

Computer Science 2212b

Team Assignment 3

Team 4

Due Date: Mar, 15, 2014

Team Members:

Zaid Albirawi

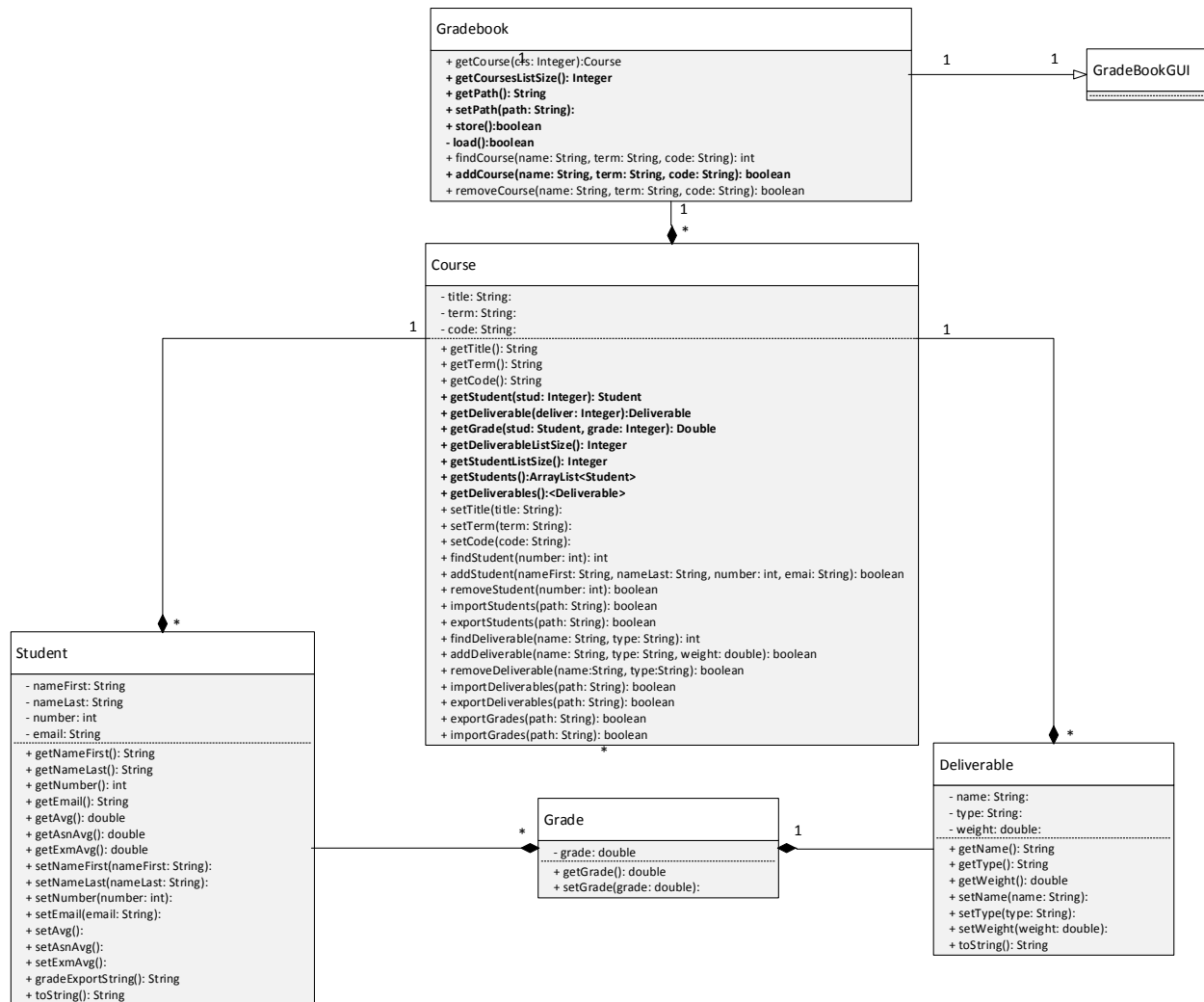
Justin Cheng

Steve Juarez

Jenna Le

Mikhail Lebedzeu

UML Diagram



User Stories

Link: <https://www.pivotaltracker.com/s/projects/1010498>

Webpage Screenshot

PivotalTracker Tracker has some new updates - [read more](#)

team4-gradebook velocity **13**

CURRENT BACKLOG ICEBOX EPICS MORE PROJECT STORIES

ADD STORY

DONE 23 Jan Pts: 0
6 Feb Pts: 16

- courses management User edits course attributes (ZA)
- courses management User adds a course (ZA)
- courses management User removes a course (ZA)
- gradebook management User changes active course (ZA)
- deliverables management User edits deliverable attributes (ZA)
- deliverables management User adds a deliverable (ZA)
- deliverables management User removes a deliverable (ZA)
- student management User edits student attributes (ZA)
- student management User adds a student (ZA)
- student management User removes a student (ZA)

