TUFFY'S TRAPPER KEEPER 2017

12/13/2017

Group Code Squad

Team Members:

Vanchhit Khare, Juan Sanchez, Samarth Shah, Jose Ureno, and Jose Urrutia

Course:

CPSC 362 Foundations of Software Engineering

Professor:

Sara Ghadami

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Jose Urrutia	9/30/2017	Created cover page, table of contents	0.1
Juan Sanchez	10/02/2017	Added project plan	0.2
Vanchiit Khare	10/05/2017	Added user stories (functional and non)	0.3
Jose Ureno	10/06/2017	Added Use-cases	0.4
Jose Ureno	10/10/2017	Added Use-case diagram	0.5
Samarth Shah	10/10/2017	Added Trello screenshots	0.6
Jose Urrutia	10/10/2017	Added planning, staging, grooming	0.7
SS and VK	10/10/2017	Added development process	0.8
JS and JU	10/11/2017	Added user manual	1.0
Jose Ureno	10/15/2017	Update pre-game planning	1.1
Vanchiit Khare	10/16/2017	Add more use-cases	1.3
Samarth Shah	10/25/2017	Add more user stories	1.5
Jose Urrutia	10/26/2017	Add more trello screenshots	1.6
Juan Sanchez	10/27/2017	Added class diagram and CRC cards	1.7
Jose Ureno	10/28/2017	Update user manual	1.8
Jose Urrutia	11/07/2017	Update revision history and table of contents	2.0
Jose Ureno	11/30/2017	Update use-cases/diagram, add State	2.2
Samarth Shah	12/05/2017	Update user stories	2.4
Juan Sanchez	12/06/2017	Update class diagram, crc, add State	2.6
Vanchiit Khare	12/06/2017	Update user manual	2.8
Jose Urrutia	12/11/2017	Finalize document	3.0

1. Business Requirements

Tuffy's Trapper Keeper 2017 aims to become the foremost organizational tool for students at California State University, Fullerton, and for students across the CSU system. Many students find difficulty in keeping track of their overall class grades and assignments. This is due to a lack of organizational skills, or perhaps a lack of professors not updating the online grading tracking system within Titanium. This stand-alone student planner will transform the way students keep track of their school work while allowing them to see their current progress within a semester. Tuffy's Trapper Keeper 2017 is a simple registry and planner that will allow students to keep track of their currently enrolled classes, assignments, tests, and grades.

1.1 Background

Although schools have online portals for students, where grades and assignments can be posted, they are not always utilized by professors. This results in students not always knowing of upcoming deadlines or current grades. There is also the digital world that we are currently experiencing, where students can be hard pressed to carry bulky planners.

1.2 Business Opportunity

The market is ripe with students who would benefit immensely from using Tuffy's Trapper Keeper 2017. This product will be designed to require no input from the school or teachers, allowing the students to self-manage their agendas and grades. This will allow a student to keep track of all upcoming events and grades throughout their classes, and even between multiple schools. Since all the program will require is a simple set up, and to be updated occasionally, students will be able to more effectively manage their class schedules and assignments. Currently, there is a lack of student ability to remain organized and updated, and having these tools in a convenient digital package. There is a market for growth within California State University, Fullerton.

1.3 Business Objectives and Success Criteria

The objective in providing this product to California State University, Fullerton students is to discover the marketability of this software and whether or not it would be beneficial to distribute the software to the entire CSU system. If implementation of the software is successful on campus, then there is high potential for the same amount of usability on other campuses. However, at this point a price would be implement for full-scale use. There is a potential revenue in the hundreds of thousands of dollars if only a fraction of the student population of each campus decides to use this software.

1.4 Customer or Market Needs

Currently, there is a need for students to remain on schedule, on task, and organized with their class requirements. The Fullerton campus provides Titanium, an online organizer for classes and their respective professors and student body. Unfortunately, Titanium requires student participation and for professors to regularly update students' grades. There is also the issue of requiring access to the World Wide Web. With Tuffy's Trapper Keeper 2017, students will need to…

1. Have access to a laptop of desktop PC
2. Remain active in maintaining information for each class
3. Remain active in updating assignment grades
4. Populate all necessary fields for each class and assignment in order for the software to be used to its full potential.

Of course, the use of Tuffy's Trapper Keeper 2017 requires that students remain updated with their classes. It will require students to enter their assignments and exams, with details of the due date and time, assignment points total and points received, and the type of grading scale used for each class.

1.5 Business Risks

One major business risk associated in developing this product is that there is marketplace competition. There are student planners in the market that provided further functionality and integration across all platforms. Our product will provide ease of use and functionality, with a simple design that does not over promise. There is also the possibility that students decide not to use our software, as well as no promote the software to their colleagues. It would wise to promote the product on campus with a small marketing campaign. Another way to speed up adoption of the software is by targeting students who already use planners and various applications to keep track of this information. By open sourcing the software we will allow the constant evolvement of the program. Eventually the program can evolve into a database of classes and assignments that can be shared among users. This will allow one student to create a class and allow multiple students to simply join the class. Hopefully this will increase the adoption rate of the program.

2. Vision of the Solution

The system built for business objectives is implemented on democratic principles. All group members should remain active in communication through multiple messaging services, as well as voice their ideas, opinions, or advice at bi-weekly group meetings. During group meetings, one member can act as the discussion leader and ensure that each member is allotted time to voice their concerns.

2.1 Vision Statement

Keep track of all your assignments and grades in one place! The purpose of Tuffy's Trapper Keeper 2017 is to serve as an organizational tool and planner to help students remain proactive and engaged in their classes. No longer will a student need to worry about different logins and nagging a professor to update their online profile. With this product, anyone can become and organized monolith of productivity.

2.2 Major Features

1) Keep track of each class during the semester
2) Set reminders for upcoming assignments, quizzes, and exams for each class
3) Input grades for accurate grade keeping
4) Create tasks that need completing
5) Include detailed information of each class including professor contact details and room number
6) Calculate GPA for the semester
7) Add detailed notes on a per class basis
8) View a calendar that displays your upcoming assignments and tasks

2.3 Assumptions and Dependencies

The biggest assumption that was made during the conception phase of this product was that users will have to be willing to remain proactive in keeping the planner updated. It is imperative that users are diligent in inputting received grades and inputting upcoming assignments and/or exams in order to receive an accurate grade calculation and necessary reminders.

3. Scope and Limitations

The major limitation with product is that all information will be stored locally on the user's laptop or desktop pc, and will not be available in a mobile application or online portal. The app is to function as a standalone desktop application, meaning keeping data mobile or sharing will be limited in that regard.

3.1 Scope of Initial Release

On the surface, this product serves as a very powerful student planner and organizational tool that can be used to remain proactive with scholarly requirements throughout a semester. The ability to track all grades from assignments, exams, quizzes, etc., and the ability to set reminders on due dates for important tasks, will provide the highest value for students.

3.2 Scope of Subsequent Releases

Later and updated releases of this product will include the ability to be used on mobile platforms, as well as creating a standalone database that is integrated into the program.

3.3 Limitations and Exclusions

Some limitations and exclusions that a stakeholder might anticipate is advertising including in the final product as a means of receiving profit from all downloads. This is a feature that will not be included in final release. A stakeholder might also anticipate distributing the product to all platforms, however this is not anticipated by developers.

4. Business Context

Major assumptions for project concept include having users who are interested and willing to use the product, and that schedules should not be an issue. As for management priorities, it is imperative that the schedule remain 100% strict as there are required time constraints for future sprits and final release.

4.1 Stakeholder Profiles

Stakeholder	***Major Value***	***Attitudes***	***Major Interests***	***Constraints***
Developers	*See potential for revenue*	*See product as avenue toward future revenue*	*Time to market and better features than competitor*	*Have a working product*
Students	*Better organization and grade tracking; improved productivity*	*Highly receptive and positive reaction toward use*	*Automatic schedule reminders and class grade calculations for all classes in one location*	*Must run on personal laptop or desktop PC*

4.2 Project Priorities

Dimension	***Driver (state objective)***	***Constraint (state limits)***	***Degree of Freedom (state allowable range)***
Schedule	*Release 1.0 to be available by 10/11, release 1.1 by 12/1*	*Limited by semester schedule ending 12/15*	*No freedom whatsoever*
Features	*Add class and assignments by 10/1, database by 10/1, basic GUI by 10/1*	*Limited by database design, GUI design*	*70% of high priority features must be included in release 1.0*
Quality	*Expected minimal and pleasant GUI design, no errors in calculations*	*Dependent on developer skill with GUI design programming*	*75% user acceptance for 1.0, 90% for 1.1, 95% for 1.2*
Staff	*Work harmoniously and on time*	*Maximum team size is 5 developers*	*Dependent on skill some freedom allowed, but expecting contributions*
Cost	*Cost should remain near $0*	*Developers do not have income, pro bono work*	*No budget acceptable*

4.3 Operating Environment

What access security controls and data protection requirements are needed?>
Our customers, the students, will be using this product in a school setting throughout their collegiate and academic career. At launch of the product, the users will be located locally around the California State University, Fullerton campus and, once distributed to other campuses, users will gravitate toward their respective campus. The users would need access to the system at time that are convenient for them, so it is imperative that this product work as a stand-alone product that does not rely on external storage necessities. This in turn requires that all inputted information should be stored on a local back-end database that users do not need to know about, so long as the product works as expected. The only combination of data that is required is the data that is stored locally, and there is not a necessity to access outside information. Of course, there will be service interruptions and downtime during product updates, but it is not imperative that users have access to their information at all times. Users should normally use the product only when necessary.

5. User Stories

The following are the user stories chosen to be completed for all sprints. Due to some limitations in database design and important functionality, some user stories had to be revised along the way of implementation.

5.1 User Story 1

As a user, I want to add a semester to the planner.

5.2 User Story 2

As a user, I want to be able to view each semester.

5.3 User Story 3

As a user, I want to add a class to a specified semester.

5.4 User Story 4

As a user, I want to be able to view a chosen class based on a semester.

5.5 User Story 5

As a user, I want to be able to view a class.

5.6 User Story 6

As a user, I want to be able to delete a class.

5.7 User Story 7

As a user, I want to update the details of a class.

5.8 User Story 8

As a user, I want to add an assignment for a chosen class.

5.9 User Story 9

As a user, I want to be able to view a specified assignment.

5.10 User Story 10

As a user, I want to be able to delete an assignment.

5.11 User Story 11

As a user, I want to be able to update the details of an assignment.

5.12 User Story 12

As a user, I want to be able to add personalized notes base on a particular class.

5.13 User Story 13

As a user, I want to be able to load older notes that belong to a particular class.

5.14 User Story 14

As a user, I want to be able to see my overall GPA.

5.15 User Story 15

As a user, I want to be able to see all of my upcoming tasks in a calendar view.

6. Non-functional User Stories

The following are the non-functional user stories chosen.

6.1 Non-functional User Story 1

As a student, I want your program to save the information I input.

6.2 Non-functional User Story 2

As a student, I want each assignments grade calculation to be correct.

6.3 Non-functional User Story 3

As someone who likes to stay organized, I want your software to help me be productive.

6.4 Non-functional User Story 4

As a user, I want the program to remain stable and not crash when entering information.

6.5 Non-functional User Story 5

As a student, I want to be able to run your product on my Apple laptop or Windows desktop.

7. Use-cases

The following are the use-cases and their description for each requirement.

7.1 View Class Schedule

ID: UC-001

Description: Give users the option to view their weekly course schedule to allow them to schedule and plan times for studying the required material and be better with their time management.

Primary Actor: User/Student

Pre-Conditions: User must have first added courses to the database using the "Add Class" option within the application.

Post Conditions:

Success end condition:

Student can view their entire class schedule along with any descriptions or links that have been saved to it.

Failure end condition:

In the case that the student has not added any courses, they will be shown an empty calendar with no reminders or changes made to any of the dates.

Main Success Scenario:

1. Student opens the application to gain access to all class details.
2. By clicking on view schedule, the user/student is presented a window with a description and meeting dates/times. Here the user can also make any changes to the course description or add any reminders/links that are relevant to the course.
3. Finally, the user has the option to save and update the course to reflect any possible changes made by the teacher.

7.2 Add Course

ID: UC-002

Description: Users are given the option to add courses for which they can have descriptions and keep important links saved.

Primary Actor: User/Student

Pre-Conditions: User must give each course a course ID (number), course name and meeting times for the week.

Post Conditions:

Success end condition:

Once the student has added a course, they have the option to make notes or save links in the description for them.

Failure end condition:

Without giving a course ID, course name and meeting time the user will be unable to save the course to the local application database.

Main Success Scenario:

1. Student opens the application to gain access to all class details.
2. Upon being added to the database, each class will then be displayed on the calendar to allow the user to easily see when and where the class meets.
3. Student can also remove courses from the database if they no longer need them.

7.3 Add Task/Deadline

ID: UC-003

Description: Adding a task or deadline for each course that has been added. This allows users to view assignments coming up in their calendar/schedule and have descriptions with requirements for those assignments.

Primary Actor: User/Student

Pre-Conditions: User will set a due date and add the assignment under the corresponding class.

Post Conditions:

Success end condition:

After setting a date for a task/assignment, the student will see it in the calendar to remind them of when it is due.

Failure end condition:

Student must add a due date before they can add a deadline otherwise it will not be saved in the database.

Main Success Scenario:

1. Student opens the application to view saved courses.
2. Clicking a course will then allow for the adding of a task/assignment.
3. Once they are under the correct course, the student can add an assignment deadline that will then show up in their calendar.

7.4 Set Course Grade

ID: UC-004

Description: Allows user to easily view their current grade in each class without needing to access their school's online portal.

Primary Actor: User/Student

Pre-Conditions: Student must have added the class to the database in order to set a grade for it.

Post Conditions:

<u>Success end condition</u>:

Once the grade is set, the user can view it easily along with the class description.

<u>Failure end condition</u>:

If the course has not been added, the student cannot set a grade for it.

Main Success Scenario:

1. Student views class schedule to get access to their descriptions.
2. Once the course window pops up, user can easily set their current course grade on the page.
3. Editing the grade will also be easy so that the user can make any updates to the current grade.

7.5 Edit Class Details

ID: UC-005

Description: Users can also change descriptions to match any changes that may have been made to their course syllabus or requirements based on a professor's changes.

Primary Actor: User/Student

Pre-Conditions: User must have added a course and a description for it before they can make any changes.

Post Conditions:

<u>Success end condition</u>:

If needed, the student can change the description along with any meeting times or grades to reflect their current status.

<u>Failure end condition</u>:

If there is no description, the student can first add one and make any required changes afterwards.

Main Success Scenario:

1. Student opens the application to view their schedule.
2. Clicking on a course will show the window with existing description and details.
3. Editing will allow the user to make any necessary updates to the details for the class.
4. Once the changes are made, they can be saved and it will update in the database.

7.6 View Classes by Semester

ID: UC-006

Description: Give users the option to view a list of their classes within a specific semester.

