provide adequate capacity for the expected workload.

Performance and Reliability

- **Capacity planning**
  - The process of estimating the expected workload of an application and ensuring that the design will have adequate capacity to meet it.
  - Capacity planning is based largely upon experience with previous solutions and existing systems

Performance and Reliability

- After deployment, the operation of a server or application can be adjusted to improve performance
- **Tuning** is the process of making changes to the configuration of a server or application in order to improve performance.
  - Most applications involve a chain of components such as web server, database server, network connections, physical hardware all which influence the overall performance of the system

Performance and Reliability

- Often one of the component will act as the constraint and limit the performance of the whole system.
- When such a component prevents the rest of the system from operating at full capacity, it is referred as to a **System Bottleneck**.
- Part of tuning is to finding and removing bottlenecks by improving the performance of the weakest components.
- Bottlenecks may be present in both H/W or S/W

Performance and Reliability

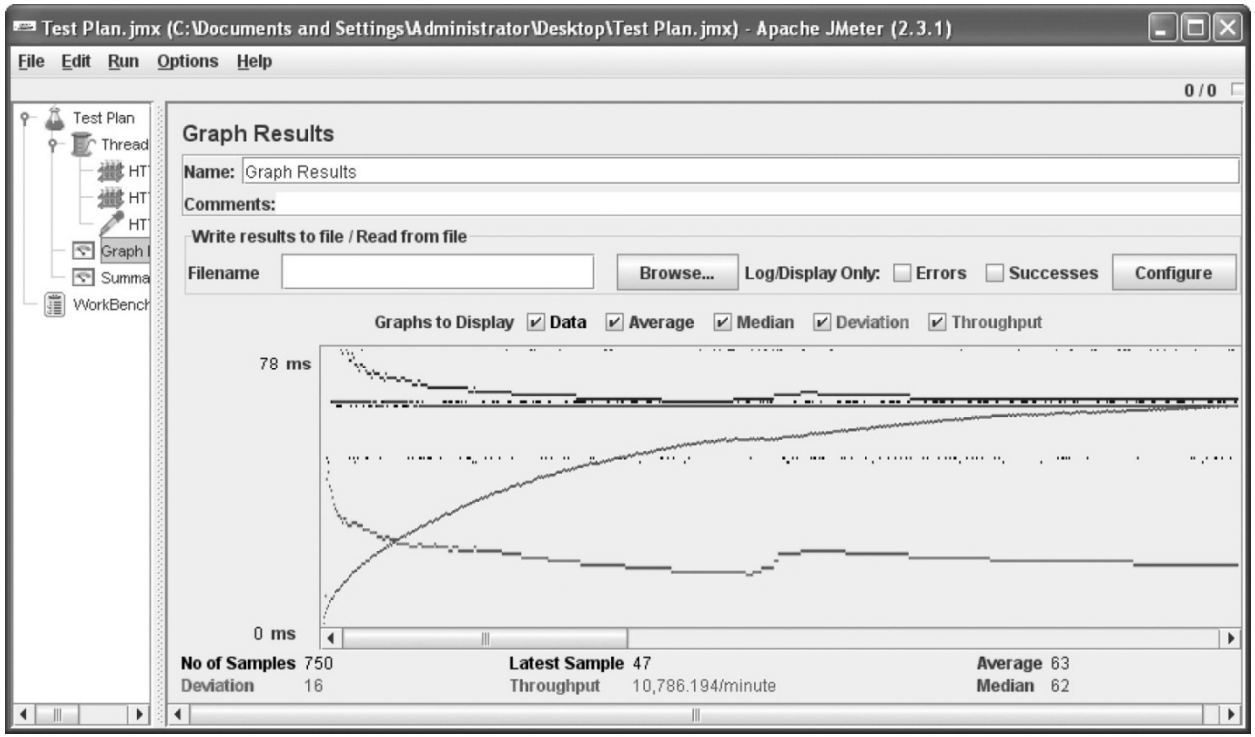
- Locating bottleneck is helped by the **availability of performance data**, but largely depends upon experience, heuristics, and **trial and error**
- There are a variety of tools to measure web server and application performance, such as JMeter.
  - Jmeter creates an artificial server or application workload that can be configured in a variety of ways to simulate a real workload.
  - It also collects response time data and presents the user with a graphical of the server or application response.