20 Feb Pts: 22

- courses management User exports deliverables (ZA)
- courses management User exports students (ZA)
- courses management User imports grades (ZA)
- gradebook management User quits program (MI)
- courses management User edits student's grades (ZA)
- gradebook management, user interface A user can save their information (JL)
- gradebook management User views student grades in the course (ZA)
- courses management User imports students (ZA)
- courses management User imports deliverables (ZA)
- gradebook management User generates a student report PDF (ST)

CURRENT 6 Mar - Current Pts: 2 of 8

Hide accepted stories

- user interface User can switch between program tabs (Grades, Setup, Reporting) (JL)
- gradebook management User sends reports to students via email (JC) **Finish**
- courses management User exports grades (ZA) **Finish**
- User can create a new student directly from the table (JL) **Finish**

BACKLOG

ICEBOX Select All

EPICS show 2 done epics

- Course Management
- Gradebook management

Acceptance Testing

| | |
|------------------------|--|
| ID | 1 |
| Testing | User creating a new course; No title was given |
| References | 4.2.1 |
| Test Case | 1. When the program is running, click the “My Courses” drop down menu in the top right corner 2. Add information into course code (term optional) 3. Press ‘Add’ |
| Expected output | Error. Program behavior: “Add course” window will not close, small error message “Please enter a course name” will show |
| ID | 2 |
| Testing | User creating a new course; No course code was given |
| References | 4.2.1 |
| Test Case | 1. When the program is running, click the “My Courses” drop down menu in the top right corner 2. Add information into course name (term optional) 3. Press ‘Add’ |
| Expected output | Error. Program behavior: “Add course” window will not close, small error message “Please enter a course code” will show |

| | |
|------------------------|--|
| ID | 3 |
| Testing | User creating a new course; Presses cancel |
| References | 4.2.1 |
| Test Case | <ol style="list-style-type: none"> 1. When the program is running, click the “My Courses” drop down menu in the top right corner 2. (optional) change course code and/or course name and/or select a term 3. Press ‘Cancel’ |
| Expected output | Program behavior: “Add course” window should disappear. No course should have been added and the previously selected course should still be selected |

| | |
|------------------------|--|
| ID | 4 |
| Testing | User creating a new course; course code and course name given, term given |
| References | 4.2.1 |
| Test Case | <ol style="list-style-type: none"> 1. When the program is running, click the “My Courses” drop down menu in the top right corner 2. Add information into course name, course code, and select a term 3. Press ‘Add’ |
| Expected output | Program behavior: The specified course has been added, “Add course” window should disappear and this course is now selected in the program. Tab beside “My courses” should spell the name of the course, along with the course code with the term appended at the end of it. This course should now come up in the “My courses” drop down menu (with the term indication) |

| | |
|------------------------|---|
| ID | 5 |
| Testing | User creating a new course; course code and course name given, term NOT given |
| References | 4.2.1 |
| Test Case | <ol style="list-style-type: none"> 1. When the program is running, click the “My Courses” drop down menu in the top right corner 2. Add information into course name and course code, do not select a term 3. Press ‘Add’ |
| Expected output | <p>Program behavior: The specified course has been added, “Add course” window should disappear and this course is now selected in the program. Tab beside “My courses” should spell the name of the course, along with the course code (no term indication). This course should now come up in the “My courses” drop down menu (Again with no term indication)</p> |

| | |
|------------------------|---|
| ID | 6 |
| Testing | User editing an existing student; Editing first or last name, or email address, leaves field blank |
| References | 4.3.2 |
| Test Case | <ol style="list-style-type: none"> 1. When the program is running, select a class that contains students already or create a student for a class 2. Double click on an information component in the row of the student you want to edit (Either first name, last name, or email address) 3. Do not enter any information in to the box 4. Press enter |
| Expected output | <p>Error.</p> <p>Program behavior: The information box will revert back to its original state before an attempt to edit it was made. User will be notified with a message outlining their mistake.</p> |

| | |
|------------------------|--|
| ID | 7 |
| Testing | User editing an existing student; invalid student number |
| References | 4.3.2 |
| Test Case | <ol style="list-style-type: none"> 1. When the program is running, select a class that contains students already or create a student for a class 2. Double click on a student number information component in the row of the student you want to edit 3. Enter into the student number field either nothing or not a number, or a number that already exists in the table 4. Press enter |
| Expected output | <p>Error.</p> <p>Program behavior: The student number box will revert back to its original state. User will be notified with a message outlining their mistake.</p> |

