# DIM3/iTECH Course Overview 2011/2012

Version 1.2
(subject to some minor changes)

Release Date 2/12/11

# Table of Contents

## Introduction

The main aim of this course is to provide students with an overview of the different stages and components of web application development. It will provide students with the skills to design and develop distributed web applications in a disciplined manner, and strengthen their understanding of the context and rationale of distributed systems.

### The aims of this course are:

- To provide an overview of the ongoing developments in web application development
- To promote the disciplined design and development of distributed web applications
- To strengthen their understanding of the context and rationale of distributed systems
- To understand the messaging and protocols used as a communication mechanisms in web applications
- To develop the ability to implement and deploy distributed web applications

### The Intended Learning Outcomes of this course are:

Upon completion of the course, students will have the ability to:

- Illustrate and describe the n-tier client-server architecture of web applications
- Explain the role of messaging and protocols within the design of web applications
- Critique and evaluate the information architecture of web applications
- Explain the opportunities, challenges and problems of developing web applications in a distributed environment
- Identify and critically analyze the requirements of a web application
- Design and specify the architecture of a web application
- Evaluate and assess specifications and designs of web applications
- Construct, build and deploy a web application

### The main Languages and Frameworks that you will be required to use within this course:

- Python, http://www.python.org/
- Javascript, https://developer.mozilla.org/en/JavaScript/Reference
- AJAX
- JQuery, http://jquery.com/
- Django (Web Application Framework), http://www.djangoproject.com
- CSS, http://www.w3.org/Style/CSS/Overview.en.html
- XML, http://www.w3.org/XML/
- XHTML, http://www.w3schools.com/xhtml/
- PuppyIR, http://sourceforge.net/projects/puppyir/
- Plus more.

**Note / Warning**: Since this course tries to cover the entire process of web application development, it will not be possible to teach any of the languages in detail. During the course and through the labs we shall touch upon each of these languages and frameworks but it will be your responsibility to learn each in sufficient detail to use them to build your own web application.

### Other Skills required/desired:

- Entity Relationship Diagrams (and database modeling and SQL skills)
- Dataflow Diagrams
- Sequence Diagrams
- UML and architectural diagrams

Again, these will not be taught explicitly in this course and are assumed knowledge for this course. If you do not understand how to create or interpret such diagrams you should research each of these.

## Course Text Book and Recommended Reading

**The course text is:**

- Web Design: a complete introduction, by Nigel Chapman and Jenny Chapman

**Recommended Reading:**

- Peter Morville and Louis Rosenfeld; Information Architecture for the World Wide Web; O'Reilly, Third Edition, Nov 2006.
- Marti Hearst, Search User Interfaces, http://searchuserinterfaces.com/

**Technical Recommended Reading:**

- Dive Into Python: http://www.diveintopython.org/
- The Python Tutorial: http://docs.python.org/tutorial/
- The Django Book: http://www.djangobook.com/en/2.0/
- Design Patterns in Python, Rahul Verma and Chetan Giridhar, http://www.testingperspective.com/wiki/doku.php/collaboration/chetan/designpatternsinpython/mvc
- Other books on Python http://wiki.python.org/moin/AdvancedBooks
- Python in 10 Minutes, http://www.korokithakis.net/tutorials/python/

## DIM3 Timetable and Assessments

Lectures 12-1pm Monday and Wednesday
Labs 2-4pm Tuesday

| DIM3 Assessment Schedule | | | |
|---|---|---|---|
| **No.** | **Assignment** | **Due Date** | **Percent** |
| **1** | Specifications and Design Report | **18/2/12** | * |
| **2** | Peer Feedback on Specifications and Design Report | **26/2/12** | 5% |
| **3** | Revised Spec and Design, plus Implementation Report | **1/3/12** | 15% |
| **4** | Demonstration of Web Application | **13/3/12** | ** |
| **5** | Exam (1.5hrs) | **Tba** | 80% |

## iTech Timetable and Assessments

Lectures 12-1pm Monday and Wednesday
Labs 2-4 Wednesday

| iTECH Assessment Schedule | | | |
|---|---|---|---|
| **No.** | **Assignment** | **Due Date** | **Percent** |
| **0** | Demonstration of Polling App | **8/2/12** | 5% |
| **1** | Specifications and Design Report | **18/2/12** | * |
| **2** | Peer Feedback on Specifications and Design Report | **26/2/12** | 5% |
| **3** | Revised Spec and Design, plus Implementation Report | **1/3/12** | 20% |
| **4** | Demonstration of Web Application | **14/3/12** | ** |
| **5** | Exam (2hrs) | **Tba** | 70% |

*** This item must be submitted. Failure to do so will result in -5% to your overall report mark (item 3). i**

**** Your web application must be demonstrated to show that a complete working system was produced. Failure to do so will result in -5% to your overall report mark (item 3).**

## Lecture Series Overview

**Introduction**
L1. Course Overview (LA)
L2. Assessments and Use Cases (LA)

**System and Information Architectures**
L3. System Architectures (LA)
L4. Web Application Framework (LA)
L5. Django
L6. Case Study: PuppyIR (LA)
L7. Information Architecture (LA)