Performance Measurement Tools

- **JMeter**, is a free desktop tool for performance measurement.
- It creates an artificial server or application workload that can be configured in a variety of ways in order to stimulate a real workload.
- It also collects response-time data and presents the user with graphical analysis of the server or application performance.
- JMeter can also perform distributed testing by simulating a distributed user base.

Performance Measurement Tools

- Tools such as Jmeter help to determine throughput by simulating one or more users and reporting performance
  - The Ascending Curve indicate throughput for a particular test session, for which throughput is measures at around 10,000 transactions per minute.



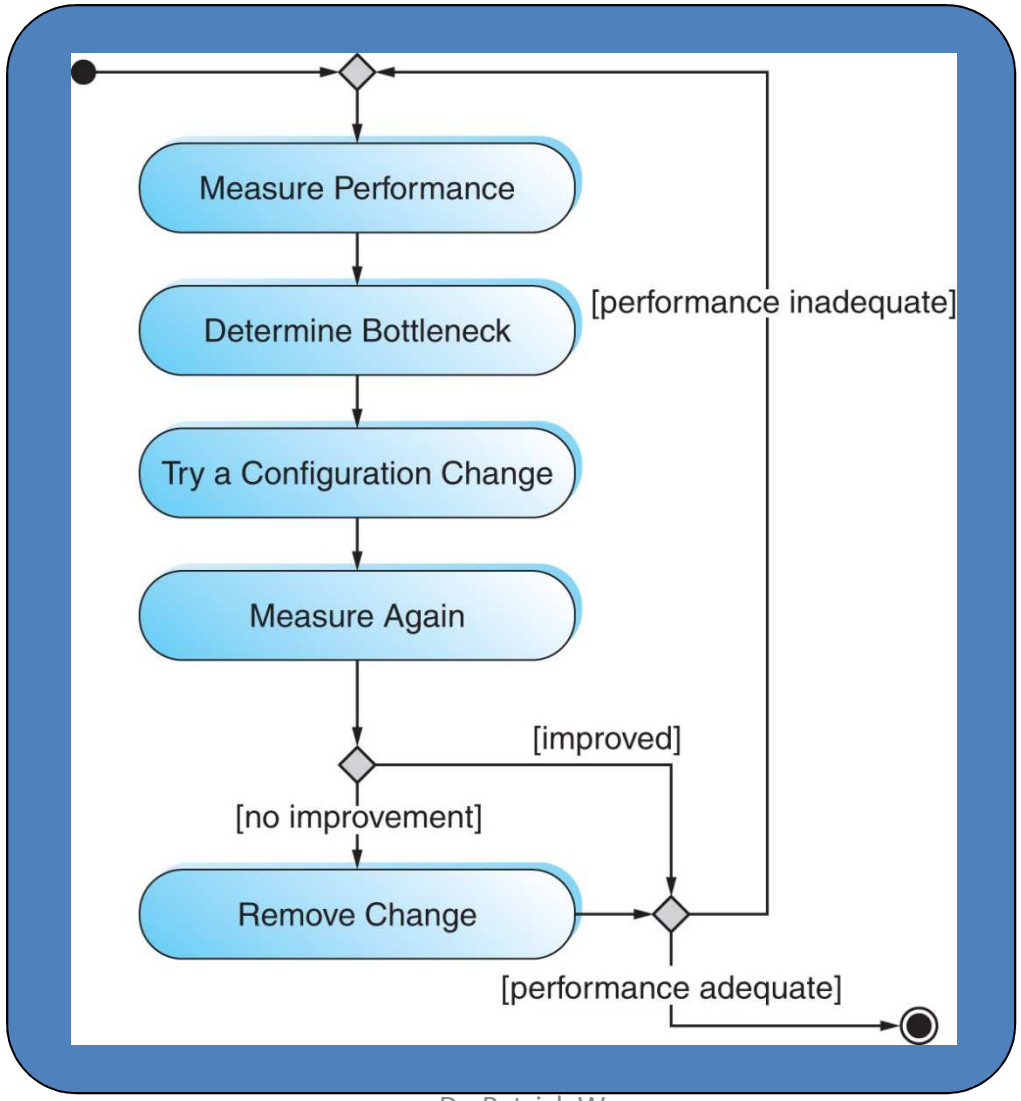
Performance Measurement Tools

- A server includes a portal server, three web servers, and a data server. The portal can handle 180,000 transactions/hour, the web servers can handle 20 transactions/second each, and the data server can handle 1800 transactions/minute.
  - Which component of the system is a bottleneck?
  - Explain your reasoning.