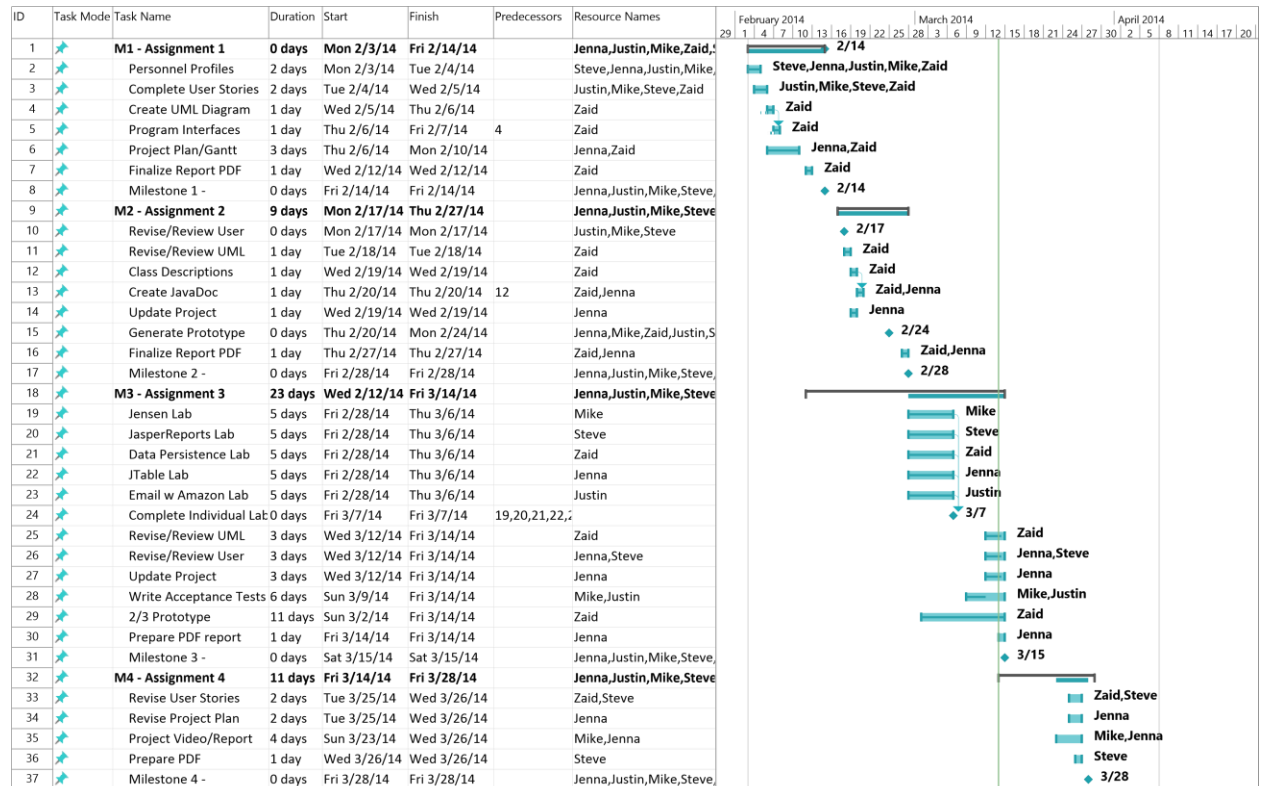
| | |
|------------------------|---|
| ID | 8 |
| Testing | User editing an existing student; valid information was input |
| References | 4.3.2 |
| Test Case | <ol style="list-style-type: none"> 1. When the program is running, select a class that contains students already or create a student for a class 2. Double click on an information component in the row of the student you want to edit (Either first name, last name, email address, or student number) 3. Enter at least one character for any field, and make certain it is an unused number for student number 4. Press enter |
| Expected output | <p>Program behavior: The information box will now contain the new information passed.</p> |

Jenkins CI

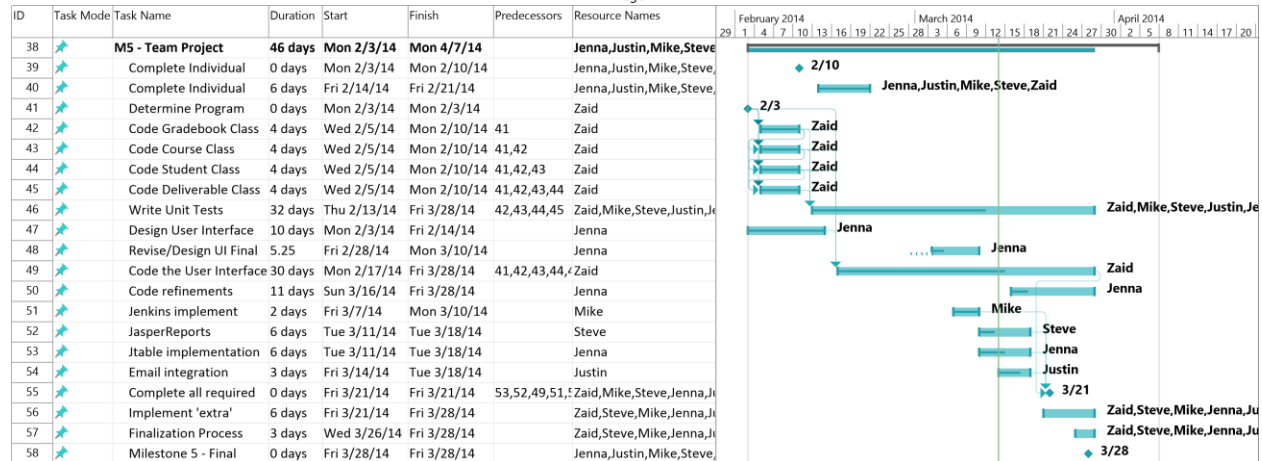
Username: ta

Password: JennaButler

Project Plan



Page 1



Page 2