Group #10

July 5th, 2016

COMP 3350
John Braico

Group Members:
Ben, Barry, Julius, Connel, Deep

Academic Assistant

Iteration 3 Hand-In

Completed Big User Stories

Select a Course to view (High)

- I want to be able to click on a course and see what kind of categories it has.
- Cost : 2 days

Detailed User Stories:

See Course categories(High)

- I want to be able to see course categories when I click on a course.
- Cost : 1 days.

See Course grade(High)

- I want to be able to see what grade I have in this course.
- Cost : 0.5 days.

See Course goal grade(Medium)

- I want to be able to see what grade goal I set for this course.
- Cost : 0.5 days.

Developer Tasks:
GUI:
- Implement an interface for viewing a course(Estimated 2, real 1 hour)

Persistence:
- Get the HSQLDB to properly store course
  and course elements.   (Estimated 3, real 2 hours)
- Update stub database to do the same   (Estimated 5, real 2 hours)

Logic:
- Add logic viewing and retrieving a course (Estimated 2, real 2 hour)

Testing   (Estimated 3, real 3 hours)

Create a Course Element (High)

- I want to be able to add a new category to a course.
- Cost: 1 days.

Detailed User Stories:

Create Course Element (High)

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

Developer Tasks:
GUI:

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

Persistence:

- Get the HSQLDB to properly store course elements (Estimated 1, real 1 hour)
- Ensure course elements are properly stored "in" the order of creation(Estimated 1, real .5 hour)
- Update stub database to do the same (Estimated 2, real 1 hour)

Logic:

- Add logic for creating a course element   (Estimated 1, real 1 hour)
  - o   Course elements will be added into courses

Testing   (Estimated 3, real 2 hours)

Manipulate a Course (High)

- I want to be able to able to change my course details and delete it.
    Cost : 3 days.

Detailed User Stories
Delete Course(High)

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

Edit Course(High)

- I want to be able to change my course's name or code.
- Cost : 1 day.

Edit Course(High)
- I want to be able to change my course's name or code.
- Cost : 1 day.

Developer Tasks:
GUI:
- Implement an interface for editing a course(Estimated 2, real 1 hour)

Persistence:
- Get HSQLDB to allow editing of course information (Keep changes persistent) (Estimated 3, real 4 hours)

Manipulate a Course Element (High)
- I want to be able to edit/delete a course category.
- Cost : 2 days

Detailed User Stories
Delete Course Element (High)
- I want to be able to delete any course category that I want to.
- Cost : 1 days.

Update Marks (High)
- I want to be able to update what mark I got in a particular category.
- Cost : 1 days.

Developer Tasks:
GUI:
- Implement an interface for viewing all course elements in the course and integrate deletion (Estimated 2, real 1 hour)
- Implement an interface for removing a course element from a course element (button, integration…) (Estimated 2, real 1 hour)
- Implement an interface for updating a course element (button, integration…) (Estimated 2, real 1 hour)

Persistence:
- Get HSQLDB to allow for persistent removal of course element (Estimated 1 hour, real 1 hour)

Testing   (Estimated 1 hour, real 1.5 hour)

View GPA(Medium)

-       I want to be see my term gpa, and overall gpa for all my terms.
-       Cost : 2 days

Detailed User Stories
View GPA (High)
- I want to be able to see my term gpa
- Cost : 1 days.

View Overall GPA(High)
- I want to be able to see my overall gpa.
- Cost : 1 days.

Developer Tasks:
GUI:
-       Implement an interface for viewing gpa in the term and all terms (Estimated 2, real 1 hour)

Logic:
-       Calculate the correct GPA.(Estimated 2, real 1 hour)

Persistence:
-       Get HSQLDB to allow for retrieval of correct gpa(Estimated 1 hour, real 1 hour)

Testing   (Estimated 1 hour, real 1.5 hour)

View Transcript(Medium)

-       I want to be see all my courses with their grades and my overall gpa.
-       Cost : 1 days

Detailed User Stories
View Courses w/grades (High)
- I want to be able to see all my courses with grades
- Cost : 0.5 days.

View Overall GPA(High)
- I want to be able to see my overall gpa.
- Cost : 0.5 days.