**System Specifics**
L8. Messaging and Protocols (LA)
L9. Client-Side Environment/Ecosystem (LA)
L10. Cascading Style Sheets (LA)
L11. Client Side Scripting (LA)

L12. Assignments/Feedback/JQuery (LA)

L13. XML – Introduction (SR)
L14. Processing XML – DOM and SAX (SR)
L15. XML – Advanced (LA)
L16. AJAX with JQuery (LA)

**Information Specifics**
L17. Web Accessibility (LA)
L18. Ambient Findability (LA)

**Other system architectures, developments and modeling**
L19. Other Web Services (SR)

**Wrap-Up**
L20. Summary (LA)

## Lab Series Overview

Week 1. Assessments and Use Case (L2), Groups and Python (in Math 516). Python is for iTech students only.
Week 2. PuppyIR Case Study (L6) in Math516, PuppyIR Tutorial: MaSe in Lab.
Week 3. Build a Polling Web App (Django Tutorial Part I+II) in Lab
Week 4. Build a Polling Web App (Django Tutorial Part III+IV) in Lab
Week 5. PuppyIR BaSe: Cascading Style Sheets in Lab
W6. Project Group Work in Lab
W7. XML Tutorial and Project Group Work in Lab
W8. Project Group Work in Lab
W9. Project Group Work in Lab
W10. Demonstrations

## Assessments

There are five assessment items for DIM3 and 6 assessment items for iTECH.

### O. Polling Application (iTech Only)

In the labs, you need to demonstrate to the tutors/lecturers your working polling web application by the end of week 5.

### 1. Specification and Design Report

Defining and describing the specifications given user requirements is a very important step in the development process. A clear specification and well thought out design will provide the basis of the development and demonstrate that you understand:

- the process of designing web applications,
- the interrelationships and messaging between components, along with
- the standards and protocols required, and
- the interoperability concerns.

**Note:** While you shall see that there is a lot of repetition in the specification and design of web applications (as you move from designing one to another), it is important to know why you are re-using and re-applying the same approach, methods, tools and diagrams in the process.

**Report Template:**

An outline of your report is provided below as a guide. You should include the details suggested below and any other details that would communicate your specification and design more clearly. Also, be sure to include all links and references that are relevant.

Within the guide below you will also see "Reflective Questions": theses are questions that you should reflect upon once you have written the section and ask yourself whether the answers to these questions are clear from your design and specification (or whether you have valid reasons for your choices, or not). If you don't have satisfactory answers then you should go back and improve your design and specifications.

1. Title and Abstract.
   a. Title, your name, and student number, course identifier
   b. Provide a concise summary of the application (main purpose, usages)
2. Aim of Application.
   a. What is the purpose of the application?
   b. Describe the design goals and objectives of the application.
   c. Functionality List: i.e. what is the required and desired functionality?
   d. Reflective Questions:
      - Is the scope of the application appropriate? Are the design goals realistic/achievable? How complex is the application? Is distribution across the web appropriate?
3. Client Interface
   a. Draw a wireframe of the user interface
      - this may require several wireframes depending on the complexity of the application and the interfaces
   b. Describe the user interface.
      - i.e. Label key input and output components: describe them.
   c. Provide a Walkthrough and explain the user interactions with application.
      - i.e. use cases

    d. Describe the interactions associated with the dynamic components on the user interface.
    e. What calls are required to dynamically update the data on the client side?
    f. Reflective Questions:
        - How does the user interface help the user achieve their goal, or complete their task? Is the user interface intuitive, appealing, usable, etc?
        - What technologies are used on the client side? What are the reasons for your choices? i.e. what is the advantages and disadvantages of using this technology? What other options are there?

4. Application Architecture
    a. N-Teir Architecture Diagram
        - i.e. data flow diagram between the interface/client, middle ware, and backend services/data repos
    b. Describe the data model i.e. what data needs to be stored or persisted by the application?
    c. What are the relationships within the data model.
        - i.e. use ER diagram and explain.
    d. Describe the backend services used (if any).
    e. Reflective Questions:
        - How have you ensured that there is a separation of concerns? What other technology could have been used instead of django? What are the advantages of using a Web Application Framework over other technology? And, what are the disadvantages?

5. Message Parsing
    a. On the architecture diagram, Identify and label the main messages that will be parsed through the application.
        - or alternatively include sequence diagrams to denote the sequence of communications parse between clients and servers.
    b. Describe the messages that are parsed back and forth through the application.
    c. For the main transactions - describe the payload of the messages
        - i.e. What are the contents of the messages? i.e. include the XML, XHTML, JSON, etc of one or two messages.
    d. Reflective Questions:
        - What is the format of the messages? Why this format? What other formats could be used, what are the advantages and disadvantages of these other formats?

6. Summary
    a. Summary of application and the technologies used and required for the application.
7. References and Links
    a. Add all relevant links and references that are used within your report.

**Examples:**
Below is a list of some sample specification and design reports in various formats (i.e. not necessarily strictly like the ones above). These examples come directly from the PuppyIR case study – some are tailored for dissemination purposes and some are for project reporting purposes. It is important for you to see a variety of different ways in which to describe the

specifications and designs of your web applications. Some are good, some are bad, and some are just plain ugly.

