

Group #10

June 21st, 2016

COMP 3350

John Braico

Group Members:

Ben, Barry, Julius, Connel, Deep

Academic Assistant

Iteration 2 Hand-In

Completed Big User Stories

Select a Term to View (High)

- I want to be able to click on a term and see what courses are in it.
- Cost : 2 days

Detailed User Stories:

See Courses(High)

- I want to be able to see courses when I click on a term.
- Cost : 2 days.

Developer Tasks:

GUI:

- Implement an interface for viewing a term (Estimated 2, real 1 hour)

Persistence:

- Get the HSQLDB to properly store terms (Estimated 3, real 2 hours)
- Update stub database to do the same (Estimated 5, real 3 hours)

Logic:

- Add logic viewing a term (Estimated 2, real 2 hour)

Testing

(Estimated 3, real 3 hours)

Create a Course (High)

- I want to be able to create a new course.
- Cost: 1 day.

Detailed User Stories:

Create Courses(High)

- I want to be able to create a course in a term.
- Cost : 1 day.

Developer Tasks:

GUI:

- Implement an interface for creating a course (inside a Term) (Estimated 2 hours, real 2 hours)

Persistence:

- Get the HSQLDB to properly store courses (Estimated 1, real 1 hour)
- Ensure courses are properly stored “in” their appropriate term (Estimated 1, real .5 hour)
- Update stub database to do the same (Estimated 2, real 1 hour)

Logic:

- Add logic for creating a course (Estimated 1, real 1 hour)
 - o Courses will be added into terms

Testing (Estimated 3, real 2 hours)

Manipulate a Term (High)

- I want to be able to change my term details and delete it.
Cost : 6 days.

Detailed User Stories**Delete Term(High)**

- I want to be able to delete any term that I want to.
- Cost : 3 days.

Edit Term(High)

- I want to be able to change my term’s semester or year.
- Cost : 3 days.

Developer Tasks:**GUI:**

- Implement an interface for removing a term (Estimated 2, real 1 hour)
- Implement an interface for editing a term (Estimated 2, real 1 hour)

Persistence:

- Get HSQLDB to allow editing of term information (Keep changes persistent)
(Estimated 3, real 4 hours)
- Get HSQLDB to allow for persistent removal of terms(Estimated 2, real 2 hours)

Delete a Course (High)

- I want to be able to delete a course
- Cost : 1 day

Detailed User Stories

Delete Course(High)

- I want to be able to delete any course that I want to.
- Cost : 1 days.

Developer Tasks:

GUI:

- Implement an interface for viewing all courses in the term and integrate deletion (Estimated 2, real 1 hour)
- Implement an interface for removing a course from a term (button, integration...) (Estimated 2, real 1 hour)

Persistence:

- Get HSQLDB to allow for persistent removal of courses (Estimated 1 hour, real 1 hour)

Testing

(Estimated 1 hour, real 1.5 hour)

Group Log

Date	Times	Description
May 12th	2:00-3:00pm	Discussed project architecture as a group.
May 13th	1:00-2:40pm	Implemented initial versions of domain specific objects as discussed in previous meeting. (Barry)
May 16th	12:00-1:00pm	Team meeting to discuss delegation of work for Iteration 1
	12:00-2:00pm	Creating prototypes and sketches. (Ben & Julius)
	1:00-2:00pm	Discussed and planned developer tasks for first Big User Story (Group)
	2:00-2:30pm	Starting a prototype GUI through Android Studio

