15-437/15-637 Fall 2014 Final Exam Study Guide

This guide is designed to be a list of topics that might appear on the final exam. Most (if not all) of the answers to these questions are in the lecture videos or the in-class notes, and the questions appear roughly (but not completely) in chronological order based on the course.

Historically, the course final exam typically consists of open-ended questions, short-answer questions, detailed implementation-based questions that require you to produce software artifacts, and sometimes multiple-choice questions. The questions in this study guide are characteristic of the open-ended and short-answer questions that might appear on the exam.

1. What is HTTP? What problem(s) does it solve? Describe the contents/parts of an HTTP request and response. Know the differences between the various HTTP methods and when each method should be used. Know the parts of a URL.
2. Understand the basic principles of internet networking. What is a MAC address? An IP address? A DNS hostname? Describe the process, in detail, of sending a request from a client computer to a web server and retrieving a response.
3. Understand the basic functions of a web server for both static and dynamic (i.e., generated from web applications) content. For dynamic content, understand the typical division of functionality between the web server, the application server, and the database system.
4. What is HTML? What advantages does it have over a plain-text representation of data? Know how to write basic HTML and compare the style and function of relatively sophisticated HTML. Understand the use of the HTML and tags.
5. What is CSS? What advantages does CSS offer over including style information directly into the HTML document? Know how to write basic CSS rules.
6. What is Twitter Bootstrap? What problems does it solve? Know at a high-level how Bootstrap works.
7. What are HTML hidden fields? Cookies? Sessions? What advantages does each method have over the others? For each method give an example of when that method is clearly better than the others. Know how sessions are typically implemented.
8. What is Model View Controller? Understand the responsibilities of each component of the MVC architecture. What advantages does MVC have over unstructured web applications? Why is MVC particularly well-suited to web applications? Given an MVC implementation (such as the Django framework) what are the disadvantages of that MVC design?
9. Know many in-depth details of Python and the Django framework, including project configuration, URL dispatch, actions, Models, the ORM query API, templates and the template language, static content, Forms, ModelForms, request and response features such as parameters, cookies, sessions, and HTTP header manipulation, reverse URL resolution, transactions, and the built-in authentication system. Understand the data interaction of a typical Django application. If given sufficient time, you should be able to design and write reasonable pseudocode for a complete Django application, end-to-end, using all of the features above. You should also be able to describe, in prose and in reasonable pseudocode, how features of the Django framework are implemented.
10. What is a MIME type? Know how to detect and set the MIME type for HTTP requests and responses, both for the HTTP protocol itself and within Django.

11. What is a multi-part HTML form? Understand high-level details of how multi-part form data is transmitted in an HTTP request and how multi-part form data can be read within a Django application.
12. What is JavaScript? What advantages are there to using JavaScript to manipulate a web page rather than sending all requests to the web server? What problems are there when using JavaScript to manipulate a page? What factors in modern browsers can make writing JavaScript challenging?
13. What is the Document Object Model? Know basic DOM traversal and manipulation in either raw JavaScript or jQuery.
14. What is Ajax? Describe, in detail, the technologies and operations used to make an Ajax application. What are the advantages and disadvantages of using Ajax as compared to synchronous requests? What are some advantages and disadvantages of using Ajax as compared to Adobe Flash, Java Applets, or other interactive client-side runtime environments? Be able to write pseudocode for a simple Ajax application in either raw JavaScript or jQuery.
15. What is jQuery? What problems does it solve? Understand jQuery selectors well enough to understand others' jQuery code to traverse and manipulate DOM elements, and be able to write jQuery pseudocode to traverse and manipulate DOM elements.
16. What are the advantages of writing an application as a web application, as opposed to a client-side application installed on a user's computer?
17. What advantages does a server-side web application have over a client-side web application?
18. What is a race condition? What does it mean for code to be "thread-safe"?
19. Understand the advantages of encapsulating data management logic as methods of the data (Model) classes, compared to embedding the data management directly into the application's (actions) logic.
20. What advantages does a database offer over storing data directly in the filesystem? The filesystem over using a database?
21. What is a B-tree? What is a heap file? What advantages and disadvantages does a B-tree have over other storage methods?
22. Understand the basic structure of a relational database and the relational model.
23. What is SQL? Be able to write basic SQL statements including simple SELECT, INSERT, UPDATE statements.
24. What is a SQL join? Be able to write basic SQL statements executing a SQL join.
25. Within a database, what is a primary key? What is a secondary index? What are the advantages and disadvantages of keeping a large number of secondary indexes on a relation?
26. What is a transaction? Understand basic terminology related to transactions (begin, commit, abort/rollback, etc).
27. What is serializability? Understand the basic ideas of transaction isolation levels.
28. What are the ACID properties? Understand how these properties are implemented in a modern database system.
29. Understand the relative costs of interacting with a typical database system, such as the cost of initially connecting to a database, the cost of executing a simple query on a database, and the cost of executing a simple update on a database.
30. You should understand some basic principles of relational database design, to be able to compare and contrast two different data layouts proposed for a data design problem.
31. For each of the ACID properties, name a technology or technique commonly used to guaranteed that property within a database system.
32. What is write-ahead logging? Understand the sequence of operations for transactional reads and writes with write-ahead logging, as well as how the log is used to recover the correct

database state during system crashes or transaction rollbacks. Explain how write-ahead logging can increase the performance of a transactional system even though it increases the number of writes and amount of data written.