- PuppyIR: OS framework HCIR 2009,
  - http://dl.dropbox.com/u/382885/PuppyIR/PuppyIR-2009-HCIR.pdf
- Specifications Report
  - http://dl.dropbox.com/u/382885/PuppyIR/PuppyIR-D4_1-SpecificationReport-1_0.pdf
- Design Report,
  - http://dl.dropbox.com/u/382885/PuppyIR/PuppyIR-D4.2-Design-Report-v1.0.pdf
- Implementation Report
  - http://dl.dropbox.com/u/382885/PuppyIR/PuppyIR-D4.3-Report-on-Implementation-and-Documentation-v1.0.pdf
- FiFi – Finding and Filtering Information for Children,
  - One page version: http://dx.doi.org/10.1145/1835449.1835569
  - Four page version: http://dx.doi.org/10.1145/1840784.1840834

**Submission Details:**
5 pages in ACM Style Format (http://www.acm.org/sigs/publications/proceedings-templates)

**Where and how to submit:**
We will be using the Aropa system to submit work. Details will be provided in lectures.

**Assessment Criteria:** This submission will be marked according to the criteria used in the Peer Assessment exercise (see below), i.e. the report needs to complete, clear, well thought out and justified. The final score for the report will be a combined average of the peer ratings and the marker's rating.

### 2. Peer Assessment – Feedback on Specifications and Design Report

Understanding and reviewing specifications and designs is an important part of developing any type of application. Thus, this assessment will require you to evaluate several other specification and design reports. This will be helpful, not only, because you will see what others have produced – but also to provide you with more examples for the case study in the exam (see below).

**Reviewer Guidelines:**
For this assessment item you will be assigned 2 other reports. You will need to evaluate each report for its completeness, clarity and thoughtfulness:

- Completeness: does the report address each of the required questions? Which parts are incomplete?  And what else should be added?
- Clarity: does it answer them clearly and concisely, i.e. could you take this specification and implement the web app?, which parts or points are unclear? What parts are just waffle? and
- Thoughtfulness: does the report discuss any aspects, which will have a major impact on design, i.e. does it provide justification for its design choices? Comment on both positive and negative aspects.
- Scope: Is the project overly complicated or complex? Is the project too ambitious? If it is a PuppyIR app does it create a theme, add a component, and include a different backend service?

Each review should:

(1) summarize the specification and design in one or two sentences,
(2) address each of the above questions (i.e. completeness, clarity and thoughtfulness), using examples from the report to evidence remarks (a paragraph each),
(3) provide suggestions on how the report could be improved, and
(4) provide a overall assessment (one-two sentences) The review should be about half a page in length.

An overall rating out of five should be assigned to each report. Five points for an excellent specification and design, that is very complete, clear and well thought out. Four points for a couple of minor things to be improved, and then if it is not complete deduct another 1 or 2 marks (depending on completeness, or lack of), and if it is not clear, deduct another 1 mark, and if it doesn't appear to be well thought out loss another mark.

If you feel that you have received a particular bad i.e. nasty reviews, or un-professional reviews then please email Leif.Azzopardi@glasgow.ac.uk with subject line: "Review-Appeal-2010-groupname", and in the body, include the offending review, and outline what the problem is.

**Assessment Criteria:** Each reviewer will be assessed on how well they evaluate the specifications and design reports with respect to the guidelines. A mark out of 5 will be given for this feedback. And you will be asked to rate the quality of other student's feedback that is provided, as an additional check.

All reviews and feedback will be monitored. Any reviews or feedback which are malicious, nasty or unprofessional will result in 0 marks for this component (and disciplinary action if appropriate).

### 3. Revised Specification and Design, along with Implementation Report

**Report Template:**

**1- The Specifications and Design Report from above.**

**2- Summary on feedback (about 0.5 to 1 page)**

a. include a summary of the feedback given (or refer to specific comments from the feedback)
b. comment on how you have revised the design (if at all) according to the comments received
c. how has the feedback helped, and has this process been helpful.

**3- Implementation Notes (1-1.5 pages)**

a. Functionality Checklist (what functionality is completed)
b. Known Issues (what kind of works, what kind of errors to do you get)
c. Plan for future development

**4- Reflective summary on the development of the application. (about ½ a page)**

a. What have you learnt through the process of development? How did the application of frameworks help or hinder your progress? What problems did you encounter? What were your major achievements?

**Submission Details:**

7 pages in ACM Style Format (http://www.acm.org/sigs/publications/proceedings-templates). The final reports for this course will be combined in a Technical Report and published as part of the School of Computing Science's Technical Report Series.

## 4. Demonstration

In the lab in week 10, you need to demonstrate to the tutors/lecturers your working web application. The web app needs to be running on one machine in the lab, and accessed on the demonstration machine (which will be on a projector for everyone to see).

## 5. Exam

The exam will consist of two parts:

(1) About 5-10 questions for DIM3 and about 10-15 questions for iTech on web application development, mainly general knowledge questions.

(2) A case study where you will be given a user scenario and briefing for a web application, and you will need to outline the design and specification of a web application. The format will be similar to your assignment but you will be asked to design a web application that is different to the one in your assignment.