May 17th	2:00-3:00pm	Ironed out SVN connection between group members. Discussed implementation specifics of Objects and for Persistence. (Group)
May 18th	1:30-2:30pm	Finished (for now) Object classes (Barry)
	2:30-3:00pm	Went over the Object classes and added minor changes(Deep).
	2:00-4:00pm	Created mockups including Terms page (Ben)
May 22nd	1:00-2:00pm	Began writing tests for domain-specific objects (Barry)
	12:00-3:00pm	Created Terms UI page (Ben)
	7:00-9:00pm	Attempted to integrate Terms page into menu fragment (Ben)
May 23rd	1:30-3:30pm	Attempted to integrate Terms page into menu fragment (Julius)
	2:00-3:00pm	Wrote more tests for objects.
	6:00 - 7:00pm	Discussed how we are going to break down responsibilities for the GUI (Julius,Ben)
May 24th	2:30-5:15pm	Made Terms page use Terms object, edited Term object, created Semester enum (Ben)
	2:00-4:30pm	Group meeting to work together with clear communication, dealt with issues regarding SVN for some time.
	9:00-10:00pm	Started writing logic class - CalculateGPA alongside tests(Deep)
May 25th	12:00-1:00pm	Discussed database and object code ideas (Ben, Barry, Connel)
May 27th	3:00 - 6:00pm	UI: From the main page created an activity page that will ask the user to

		select a term and a year from a dropdown menu. (Julius)
	4:00-5:00pm	Finished logic and its tests.(Deep).
May 29th	3:00-5:30pm	Integrated the new Term's page with the Term's view page, cleaned up unused UI files (Ben)
	6:30-8:00pm	Debugging some commit error, ended up having to re-check out and redo all my work in steps to be able to commit (Ben)
	7:00-9:00pm	Completing remaining tests(Deep).
May 30th	12:00-2:00pm	Duplicate checking for Term creation (Ben, Julius)
	3:00-5:00pm	Refactored some objects/archetecture, Added sorting to the Terms, Added .equals and .compareTo to the Term object, Added a Comparator for Term (Ben)
	5:30-9:00pm	Changes to business objects as well as setting up tests and working on Course, CourseElement, AcademicObject and Event tests (Barry).
May 31st	11:30-12:30pm	Team meeting (Group minus Julius)
	2:00 - 2:30pm	Changed padding on the spinners, buttons. Attempted changing text size Moved CreateTerm and ViewTerms into Presentation folder (Julius and Ben)
	2:30 - 3:00pm	Team Meeting (Group including Julius)
	3:30-5:00pm	Reorganized the architecture. Connected the GUI-Logic-Persistence together. Made Icon. Made stub

		databased and refactored the Persistence. (Ben)
	8:00-9:00pm	Made spinner text size larger by creating a new layout xml file and increased padding in the listview (Julius)
June 1st	12:00-1:00pm	More tests for various Objects and getting junit 4 "AllTests" to run smoothly. (Barry)
	4:00-6:00pm	Finished all testing(Deep).
June 2nd	10:00-11:00am	Updates to various tests (Barry, Deep)
	11:00-11:40am	Fixed crash that occurred due to adding new terms (Barry, Ben)
June 8th	11:00-12:30pm	Developer tasks for iteration 2 (Barry)
June 9th	2:00-2:40pm	Meeting to discuss/assign workload for iteration 2 (Julius, Barry, Connel)
	3:00-3:30pm	Discussed the use of interfaces and implemented a persistence interface (Connel and Barry)
June 14th	2:00-3:00pm	Met to discuss HSQL database implementation and clarify development priorities / workload (Deep, Connel, Barry)
	2:30-5:00pm	SQLDB initial setup (Connel)
	8:00-9:00pm	Created a new activity in order to delete courses in listView for terms. The delete page activity is now linked with the main term page activity. (Julius)
June 15th	10:30-12:40pm	Fixed ownership of private, mutable variables. Checked for LSP breakage with AcademicObject inheritance. Implemented initial database dependency injection. Began work on ViewCourses and CreateCourse