Primary Actor: User/Student

Pre-Conditions: User must have first added a course under the selected semester.

Post Conditions:

Success end condition:

Student can view a semester with a list of the classes categorized under it.

Failure end condition:

In the case that the student has not added any courses, the semester class list will be empty and only have the name.

Main Success Scenario:

1. Student opens the application to gain access to all class details.
2. User will then click on the semesters tab and be presented with a view of all semesters in which they have taken classes.
3. Finally, clicking on a particular semester will show all classes taken during that time.

7.7 Submit Assignment

ID: UC-007

Description: User can submit an assignment/task for a class and enter the grade received out of total points possible.

Primary Actor: User/Student

Pre-Conditions: User must have a class under which they can add an assignment.

Post Conditions:

<u>Success end condition</u>:

Program will calculate a grade based on the data submitted by the user. It will then be added to the total course grade.

<u>Failure end condition</u>:

In the case that the student does not enter any data, the assignment will not be submitted.

Main Success Scenario:

1. Student opens the application to gain access to all class details.
2. User can then click on a specific course to see the list of submitted assignments.
3. Clicking on an assignment will then show the calculated grade.

7.8 Add Semester

ID: UC-008

Description: Give users the option to add a semester (Ex: Spring 2017, Fall 2018, etc). under which they can add classes taken.

Primary Actor: User/Student

Pre-Conditions: User must first have classes that they can add to a specific semester.

Post Conditions:

Success end condition:

Student can add semesters under which they can add courses that were taken during that time period.

Failure end condition:

In the case that the student does not add any data for a semester, the option will not be added to add courses under it.

Main Success Scenario:

1. Student opens the application to gain access to all class details.
2. Student then uses the “add semester” option to enable the adding of courses.
3. Finally, the user will add courses that were taken during this time period to make it easier to organize classes.

7.9 Delete Semester

ID: UC-009

Description: Give users the option to remove semesters in the case that they no longer need to view the data for those courses.

Primary Actor: User/Student

Pre-Conditions: User must have first added a semester before they can remove it.

Post Conditions:

Success end condition:

Student can remove any classes that were taken during a semester to eliminate any clutter on their home page.

Failure end condition:

In the case that the student has not added a semester, they will first have to add it before they are allowed to delete it.

Main Success Scenario:

1. Student opens the application to gain access to all class details.
2. Student clicks on a specific semester that they no longer need/want access to.
3. Using the “delete semester” option, the user will remove any associated classes from the database.

7.10 Add Personalized Notes

ID: UC-010

Description: Give users the option to add notes to an assignment or class to enhance description.

Primary Actor: User/Student

Pre-Conditions: User must have first added an assignment or course for which they would like to add notes.

Post Conditions:

<u>Success end condition</u>:

Student can add a description or notes to an assignment to remind or help them with completion.

<u>Failure end condition</u>:

If a class or course has not been added, the user will not be able to add notes.

Main Success Scenario:

1. Student opens the application to gain access to all class details.
2. Student can then click on an assignment or class for which they need to add notes.
3. Once the class or assignment has been selected, user can add notes in the description.

7.11 View Calendar

ID: UC-011

Description: View a calendar with highlighted dates when assignments are due

Primary Actor: User/Student

Pre-Conditions: Current calendar month must have existing assignments in order to highlight a date.

Post Conditions:

<u>Success end condition</u>:

Student can see a window with a calendar for the current month to remind them when assignments are to be completed.

<u>Failure end condition</u>:

If an assignment has not been added for the current month, the windows will be blank.

Main Success Scenario:

1. Student opens the application to track assignments.
2. Student can then click on “Calendar” button in sidebar.
3. Student will then see highlighted dates on which they have an assignment due.

7.12 Add Assignment Score

ID: UC-012

Description: User can submit the score they received on a particular exam or assignment.

Primary Actor: User/Student

Pre-Conditions: User must first submit assignment before having the option to give a score.

Post Conditions:

<u>Success end condition</u>:

Once a score has been submitted along with the total points possible, a percentage will appear in the window.

<u>Failure end condition</u>:

If either the score or total possible is left blank, the score will not be updated.

Main Success Scenario:

1. Student opens the application to gain access to a list of assignments.
2. Student can edit the assignment to add their score.
3. Once the score has been submitted, the percentage will be calculated.

7.13 View Course Grade

ID: UC-013

Description: User can view calculated grade for a particular course

Primary Actor: User/Student

Pre-Conditions: User must submit assignments along with their scores to see a calculated grade.

Post Conditions:

<u>Success end condition</u>:

After graded assignments have been submitted, program will calculate a grade for the course.

<u>Failure end condition</u>:

If scores have not been submitted for any assignments in the course, a grade will not be calculated.

Main Success Scenario:

1. Student opens program to view course progress.
2. After submitting an assignment, student can add score to assignments to upgrade course progress.
3. All assignments will be used to calculate a final grade.

7.14 View GPA

ID: UC-014

Description: Any classes with submitted assignments and calculated grades will be used to calculate GPA for a given semester.

Primary Actor: User/Student

Pre-Conditions: User must submit assignments along with their scores to see a calculated grade.

Post Conditions: If scores are calculated for every class, the program will update semester GPA.

<u>Success end condition</u>:

Once a grade has been calculated for every class in a semester, it will be used to calculate the semester GPA.

<u>Failure end condition</u>:

In the case that a course does not have a grade or that no courses have a grade, the program will not use those courses to calculate GPA.

Main Success Scenario:

1. Student opens program to view course list.
2. After grades have been submitted for all new assignments, grades will update for each class.
3. Overall semester GPA will be calculated.

7.15 Load Older Notes

ID: UC-015

Description: User can view past notes saved to a class.

Primary Actor: User/Student

Pre-Conditions: Notes for a specific class must first be saved before they are available to load.

Post Conditions:

<u>Success end condition</u>:

After completing a note for a class, the user can save it to allow them to view it at a later date.

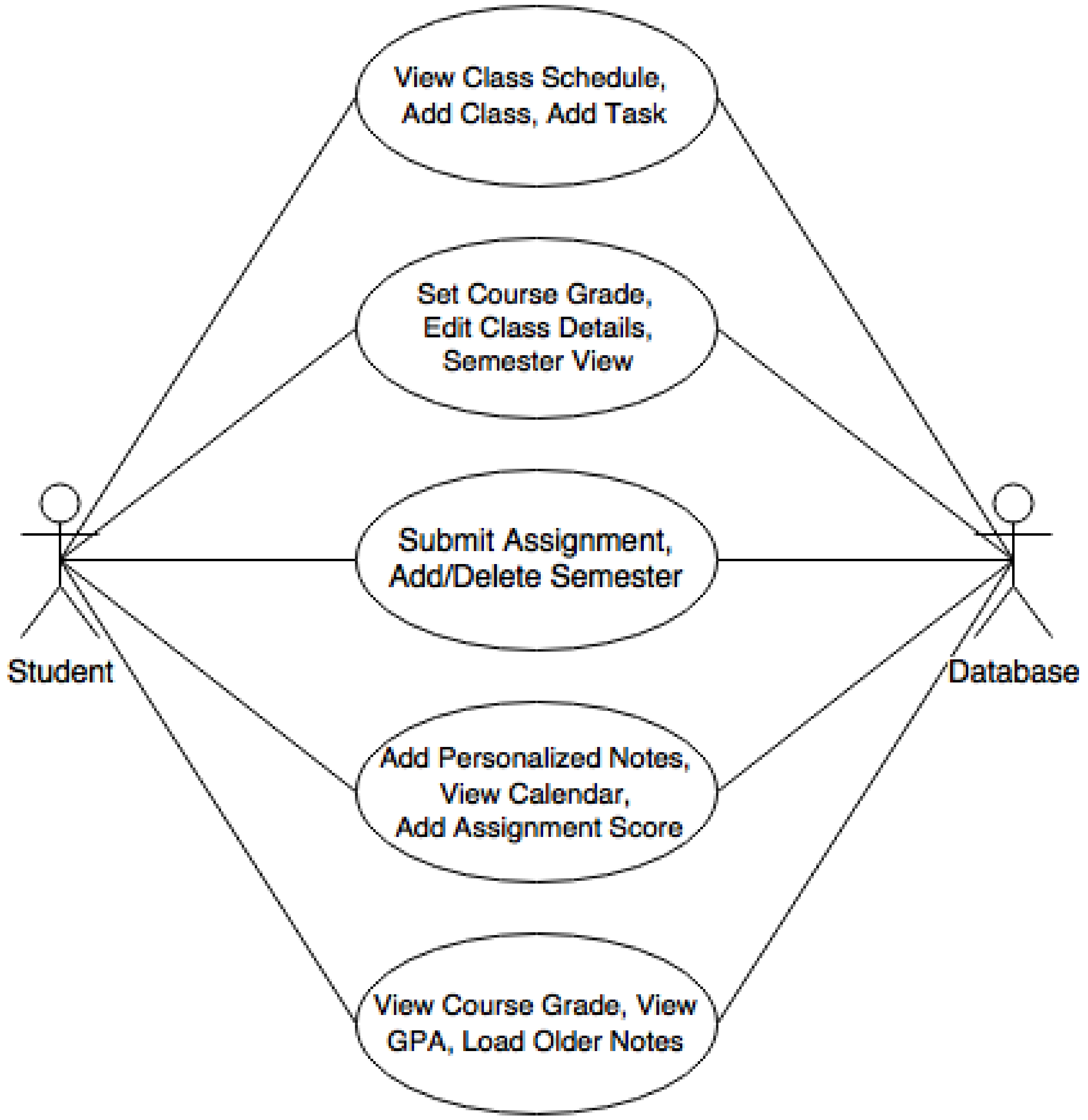
<u>Failure end condition</u>:

If a note is started but not saved to a course, it will not load when viewing a list of past notes.

Main Success Scenario:

1. Student opens program to view course list.
2. User clicks on list of notes saved to courses.
3. Older notes are then loaded for viewing by the user.

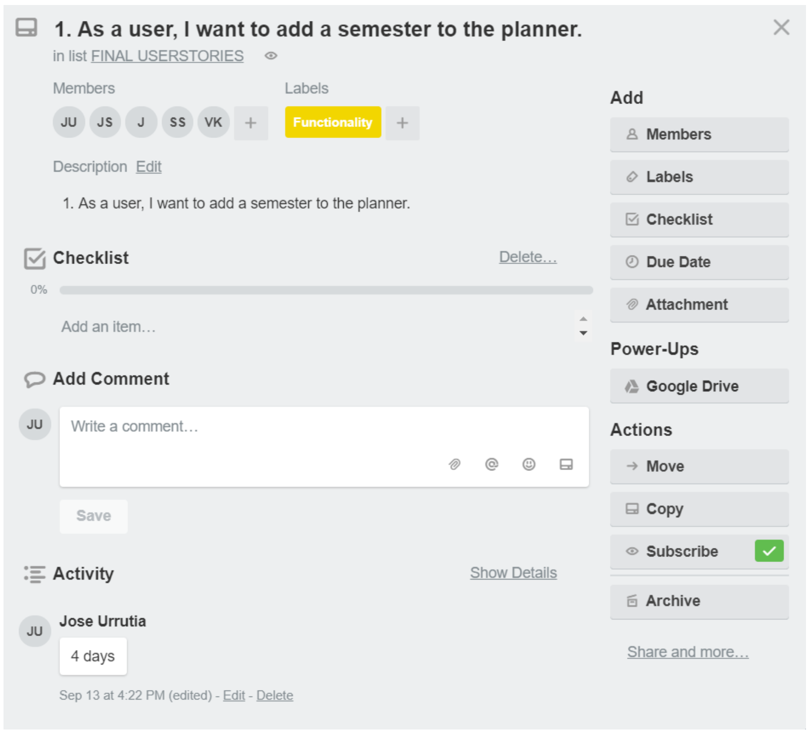
8. Use-case Diagram


View Class Schedule, Add Class, Add Task
Set Course Grade, Edit Class Details, Semester View
Submit Assignment, Add/Delete Semester
Add Personalized Notes, View Calendar, Add Assignment Score
View Course Grade, View GPA, Load Older Notes
Student
Database

9. Sprint Backlog Screen Shots

The following are the screenshots of our Trello Kanban-Board.