In the case of iTECH, a significantly more complicated use case and briefing will be provided.

## Description of Cases for Web Applications

### 1. (A, B, C) PuppyIR Project - Create your own search service using puppyIR

PuppyIR provides most of the infrastructure for a basic web search engine accessed via a distributed information service. However it only has a few possible configurations available. If you choose this option you will need to design your own puppyIR service that extends or modifies the BaSe web application in some way.

You should try to design your own service so that you create (1) a layout and look and feel for your service, (2) an additional query or document processing feature (i.e. add a component within the query/document pipeline for your application), and (3) connect to a different search service or use different search modes (i.e. other search engine or media). Example PuppyIR based Web Applications:

A. **GeoFun – Explore the World**
   - This mashup lets school children select a location on the world map, or type in the location of a place, and provides a "homework" like report about the place.

B. **FanMag – Collage together a bunch of pages about a favorite celebrity or teen idol.**
   a. This mashup will take a pre-selected teen idol/celebrity and create an instant and update to date fan-zine.
   b. The mashup should also let users comments add comments and gossip of there own.

C. **Create a academic search engine by connecting to Google Scholar or Microsoft Academic search (AcaSe)**

   –Backend connects to Microsoft Academic search
   –Include additional query fields, year and author
   –For a given session, enable user to select results, and display list of results to user, clear results and remove results
   –Considering modifying the presentation of the search result

### (D) iHero

Social media applications are great for disseminating information quickly. The idea behind this app is that "heros" can report incidences as they occur, uploading details of riots, crimes, or disasters in progress, to help officials and authorities respond to such emergencies or incidents. The app will need to capture location and other details of the incidence in a quick and easy way, perhaps even allowing pictures and video to be uploaded. Visitors to the site would be able to see emerging incidences on a map.

### (E) rate-this-course.com: start up your own website warning students to avoid taking courses (like this one).

You can imagine that users of this site would like to know how good or bad the courses that they are interested in doing are, and what they are like. Those that have taken the course would like to provide reviews to share their experiences (good and bad) with others (this is perhaps a bit altruistic, and might require some motivation or incentive, but you reckon it would be possible to get folk interested). But as they say, "Build it, and they will come."

- This app should allow users of the site to enter course details (if needed), and provide a rating and comments.
- For course details it should include the university/college, course code, name.
- For the ratings, the user needs to enter when the course was taken (semester, year), their overall rating of the course, and comments.
- The site should let users search for courses
- List courses at a specific university/college
- See a list of the latest ratings/comments
- See a list of the best rated courses

## L1 – Introduction

This lecture will provide an overview of the course aims, intended learning outcomes and the design of the course. It will explain the rationale behind the course and along with a gist of the courses content. This will include an overview of a number of examples of real life web applications, where we shall high light the main problems and questions that we shall seek to address and answer through the lecture series.

### Outline:

1. Overview of Course, and design of the course
2. Our expectations, and where to find resources for this course
3. The lecture guide for this course
4. Examples of Internet web applications
5. The main issues, concerns and problems in developing web applications

### Intended Learning Outcomes:

At the end of this lecture you should be able to:

1. Explain the structure of course and the course objectives/aims
2. List different examples of web applications
3. Explain the complexity of web applications
4. Identify the main topics and points of concern when developing web applications

### Recommended Reading

Chapman and Chapman, Chapter 1 – Web Experiences

## L2 – Web Application Use Cases: Assessments and Assignments

This lecture will provide an overview of the course assessments that you are required to undertake and complete. It will explain each assessment item, and provide an overview of aims and objectives of each item and how it fits within the course. Each assessment item emphasizes the development of transferrable design skills and professional skills development. A number of web application use cases will be presented – where one will need to be chosen as the focus of the case study i.e. you will have to pick one of the use cases, and as part of the assessment you will then go on and specify, design and build a web application based on one of these use cases.

### Outline:

1. Overview of the course assessment and how it links together
2. What is involved in each assessment item
3. An overview of a number of web application use cases
4. Question and Answer session on use cases

### Intended Learning Outcomes:

At the end of this lecture you should be able to:

1. Explain the assessment procedure for this course
2. Describe the different use case scenarios
3. List the main criteria for success

## L3 – System Architecture

This lecture will provide a high level overview of the architecture of distributed web applications. The lecture will describe how applications have been made in the past, develop your awareness of the different architectures that have and are still being used to develop applications, and observe that the process for building web applications is similar and involves a lot of redundancy (from specification and design to development and implementation). This last point will be picked up in Lecture 4, where frameworks are employed to ease the development burden. Also, the lecture will show the difference between old school software engineering and web application engineering

### Outline:

1. The n-tier architecture of a web application
    a. Single tier
    b. Two tier
    c. N-tier

### Intended Learning Outcomes:

At the end of this lecture you should be able to:

1. Describe the architecture of a web application
2. Explain the advantages of the Tiers in the architecture
3. Illustrate the different system architectures
4. Provide reasons for separating the different types of concerns

### Recommended Reading:

- Chapman and Chapman, Chapter 8 – Web Applications
- 3 Tier Architecture
  http://dotnetslackers.com/articles/net/IntroductionTo3TierArchitecture.aspx
- N-Tier Architecture
  http://www.sun.com/software/whitepapers/wp-ntier/wp-ntier.pdf

## L4 – Web Application Frameworks

This lecture will describe the Web Application Frameworks. This Python based framework stops you wasting valuable time on CRUD. In this lecture, we will present and overview of the framework and the main components within the framework to provide the theory to accompany the practical work.