		interfaces (Barry)
June 16th	2:00-6:00pm	Deleting terms (Ben & Julius)
	11:00-11:30pm	Course comparator (Barry)
	2:30-5:00PM	SQLDB add/remove term (Connel)
June 17th	6:00-10:35pm	Course related GUI components, logic elements and integration as well as testing of new code (Barry & Deep)
	7:00-10:00pm	SQLDB update/add/delete (Connel)
June 18th	10:30-12:00pm	Tests for CourseComparator, skeleton code for all tests in TermLogic and CourseLogic. Small changes to logic classes to facilitate testing (Barry)
	12:00-5:00 pm	Implementing more GUI elements (Deep).
June 19th	10:00-12:00pm	Tests for TermLogic & CourseLogic (Ben)
June 19th	3:30-8:30pm	Large amount of progress on database, dependency injection, GUI (updates to editing and deleting terms / implementing Course related elements) and overhaul to test package (Barry, Connel, Deep)
	12:00-3:00pm	All GUI elements working.
	9:00-10:30pm	SQLDB refactoring code (Connel)
June 20th, 2016	12:30-4:30pm	All testing and AllTests updates completed. Helping with logical issues fixed across the project (Barry).
	1:30-3:00pm	SQLDB fixing small bugs (Connel)
	5:00pm-Midnight	Finishing touches on GUI elements (Deep).
June 21st, 2016	2:00-4:00pm	Final review and touches (Whole Group)

readMe.txt

The electronic submission includes the following files in the zip:

- AcademicAssistant (project folder)
- readme.txt

The project is the AcademicAssistant folder. To run the project, use open project in Android Studio and open the trunk folder. This should open up the project code in Android Studio where it can be built and run.

There are two folders where the source files exist main/java and test/java. main/java has the all project source files that are not test files. It includes the following packages:

1. Application
 - Main.java
 - Services.java
2. Business — *includes all the logic java files including*
 - CalculateGPA.java
 - CourseComparator.java
 - CourseLogic.java
 - TermComparator.java
 - TermLogic.java
3. Objects — *includes all the domain-specific objects*
 - AcademicObject.java
 - Course.java
 - CourseElement.java
 - Event.java
 - Semester.java
 - Student.java
 - Term.java
4. Persistence — *includes the stub database files*
 - Persistance.java
 - SQLDB.java
5. Presentation — *includes all the GUI files*
 - CreateCourses.java
 - CreateCourse.java
 - CreateTerm.java
 - DeleteCourses.java
 - deletePage.java

- EditTermPage.java
- ViewTerm.java
- ViewTerms.java

test/java includes all the JUnit test files. It includes the following packages

1. Business

- CalculateGPATest.java
- CourseComparatorTest.java
- CourseElementLogicTest.java
- CourseLogicTest.java
- TermComparatorTest.java
- TermLogicTest.java

2. Objects

- AcademicObjectTest.java
- CourseTest.java
- EventTest.java
- SemesterTest.java
- StudentTest.java
- TermTest.java

3. Persistence

- PersistenceTest.java
- StubDB.java

AllTests.java

This iteration implemented a number of changes to the features introduced in iteration one. The two user stories that built onto these features were “Select a Term to View or Edit” and “Manipulate a Term”. These give the user the ability to select a term and view the courses inside the term, edit the term and delete the term entirely. The new features implemented in this iteration were the stories “Create a Course” and “Delete a Course”. These allow you to go from viewing a term to adding a new course to the term and also deleting courses from the term. These new features came with matching GUI and logic classes.

Structurally, the implementation of the HSQL database (including the dependency injection) was a major change to the project. This changed any use of the stub database to instead use the implemented Services and Main java files to choose the database you want to use (HSQL by default). This also led to StubDB being moved to the tests package of the project.

Finally, for each of the new logic classes we implemented a new set of matching tests. We also included a full set of tests for the Persistence (by injecting the stub database). This is in the tests package along with the StubDB.

Important! Please note: We have noticed throughout testing the application that in order to ensure that items are successfully added or deleted from the HSQLDB database, you need to give the device (or simulator) a few seconds before closing the app via the app switcher immediately after performing the addition or deletion. We don't know why this issue is occurring exactly,

The group log is attached in the hand in, but to check for periodic updates, the following Google Docs link can be used:

https://docs.google.com/document/d/1_vR5N7kjinzSanXCkzFDOPc9pszXQYUnvqUrrMAaBZ4c/edit?usp=sharing

Our repository can be found at the following link:

svn+ssh://aviary.cs.umanitoba.ca/home/student/yakimchb/softeng_g10/repo