Tuning Process

- A general methodology for server or application tuning is to:
  - **Measure performance** and use the data to find the bottleneck,
    - Helped by the availability of performance data, but largely depends upon experience, heuristics and trial and error
  - **Reconfigure the bottleneck component**
  - **Measure again to see if the change helped** to improve performance.

Tuning Process



Techniques for web server Tuning

- There are several most effective techniques for web server tuning.
- **Thread pool configuration**
  - Incoming requests are handled by individual threads that are managed by the server as a thread pool.
    - Threads require system memory in which to operate.
    - As more threads are added to the thread pool, the amount of memory available to each is diminished, which can reduce performance
  - Therefore, the number of threads must be adequate to handle workload, but low enough so that each thread has adequate memory available.

Techniques for web server Tuning

Session Time-out

- Sessions are established when clients contact the server and are maintained during a series of client-server interactions
- If a client explicitly logs out of an application, the session can be discontinued, but more commonly sessions are automatically dropped after a fixed time-out period.
- Setting the time-out period to shorter interval reduces the number of sessions that must be maintained by the server, each of which also requires a memory allocation.

Techniques for web server Tuning

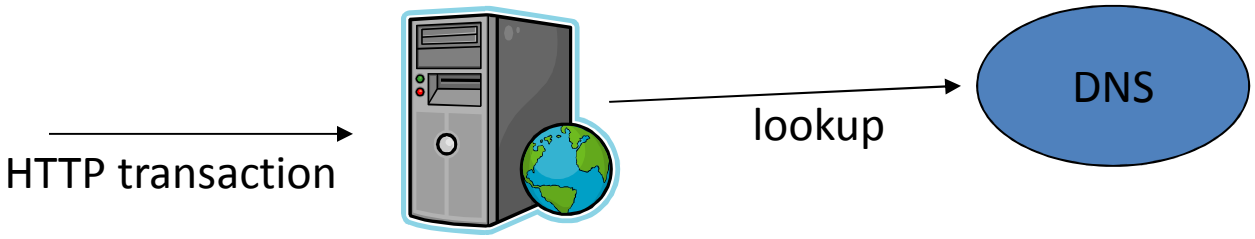
- **Session Time-Out**
  - Sessions are created when clients first contact the server
  - Sessions expire automatically after the specified time-out period (e.g., 5-30 min)
  - **Longer time-out** => more sessions stored on the server, more memory used
    - Longer time-outs are also security risk since unattended workstations invite misuse
  - **Shorter time-out** => less convenience for users who are multitasking and would require to leave application temporarily
- **Finding the best session time-out interval is therefore a three-way trade-off among convenience, performance and security**

Techniques for web server Tuning

- **DNS Reverse lookup**
  - Many servers keep a log of incoming transactions for security, debugging, or performance-monitoring purposes.
  - Incoming transactions carry client's IP address which can translate to a URL via a reverse DNS lookup (translating IP to URL) before the transaction is logged.
  - The lookup consumes CPU time and network resources, and if it is not necessary then it should be disabled.

Tuning: DNS Reverse Lookup

- Web servers may keep a log of incoming transactions, including the source URL of each transaction
- This requires translating IP address from an incoming transaction to a URL via reverse DNS lookup



- Eliminating the DNS lookup reduces the server workload

Techniques for web server Tuning

- **Database connection pooling**
  - Creating and destroying a database connection requires significant processing time.
  - Instead of opening and closing a database connection each time a database transaction is executed,
    - A pool of connections can be established when transaction is initiated and destroyed when the application shuts down
    - Application components needing to execute a database connection can then borrow a connection from the pool and return it upon completion of the transaction.
      - As a result, no connection creation or destruction is required while the application is running.
      - The size of the connection pool is subject to capacity and available memory.