Developer Tasks:
GUI:
- Implement an interface for viewing gpa from all terms (Estimated 2, real 1 hour)
- Implement an interface for viewing all courses with grades (Estimated 2, real 1 hour)

Logic:
- Calculate the correct GPA.(Estimated 2, real 1 hour)
- Get the correct grade for courses (Estimated 2, real 1 hour)

Persistence:
- Get HSQLDB to allow for retrieval of correct gpa(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)
	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)
June 17th	6:00-10:35pm	Course related GUI components, logic elements and integration as well as testing of new code (Barry & Deep)

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	9:00-12:00pm	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.
June 20th, 2016	12:30-4:30pm	All testing and AllTests updates completed. Helping with logical issues fixed across the project (Barry).
	5:00pm-Midnight	Finishing touches on GUI elements (Deep).
June 21st, 2016	2:00-4:00pm	Final tweaks (Whole Group)
June 23rd, 2016	2:30-5:30pm	Modifying the Peristence interface for the methods needed for the next iteration (Connel)
June 28th, 2016	3:30-5:30pm	Finished modifying the Persistence interface and started on the StubDB for the new courseElement methods (Connel)
	8:00-10:00pm	Adding courseElement methods to the StubDB (Connel)
June 30th, 2016	2:00-4:00pm	Meeting to discuss Iteration 3 (Entire group)
	12-8 pm	GUI & Logic stuff(Deep).
July 1st, 2016	2:00-5:00pm	Modifying the script for the SQLDB to

		accommodate the new courseElements and started on the methods (Connel)
July 2nd	11:00-3:00pm	Continued adding courseElements to the SQLDB (Connel)
	10-6pm	GUI & Logic stuff(Deep).
July 3rd	3:00-6:00pm	Setup of Robotium for Application Testing. Did application testing for Add/Delete Term (Julius)
	7:00-8:00pm	Presentation planning (Ben)
	8:00-9:00pm	Setup of integration test structure, planning and skeleton code (Barry)
	10-4pm	GUI & Logic stuff(Deep).
July 4th, 2016	9:30-11:30am	Finishing up courseElements in the SQLDB (Connel)
	9:30-4:30pm	Integration tests (Barry)
	12:30-1:30pm	Integration testing of SQLDB (Barry,Connel)
	5:00-6:30pm	Fixing bugs with courseElements in the SQLDB (Connel)
	8:00-9:30pm	Troubleshooting bugs with courseLogic and Persistence (Deep, Connel)
	10:00-12:30am	Testing SQLDB and refactoring/cleaning up code (Connel)
	12-8pm	Bunch of GUI stuff(Deep)
July 5th	9:30-12:30	More integration testing and figuring out how to make them work nicely with the HSQLDB (Barry)
	10-5pm	GUI & Logic stuff(Deep)
	10:00-11:00	Code review (Ben)
	11:45-12:15pm	Added Application Testing through

		Robotium for Add/Delete Course (Julius)
	1:00-2:15	Separating Edit/Create header/button text from one another for Courses and Terms (Ben)
	1:00 - 3:00pm	Added GPA acceptance test, Add/Edit/Remove Course Element Acceptance test, Transcript Acceptance test (Julius)
	1:00-4:00pm	Fixed error in SQLDB to do with nextPersistentID (Connel)
	2:00-4:00	Unit tests for new features, finishing touches for integration tests on the HSQLDB (Barry)
	2:15-3:45pm	Added transcript big user story (Ben)
	4:45-5:30	Code review (Ben, Connel)

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
- CourseElementLogic.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
- CreateCourseElement.java
- CreateTerm.java
- DeleteCourses.java
- DeleteCourseElement.java
- DeleteTerm.java
- EditCoursePage.java
- EditTermPage.java
- HomePage.java
- SetGradeGoal.java
- ViewCourse.java
- ViewCourseElement.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
4. Integration Tests
    - BusinessPersistenceSeamTest.java
    - DataAccessIntegrationTest.java
5. Persistence
    -

AllTests.java
RunUnitTests.java
RunIntegrationTests.java

Acceptance/java includes the AcceptanceTests.java which is the java file with the robotium tests.

This iteration implemented a number of new big user stories. We implemented 4 new big user stories. They include "Select a Course to View or Edit", "Manipulate a Course", "Create a Course Element.", "Manipulate a Course Element".These give the user the ability the select a course, view course information, edit the course and delete the course. User can also add, delete and maintain course elements. GPA and grades are now accessible in terms and courses.