### Outline:

1. What is a Web Application Framework
    a. Definition
    b. Characteristics
    c. Model View Controller
    d. Benefits and limitations

### Intended Learning Outcomes:

At the end of this lecture you should be able to:

1. Describe and discuss web application frameworks
2. List and describe alternative web application frameworks
3. Illustrate and explain the Model View Controller Design Pattern

### Recommended Reading

- Chapman and Chapman, Chapter 1 – Web Experiences
- Model View Controller, http://en.wikipedia.org/wiki/Model-view-controller
- Understanding MVC, Jeff Atwood, http://www.codinghorror.com/blog/2008/05/understanding-model-view-controller.html

### Recommended Viewing

- Ruby vs. Django. (3mins) http://www.youtube.com/watch?v=PLUS00QrYWw
- Pro's and Con's of MVC, (2mins), http://vimeo.com/7404134
- MVC Song (6mins) http://www.youtube.com/watch?v=YYvOGPMLVDo
- MVC Public Service Announcement No.1, (2 mins), http://www.youtube.com/watch?v=91C7ax0UAAc

## L5 – The Django WAF

This lecture will describe the Django Web Application Framework. This Python based framework stops you wasting valuable time on CRUD. In this lecture, we will present and overview of the framework and the main components within the framework along with some practical demonstrations.

### Outline:

1. The Django Framework
   a. Model View Template Design Pattern
   b. Django Architecture
   c. Django Modules and Creation Steps
   d. Sample Application

### Intended Learning Outcomes:

At the end of this lecture you should be able to:

1. Describe the architecture and main modules of Django
2. Setup a simple web application

### Recommended Reading

- For more information (Documentation, Download and News) http://www.djangoproject.com/
- A Good book to learn Django, http://www.djangobook.com/
- Design Patterns in Python, Rahul Verma and Chetan Giridhar, http://www.testingperspective.com/wiki/doku.php/collaboration/chetan/designpatternsinpython/mvc

### Recommended Viewing

- Django: Web Development for Perfectionists with Deadlines (1hr) http://www.youtube.com/watch?v=p-WXiqrzAf8
- Leah Culver talks Python and Django (10mins) http://www.youtube.com/watch?v=w84a7_QL_pA
- Django framework demo (5mins) http://www.youtube.com/watch?v=CpXz_LsBDBo

## L6 – Case Study – PuppyIR: Developing information services for children

This lecture will detail a current Open Source framework dedicated to creating information services (i.e. web applications) for children. The specification and design of the framework will be presented, along with an overview of a number of different demonstrators. This case study will provide numerous insights into the design and development process.

### Outline:

1. The PuppyIR project
2. The Framework
    a. Use case of the meta-level user (i.e. developer)
    b. The design goals of the framework
    c. The specification and design of the Framework
    d. The architecture
3. Example Demonstrators
    a. SuSe
    b. FiFi

### Intended Learning Outcomes:

At the end of this lecture you should be able to:

1. Describe the PuppyIR Framework
2. Describe the main architectural components with in the framework
3. Illustrate basic data flows of messages and communications through the n-tier architecture
4. Design and draw a basic puppyIR service
5. Assemble and Compose their own puppyIR service
6. Discuss the benefits of Top-Down or Bottom Approach Design

### Recommended Reading

- PuppyIR: OS framework HCIR 2009,
    - http://dl.dropbox.com/u/382885/PuppyIR/PuppyIR-2009-HCIR.pdf
- Specifications Report
    - http://dl.dropbox.com/u/382885/PuppyIR/PuppyIR-D4_1-SpecificationReport-1_0.pdf
- Design Report,
    - http://dl.dropbox.com/u/382885/PuppyIR/PuppyIR-D4.2-Design-Report-v1.0.pdf
- Implementation Report
    - http://dl.dropbox.com/u/382885/PuppyIR/PuppyIR-D4.3-Report-on-Implementation-and-Documentation-v1.0.pdf
- FiFi – Finding and Filtering Information for Children,
    - One page version: http://dx.doi.org/10.1145/1835449.1835569
    - Four page version: http://dx.doi.org/10.1145/1840784.1840834

### Recommended Viewing

- Who let the dogs out http://www.youtube.com/watch?v=He82NBjJqf8

## L7 – Information Architecture

This lecture will take a step back from web application development and first consider the impact of web application development from the point of view of Information Architecture i.e. how does this web application fit within the context of the web site / organization.