Techniques for web server Tuning

- **Bottleneck** occurs when one component of a system prevents the rest of the system from operating at full capacity; *this component acts as a constraint and limits the performance of the whole system*.
  - **Bottlenecks** are often related to hardware and can be mitigated by changes such as processor reconfiguration, increasing physical memory, or changing a disk controller.
- If application uses a virtual machine, such as Java Virtual Machine, configuration changes may be possible there as well.
- **In the end, tuning can provide only marginal improvements in performance and efficiency.**
- Continued growth in the workload of a web application will eventually **dictate a redesign of its architecture and deployment on more powerful platforms**.

Media and Compression

- To provide interesting and useful user interface, most websites frequently use images, video and audio clips.
- These include:
  - JPEG (Joint photographic Experts Group), MPEG (Moving pictures Experts Group), QuickTime, RealPlayer, MP3, MIDI, WAV (Windows Waveform), and others.
- Compression algorithms re-encode image data into more compact representations of the same information.
- There are two types of compression methods: **lossy** and **lossless**.
  - ***Lossy* compression** creates smaller files by discarding (losing) some information about the original image. It removes details and color changes it deems too small for the human eye to differentiate.
  - ***Lossless* compression**, on the other hand, never discards any information about the original file.

Media and Compression

- To reduce size and transmission time, all theses formats employ compression.
- The degree of compression is generally variable and provides trade-off between media quality and size
- A high quality image might look nice to a user, but if it takes too long to download, it is unlikely that many users will wait to see it.
  - ***Lossless compression*** *reduces data's density by making it more concise, maintaining original quality.*
  - ***Lossy compression*** *erases minute portions of data, decreasing the size of media but sacrificing some aspects of the original composition.*
- Determine the best media size and degree of compression in order to satisfy the requirements without causing long download times

Compression

- HTML can also be compressed
  - HTTP compression allows content to be compressed on the server before transmission to the client.
- For resources such as text this can significantly reduce the size of the response message, leading to reduced bandwidth requirements and download times.
  - Compression is particularly useful where secure SSL connections are used, because it reduces the amount of content that has to be encrypted on the server and decrypted by the client.
- Two compression algorithms are commonly used - **deflate** and **gzip**.
- HTTP clients indicate their support of compression using the **Accept-Encoding** header.

Compression

- Compression is controlled by HTTP headers:
  - **Accept-Encoding:** request header specifying which compression encodings the client understands
  - **Content-Encoding:** response header specifying which encoding was used
    - **Accept-Encoding: gzip, deflate**
- Transfer encoding can take place between proxies, caching servers regardless of what the client and server do.
- Content encoding, however, is always done between client and server only.
  - A server will only compress content for clients that support compression and will set the **Content-Encoding** header so that the client knows which algorithm to use when reading the response body

Design for reliability

- Reliability of a computer system is its ability to meet performance target over a period of time.
- One common and simple performance target is **availability** (the portion of clock time during which a system is available)
- If a system is available for 22 out of 24 hours, then its availability for the day is 22/24 or 91.7%.

Reliability

- **Reliability:** the ability of a system to meet its performance target over time
  - the target may be less than 100% availability
- **Availability:** the percentage of time during which a system is available
- **Outage:** the period of time during which a system is not available

Reliability

- **Example:**
  A system has a performance target of 98% availability over any 24-hour day
  - Daily availability figures for 5 days:
    - 99%, 100%, 95%, 98%, 100%
  - Reliability over 5 days = 4 / 5 = **80%**
- This is the portion of clock time during which the system is available.
- It is measured as the percentage of clock time an application is usable.
  - For example, if a system is available for 22 out of 24 hours, then its availability for the day is (22/24), or 91.7%.

Design for reliability

- Availability during peak demand periods may be more important than off-hour availability.
- It also makes the difference in perceived availability whether downtime is continuous (1-2 hours) or repeated (10-12 minutes).
- An outage is the period of time during which a system is not available
- Web applications availability requirements are typically very high as the user base may span several time zones

Design for reliability

- Each time there is outage means decreased customer satisfaction and reduced income for commercial sites
- High target such as 99% availability entails more that 7 hours of outage each month which would be costly.
- The following list includes the amount of time that each of four web applications was unavailable in a 24-hour period. Calculate the availability of each application as a percentage, to two decimal places precision.
  - 5 minutes
  - 32 seconds
  - 1.5 hours
  - 48 minutes