9.1 Screenshot 1



1. As a user, I want to add a semester to the planner.

in list FINAL USERSTORIES

Members: JU JS J SS VK

Labels: Functionality

Description Edit

1. As a user, I want to add a semester to the planner.

Checklist Delete…

0%

Add an item…

Add Comment

Write a comment…

Save

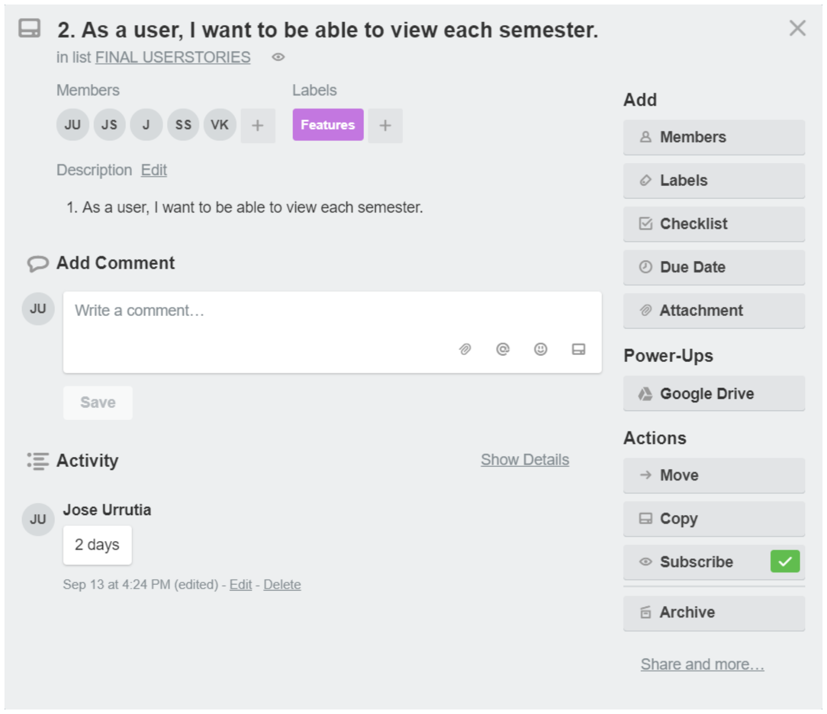
Activity Show Details

Jose Urrutia

4 days

Sep 13 at 4:22 PM (edited) - Edit - Delete

9.2 Screenshot 2



2. As a user, I want to be able to view each semester.

in list FINAL USERSTORIES

Members: JU JS J SS VK

Labels: Features

Description Edit

1. As a user, I want to be able to view each semester.

Add Comment

Write a comment…

Save

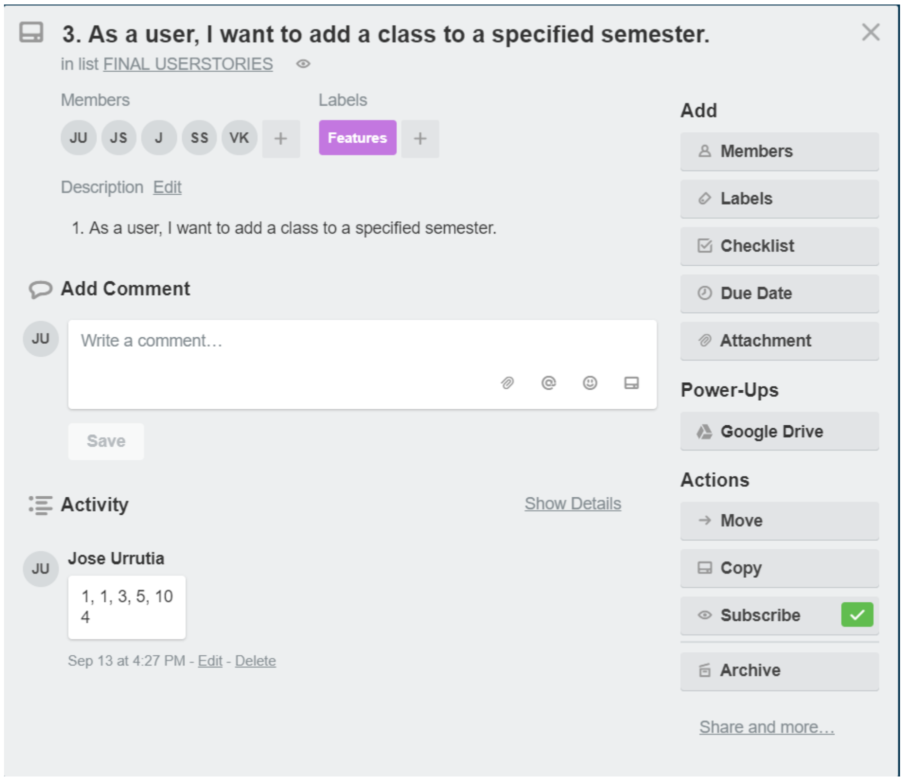
Activity Show Details

Jose Urrutia

2 days

Sep 13 at 4:24 PM (edited) - Edit - Delete

9.3 Screenshot 3



3. As a user, I want to add a class to a specified semester.

in list FINAL USERSTORIES

Members: JU JS J SS VK

Labels: Features

Description Edit

1. As a user, I want to add a class to a specified semester.

Add Comment

Write a comment…

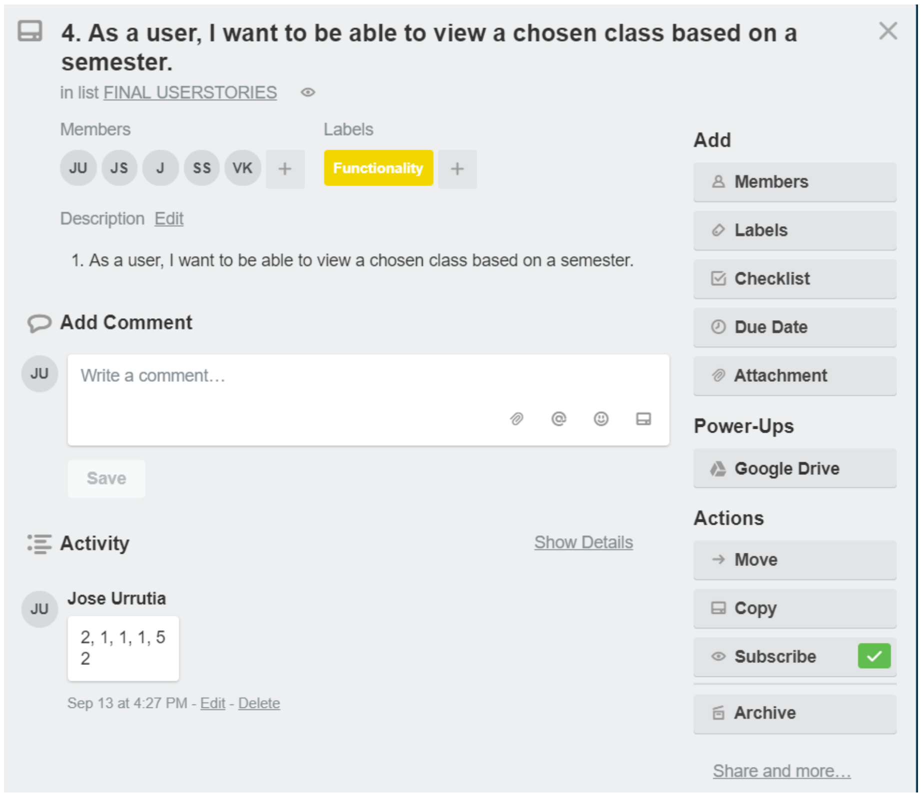
Activity — Show Details

Jose Urrutia

1, 1, 3, 5, 10
4

Sep 13 at 4:27 PM - Edit - Delete

9.4 Screenshot 4



4. As a user, I want to be able to view a chosen class based on a semester.

in list FINAL USERSTORIES

Members: JU JS J SS VK

Labels: Functionality

Description Edit

1. As a user, I want to be able to view a chosen class based on a semester.

Add Comment

Write a comment…

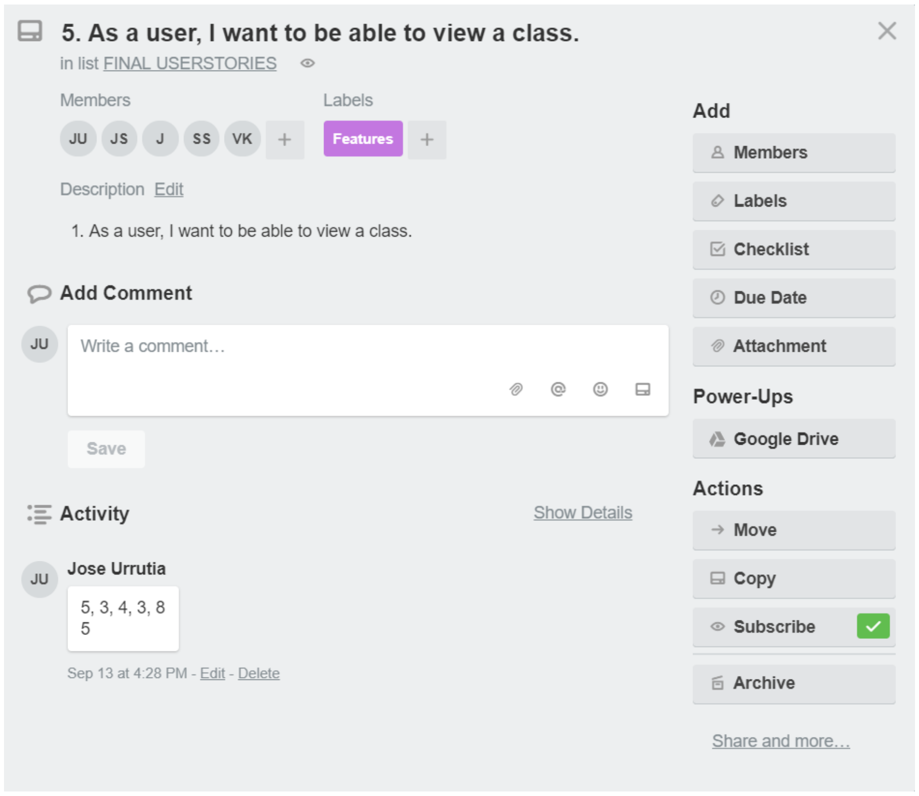
Activity — Show Details

Jose Urrutia

2, 1, 1, 1, 5
2

Sep 13 at 4:27 PM - Edit - Delete

9.5 Screenshot 5


5. As a user, I want to be able to view a class.

in list FINAL USERSTORIES

Members: JU JS J SS VK

Labels: Features

Description Edit

1. As a user, I want to be able to view a class.

Add Comment

Write a comment…

Save

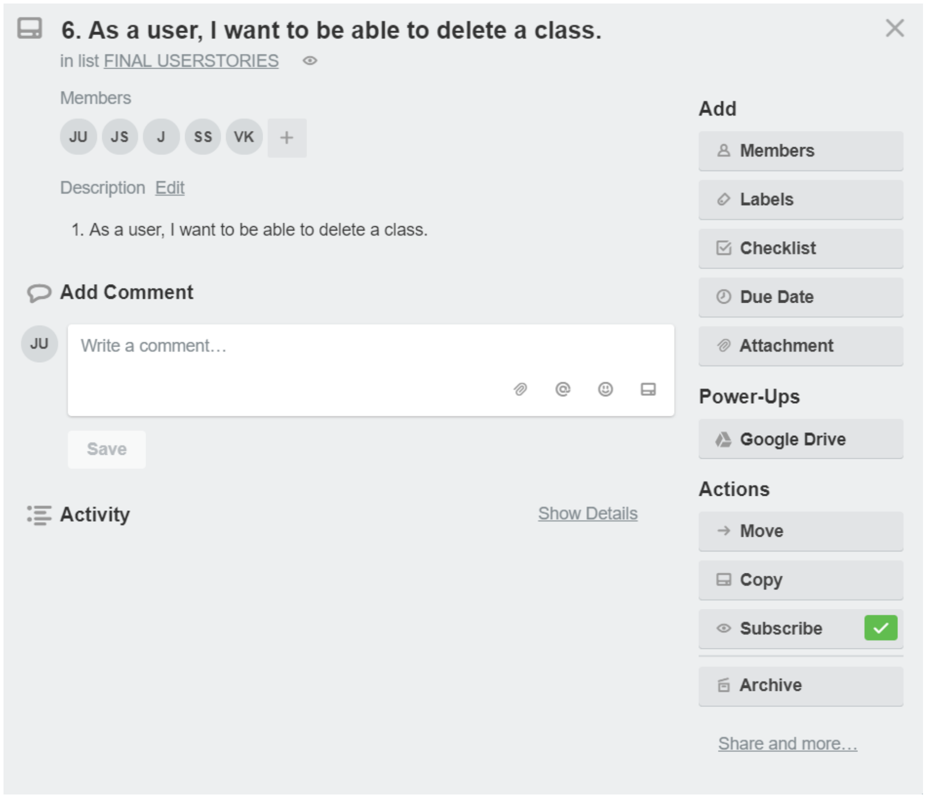
Activity Show Details

Jose Urrutia

5, 3, 4, 3, 8
5

Sep 13 at 4:28 PM - Edit - Delete

9.6 Screenshot 6


6. As a user, I want to be able to delete a class.

in list FINAL USERSTORIES

Members: JU JS J SS VK

Description Edit

1. As a user, I want to be able to delete a class.

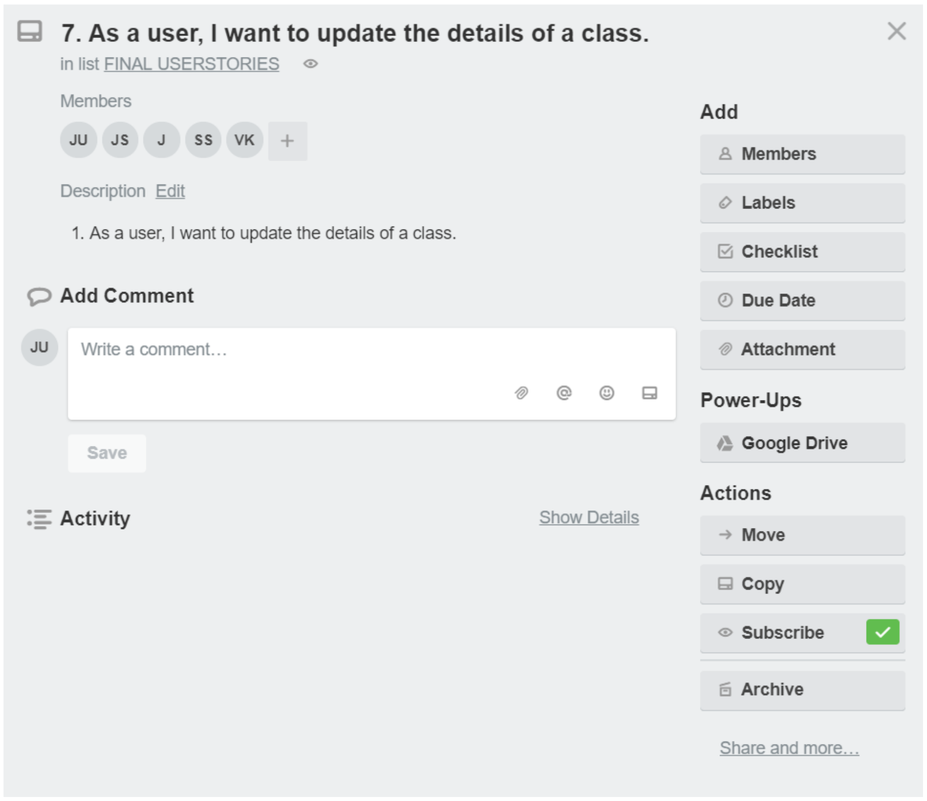
Add Comment

Write a comment…

Save

Activity Show Details

9.7 Screenshot 7


7. As a user, I want to update the details of a class.

in list FINAL USERSTORIES

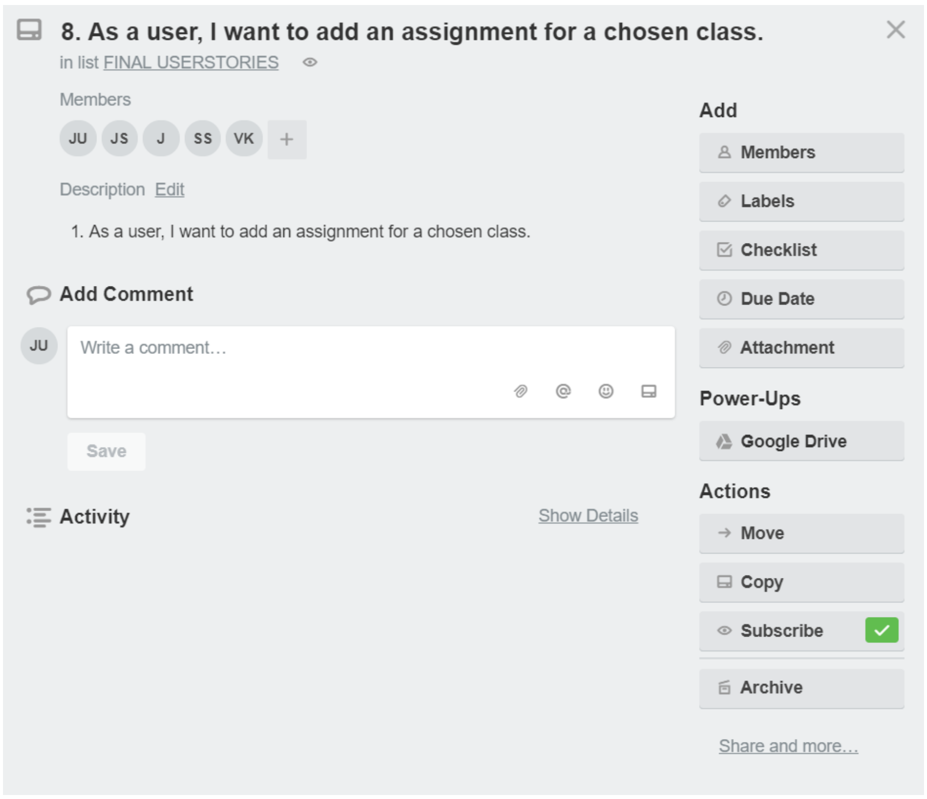
Members

JU JS J SS VK

Description Edit

1. As a user, I want to update the details of a class.

9.8 Screenshot 8


8. As a user, I want to add an assignment for a chosen class.

in list FINAL USERSTORIES

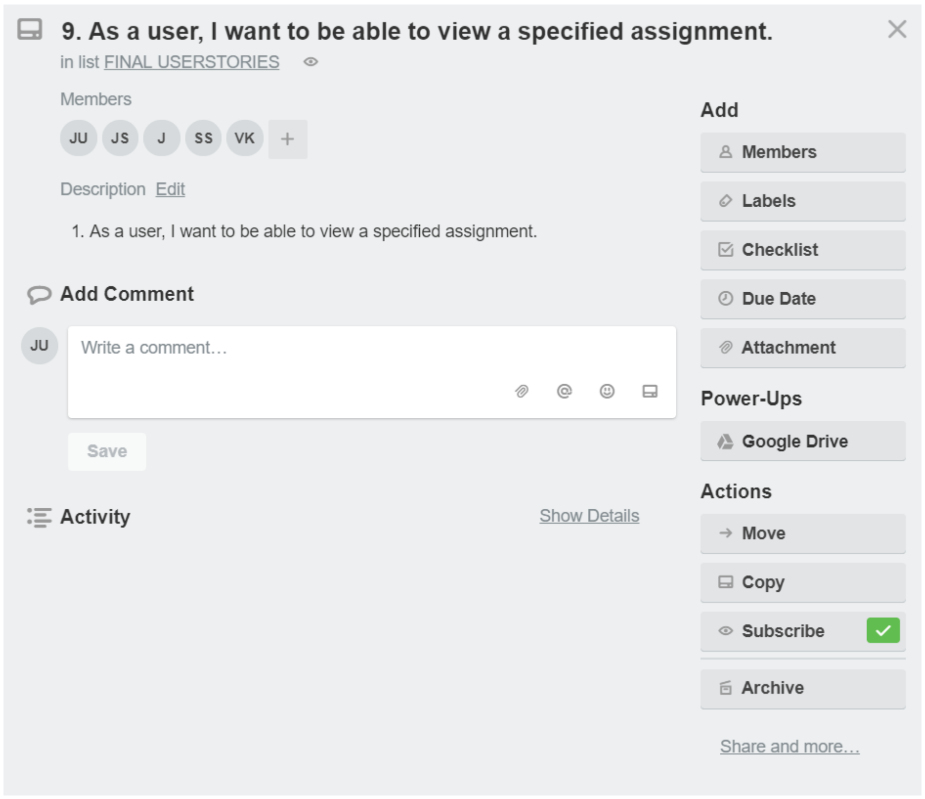
Members

JU JS J SS VK

Description Edit

1. As a user, I want to add an assignment for a chosen class.

9.9 Screenshot 9



9. As a user, I want to be able to view a specified assignment.

in list FINAL USERSTORIES

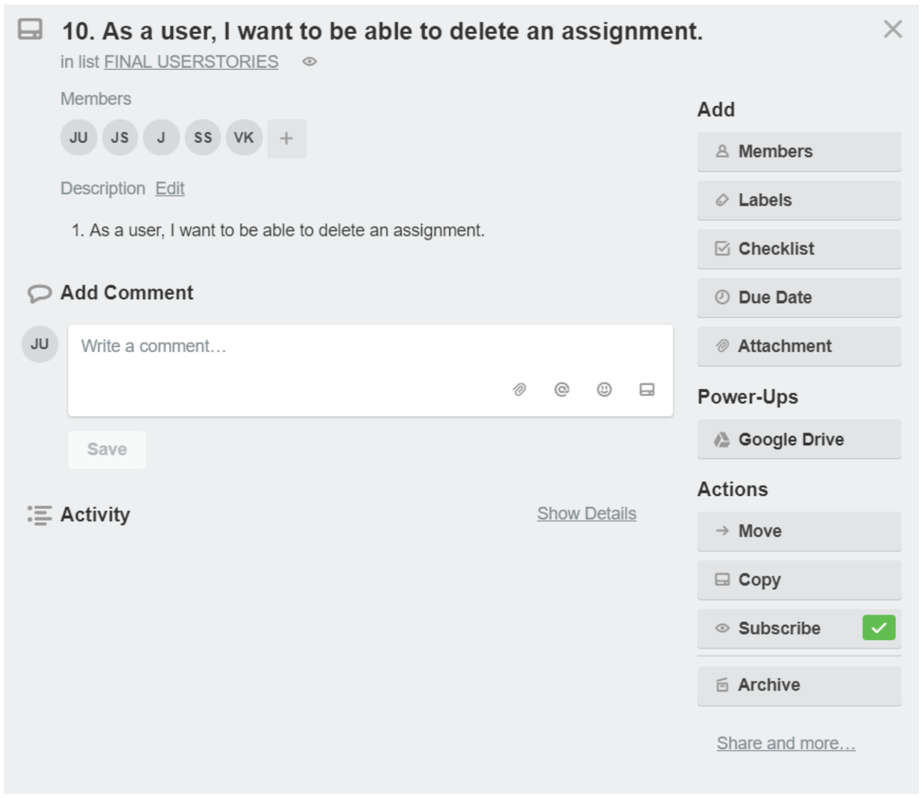
Members

JU JS J SS VK

Description Edit

1. As a user, I want to be able to view a specified assignment.

9.10 Screenshot 10



10. As a user, I want to be able to delete an assignment.

in list FINAL USERSTORIES

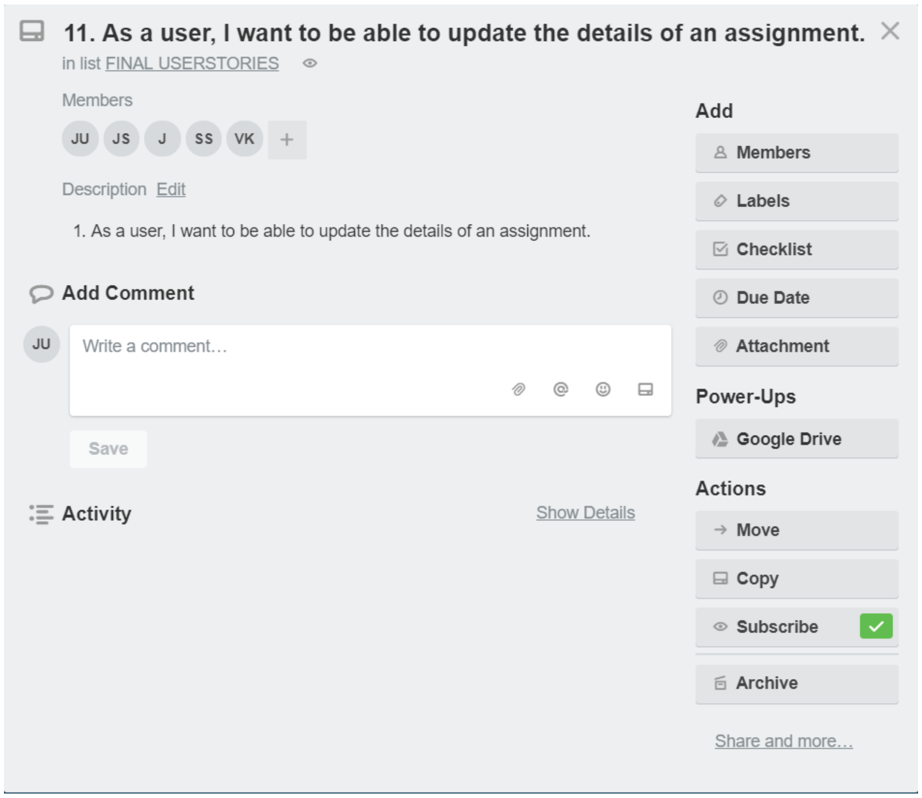
Members

JU JS J SS VK

Description Edit

1. As a user, I want to be able to delete an assignment.