33. What is two-phase locking? Explain the sequence of operations for transactional reads and writes with two-phase locking. How is two-phase locking typically enforced?
34. What is deadlock? Describe two methods to avoid deadlock when two-phase locking is used.
35. Understand fundamental aspects of object-relational mapping tools. What advantages and disadvantages are offered by an object-relational mapping tool over directly writing the SQL? Be able to convert specific Django query API calls into SQL statements and vice-versa, for simple SQL statements such as the kind you are expected to know above.
36. What is unit testing? Understand how to use basic features of the unit testing framework within Django.
37. What problems can be caused by a user embedding HTML tags in an input field? Understand what features of the Django framework usually prevent HTML injection attacks.
38. What is a Cross Site Request Forgery (CSRF) attack? How does the Django framework help prevent CSRF attacks?
39. What problems can be caused by a user embedding SQL syntax characters in an input field? Understand what features of the Django framework usually prevent SQL injection attacks, and what features of the Django framework require you to manually prevent SQL injection attacks.
40. What is a one-way function? Describe how one-way hash functions can be used to provide security for data.
41. What is a salt?
42. What is network sniffing? Spoofing?
43. Understand the details of public key cryptography, including public and private keys, encryption, and digital signatures.
44. Understand the details of private key cryptography. What are the advantages and disadvantages of public key vs. private key cryptography. How does a hybrid cryptosystem address the shortcomings of public and private key cryptography?
45. What is SSL/TLS? HTTPS? What properties does SSL/TLS provide? Understand, in detail, how SSL/TLS works and each step of the SSL/TLS protocol.
46. What is an X.509 certificate? What is the most relevant information it contains, and how is this information used to guarantee the authenticity of a certificate?
47. Understand the strengths and weaknesses of SSL/TLS for web security. How can its security be violated?
48. What is a Certificate Authority, and what function do they provide? How does a web client verify the authenticity of a Certificate Authority? Understand the idea of a chain of trust and Public Key Infrastructure.
49. Describe distinct methods to distribute user requests evenly among a collection of static web servers. For each method, what are the advantage and disadvantage of that method as compared to the other methods?
50. What are the advantages and disadvantages of static web caching as compared to distributing HTTP requests among a collection of static web servers at the content provider?
51. What is a Content Distribution Network (CDN)? What are the advantages and disadvantages of using a CDN as opposed to a traditional collection of web servers and ad-hoc static web caches?
52. Why is scaling web applications harder than scaling static web content? Describe one common method to distribute requests evenly among a collection of web and application servers. What is the problem with this common method?

53. Understand database partitioning, and how it can increase the performance of a database server. What problems are caused by partitioning a database?
54. What is database replication, and how does it increase database scalability? What problem does replication have?
55. What is consistent hashing? Understand how consistent hashing can be implemented by partitioning a ring-like hash space into buckets that store contiguous ranges of hash values.
56. What is memcached? Understand the basic architecture of a memcached cluster and the advantages and disadvantages of requiring client-side libraries to manage a cache cluster.
57. How can increasing the concurrency of requests increase database performance? What problems can increased concurrency cause?
58. Why should one tune the performance of their web application? Why shouldn't one tune the performance of their web application?
59. What is a performance profiler?
60. How do you determine which parts of your web application that you should tune? Why is it important to test and tune your web application at high system loads? What are the common resource bottlenecks? Of these, which resource is usually the bottleneck for web applications? What is the 80/20 (or 90/10) rule?
61. How might you try to increase the performance of an I/O-bound web application?
62. What factors influence how long it takes to read or write a program's instance variable? What factors influence how long it takes to read or write a data element in a database?
63. How can increasing the concurrency of a web application eliminate I/O as a performance bottleneck? What are the advantages and disadvantages of batching I/O operations together for scalability and performance?
64. Describe several approaches to reduce browser-to-server network I/O.
65. Aside from increasing the number of concurrently-executing threads, how can one increase the effective concurrency of a web application?
66. What is accessibility? Understand several basic strategies for making a device-independent web page accessible.
67. What is a think-aloud study? Understand the usefulness of think-aloud studies for obtaining detailed feedback on the usability of an application, and know how to run a basic think-aloud study.
68. Compare advantages and disadvantages of common modern web application server deployment methods, including local deployment, Heroku, and Google App Engine and/or AWS.
69. What is a version control system? You should know the basic features of Git and how you can use Git to manage a concurrent development process among a large team of developers.
70. Compare and contrast Google's App Engine DataStore data model to relational systems. What are the limitations of the DataStore? What are its strengths? What are entity groups, and what is an ancestor query?
71. Describe the architecture of Google's App Engine DataStore, including its relationships to BigTable.
72. Describe the architecture of Google's BigTable, including its relationships to GFS and Chubby. What are the strengths and limitations of GFS? What are the strengths and limitation of Chubby? How does BigTable achieve its design goals given those limitations?
73. Describe how reads and updates are implemented in a log-structured merge tree.