CS 350 – Software Engineering

October 18, 2017

Week 9, Class #25

Detailed Design

Mark Seaman

MWF – 10:00-11:30 - Kepner 0095F

This Week – Design

- Monday, 10-16
  - Lecture – Architecture
- Wednesday, 10-18
  - Lecture – Detailed Design
- Friday, 10-20
  - Lecture – Data Design

Next Week

Code

Version Control Exercise

- Create a folder in the Exercise Results with your BearID
- Upload your Project Plan and Technology Plan to your folder
- Start on your Design Plan

Design

- ✧ Problem solving process
- ✧ Suppress most details to focus on the critical ones

- ✧ Stages of Design
  - Architecture
  - Interfaces
  - User Experience – UX
  - Data

Design

✧ Architecture

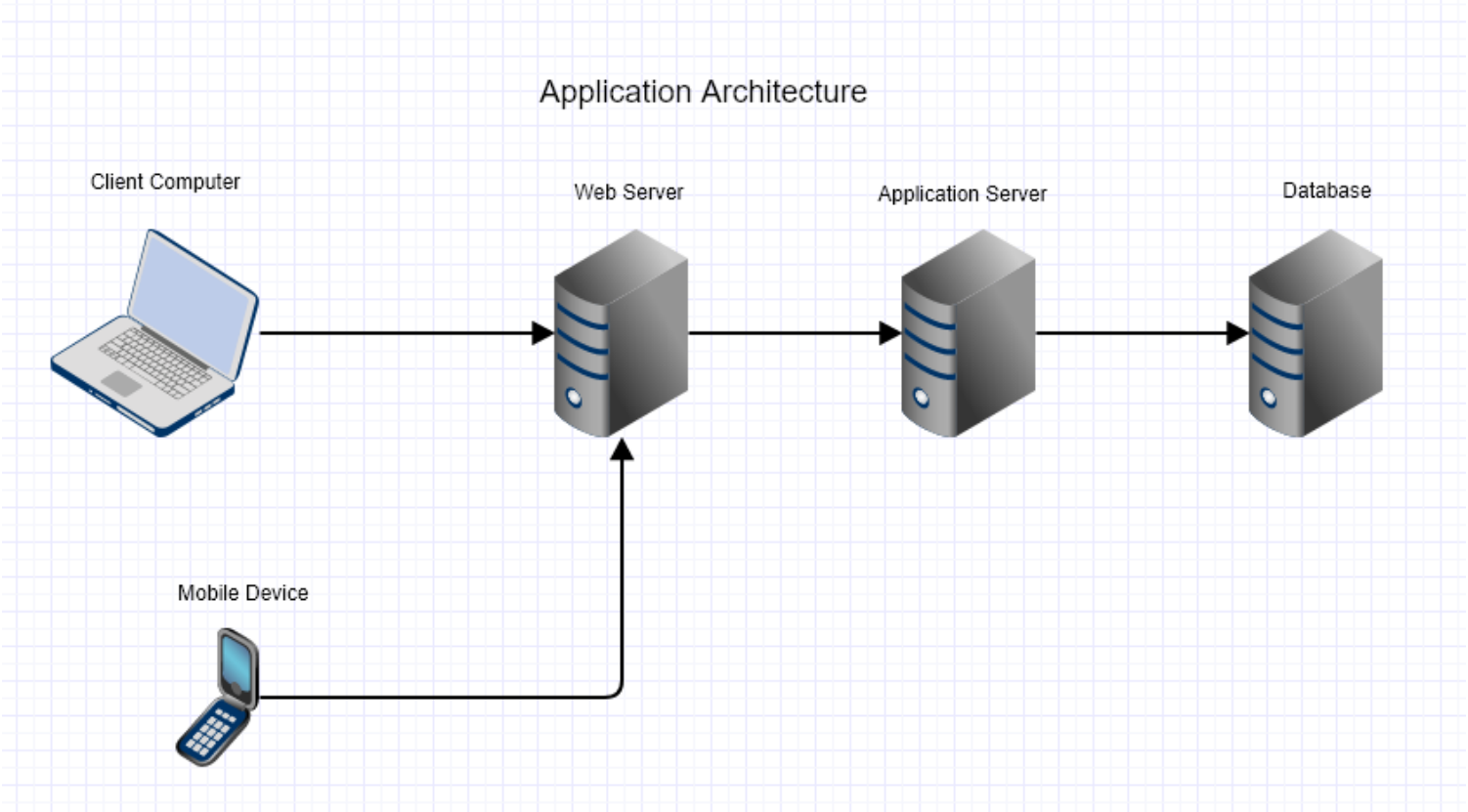
- System partitioning
- Major components
- Data flow diagrams
- Control flow

✧ Interfaces

- Each interface is described as a function stack
- Define the signatures

✧ Data

App Architecture



Architecture Process

- ✧ Partition the system
- ✧ Define interfaces
- ✧ Create a test plan
- ✧ Manage dependencies

World Press Architecture

- Computer client - Web browser
- Web server - http, urls to web pages
- Email server - SMTP
- Application server – functions
- Database - CRUD

Web Server Interface – high-level

- Login
  - register (name, email, password)
  - login (name, password)
- Author
  - add_article (user, title, body)
  - edit_article (user, title, body)
  - delete_article (user, title)
- Reader
  - get_article (title)
  - list_article (user)

Web Server Interface – high-level

- ✧ News (ignore this for now)
  - register (email)
  - verify_email (token)
  - unsubscribe (email)

Power of Wishful Thinking

- ✧ Top-down design
- ✧ Bottom-up construction
- ✧ Middle-out testing

Web Server Interface - Functions

✧ Login

- def register (name, email, password):
  - return render("register.html", name, email, password)
- def login (name, password):
  - return render("login.html", name, email, password)

Web Server Interface - Functions

✧ Author

- def add_article (user, title, body):
  - return render("article_add.html", user, title, body)
- def edit_article (user, title, body):
  - return render("article_edit.html", user, title, body)
- def delete_article (user, title):
  - return render("article_delete.html", user, title)

Web Server Interface - Functions

✧ Reader

- def get_article (title):
  - return render(“article_get.html”, title)
- def list_article (user):
  - return render(“article_list.html”, user)

Data Design - CRUD

- User
  - name, email, password
- Article
  - user, title, body
- Subscriber
  - email, user