THE GEORGE WASHINGTON UNIVERSITY

WASHINGTON, DC

8. Project Intro

CSCIW 2541 Database Systems & Team Projects

Gabe

Slides adapted from Prof. Bhagi Narahari, Wood,; and Silberschatz, Korth, and Sudarshan

The Project

A significant part of this course is a large database systems **team** project.

In the project you will design & implement a database system — Full stack development:

- Front End (HTML/CSS & optional Javascript)
- Application server – in Python with Flask
- DBMS backend – MySQL

The project will involve working in teams of 2-3.

University Management System

You will build a web application to manage many aspects of a university

1. Application processing system (APS)

2. Course registration system (REGS)

3. Advising system (ADS)

Application Processing System (APS)

- Think: grad student application to a department
  a. Applicant (registration, etc…) applies
     - Can check status at any point
  b. Recommendation letters are submitted (emulating email)
  c. Multiple reviewers evaluate
  d. A single chair/director decides
     - admit
     - admit with aid
     - reject
  e. Update admission status of applicant

Course Registration System (REGS)

- Students create accounts
- Can add/drop courses
- Class history for students
  - Grades
  - Classes, time, place
- Faculty assign grades
- Retrieve transcript
- Check for schedule conflicts
- Add course offerings in valid slots
  - location conflicts
- PhD students have limited registrations

Advising System (ADS)

- DEGREEMAP-ish
- Student registration
- Assumes students have a class record (manual)
- Students have a class plan
  - checked for completeness
- Faculty advisor views and approves plan
- Graduation audits
  - check if degree requirements are met
  - advisor must still approve
- Alumni tracking

Academic Honesty

- You **cannot** share or view other student’s code outside of your group
  - in the class or not
- Online documentation
  - You can read/use online documentation
  - Use general advice online not specific to your assignment
  - Emulate “real world”
  - You *cannot* use automated assistants to generate/suggest code
- This policy replaces conflicting AH policies for the class

Two Phases: Impl + Integration

Project broken into 2 phases:

- Phase 1: teams build an application assigned to the team
  - Includes reports containing the design
  - Demo of the Application
- Phase 2: Work in new teams to integrate different applications and produce the final project assigned to you
  - Teams chosen by me based on teamwork in Phase 1
  - This requires integration and NOT redesign

Team Project

You **have** to work in teams

- Each team member required to ‘produce’ equitable share ‘product’
- Teamwork will be assessed…
  - Not all team members may get the same grade on the project!
  - You must bring teamwork issues to attention of the instructor

Mentors and Grading

Mentors

- "Mentor meetings" in the schedule
- One of the class periods/week
- Track progress, teamwork

Grades

- Mentor meetings
- Demo

Trello & Github

Team Project….Warnings!

Requirements (for each Phase) will be provided, and your project must meet minimum requirements

- This only gets you a 80% - need to innovate to earn more!

You have to submit a working project

- No partial credit if your project does not work – you get a zero!

What is a working project ?

Must meet all the specifications

Must have correct and complete workflow

- Workflow specified in the assignment

“Reasonable” user interface

- Easy to use
- You have to come up with “easy to use” based on user interfaces you have seen/used.

Extra features – after you have a working system

Working in Teams

Each team works on their assigned project

Team members have to take lead on some aspect of the project

- Workload distribution is important
- Grades for team members can vary based on the project evaluation of each aspect of the “product” you produce

Every team member has to contribute to all aspects of ‘full stack’ development

- You can’t work on just HTML/CSS !!

We will be using some of the weekly class sessions (lecture or lab) for teamwork – but this is not enough to work as a team!

Warning: Team work

Team members must:

- Be respectful
- Divide work up evenly between all members
- Work on both front and backend

Teamwork Rules

Extensive set of rules you must follow

- Communication – early and often
- Respect – responsibility and follow-through
- Planning – everyone on the same page
- Allocating work – clear division of labor/plan
- No procrastination – makes everything else impossible
- Early problem notification – let me help you
- Flexibility – different working styles/schedules
- Professionalism – co-worker not college kid, SEH not dorm
- No type-casts – everyone does interesting work

Teamwork Traps

Extensive set of rules you must follow, and **common traps**

- Radio silence
- “do the documentation” “...project management”
- “I’ll just do it all”
- Altering other’s code without their approval
- Their code, your commit
- Falling behind, but not helping a back-up plan
    - “I’ll have it done tomorrow…”
- Gossip
- Asserting organization – need consensus on, e.g. when to meet

Team Assignment

You will form a team of **2-3 students** or you can be randomly assigned, discuss NOW!

Who wants to be on a 2 person team?

Weekly Tasks Timeline

The Phase1 projects are due early April

- You will give a demo, and work through a series of testing steps

Weekly checkins – required!

- Update your Mentors/Program Managers
  - in the classroom, office hours,….

Important: Asking Clarification Questions

This is a mimicry of a ‘real world’ DB system design and delivery

- Client gives project you have only a few opportunities to clarify
  - You cannot assume client is available 24-7 to answer your emails

Each team is allowed FIVE questions by March 8th

- You must post as an Issue in Github
- You should NOT post your project clarification question to Slack
  - This violates policies

Each team is allowed FIVE additional questions on March 24th

Posts to Slack can only ask about general Python, MySQL, etc. issues and NOT about your project

Project Deployment

Your final project will be deployed on **our** server

You should have a separate development environment

- Install Python+MySQL on your laptop…but this only works for you
- AWS !!