Design for reliability

- Sources of outages (point of failure)
  - **External Network:** Internet connection provided by an ISP, Internet backbone
  - **Internal Network:** Local area components
  - **Server Platform:** Processor Data Units, power units
  - **Application Software:** Web application
  - **External Software agents:** other required services such as banking systems
  - **Internal environment:** Power supply, heating/cooling

Points of Failure

- **Single Point of Failure (SPOF):** any system component, the failure of which can cause the entire system to fail
- SPOFs can be eliminated through:
  - **standby component** (or service): an idle component that can be inserted in place of a failed component
  - **replicated component** (or service): one of a set of components that work in parallel so that if one fail, the others continue to share the workload
- Each single point of failure (SPOF) must be eliminated in order to avoid an outage of the entire system.

Design for reliability

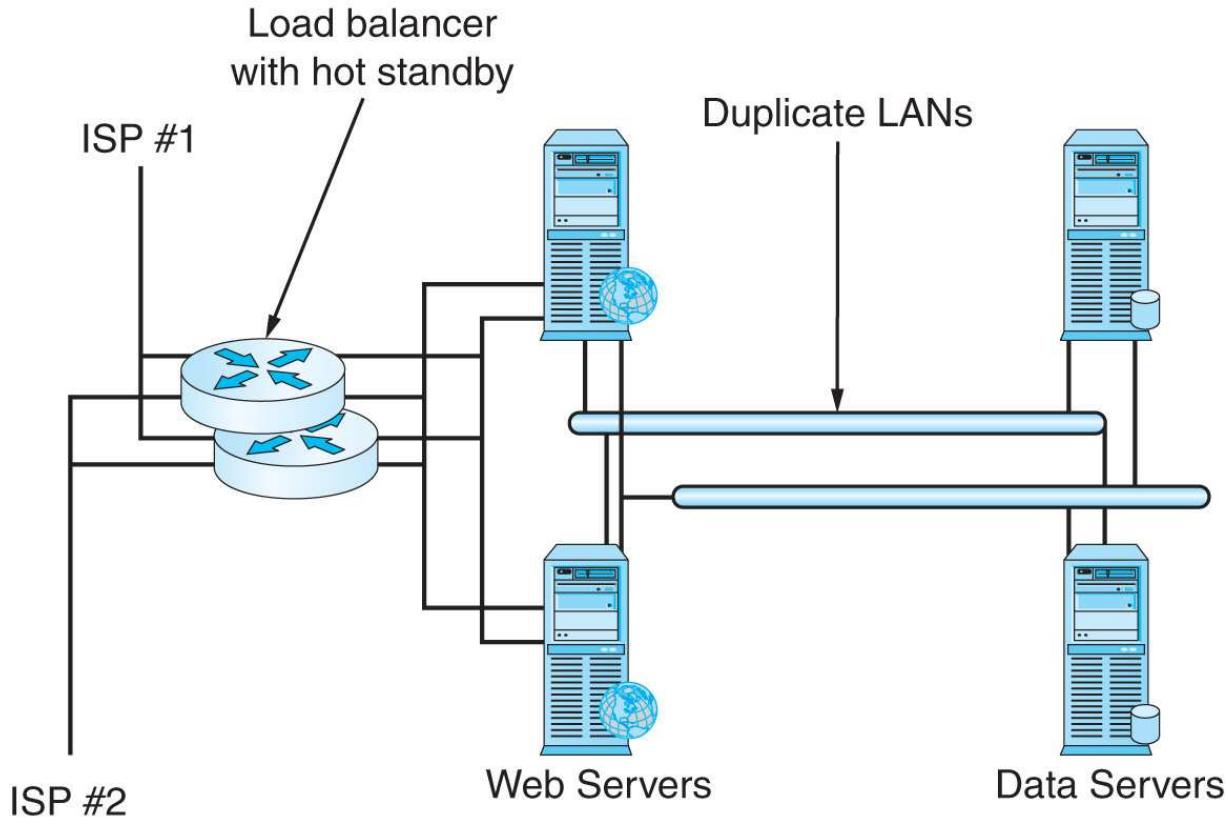
- To ensure availability, each point of failure must be identified through a risk analysis process and plans must be made to mitigate each risk
  - Either by preventing its occurrence or
  - By planning an adequate compensation in case it occurs
- One way to do this is through
  - **Redundancy:** Provision of alternative component that can replace or compensate for failure of a primary component