9.11 Screenshot 11



11. As a user, I want to be able to update the details of an assignment.

in list FINAL USERSTORIES

Members

JU JS J SS VK

Description Edit

1. As a user, I want to be able to update the details of an assignment.

Add Comment

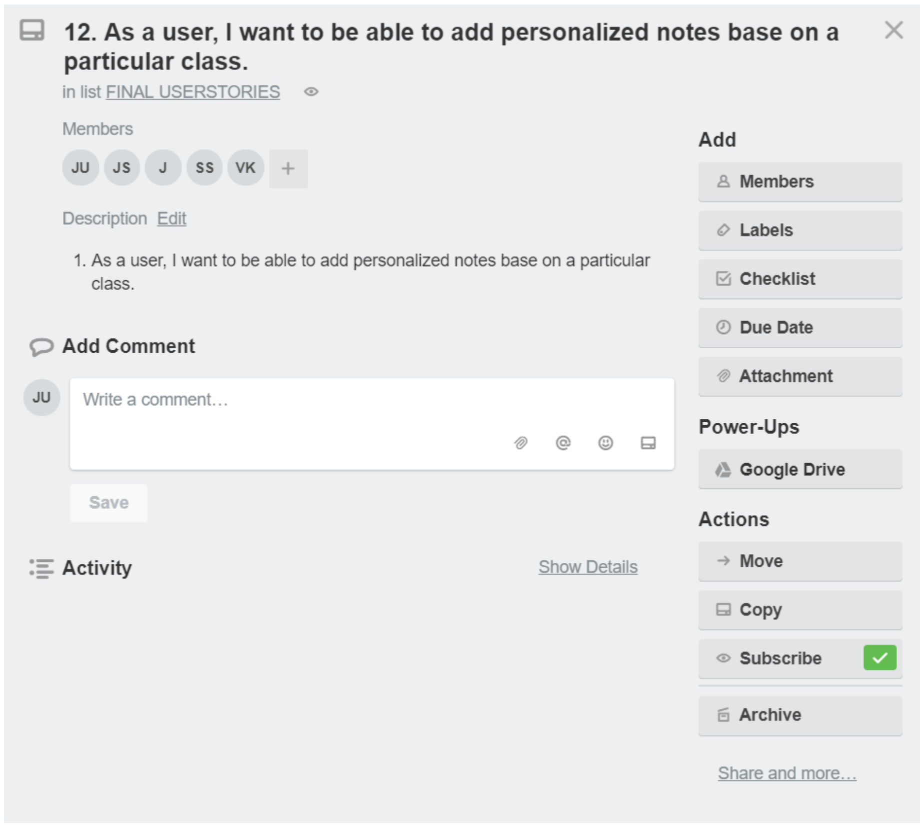
Write a comment…

Save

Activity

Show Details

9.12 Screenshot 12



12. As a user, I want to be able to add personalized notes base on a particular class.

in list FINAL USERSTORIES

Members

JU JS J SS VK

Description Edit

1. As a user, I want to be able to add personalized notes base on a particular class.

Add Comment

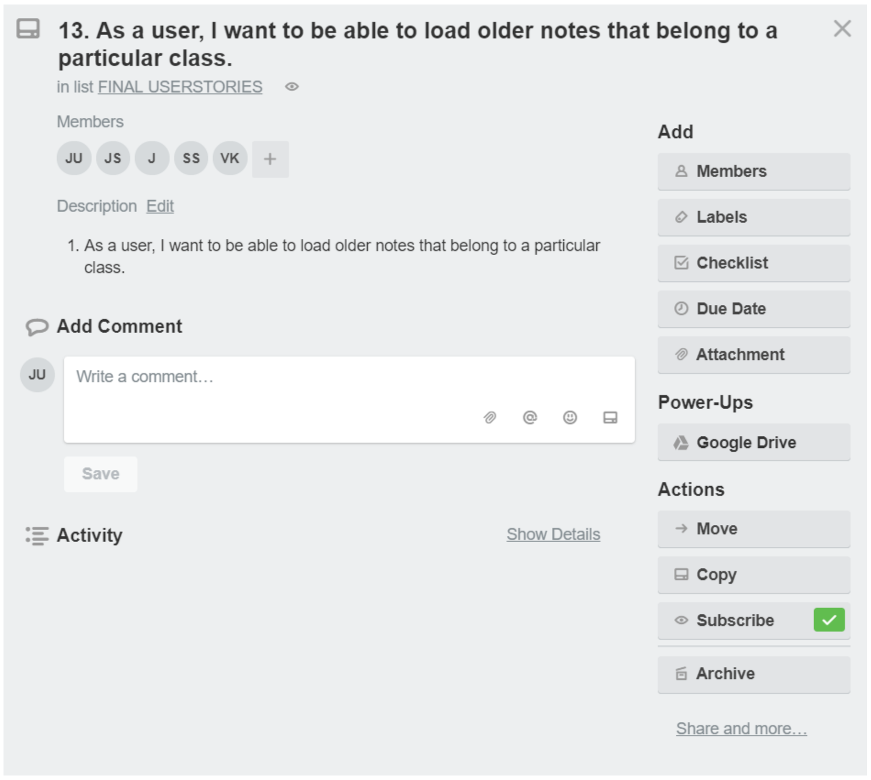
Write a comment…

Save

Activity

Show Details

9.13 Screenshot 13


13. As a user, I want to be able to load older notes that belong to a particular class.

in list FINAL USERSTORIES

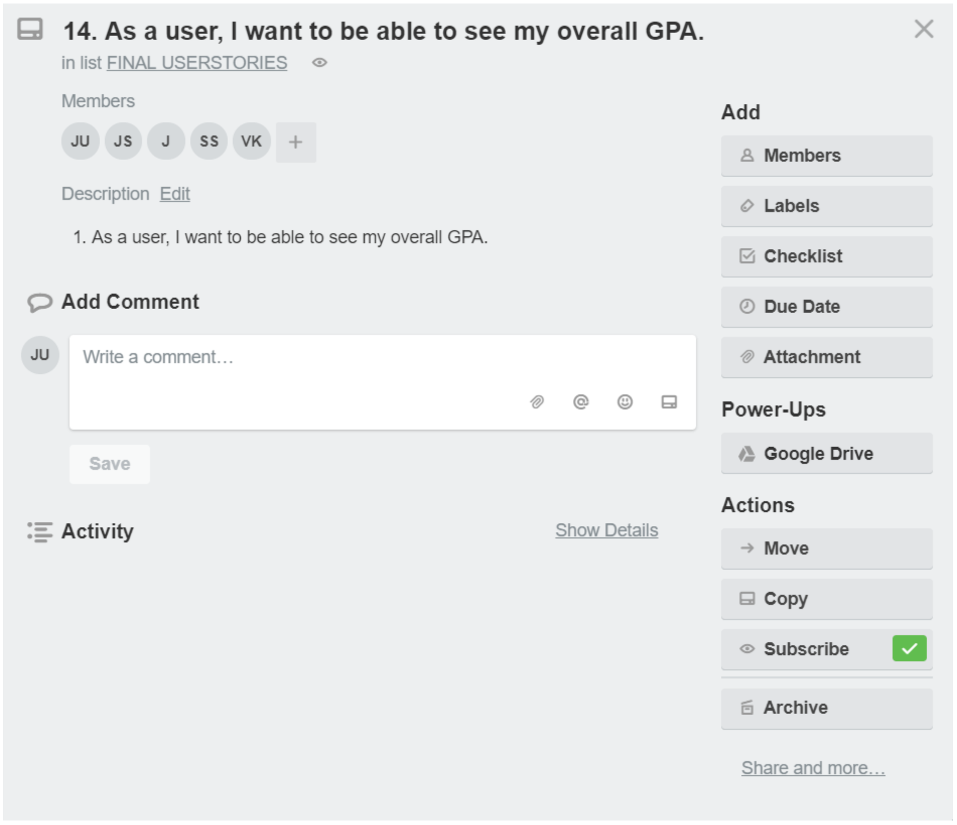
Members

JU JS J SS VK

Description Edit

1. As a user, I want to be able to load older notes that belong to a particular class.

9.14 Screenshot 14


14. As a user, I want to be able to see my overall GPA.

in list FINAL USERSTORIES

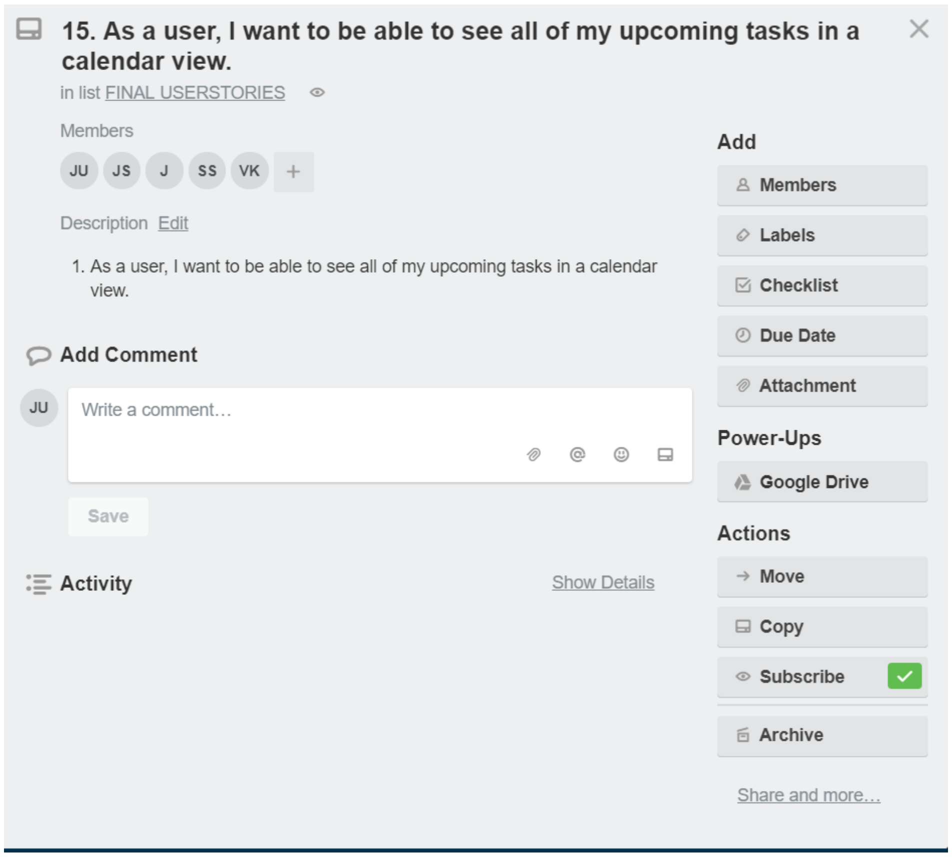
Members

JU JS J SS VK

Description Edit

1. As a user, I want to be able to see my overall GPA.

9.15 Screenshot 15



15. As a user, I want to be able to see all of my upcoming tasks in a calendar view.

in list FINAL USERSTORIES

Members

JU JS J SS VK

Description Edit

1. As a user, I want to be able to see all of my upcoming tasks in a calendar view.

Add Comment

JU

Write a comment…

10. Pre-game Planning

During our pre-game planning phase, our group read each proposed story and each member made a comment about that particular story. We wanted to discuss how important each story was to the overall product, as well as which stories were most important to implement by the first iteration due date, and why they were important. Eventually, we decided on fifteen stories that were important due to their potential for setting the foundation for this product. However, we were aware of the potential for some of the user stories requiring change as the programming process for some stories was very ambitious.

After finalizing the five user stories for the first iteration, we continued onto the poker game. Some members have more experience with our preferred programming language (Java), while other members are beginning their familiarization. Overall, the poker game allowed us to come to a consensus on how long we expected each user story to be completed. What was not anticipated or discussed was the schedule of other classes affecting our production development. In later sessions, we were able to better understand how our schedules synced and decide on future meetings that are mandatory.

During our second and third sprints, we maintained the same process of team planning and decisions in order to finalize the second group of user stories and the third group of user stories. We made sure that they were obtainable goals and something that could be implemented within our code. Of course, these user stories needed to contribute to the experience of the software, but it was imperative that these functionalities remained consistent throughout the coding process.