### Outline:

1. What is Information Architecture?
2. Why Information Architecture matters?
3. Information Architecture Concepts
    a. Complex systems, Invisible work, Knowledge networks, Information seeking behavior
4. Information Architecture Systems
    a. Searching systems, Navigation systems, Semantic networks
5. Information Architecture Deliverables
    a. Wireframes, Blueprints, Dataflow Diagrams, Interaction Diagrams
6. Infamous Three Circles
    a. Context, Content, Users
7. The Anatomy of an Information Architecture
    a. Visualizing IA
    b. IA Components
    c. Browsing aids
    d. Search aids
    e. Contents and Tasks
    f. Invisible Components
8. Top-Down and Bottom-Up approaches to Information Architecture (Morville and Rosenfield, p44,47,388)

### Intended Learning Outcomes:

At the end of this lecture you should be able to:

1. Describe and define Information Architecture
2. Describe the major concepts, systems and deliverables of IA
3. Assess how the different component affect the IA of a site
4. Compare and contrast the different approaches to IA.

### Recommended Reading:

Chapters 1-4 from Morville and Rosenfield
Chapman and Chapman, Chapter 10 – Web Page Design, Chapter 11 – Web Site Design

### Recommended Viewing:

What is an Information Architect (2mins)
http://www.youtube.com/watch?v=nv59MOxrOGA

A short presentation by students on Information Architecture (10mins)
http://www.youtube.com/watch?v=ARFw-Thv1Xk

## L8 – Messaging and Protocols

This lecture will talk about the lifeblood of the Internet programming and Internet technology. This lecture will explain the main protocols that are employed that facilitate communication across the Internet, along with the main mark up language used to wrap communications.

### Outline:

1. Seven Layers ISO model
    a. Internet layer
    b. Transport layer
    c. Application layer
        i. HTTP
        ii. User Agent Protocols
        iii. Other Protocols
2. HTTP
    a. Request and response Messages
    b. Get, Post, Head, etc methods
    c. MIME and the 8 content types
    d. Stateless vs. Stateful protocols
3. XHTML
    a. What is XHTML
    b. Difference between HTML and XHTML
    c. Syntax
        i. Elements
        ii. Attributes
        iii. Basic Structure
        iv. Layout
    d. Other Markup Languages
4. Inter-Process Protocols
    a. Remote Procedure Calls
    b. REST
    c. XML-RPC
    d. SOAP

### Intended Learning Outcomes:

At the end of this lecture you should be able to:

1. List the main three layers of internet Seven layers ISO Model and describe their purpose
2. Describe and explain a number of protocols
3. Describe the main methods used in HTTP
4. Explain the request-response interaction
5. Compare and contrast state and stateless protocols
6. Recognize key elements, attributes and structures within XHTML
7. Differentiate between Inter-Process Protocols

### Recommended Reading

- Chapman and Chapman, Chapter 2 – Web Technology (p35-69)
- Chapman and Chapman, Chapter 3 – Markup

## L9 – Client-side Environment

This lecture will introduce, define and discuss the client-side environment of web applications.

### Outline:

1. Client-side in terms of the N-Tier architecture
2. Differences from *traditional* desktop software
3. Components of a Web Browser
    a. Document Object Model (DOM)
    b. Event Model
        i. Observer Pattern
        ii. Event Propagation
4. Impact of web browser on development
5. Future of the web browser

### Intended Learning Outcomes:

At the end of this lecture you should be able to:

1. Explain the client-side environment of web apps
2. Identify the major components of a web browser
3. Discuss how the components affect web application development
4. Assess the benefits and limitations of the web browser
5. Debate the future of the web browser/application

### Recommended Reading

- Chapman and Chapman, Chapter 7 – DOM Scripting

### Recommended Viewing

- An Inconvenient API: Theory of the DOM (parts I-III) - http://video.yahoo.com/watch/111582/992708

## L10 – Cascading Style Sheets (CSS)

This lecture will introduce, define and discuss the cascading style sheets (CSS) and their role within the development of web applications.

### Outline:

1. Syntax and Semantics of CSS
    a. Advantages of CSS and the separation of concerns
    b. Rules of CSS
2. Typography
    a. Fonts
    b. Text Decoration
    c. Text Size
3. Layout
    a. The box model
    b. Borders
    c. Padding
4. Positioning
5. Examples

### Intended Learning Outcomes:

At the end of this lecture you should be able to:

1. Explain the reasons for using CSS
2. Specify the style and appearance of text
3. Specify and control the layout of elements within in a page
4. Describe the CSS box model

### Recommended Reading

- Chapman and Chapman, Chapter 4 –Style sheets
- W3C's Cascading Style Sheets page: http://www.w3.org/Style/CSS/
- CSS Tutorial, that contains examples, quizzes, and a reference guide: http://www.w3schools.com/css/default.asp
- CSS Zen Garden: http://www.csszengarden.com/

### Recommended Viewing

- You Gotta Keep'em Seperated (3mins)
  http://www.youtube.com/watch?v=eFwkv14u3b4

## L11 – Client-side Scripting - JavaScript

This lecture explains how the browser can be manipulated programmatically with client-side scripting.

