ASN3

Anonymous Social Network phase 3

Shanky Balani
Brian Burns
Jonathan Grimes
Ben Kornfuehrer
Sang Yun

CS 373: Software Engineering, Fall 2010

Outline

- Demo
  - Browser
  - User
  - Admin
- Issue Tracker
  - Bugs Fixed
  - Outstanding Issues / Future Improvements
- Why it's the best
- What we learned

Browser Demo

- sort data
- search/filter data
- paging
- accordion
- signup

User Demo

- login
- view your data
- rate a class
- edit a rating
- delete a rating
- note updated aggregate scores after each operation
- change password
- logout

Admin Demo

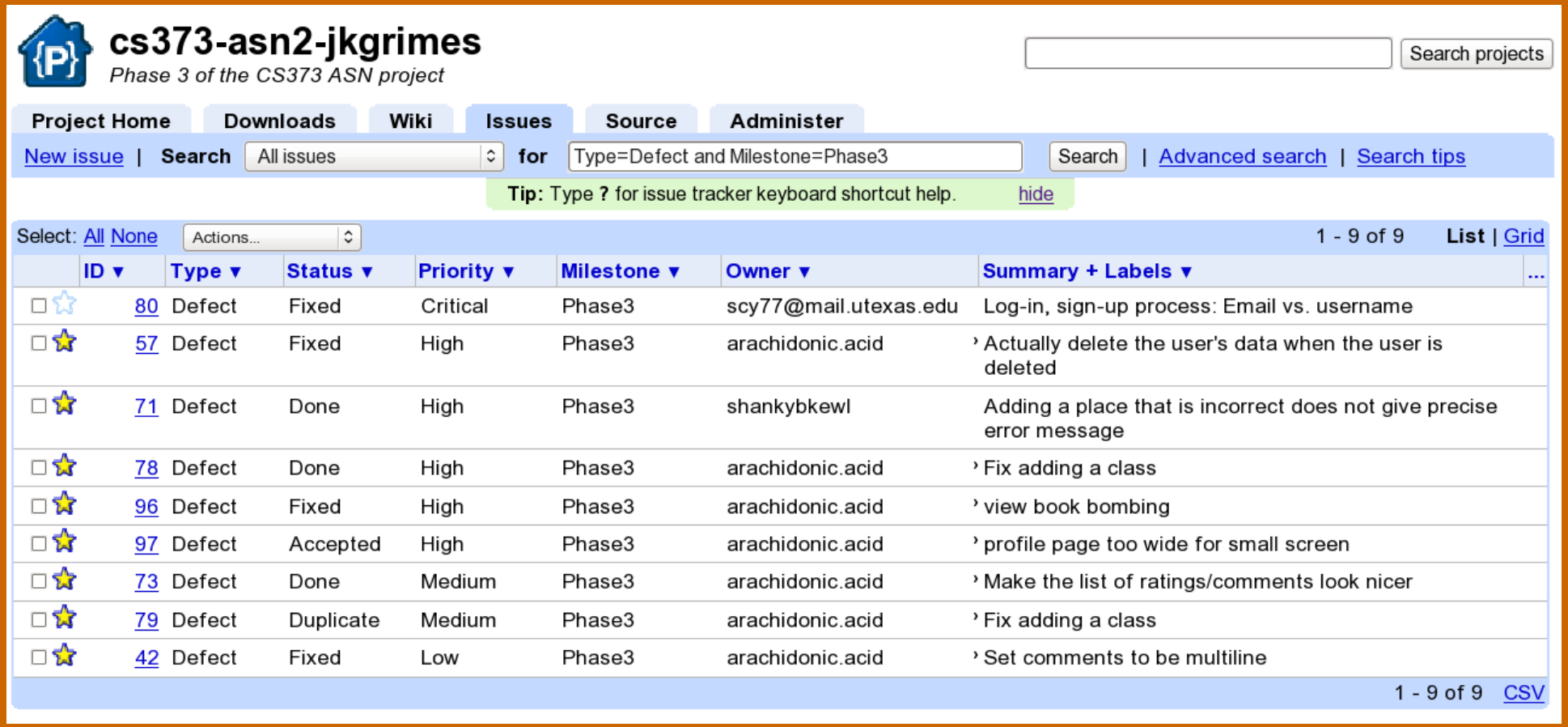
- login as admin
- edit a class
- delete a class
- show list of students
- delete a student