11. Staging or Grooming

As a group, we immediately identified that the backbone for this project was going to be the design of the database structure. The database structure is essential for this program to run because it allows for the storage of user input, as well as the ability to recall information that is needed to perform a grade calculation or display information to the user. Next, we discussed the exact design that we wanted to implement. There was a vast difference of opinion on how the product should look or function. However, we had a mutual agreement on a basic design that will be allowed further redesigns and interpretations.

The next step in staging was ensuring that all developers familiarized themselves with the Java development platform and how it works. This was crucial to the success of the product because the development team could not rely solely on the expertise of the lead programmer. Once these plans were in place, we proceeded with the first stages of development.

The second stage of development brought about an overhaul of the user interface experience. Our idea was to have a design that was simplistic and easy to use, and pleasing to the eye. We could accomplish that goal during the second sprint. However, it was difficult to make sure that all windows remained functional. The database was redesigned somewhat, but nothing major was changed.

The third stage of development was used to add our remaining user stories, but development was rather simple during this phase because the majority of the coding was completed and we had a formula set to implement designs.

12. Development Process and Documentation

The development process describes the system set in place by our team to meet deadlines and work together. Documentation was important as well, as this allows the entire team to evaluate progress and changes.

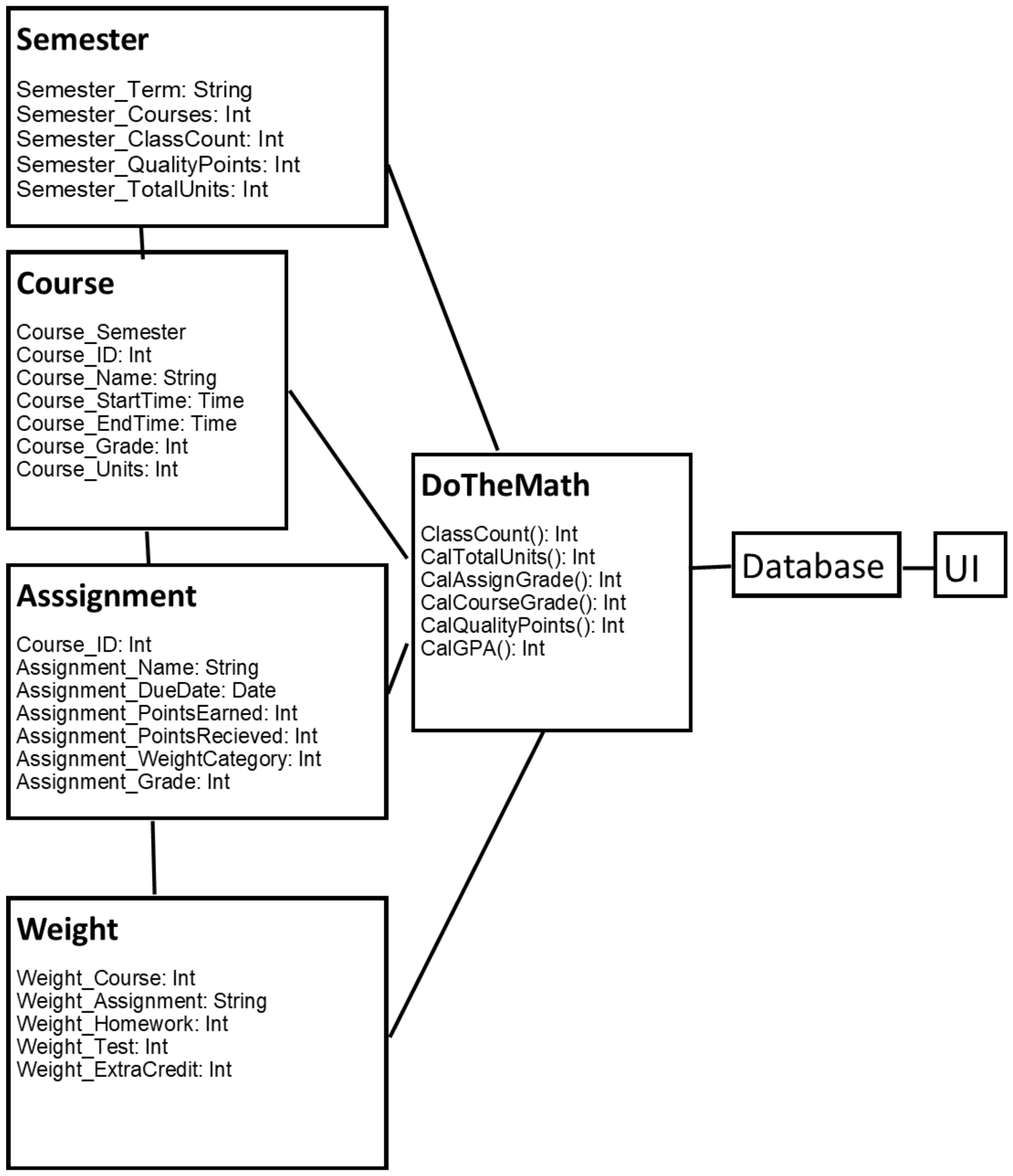
12.1 Development Process

The first important step in developing the product was designing the database. Those team members who were familiar with database designs and structures came up with a rough design that would allow all data to be connected and work in conjunction. It was imperative that this process be thorough and complete because any errors in the design of the database could result in future setbacks and allow certain features of the product to not function properly. Next, the team familiarized themselves with designing a graphical user interface within Java, as well as how to link the user interface to the back-end database system. Once crucial step implemented for the development process was to ensure that the lead programmer perform a code walkthrough of each team members implementations to guarantee proper functionality. After each member found a productive workflow for designing a window and creating the code that brings functionality, the lead programmer assisted each member with implemented the back end database.

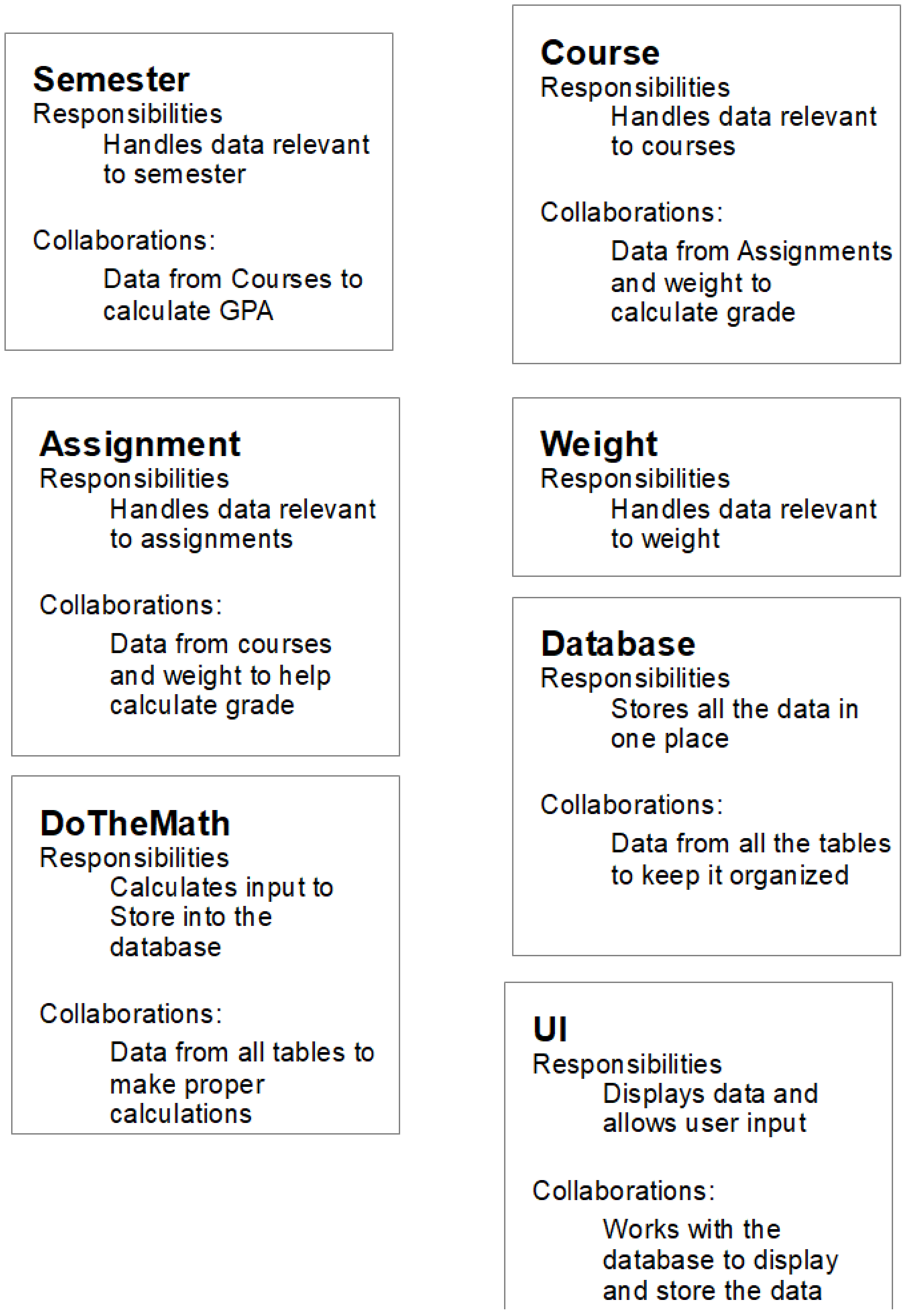
12.2 Documentation

The documentation was imperative to the development process. It allowed the team to understand what functionality or designs have been implemented, and what needs to be developed. Once a member completed part of the program, they documented exactly what they changed and when.