### Outline:

1. Client-side Scripting
2. Manipulating the Document Object Model (DOM)
3. Overview of JavaScript

### Intended Learning Outcomes:

At the end of this lecture you should be able to:

1. List what can be achieved using DOM manipulation
2. Explain how the DOM can be manipulated
3. Compare JavaScript to other programming languages

### Recommended Reading

- Chapman and Chapman, Chapter 7 – DOM Scripting
- Eloquent JavaScript - http://eloquentjavascript.net/contents.html
- JavaScript: The world's most misunderstood programming languages - http://javascript.crockford.com/javascript.html

### Recommended Viewing

- The JavaScript Programming Language (Douglas Crockford) - http://www.learnwebdesignonline.com/videos/programming/javascript/yahoo-douglas-crockford

## L12 – JQuery / Assignments / Giving Feedback

This lecture continues on from the previous lecture on Client Side Scripting and introduces JQuery. JQuery is javascript library that makes life on the client side a lot easier. This lecture will also go over your assignments and discuss ways to provide constructive feedback on reports.

### Outline:

1. Improving JavaScript development using jQuery
   a. Motivation for jQuery
   b. Using jQuery: Patterns and Practices
   c. Integrating jQuery into Django
2. Assignments
3. Giving and Receiving Feedback

### Intended Learning Outcomes:

At the end of this lecture you should be able to:

1. Describe the basic pattern of the JQuery Language
2. Assess the benefits and limitations of using jQuery
3. Provide constructive and professional feedback on specifications and designs

### Recommended Viewing

- Intro to jQuery (John Ressig) - http://tv.adobe.com/watch/ajax-experience-2008/intro-to-jquery-by-john-resig/
- JavaScript and jQuery (John Resig) - http://www.youtube.com/watch?v=GKfHdOrR3lw&feature=related

## L13 – Introduction to XML and XHTML

This lecture will provide an overview of XML and XHTML introducing the main XML concepts.

### Outline:

1. What is XML?
2. Overview of XML and its Structure
    a. Elements
    b. Attributes and Attribute Types
    c. Entities
2. Overview of XHTML
    a. Frameset, Transitional and Strict
    b. Elements, Attributes, etc
    c. Structure and Layout
3. Alternative: JSON
    - JSON introduction
    - Differences, pros and cons

### Intended Learning Outcomes:

At the end of this lecture you should be able to:

1. Identify XML and describe its structure
2. Analyze an XML document and explain the elements, attributes and entities
3. Explain the differences between the different flavors of XHTML
4. Explain the differences between XHTML and HTML: and point out the advantages of XHTML
5. Discuss alternatives to XML and the advantages and disadvantages of XML and alternatives.

### Recommended Reading

- Chapman and Chapman, Chapter 3 - Markup
- XML Tutorial: http://www.tizag.com/xmlTutorial
- What is XML Tutorial: http://www.w3schools.com/xml/xml_whatis.asp
- A Technical Introduction to XML: http://www.xml.com/pub/a/98/10/guide0.html
- XML in a Nutshell: http://oreilly.com/catalog/9780596007645/index.html

## L14 – XML DOM and SAX Parsing / Processing XML

This lecture will review the different models for parsing XML i.e. DOM and SAX. This lecture will also include examples where we parse web results from Yahoo! BOSS API. The examples processing search results will be in Python using DOM and SAX.

### Outline:

1. The Simple API for XML (SAX)
2. The Document Object Model (again)
3. How (or how not to) parse XML
    a. Example processing Yahoo! BOSS API Results
    b. In DOM
    c. In SAX
4. DOM vs SAX

### Intended Learning Outcomes:

At the end of this lecture you should be able to:

1. Describe DOM and SAX
2. Compare and contrast DOM and SAX models
3. Process XML using DOM

### Recommended Reading

- Should I use DOM or SAX? http://developerlife.com/tutorials/?p=28
- http://oreilly.com/catalog/pythonxml/chapter/ch01.html
- A good overview of parsing xml in python: http://diveintopython.org/xml_processing/
- Discussion on DOM vs. SAX: http://www.xml.com/pub/a/2001/11/14/dom-sax.html
- Parsing XML with SAX and Python Tutorial: http://www.devshed.com/c/a/Python/Parsing-XML-with-SAX-and-Python/

## L15 – Advanced XML

This lecture will describe some of the advanced XML concepts and features. This lecture is mainly to provide an overview the breadth of additional services and advanced functionality that XMLites have developed.

### Outline:

1. XML Schema
2. Data Types
    a. attributes
    b. Complex types
    c. child elements
3. The X's
    d. XPointer
    e. XPath
    f. XQuery
    g. XSLT
4. XML and Databases
    h. data-centric and document-centric
    i. Custom Databases for XML
    j. Native Databases

### Intended Learning Outcomes:

At the end of this lecture you should be able to:

1. Describe XML Schema and how it is different from DTDs
2. List and describe the different X's, their use and role in XML processing
3. Explain the different ways to transport, store and access XML data

## L16 – AJAX

This lecture will introduce the set of technology collectively known as AJAX, discussing its impact on web application development and user experience.

### Outline:

1. The rise of web applications
   a. Background
   b. Features
   c. Examples
2. History of Ajax
3. Asynchronous vs. Synchronous models of communication
4. Overview of Ajax using jQuery
   a. Comparison with JavaScript
   b. Handling data formats (XML and JSON)
   c. Ajax/jQuery patterns
5. Debate: Web Apps vs. Desktop Apps

### Intended Learning Outcomes:

At the end of this lecture you should be able to:

1. Explain the emergence of web applications
2. Identify what differentiates a web application from a web page and a traditional application
3. Describe the components of Ajax
4. Illustrate the differences between synchronous and asynchronous communication in the context of web apps
5. Debate the future of web application development

### Recommended Reading

- Ajax: a new approach to web applications - http://www.adaptivepath.com/ideas/essays/archives/000385.php

## L17 – Web Accessibility

This lecture will discuss web accessibility issues from the point of view of the Web Accessibility Initiative.

