CS1320
Creating Modern Web and Mobile Applications

Lecture 36

SPHERE-E Web Architecture

Sphere-E demo

- Show working set usage
- Show scenario usage

Underlying Data

- Given data on products
  - Product information
  - LCA (environmental impact) data
  - As a MYSQL database dump
  - At 3-6 month intervals
- Stored in SQL database
  - Postgresql for test version
  - Mysql for beta version
  - Mysql for production version

Application Data

- Kept in a SQL database (postgresql and mysql)
- Queried and updated as needed
- About 40 tables
  - Initialization script
  - Update script

Boilerplate Files

- Config.js
  - Configuration information
  - Basic language extensions
- Log.js
- Email.js
- Database.js
- Initialize.js
- Renderpdf.js

Front vs. Back End

- Do as much as possible front-end
  - All data is too much
  - Data for one type of product is okay
- Sorting, filtering, selections
  - All done in the front end
  - Values saved by the back end
- New page (consider vs evaluation …)
  - Done in the back end

Dispatch

- Server.js
  - Sessions
  - Logging
  - Middleware (authentication of various sorts)
  - Error handling (lots)
  - Forever to keep it up

Sessions

- Session management
  - Sessions maintained by express (session & cookie), redis
    - Allows sessions to persist over system crashes
    - Easier than creating our own tables
    - Stores all the information
  - Updated by login middleware
  - Updated by explicit login, logout

Authentication

- Auth.js
- Roles and permissions
- Login, registration
- Sessions
- Forget login, forget password
- Display user settings

Authentication Database Tables

- AuthUser
  - User id, password, email,name, address, …
- AuthUserGroup
  - User id -> group id
- AuthPermission
  - List of possible permissions with names
- AuthGroupPermission
  - Groupid -> { permissions }
- AuthUserPermission
  - Userid -> { permissions }
- AuthRolePermission
  - Roleid -> { permissions }
  - Role determined by the current project and user id

Authentication Extras

- AuthAccount
  - List of accounts, each paying for a set of users
  - # setas, start and end date, billing info
- AuthUserAccount
  - Account -> { users }
- AuthAllowUser
  - List of emails of users allowed to register (w/ group, account)
  - With user-specific secret code
  - Set up when email is sent to the user inviting them to join
- AuthForgotPassword
  - User id and secret code (sent to user)

Authentication

- Database stores H(user + H(password))
  - Sent from registration page
- Login sends
  - H(session + H(user + H(password)))
    - Unique for this user and this session; session is random string
  - System computes H(session + <stored password>)
    - To check if password is correct
- System doesn't know user's password

Data Management

- Data.js
  - Read the whole database in from SQL
  - Convert to internal structures
- Get all categories
- Get all products for a category
  - Limit by geography
- Compute benchmarks

Static Pages

- General.js
- FAQ, about pages, home, terms
- Compute FAQ html from simpler form

Project Management

- Project.js
  - Middleware for project commands
  - Project home page
  - New project, edit project
  - Delete project
  - Manage project users and roles
- Activity.js
  - Record and display latest project activities

Working Sets

- Workingset.js
- Middleware for working set commands
- Display working set
  - Consideration, evaluation, cost, generation pages
- Edit working set
  - Set selections, filters, sort order
  - Set properties (amounts, etc)
  - Set product properties (costs)
  - Add comments

Scenarios

- Scenario.js
- Single page for scenario
    - Handle all updates in the front end
- Setting properties
    - Costs, amounts, importance