Design for reliability

- Redundant components can be configured as:
  - A **standby component** or service is generally idle and is put into service only upon failure of corresponding primary component such as a standby disk unit which mirror the work on a primary disk and takes over the function immediately upon failure of the primary
  - A **replicated Component** functions in parallel to its corresponding primary component, sharing the workload under normal circumstances and taking overall or part of the workload of a failed unit after failure.
    - A **load balancer** that distributes transactions over several servers such that all servers are kept equally busy

Built In Redundancy

- Redundant components eliminate SPOFs



Reliable Operations

- Reliability also depends on
  - Proper operations documentation
  - Well-trained administrative / operations staff
  - Physical security
  - Reliable environment: (electrical power, air coolers, etc.)

Testing

- The purpose of software testing is not show that a system works, but rather to find the errors in a system
  - There are *always* errors!

Characteristics of Web Applications

- **Content-intensive:** a great deal of information
- **Multiple, dynamic user configurations** (web browsers)
- **Multiple layers** (client, server, application, database)
- **Dependent on external networks**

Web Application Testing

- There are two perspectives on the objectives of software testing
  - Testing **to show** that the application works correctly,
  - To find and **remove errors** in the application
- Testing should be concurrent with and integral part of coding
- A model is only complete if it passes all the prescribed tests.
  - **Daily testing** of complete application is another common practice that helps to find errors early in the development process

Web Application Testing

- **There are few characteristics that makes web application testing different**
  - Web application are usually **context sensitive**:
    - There is great deal of information in the form of web pages, documents, and databases that may contain mistakes, omissions, incorrect labels, or redundancy
  - Web application must operate under **multiple, dynamic configurations.**
    - Users employ various operating systems, and web browsers which have different characteristics.

Web Application Testing

- There are **multiple layers** to web application
  - Client-server architectures typically include several layers and operate on layered network protocol.
  - Finding an error often requires analysis of the operation of application through multiple layer software and protocols in order to determine if an error originated in a particular layer or was passed through.
- **Networks** are critical part of web applications.
  - Web applications are built on top of the internet, and depend upon correct internet operation.
  - What appears as a failure of a web application can in fact be a network problem originating in the user or host ISP systems or elsewhere on the internet

Web Application Testing

- There are several established approaches to testing.
- Each bringing a different perspective, objectives and a different set of techniques to the testing process
  - Acceptance Testing
    - **Objective:**
      - To judge users' level of satisfaction with the application and perceived utility of the application from their perspective
      - Done late in the development process when significant amount of application functionality is complete (time required to complete transactions).
    - **Strategies:**
      - Present prototypes to users early in the development process. Collect qualitative feedback or measurement of usefulness
      - Allow users to try a nearly complete application or part therefore in an experimental environment (Alpha testing)
      - Allow users to try a complete application in a real, but small-scale and controlled environment (beta testing)

Acceptance Testing

- Evaluate user satisfaction with the system

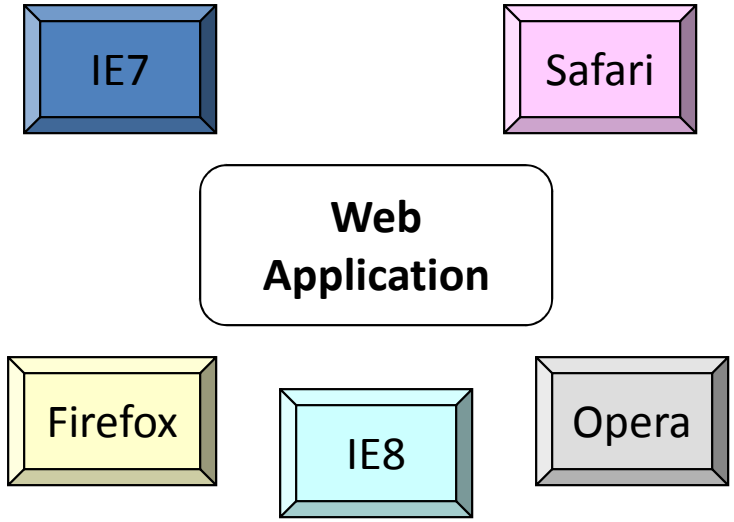
- Strategies:
  - Present prototypes during early development
  - Alpha testing: users test a nearly complete system, in a mock environment
  - Beta testing: users test a completed system, in a trial environment