- run tests
- import data
- export data
- clear data
- console

Issue Tracker

- major bugs fixed
- outstanding issues
- future improvements (phase 4)

Issue Tracker: Bugs Fixed



cs373-asn2-jkgrimes

Phase 3 of the CS373 ASN project

Search projects

Project Home | Downloads | Wiki | Issues | Source | Administer

New issue | Search: All issues for Type=Defect and Milestone=Phase3 Search | Advanced search | Search tips

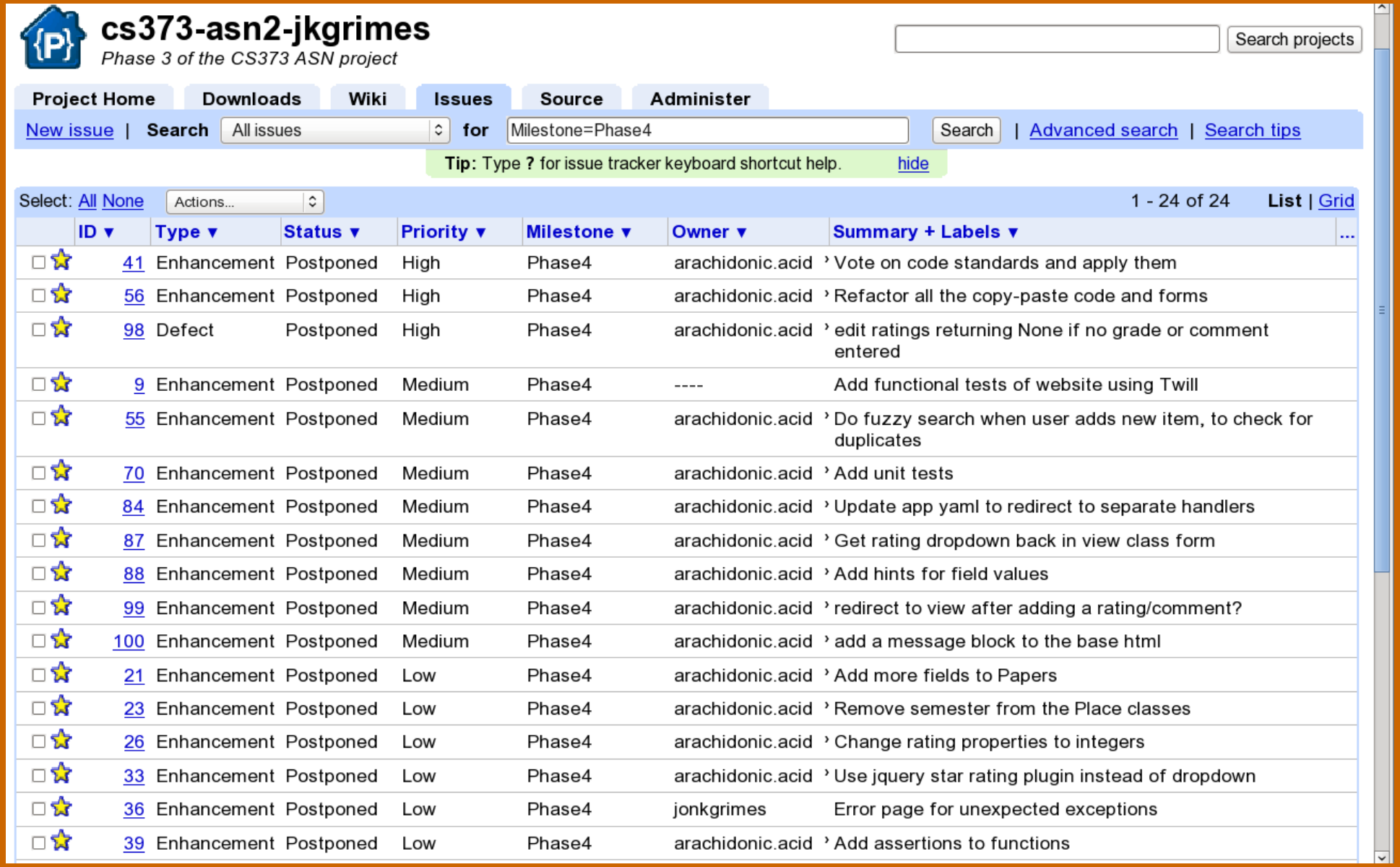
Tip: Type **?** for issue tracker keyboard shortcut help. hide

