CS 350 – Software Engineering

October 23, 2017

Week 9, Class #27

Dev Loop

Mark Seaman

MWF – 10:00-11:30 - Kepner 0095F

This Week – Code

✧ Monday, 10-16

- Lecture – Dev Loop

✧ Wednesday, 10-18

- Lecture – Pair Programming

✧ Friday, 10-20

- Lecture – User Stories

Last Week

Design

Code

Next Week

Test

Upcoming Actions

- Markdown Exercise 10/20
- Design Plan 10/23
- Development Exercise 10/25
- Exam on Friday, 10/27

Test-Driven Development

- Each feature
  - Select a feature
  - Write a failing test
  - Write just enough code to pass test
  - Refactor until beautiful

Development Exercise

✧ Setup your development loop

- Edit
- Test
- Commit

✧ Create command-line scripts

- Edit
- Browse
- Github
- Google
- Explore

Development Exercise

✧ Utilities - Files

- Text
- JSON
- CSV
- HTML
- CRUD

Power of Wishful Thinking

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

Development Exercise

✧ Exercise Instructions

- https://github.com/UNC-CS350
- CS350
- Exercises
- Development_Exercise.md

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)

Version Control Exercise

- Create a folder in the Exercise Results with your BearID
- Convert your plan into Markdown
  - Project Plan
  - Technology Plan
  - Design Plan

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)