Do all our development on your dev platform, and push final code to production machine

Yes, a bit more work thrown your way…BUT you will learn important new tools/skills

How to complete the project and learn

Start early

- This is a substantial project – waiting to last minute is recipe for disaster

Communicate regularly with the team

Bring team issues to the attention of the instructors ASAP

The project is deliberately open ended in some aspects

- You have to think through and come up with solutions or design decisions
  - Design decisions should be justifiable based on common practices, sample systems, and/or constraints

Testing, Testing, and more testing!

Homework….due after exam

Meet your team

Ensure everyone has AWS and VS Code setup

- Attend Office hours!

Read the project specifications individually BEFORE you meet as a team

- Sometimes “group think” can put blinders on our creativity and ability to identify problems…
  - So individual reading following by team discussion

University Management System

You will build a web application to manage many aspects of a university

1. Application processing system (APS)

2. Course registration system (REGS)

3. Advising system (ADS)

Demo time!

Shopping Cart HW

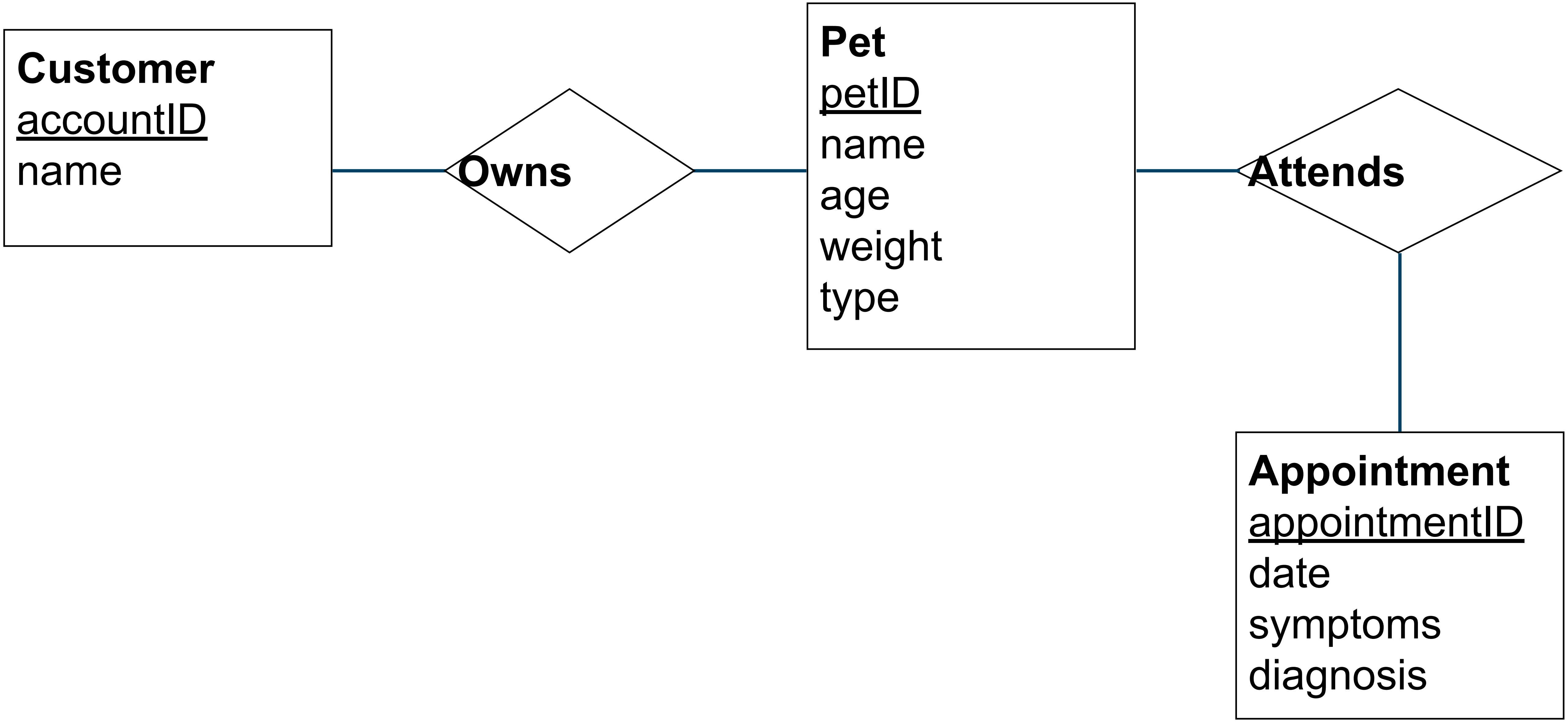
Mini-project to bring together all the technical skills we have covered in Labs and HWs

Build a simple web store. Users can…

- Display products divided up by category
- Search for products
- Add products to a temporary shopping cart
- Checkout to finalize a purchase and track quantities
- View a history of past orders

ER Modeling Reminder

A veterinary clinic wants to track information about its customers (human and animal). Pet owners have a name and account ID. Pets have a name, age, and weight. Whenever a pet comes for an appointment we must record a date, symptoms, and diagnosis.



Customer
accountID
name

Owns

Pet
petID
name
age
weight
type

Attends

Appointment
appointmentID
date
symptoms
diagnosis

Today…..

1. Meet your team members
   - And your mentor!
2. Find a weekly meeting time and submit to mentor/Git
   - on campus (not in a dorm room!)
3. Pick your team name and submit to us by 2pm
4. You have deliverables in a week…so plan your schedule
   - No kidding!
5. Learn some more Github features
   - You will need these for your teamwork….things can go wrong ☺
6. Get some sleep today…tomorrow is an all new day !