Select: All None Actions... 1 - 9 of 9 **List** | Grid

	ID ▼	Type ▼	Status ▼	Priority ▼	Milestone ▼	Owner ▼	Summary + Labels ▼ ...
☐ ☆	80	Defect	Fixed	Critical	Phase3	scy77@mail.utexas.edu	Log-in, sign-up process: Email vs. username
☐ ★	57	Defect	Fixed	High	Phase3	arachidonic.acid	Actually delete the user's data when the user is deleted
☐ ★	71	Defect	Done	High	Phase3	shankybkewl	Adding a place that is incorrect does not give precise error message
☐ ★	78	Defect	Done	High	Phase3	arachidonic.acid	Fix adding a class
☐ ★	96	Defect	Fixed	High	Phase3	arachidonic.acid	view book bombing
☐ ★	97	Defect	Accepted	High	Phase3	arachidonic.acid	profile page too wide for small screen
☐ ★	73	Defect	Done	Medium	Phase3	arachidonic.acid	Make the list of ratings/comments look nicer
☐ ★	79	Defect	Duplicate	Medium	Phase3	arachidonic.acid	Fix adding a class
☐ ★	42	Defect	Fixed	Low	Phase3	arachidonic.acid	Set comments to be multiline

1 - 9 of 9 CSV

Issue Tracker: Phase 4



cs373-asn2-jkgrimes

Phase 3 of the CS373 ASN project

Search projects

Project Home | Downloads | Wiki | Issues | Source | Administer

New issue | Search All issues for Milestone=Phase4 Search | Advanced search | Search tips

Tip: Type **?** for issue tracker keyboard shortcut help. hide

Select: All None Actions... 1 - 24 of 24 **List** | Grid

ID	Type	Status	Priority	Milestone	Owner	Summary + Labels
41	Enhancement	Postponed	High	Phase4	arachidonic.acid	Vote on code standards and apply them
56	Enhancement	Postponed	High	Phase4	arachidonic.acid	Refactor all the copy-paste code and forms
98	Defect	Postponed	High	Phase4	arachidonic.acid	edit ratings returning None if no grade or comment entered
9	Enhancement	Postponed	Medium	Phase4	----	Add functional tests of website using Twill
55	Enhancement	Postponed	Medium	Phase4	arachidonic.acid	Do fuzzy search when user adds new item, to check for duplicates
70	Enhancement	Postponed	Medium	Phase4	arachidonic.acid	Add unit tests
84	Enhancement	Postponed	Medium	Phase4	arachidonic.acid	Update app yaml to redirect to separate handlers
87	Enhancement	Postponed	Medium	Phase4	arachidonic.acid	Get rating dropdown back in view class form
88	Enhancement	Postponed	Medium	Phase4	arachidonic.acid	Add hints for field values
99	Enhancement	Postponed	Medium	Phase4	arachidonic.acid	redirect to view after adding a rating/comment?
100	Enhancement	Postponed	Medium	Phase4	arachidonic.acid	add a message block to the base html
21	Enhancement	Postponed	Low	Phase4	arachidonic.acid	Add more fields to Papers
23	Enhancement	Postponed	Low	Phase4	arachidonic.acid	Remove semester from the Place classes
26	Enhancement	Postponed	Low	Phase4	arachidonic.acid	Change rating properties to integers
33	Enhancement	Postponed	Low	Phase4	arachidonic.acid	Use jquery star rating plugin instead of dropdown
36	Enhancement	Postponed	Low	Phase4	jonkgrimes	Error page for unexpected exceptions
39	Enhancement	Postponed	Low	Phase4	arachidonic.acid	Add assertions to functions