13. Class Diagram



14. CRC Cards



Semester
Responsibilities
Handles data relevant to semester

Collaborations:
Data from Courses to calculate GPA

Course
Responsibilities
Handles data relevant to courses

Collaborations:
Data from Assignments and weight to calculate grade

Assignment
Responsibilities
Handles data relevant to assignments

Collaborations:
Data from courses and weight to help calculate grade

Weight
Responsibilities
Handles data relevant to weight

DoTheMath
Responsibilities
Calculates input to Store into the database

Collaborations:
Data from all tables to make proper calculations

Database
Responsibilities
Stores all the data in one place

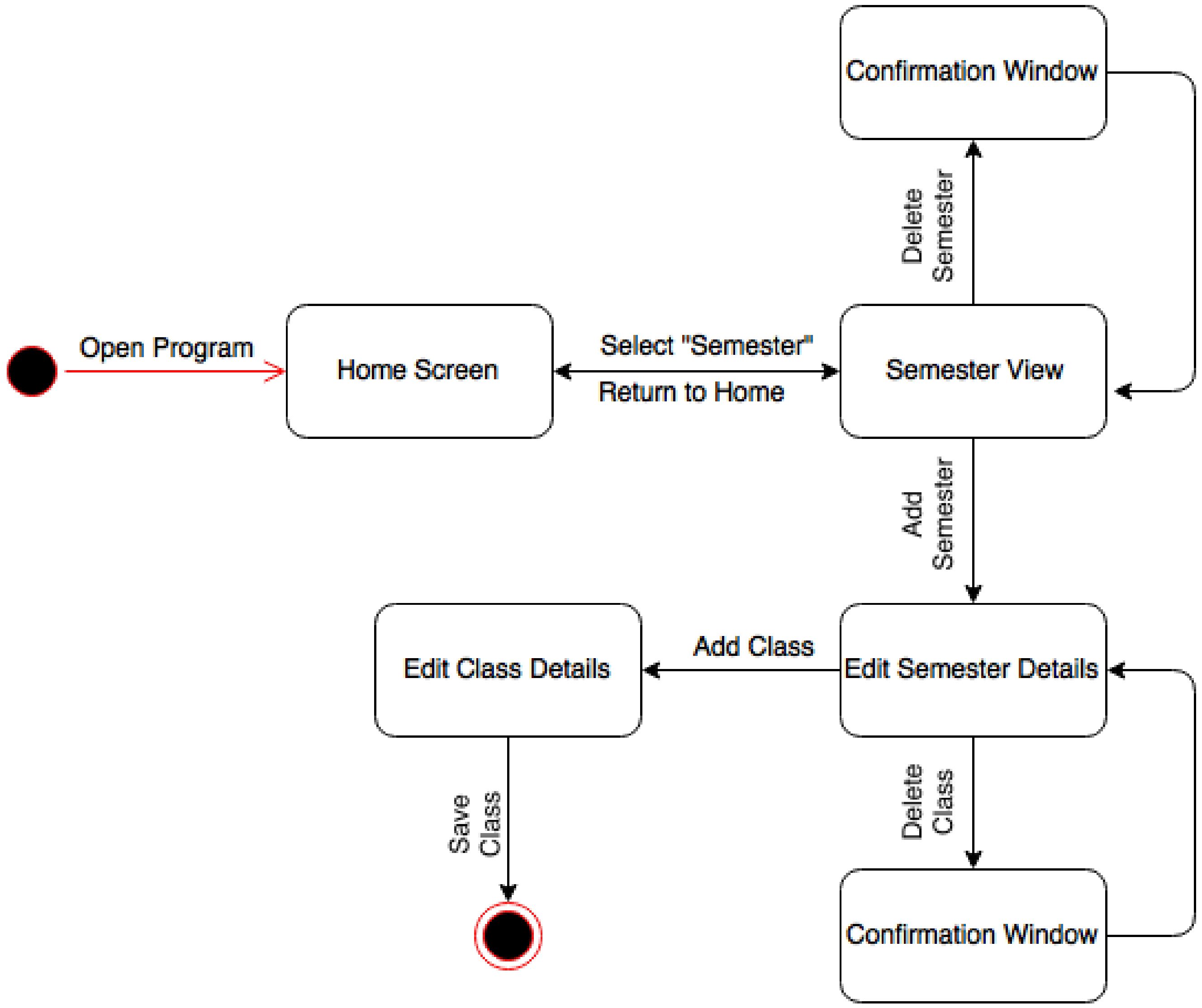
Collaborations:
Data from all the tables to keep it organized

UI
Responsibilities
Displays data and allows user input

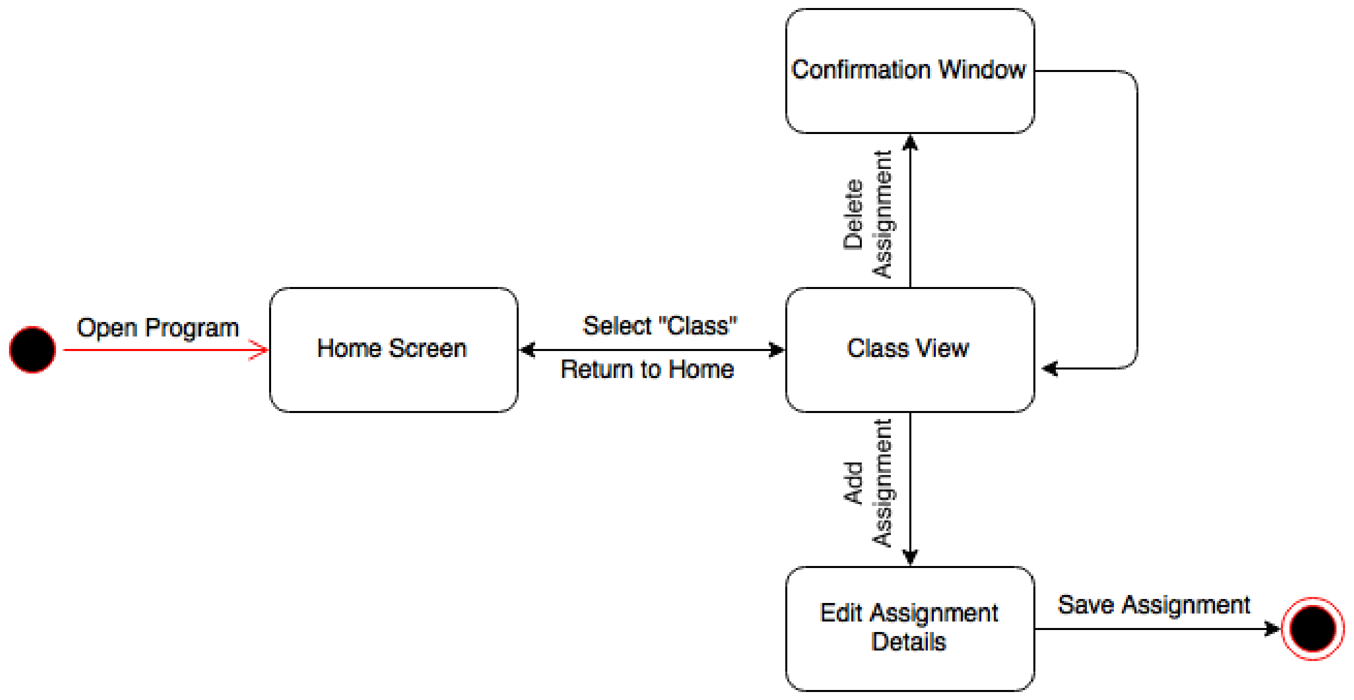
Collaborations:
Works with the database to display and store the data

15. State Diagrams

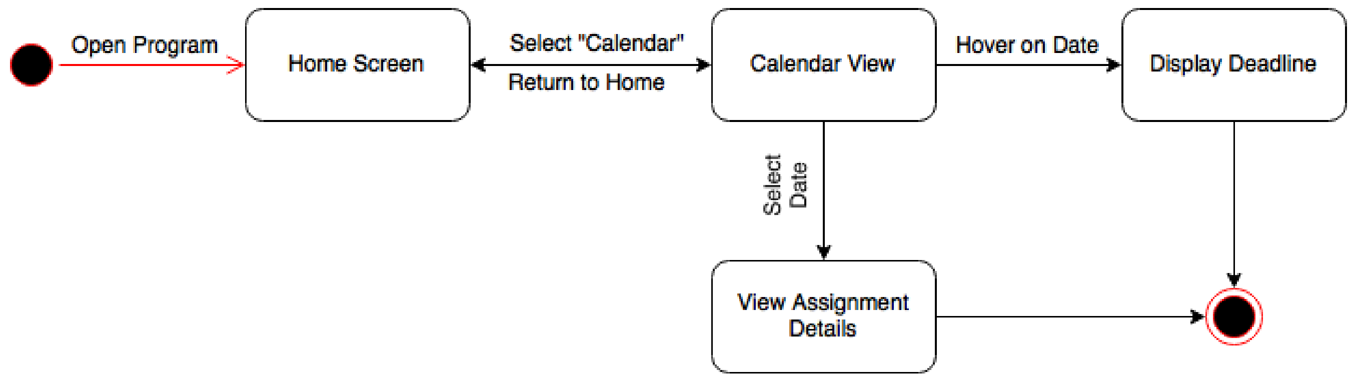
15.1 Semester Window



15.2 Class Window



15.3 Semester Window



16. Test Cases

16.1 Test Case 1

Test Case #1:	Test Case Name: Add Semester
System: Tuffy’s Trapper Keeper	Subsystem: N/A
Designed By: Samarth	Design Date: 9/30/17
Executed By: Jose	Execution Date:
Short Description: Add a semester that can hold classes	

Pre-Conditions:

None

Step	Action	Expected System Response	Pass/Fail	Comment
1	Click ‘Semester’ button	System will show add semester	P	
2	Click ‘Add’ button	Add semester window opens	P	
3	Click ‘Save’ button	System will save in database	P	

Post-Conditions:

1. The new semester is saved in the database.

16.2 Test Case 2

Test Case #2:	Test Case Name: Add class
System: Tuffy’s Trapper Keeper	Subsystem: N/A
Designed By: Vanchiit	Design Date: 09/31/17
Executed By: Juan	Execution Date:
Short Description: Add a class to a specific semester	

Pre-Conditions:

1. There is already a semester created

Step	Action	Expected System Response	Pass/Fail	Comment
1	Click ‘Classes’ button	System displays ‘add class’	P	
2	Click ‘Add’ button	Add class window opens	P	
3	Click ‘Save’ button	System will save in database	P	

Post-Conditions:

1. The new class is saved in the database

16.3 Test Case 3

Test Case #3:	Test Case Name: View Semester
System: Tuffy's Trapper Keeper	Subsystem: N/A
Designed By: Jose	Design Date: 10/01/17
Executed By: Samarth	Execution Date:
Short Description: view the list of semesters	

Pre-Conditions:

1. The user has already created a semester

Step	Action	Expected System Response	Pass/Fail	Comment
1	Click 'semester' button	System loads table with semesters	P	
2	Click 'view' button	Class list window opens	P	

Post-Conditions:

1. The window will display the classes for that particular semester.

16.4 Test Case 4

Test Case #4:	Test Case Name: View Classes
System: Tuffy’s Trapper Keeper	Subsystem: N/A
Designed By: Jose	Design Date: 10/03/17
Executed By: Jose	Execution Date:
Short Description: view the list of classes	

Pre-Conditions:

1. The user has created a semester that can hold classes
2. The user has created classes within that semester

Step	Action	Expected System Response	Pass/Fail	Comment
1	Click ‘classes’ button	Shows table of classes	P	
2	Click ‘view’ button	Class window displays	P	

Post-Conditions:

1. A window will display that shows the class information

16.5 Test Case 5

Test Case #5:	Test Case Name: Remove Class
System: Tuffy’s Trapper Keeper	Subsystem: N/A
Designed By: Juan	Design Date: 11/15/17
Executed By: Vanchiit	Execution Date:
Short Description: Delete a class from the database	

Pre-Conditions:

1. The user has created a class

Step	Action	Expected System Response	Pass/Fail	Comment
1	Click ‘classes’ button	Table of classes displays	P	
2	Click ‘delete’ button	Class removed	P	

Post-Conditions:

1. The class should be removed from the database.

16.6 Test Case 6

Test Case #6:	Test Case Name: Add Assignment
System: Tuffy’s Trapper Keeper	Subsystem: N/A
Designed By: Samarth	Design Date: 11/20/17
Executed By: Juan	Execution Date:
Short Description: Add a new assignment for a class	

Pre-Conditions:

1. The user has created a class

Step	Action	Expected System Response	Pass/Fail	Comment
1	Click ‘Add Assignment’	Add assignment window displays	P	
2	Click ‘save’ button	Database saves new assignment	P	

Post-Conditions:

1. The database should save the assignment in the database

16.7 Test Case 7

Test Case #7:	Test Case Name: View Assignment
System: Tuffy's Trapper Keeper	Subsystem: N/A
Designed By: Jose	Design Date: 11/16/17
Executed By: Vanchiit	Execution Date:
Short Description: View an assignment that is in the database	

Pre-Conditions:

1. The user should have created an assignment

Step	Action	Expected System Response	Pass/Fail	Comment
1	Click 'show assign'	Table with assignments displays	P	

Post-Conditions:

1. A table will populate with the list of assignments for that particular class.

16.8 Test Case 8

Test Case #8:	Test Case Name: Add notes
System: Tuffy’s Trapper Keeper	Subsystem: N/A
Designed By: Juan	Design Date: 11/30/17
Executed By:Jose	Execution Date:
Short Description: Save notes for a specific class	

Pre-Conditions:

1. The user should have created a semester

Step	Action	Expected System Response	Pass/Fail	Comment
1	Click ‘notes button	Display notes window	P	
2	Click ‘save’ button	Database saves note entry	P	

Post-Conditions:

1. The database should save the notes in the database.

16.2 Test Case 9

Test Case #9:	Test Case Name: Remove Assignment
System: Tuffy’s Trapper Keeper	Subsystem: N/A
Designed By: Vanchiit	Design Date: 11/30/17
Executed By: Jose	Execution Date:
Short Description: Delete an assignment	

Pre-Conditions:

1. The user should have already created an assignment.

Step	Action	Expected System Response	Pass/Fail	Comment
1	Click ‘show assignme’	Display list of assignments	P	
2	Click ‘delete’ button	Assignment is removed	P	

Post-Conditions:

1. The database removes the assignment entry.

16.10 Test Case 10

Test Case #10:	Test Case Name: View Calendar
System: Tuffy’s Trapper Keeper	Subsystem: N/A
Designed By: Juan	Design Date:12/01/17
Executed By: Samarth	Execution Date:
Short Description: Display a calendar populated with assignments	

Pre-Conditions:

1. The user must create assignments

Step	Action	Expected System Response	Pass/Fail	Comment
1	Click ‘calendar’	Calendar displays	P	

Post-Conditions:

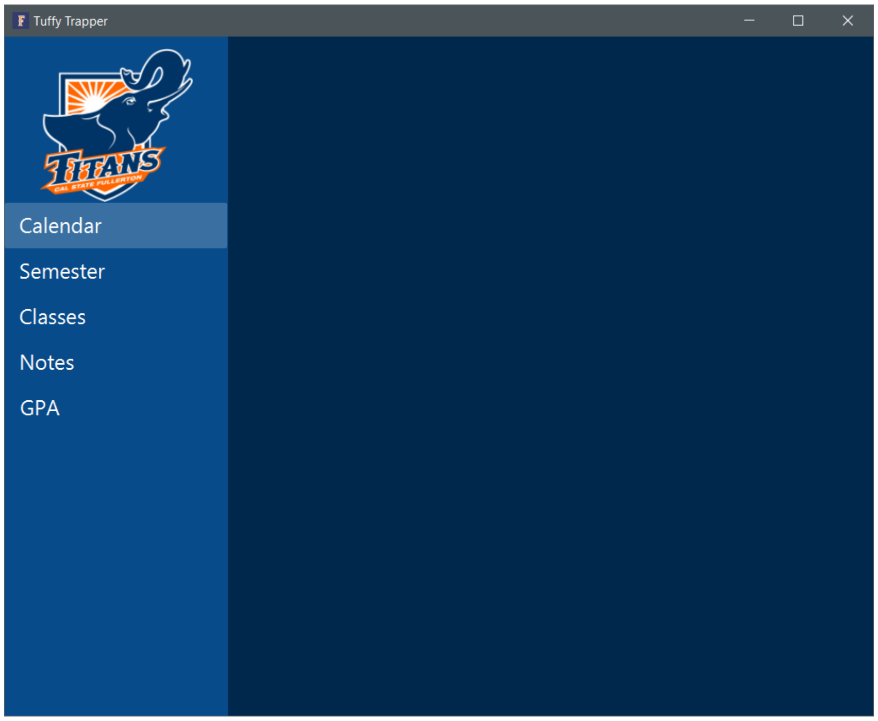
1. The calendar displays the assignments in a calendar view.

17. User Manual

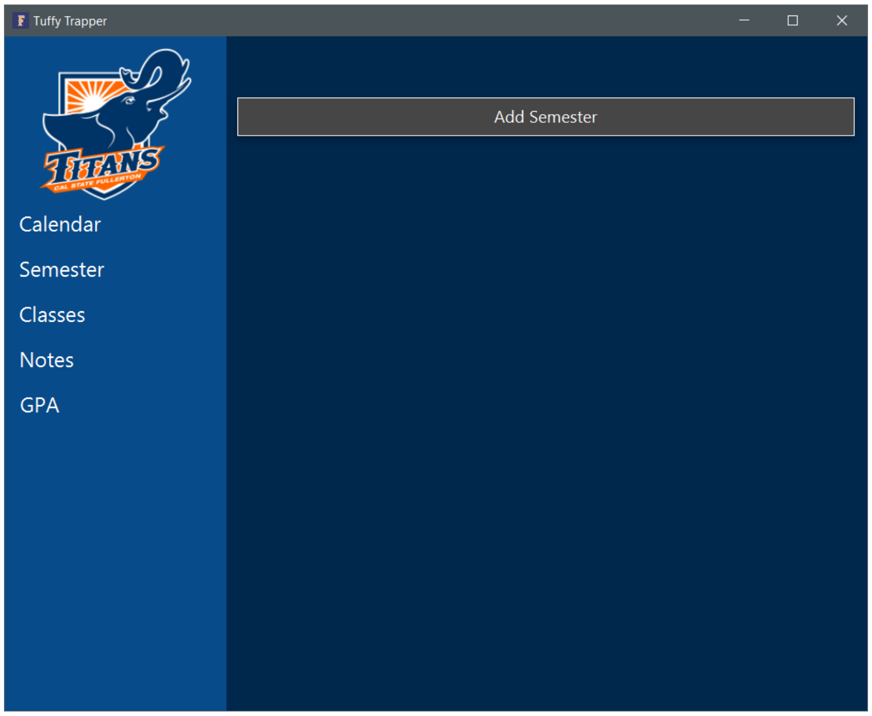
The following will provide details on how to use Tuffy's Trapper Keeper 2017.

17.1 Creating a semester

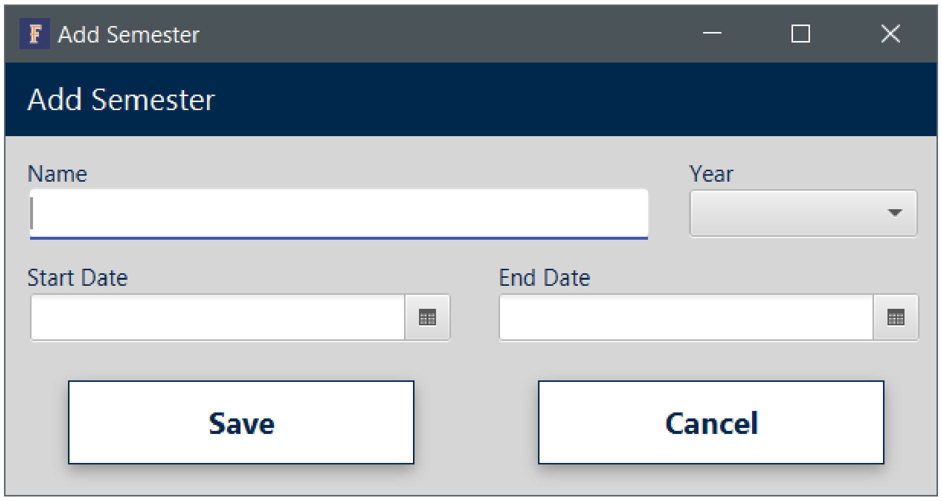
Here is a view of the main screen



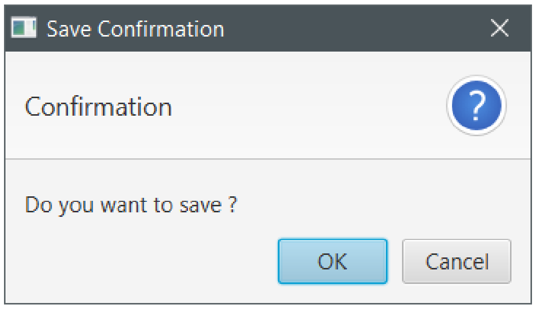
In order to create a new semester, first click on "Semester" and then click "Add Semester."



A new window will pop-up and you can now fill out the information for your semester.

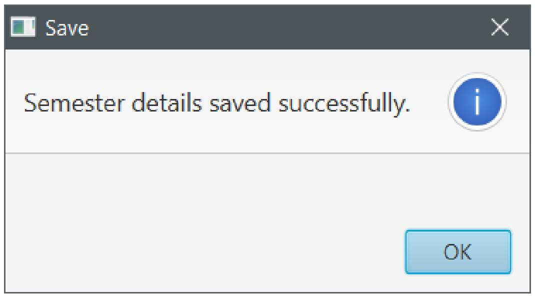


When complete, please click save. You will see the following screen.



Do you want to save ?

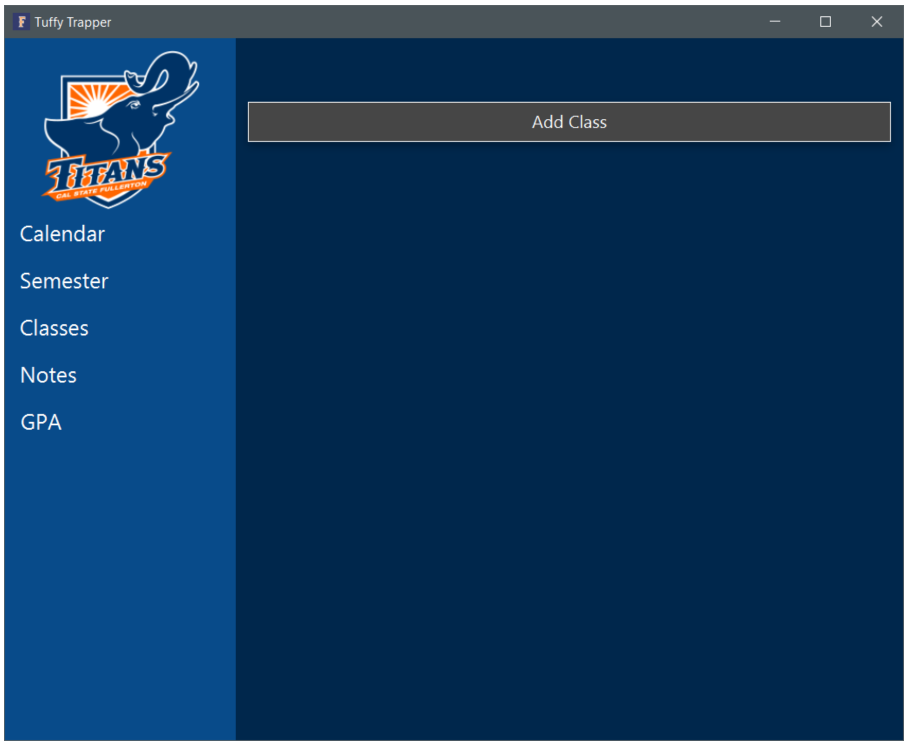
Click “OK” to confirm.



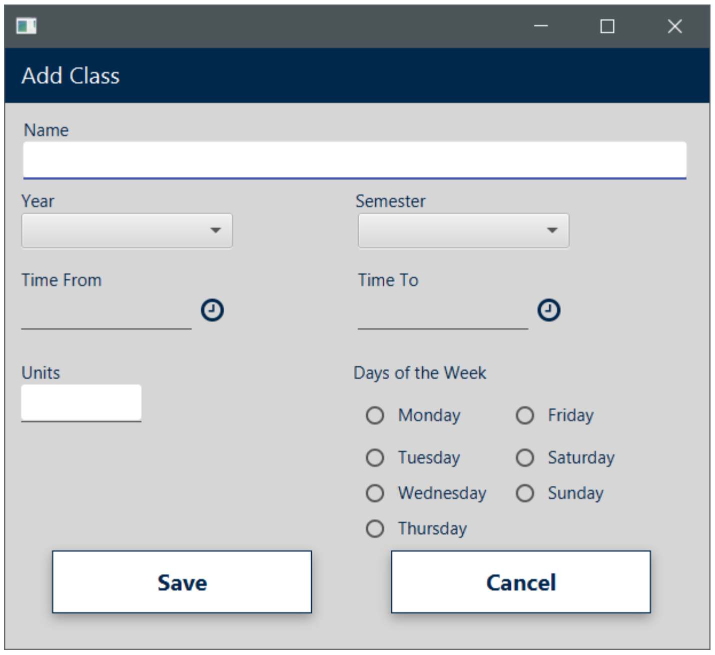
Semester details saved successfully.

17.2 Creating a class

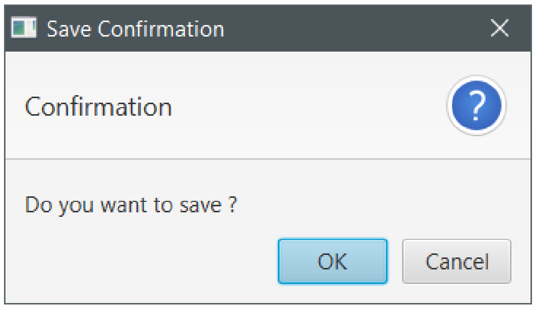
From the main screen, click on “Classes” and then click on “Add class” button.



The following window will pop-up.



Enter the information for your class. When complete, please click save. You will see the following screen.

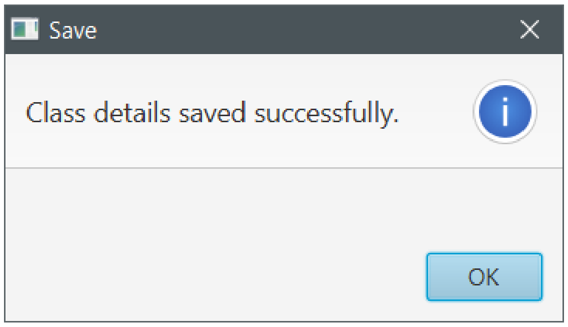


Save Confirmation

Confirmation

Do you want to save ?

Click “OK” to confirm.

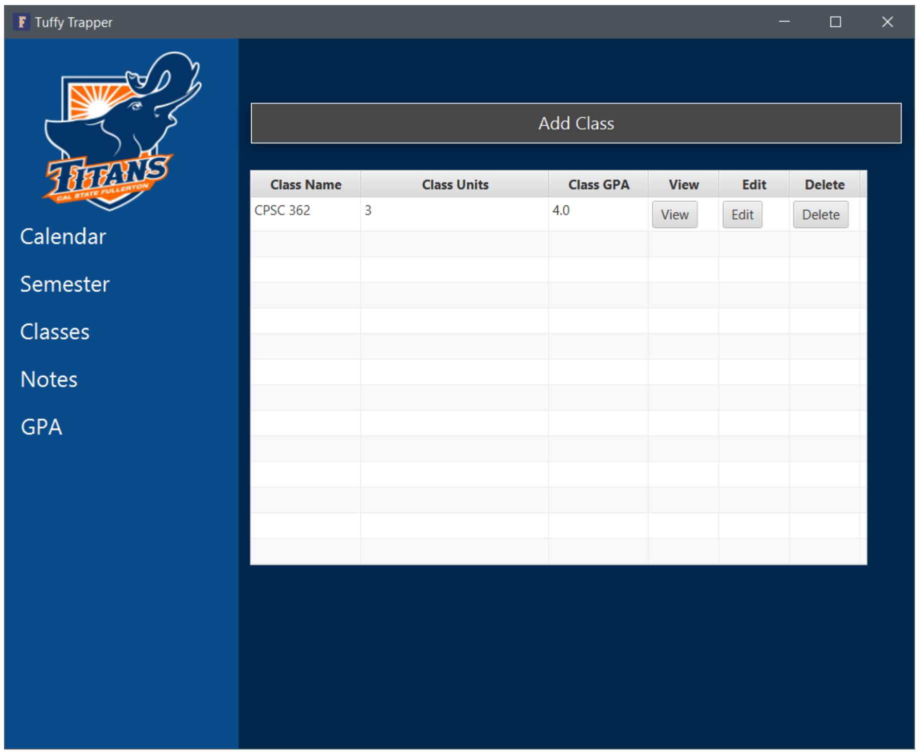


Save

Class details saved successfully.