Web Application Testing

- Compatibility/Configuration Testing
  - **Objective:**
    - To find instances of incompatible client software configurations.
    - Developers usually have no direct control over applications used on the client side of web application.
  - **Strategies:**
    - Set up virtual machines to test various operating systems and their different versions.
    - To test Several browsers and several versions of each on every operating system
    - Test different versions of plugins and helper applications that are required to use the application

Compatibility/Configuration Testing

- Find instances of software incompatibility
  - especially due to client (web browser) differences and configuration
- Strategies:
  - setup virtual machines for different browsers
  - test multiple versions of each browser
  - test plugins (e.g.,) PDF reader



Web Application Testing

- Content Testing
  - **Objective:**
    - To find errors in web pages, documents, or data that will be presented to users.
    - Errors may be in the form of typographic errors, incorrect information, missing information, obsolete information, or redundancy.
  - **Strategies:**
    - Automated spell-checking and grammar-checking software can be used to regularly scan content
    - Because content usually changes overtime, content testing is an on-going responsibility,

Content Testing

- Find errors in web pages, documents, or data that users will see
- Strategies:
  - Automated spell-checking, grammar-checking
  - Continuous content review by production staff

Functional Testing

- Find instances of a component failing to perform in accordance with specifications
- Strategies:
  - Black-box testing: test each specification
  - White-box testing: test each bit of code
  - Error injection: force errors in order to test error-handling mechanisms
  - Use testing tools
  - Maintain a test suite that can be used repeatedly
  - Regression testing: test new versions against old versions

Functional Testing

- **Objective:**
  - To find instances of a module or component failing to perform accordance with stated requirements or specifications.
- **Strategies:**
  - Test each stated specification or equipment (black-box testing)
  - Exercise each segment of code or each operation (White-box testing)
  - Force errors to occur to test error-handling operation (error injection)
  - Use test-management software to develop a test suite and add new tests to suite as they are developed
  - When functional errors are reported, add a test to detect each error before correcting the error.
  - As new versions of the software are developed, compare functionality and outputs of the old and new versions (regression testing)

Interface Testing

- Ensure that users can effectively use the application and that appropriate help and guidance are provided

- Strategies:
  - Check hyperlinks for correct navigation
  - Check that all help functions are correct and appropriate for the user
  - Check all error responses
  - Ask selected users to try the application and record their performance and reactions

Interface Testing

- **Objectives:**
  - To ensure that users can interact effectively with the application, regardless of user level, that appropriate help and guidance is provided, that the interface is pleasant to work with, and that all navigation controls work correctly
- **Strategies:**
  - Check hyperlinks to ensure that they point to the correct destination
  - Check all help functions to ensure they are correct and appropriate for the situation
  - Check all error responses to ensure they provide adequate guidance
  - Ask selected users to try a prototype or completed application and record their performance and opinions

Performance Testing

- Determine if the application has the capacity to support the expected workload

- Strategies:
  - Establish peak and average capacity expectations
  - Use performance measurement tools to place the application under the expected loads
  - Place the application under increasing load until it fails (stress testing), to determine the maximum capacity

Performance Testing

- **Objective:**
  - To determine if the application will support the expected transaction workload.
- **Strategies:**
  - Establish peak and average workload expectations.
  - Use performance-measurement tools to place the application under expected load and measure its response time
  - Use performance-measurement to place the application under increasing loads until it fails in order to determine its maximum peak capacity (stress testing)

Security Testing

- Determine if the application is vulnerable to know attacks
- Strategies:
  - Establish a threat model during development
  - Use common documented attack strategies against the system to search for vulnerabilities

Security Testing

- **Objective:**
  - To determine if the application is vulnerable to attacks that violate its security (confidentiality, integrity, authentication, mechanisms)
- **Strategies:**
  - Establish a threat model during requirements gathering, and use that to guide testing as well
  - Experience helps! Use documented exploits and attack strategies, or hire a security consultant who is familiar with them