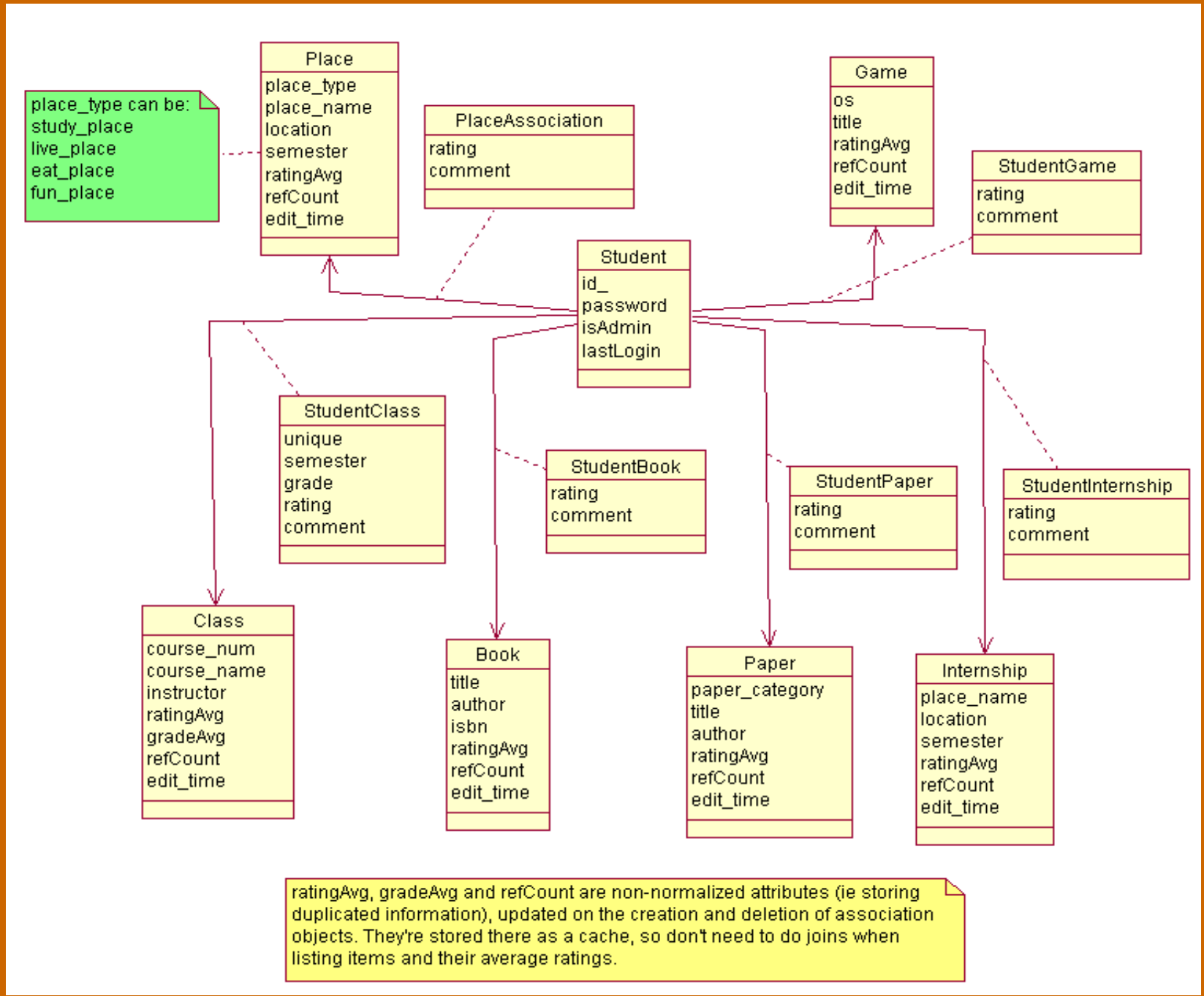
Why ours is the best

- speed - implies scalability
- DataTables plugin - filter, search, sort, paging
- aggregation done at time of input for speed
- lets you rate existing items
- averages over course+instructor, not unique #
- import does semi-fuzzy search for existing items

What we learned

- jQuery is awesome
- using decorators for authentication
- copy-pasting code is bad - better to do the abstraction up front.
- Twill for website testing
- who needs relational databases?
- making websites is not so bad
- do site design up front, with wireframes, etc
- finally figured out how to use Git
- Rails is nicer than AppEngine
- don't need to be tied to AppEngine - can abstract all the data and ui aspects, host same app on Amazon, or use Rails, (see AppScale and Active Cloud DB projects)

