

Instructions for Spring Quarter Acceptance Testing

General Information

There is a simple report template in the file `testing/acceptance/templates/report.html`. Suppose you are performing acceptance testing during week N for department X. Make a copy of `templates/report.html` into `X/weekSN-report.html`. For example, if you're doing acceptance testing during week 3 for the CHEM department, do this in the `testing/acceptance` directory:

```
cp templates/report.html CHEM/weekS3-report.html
```

To perform your acceptance testing:

1. `svn update` your working project directory
2. fill in the file `finals-week-report.html`
3. commit the file when you're done.

The report template is a very simple HTML file. You can edit the file directly in a plain text editor, or use some form of word processor and save it as an HTML file. In either case, make sure you fill in all seven sections of the report.

The objective of this acceptance testing process is to re-create the Winter 2012 schedule that was published for your department. The title for each section of the report describes the actions to be performed for that section. This is exactly the same acceptance test procedure we did earlier in the quarter.

For each section of the report, if the actions can be performed 100% successfully, then say "All actions performed successfully." If there are any problems, describe them clearly. A "problem" is a bug, sub-standard performance, or any aspect of program behavior that is not 100% satisfactory. Post a Jira issue for any problems, if an issue for that problem has not already been posted.

Details for Each Section of the Report

The following are further specific instructions for each section of the report:

1. **Create a new schedule for Winter 2012.** Save an empty version of that schedule and re-open it. Use this newly-created schedule for the rest of the testing.
2. **Enter all of the necessary resource data** for the Winter 2012 schedule. This includes all instructors, courses, rooms, and preferences.

If you want to force the generation of a correct schedule, you can try to enter specific instructor preferences to ensure that each instructor gets exactly the classes that are scheduled for that instructor in the Winter 2012 schedule. It would be better to enter preferences more broadly, such that Winter 2012 scheduled items are within the preference ranges, but preferences don't dictate a particular schedule.

3. **Generate the schedule.** In addition to any reported problem, describe the quality of the schedule that was automatically generated, in particular:
 - a. Did it contain all classes?
 - b. Were they at reasonable times?
 - c. Were there any scheduling conflicts?

If the exact Winter 2012 was generated directly, say "Winter 2012 schedule successfully generated, with no edits." If there were one or more problems, include links to the Jira issues that were posted for the problems.

4. **Edit the generated schedule** to get as close as possible to the published Winter 2012 schedule. Describe how close you were able to get to the actual Winter 2012 schedule for the department, using the Scheduler program editing functionality. Again, our objective here is to use the Scheduler program to create precisely the schedule that is published by the department for Winter 2012.

After the preceding edits are done, perform some additional edits to test the schedule editing functionality. In particular, move courses around in the schedule to test the incremental constraint and preference validation.

5. **Save the schedule**, exit the program, restart, and re-open the saved schedule. Save a copy of the schedule on a different file, exit, restart, and re-open both files.
6. **Describe any quality measures** that were not achieved during the preceding testing. The quality measures are those listed under the revised Week 5 features list in the project file: `requirements/matrix.html`.

If all quality measures are achieved satisfactorily, say "All quality measures met." If not all quality measures were covered during the preceding testing, then indicate this. For example, if you did not need to use course suffixing for your department, say "Use of course suffixing not required for X department."

7. **Describe any other problems or issues** that were not covered in the preceding testing steps, including overall UI look-and-feel and usability. Include links to any new Jira issues that you post.

Details for Entering Resource Data

To enter the resource data for your department, use the Winter 12 schedule published at schedules.csc.calpoly.edu. The following tables indicate how to convert the data in the `schedules` table to the scheduler program resource pages.

Courses:

Scheduler Program	<code>schedules.calpoly.edu</code>
Department	1st part of Course
Catalog Number	2nd part of Course
Course Name	hover over Course link
# of Sections	max value of Sect for any given course
WTU	4 if no lab with course, 5 if lab with course
SCU	number of units in the catalog for the course
Day Combinations	Days
Hours Per Week	Compute from values in Days, Start, and End
Max Enrollment	ECap
Course Type	Type
Assoc	<i>doesn't work so don't enter anything here</i>

Instructors:

Scheduler Program	<code>schedules.calpoly.edu</code>
Last Name	last part of Instructor
First Name	first part of Instructor
User Name	Goto the Directory Information channel on portal and copy/paste the instructor's name into the search box
Max WTU	12 unconditionally
Disabilities	say yes for exactly one department instructor

Preferences	<p>For courses prefs, put the ones they actually teach in Winter 12 as 'Preferred'; put an even spread of 'Acceptable', 'Not Preferred', and 'Not Possible' the rest of the courses.</p> <p>For times prefs, put two hours around the times that actually got assigned as 'Preferred'; put an even spread of 'Acceptable', 'Not Preferred', and 'Not Possible' the rest of the times.</p>
-------------	---

Rooms:

Scheduler Program **schedules.calpoly.edu**

Location	Location
Type	Type
Max Capacity	LCap
Equipment	Be creative