Also a decent bit of low and medium user stories did not get implemented. We believe the scope was quite large for the amount of time we had and feel that we did enough in the time allotted. The user stories include "Create Due Dates", 'View Due Dates", "View timeline of your degree."

AcademicAssistant is the project folder.

Root cause analysis -

5 Whys : Adding Course elements gives errors.
1. Can't add course elements with any similar private fields, even the ones that shouldn't be compared for duplicates.
2. The database saw the course elements duplicates(when they were not) so did not add.
3. The new element was being set to the same persistent id as an element already in the database, therefore the duplicate.
4. On database load, unique persistent ids were not being incremented properly.
5. Database was done in iterations which led to different objects being handled in different layers. Interactions b/w the layers messed up the persistent ID.
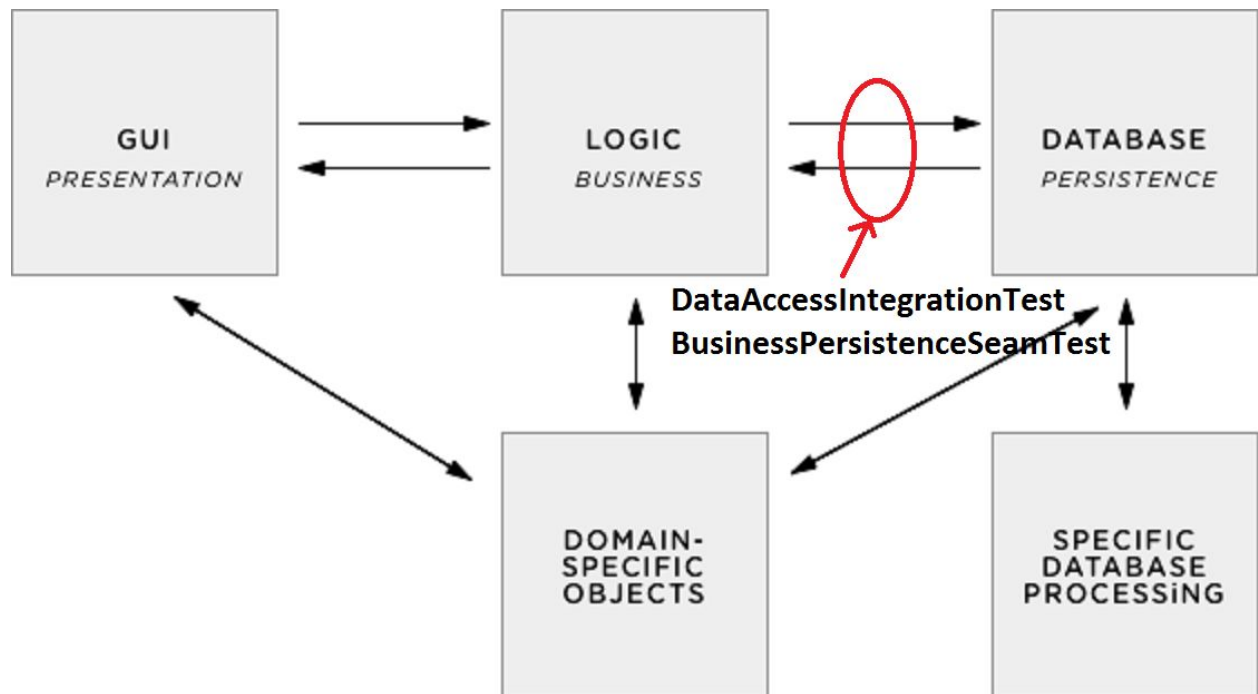
- Test structure
  - Suites for each package (Objects, Business, etc..)
  - Suites for each type of test (Unit, Integration…)
  - Acceptance tests
  - Updated AllTests to accommodate new tests

NOTE FOR README:

In order to successfully run the integration tests package, ensure the Project Files app>database>AADB.script file must match the content of the app>database>AADB.script.orig file. This fixes an error we had involving the locally stored HSQLDB to clear the database each time between tests. Otherwise running the integration tests twice causes them to fail, as the database already contains items and duplicates are found.

Revised Architectural Diagram

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, but it still exists from the last iteration. It's pretty infrequent though.

Our repository can be found at the following link:
svn+ssh://aviary.cs.umanitoba.ca/home/student/yakimchb/softeng_g10/repo