UML


place_type can be:
study_place
live_place
eat_place
fun_place
Place
place_type
place_name
location
semester
ratingAvg
refCount
edit_time
PlaceAssociation
rating
comment
Game
os
title
ratingAvg
refCount
edit_time
StudentGame
rating
comment
Student
id_
password
isAdmin
lastLogin
StudentClass
unique
semester
grade
rating
comment
StudentBook
rating
comment
StudentPaper
rating
comment
StudentInternship
rating
comment
Class
course_num
course_name
instructor
ratingAvg
gradeAvg
refCount
edit_time
Book
title
author
isbn
ratingAvg
refCount
edit_time
Paper
paper_category
title
author
ratingAvg
refCount
edit_time
Internship
place_name
location
semester
ratingAvg
refCount
edit_time
ratingAvg, gradeAvg and refCount are non-normalized attributes (ie storing duplicated information), updated on the creation and deletion of association objects. They're stored there as a cache, so don't need to do joins when listing items and their average ratings.

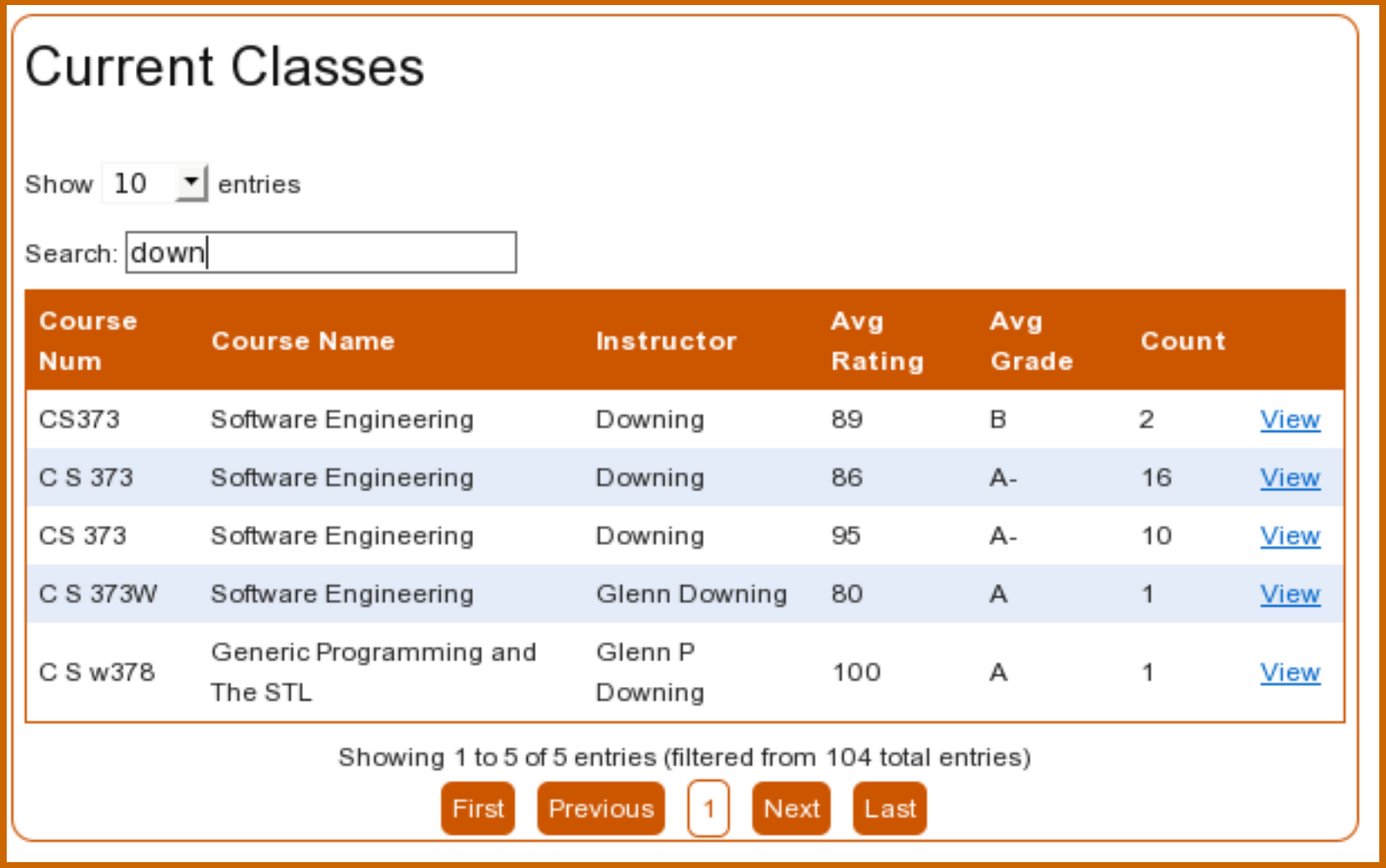
Aggregation

```
class StudentBook(db.Model):
    student = db.ReferenceProperty(Student)
    book = db.ReferenceProperty(Book)
    rating = db.StringProperty(validator=validate_rating)
    comment = db.TextProperty()

    def put(self):
        db.Model.put(self)    # call superclass
        self.getAvg() # update averages

    def getAvg(self):
        book = self.book
        links = book.studentbook_set
        ratings = [int(link.rating) for link in links]
        n = len(ratings)
        ratingAvg = None if n==0 else sum(ratings) / n
        book.ratingAvg = ratingAvg
        book.refCount = n
        book.put()
```