### Outline:

1. Problems with accessing web content
    a. Vision
    b. Hearing
    c. Movement
    d. Cognition
    e. Age-related
    f. Normal
2. Catering for different users
3. Common Mistakes
4. Web Accessibility Initiative
    a. Overview
    b. WAI Content Accessibility Guidelines
    c. Philosophy: Leave no user behind
5. Assistive Technologies
6. How to make web sites more accessible

### Intended Learning Outcomes:

At the end of this lecture you should be able to:

1. Recognize and list the main problems users face when accessing content
2. Describe the main philosophy behind the WAI
3. List and describe the different types of assistive technology
4. List the main ways in which content/functions can or should be made to be accessible
5. List the common mistakes that should be avoided

### Recommended Reading

- Chapman and Chapman, Chapter 9 – Web Accessibility
- Web Accessibility Initiative, http://www.w3.org/WAI/
- WAI Guidelines and Techniques, http://www.w3.org/WAI/guid-tech.html
- Wikipedia Overview of Web Accessibility, http://en.wikipedia.org/wiki/Web_accessibility
- http://www.toptenblogtips.com/10-misconceptions-about-web-accessibility/

### Recommended Viewing

- An Introduction to Web Accessibility, (30 mins) http://www.viddler.com/explore/boagworld/videos/6/
- The world denied (3 mins), http://www.youtube.com/watch?v=SRNdTIFkG1w
- Through the eyes of a screen reader, (5mins), http://www.youtube.com/watch?v=bQpNYDvQ010
- WebAnywhere: http://webanywhere.cs.washington.edu/

## L18 – Ambient Findability

This lecture will introduce and define Ambient Findability and how it is important in the development of web sites and web applications.

### Outline:

1. what is ambient findability
2. Information Interaction
    a. what is information
    b. information retrieval
    c. the people problem
    d. information interaction
3. The User Experience honeycomb
    e. Useful
    f. Usable
    g. Desirable
    h. Findable
    i. Accessible
    j. Credible
    k. Valuable
2. Push and Pull
    a. Design
    b. Design Mantra: You are not the user, The experience is the brand, You cant control the experience
    c. Web site real-estate
    d. Findability Hacks
    e. SEO
    f. The importance of search
    g. Five Stage Model of consumer buying process
    h. Consumer decision making process

### Intended Learning Outcomes:

At the end of this lecture you should be able to:

1. Define and describe ambient findability
2. List the main factors that affect the users experience
3. Describe the user experience honeycomb
4. Evaluate a site taking into consideration the user
5. Argue the role of search in the web application development

### Recommended Reading

Ambient Findability, Peter Morville, Chapter 1,3,5

### Recommended Viewing

- Ambient Findability (60 mins), http://video.google.com/videoplay?docid=-8085529791307393357#
- The future of Search (50 mins), http://vimeo.com/9687950

### Other links:

http://findability.org/

## L19 – Web Services

Web Services are software components "designed to enable interoperable machine-to-machine interaction in networks". [W3C] This lecture will provide an overview of web services and the associated technologies. The standard web services model as well as the life-cycle according to the W3C definitions will be explained. Furthermore, we will discuss competing technologies such as SOAP and REST and outline possible benefits and drawback of each technology by looking at sample implementations using web services, such as Amazon's Product Advertising API.

### Outline:

1. What are Web Services?
    a. Definition of Web Services
    b. Importance of Web Services
    c. Benefits of using Web Services
2. Standard Web Service Model
    a. SOAP – Information Exchange
    b. UDDI – Discovery and Integration
    c. WSDL – Description
3. Competing "Technologies"
    a. Document-Oriented and Procedure-Oriented
    b. SOAP vs. REST
4. Example Implementation of Web Services
    a. Consuming a Web Service with SOAP
    b. Consuming a Web Service with REST

### Intended Learning Outcomes:

At the end of this lecture you should be able to:

1. Describe the role of a web service
2. Illustrate the standard web services model and describe associated technologies
3. Explain the process for the creation and life cycle of a web service according to the standard definition of W3C
4. Debate the differences between SOAP and REST, and when to use them.

### Recommended Reading

- Web Services architecture overview (IBM)
  http://www.ibm.com/developerworks/webservices/library/w-ovr/
- Web Services: Principles and Technology by Michael Papazoglou, Addison Wesley, ISBN139780321155559
- Web Services: Concepts, Architectures and Applications (Data-centric Systems and Applications) by Gustavo Alonso, Fabio Casati, Harumi Kuno, and Vijay Machiraju
- W3C - Web Service Architectures
  http://www.w3.org/TR/ws-arch

## L20 – Summary

This lecture will recap some the main points of the course, review the structure of the exam, followed by questions and answers.

### Outline:

1. Summary
2. Exam structure
3. Questions and Answers