DataTables plugin



Current Classes

Show 10 entries

Search: down

Course Num	Course Name	Instructor	Avg Rating	Avg Grade	Count	
CS373	Software Engineering	Downing	89	B	2	View
C S 373	Software Engineering	Downing	86	A-	16	View
CS 373	Software Engineering	Downing	95	A-	10	View
C S 373W	Software Engineering	Glenn Downing	80	A	1	View
C S w378	Generic Programming and The STL	Glenn P Downing	100	A	1	View

Showing 1 to 5 of 5 entries (filtered from 104 total entries)

First Previous 1 Next Last

DataTables plugin

```
{% extends "../_base.html" %}
{% block scripts %}
<script src="/static/jquery.dataTables.js" type="text/javascript"></script>
{% endblock %}
{% block content %}

<h2>Current Books</h2>
<table id="data">
  <thead>
    <tr>
    <th>Title</th>
    <th>Author</th>
    <th>ISBN</th>
    <th>Avg Rating</th>
    </tr>
  </thead>
  <tbody>
  {% for book in books %}
    <tr>
      <td>{{ book.title }}</td>
      <td>{{ book.author }}</td>
      <td>{{ book.isbn }}</td>
      <td>{{ book.ratingAvg }}</td>
    </tr>
  {% endfor %}
  </tbody>
</table>
<script type="text/javascript">
$(function() {
  $("#data").dataTable({ "sPaginationType" : "full_numbers"});
});
</script>
{% endblock %}
```

Authentication decorators

```
def authenticate(f):
    def check_session(self):
        session = Session()
        if 'student_id' not in session:
            return doRender(self,'not_auth.html')
        else:
            return f(self)
    return check_session

class AddBook(webapp.RequestHandler):
    def get(self):
        form = BookForm()
        doRender(self,'book/add.html',{'form':form})

    @authenticate
    def post(self):
        form = BookForm(self.request.POST)
        if form.is_valid(): # checks values with validation functions
            book = form.save()
            self.redirect("/book/view?id=%d" % book.key().id())
        else:
            doRender(self,'book/add.html',{'form':form, \
                'error':'ERROR: Please correct the following errors and try again.'})
```

Twill

```
# a simple twill test script

# check home page
go localhost:8080
code 200
find "@nonymous Social Network"

# look at list of books
follow Books
code 200
find "Current Books"

# try to login and fail
follow Login
fv 1 1 brian000
fv 1 2 wrongpw
submit
code 200
find "Invalid"
```

Twill ctd

```
# add a book
follow Books
follow Add

# >> showforms
# Form #1
# ## ## __Name_____________________ __Type___ __ID________ __Value__________________
# 1     title                  text      id_title
# 2     author                    text      id_author
# 3     isbn                   text      id_isbn
# 4     _id                    hidden    (None)
# 5  1  None                      submit    (None)       Add Book

fv 1 1 "Twill: The Book"
fv 1 2 "Anonymous"
fv 1 3 "123456789"
submit
code 200
```

Twill ctd

> twill-sh -q test/twill

[runs all the test scripts in the specified folder, alerts you to any errors found]