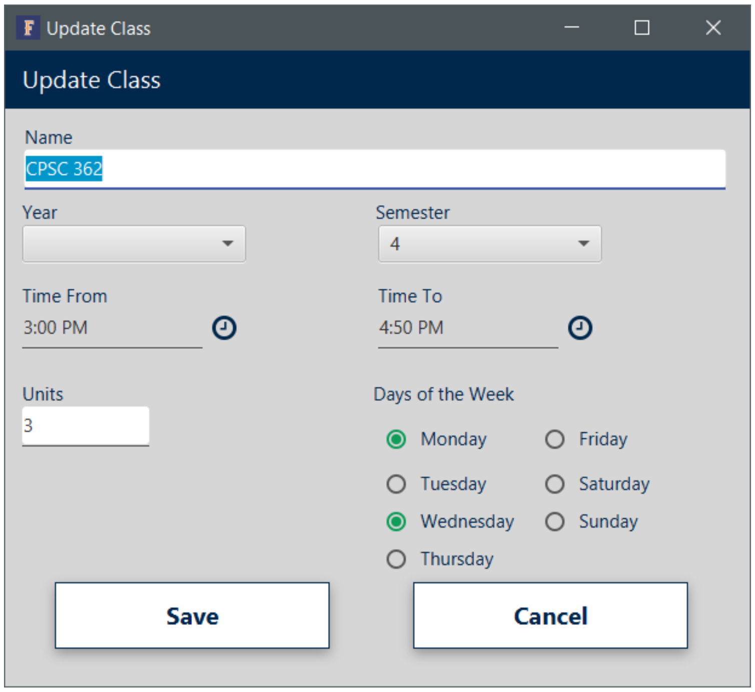
17.3 Updating a class

From the main screen, click on "classes."



Class Name	Class Units	Class GPA	View	Edit	Delete
CPSC 362	3	4.0	View	Edit	Delete

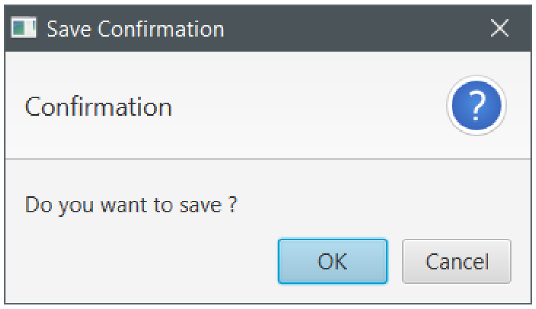
Click on the "Edit" next to the class that you want to update.



Name	CPSC 362
Year	
Semester	4
Time From	3:00 PM
Time To	4:50 PM
Units	3

Days of the Week: [✓ Monday] Tuesday [✓ Wednesday] Thursday Friday Saturday Sunday

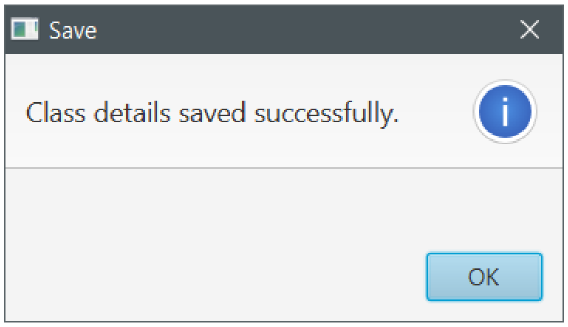
A new window will display, allowing you to edit the class information. When done, click "Save." You will see the following screen.


Save Confirmation

Confirmation

Do you want to save ?

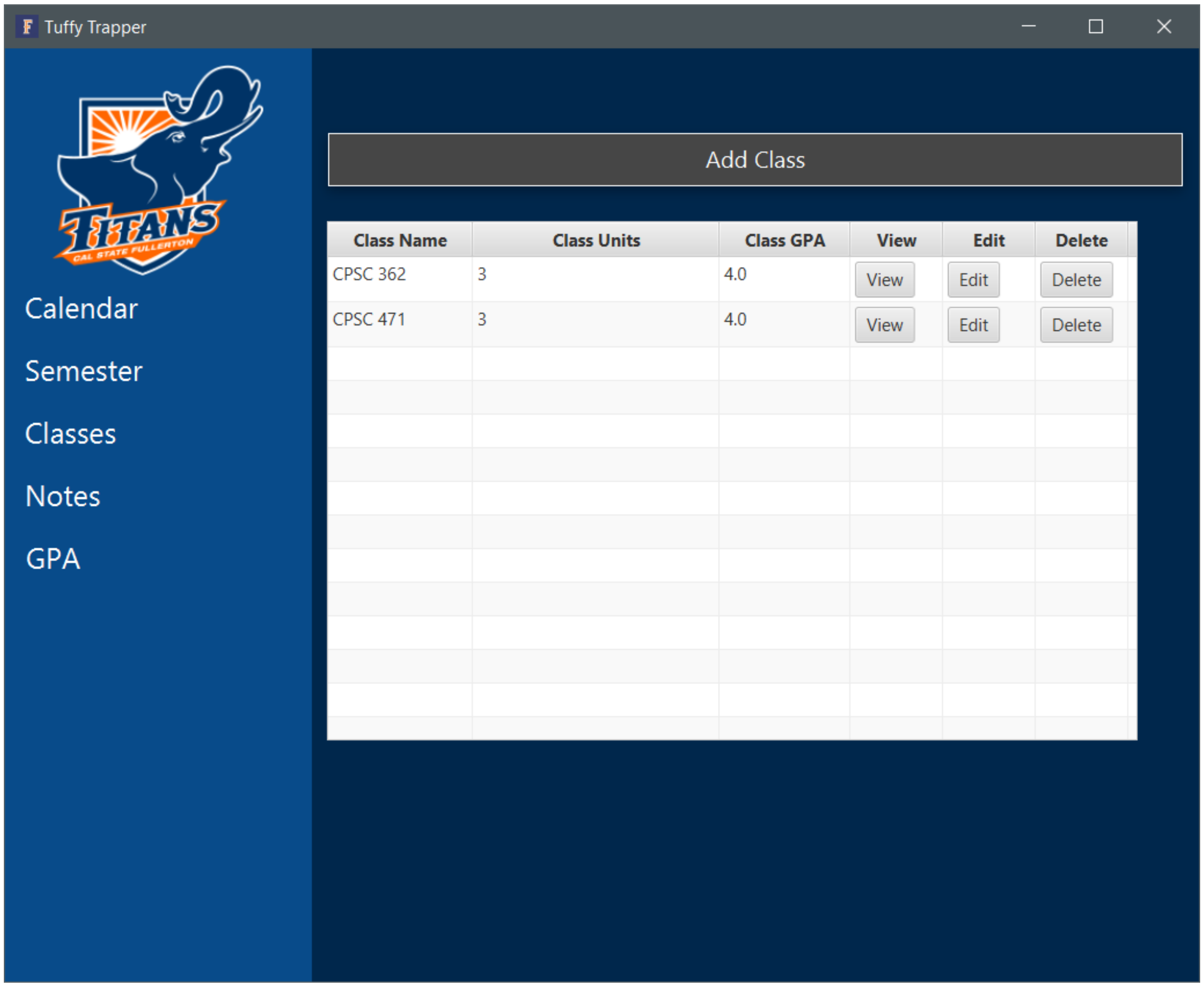
Click “OK” to confirm.


Save

Class details saved successfully.

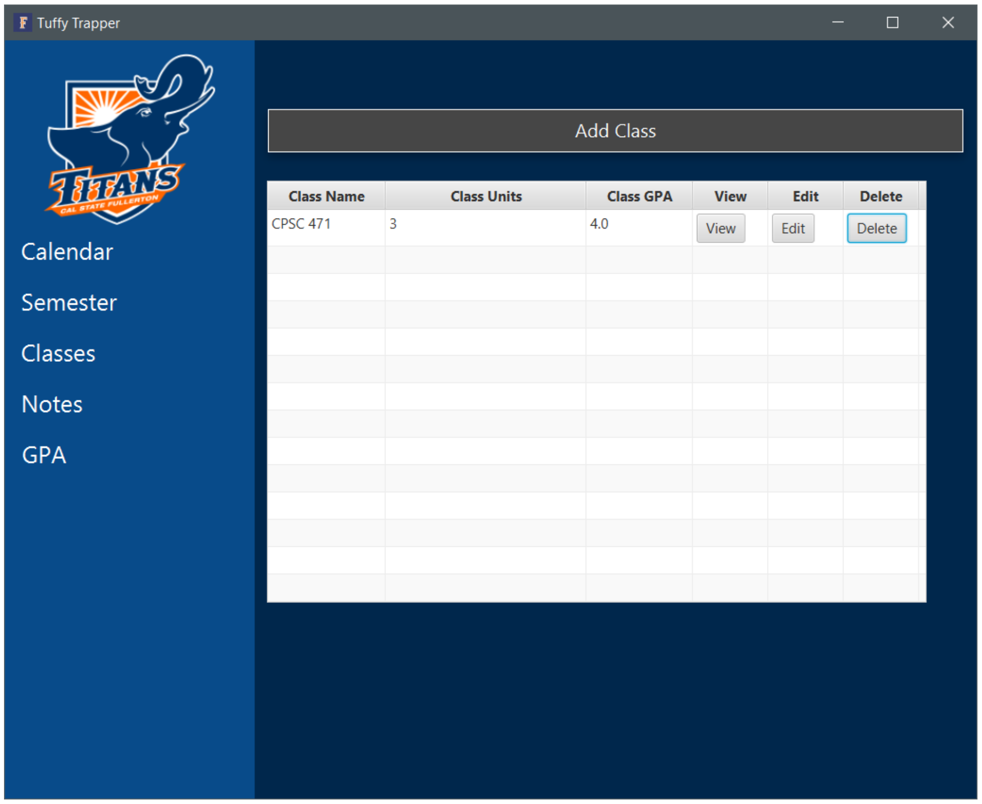
17.4 Deleting a class

From the main menu click on “Classes.”



Class Name	Class Units	Class GPA	View	Edit	Delete
CPSC 362	3	4.0	View	Edit	Delete
CPSC 471	3	4.0	View	Edit	Delete

Then click on “Delete” next to the class you want to remove.

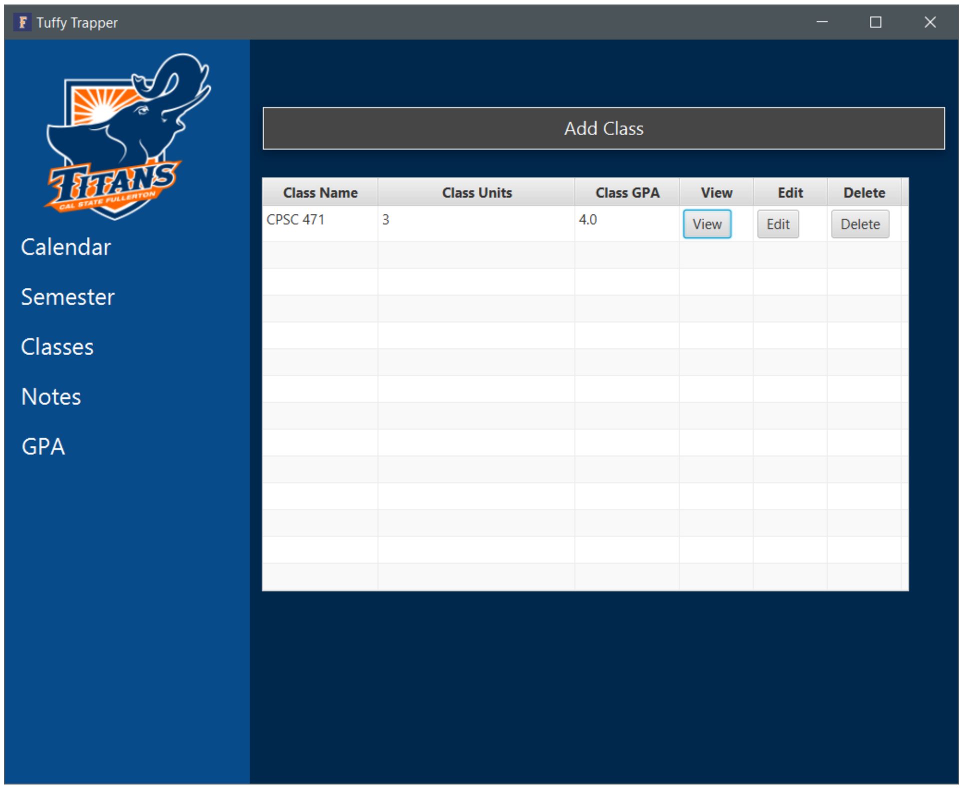


Class Name	Class Units	Class GPA	View	Edit	Delete
CPSC 471	3	4.0	View	Edit	Delete

The class will be removed from the database.

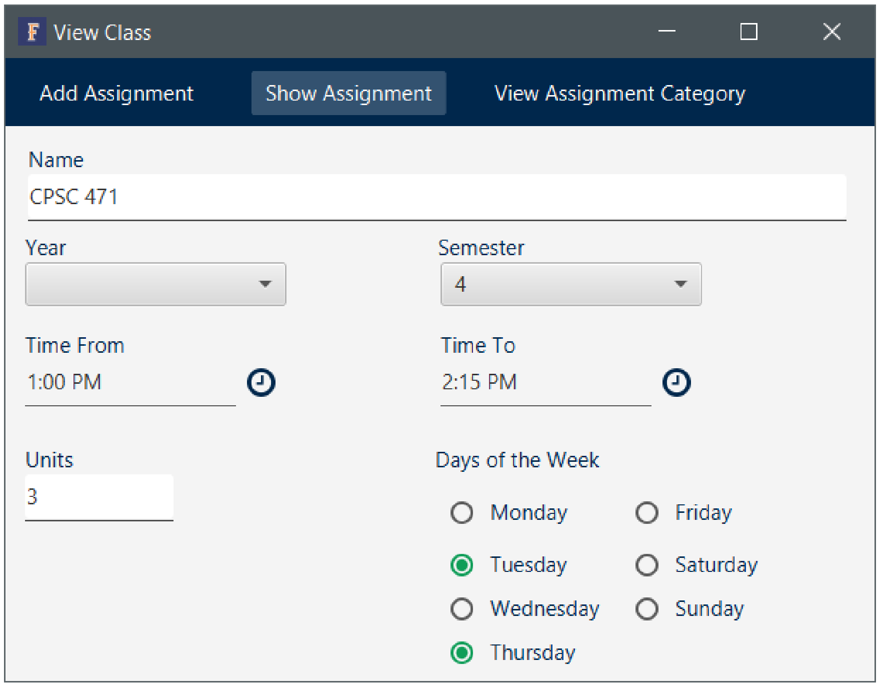
17.5 Add assignment

From the main screen, click on “Classes” and then click on “View” next to the class you want to add an assignment for.



Class Name	Class Units	Class GPA	View	Edit	Delete
CPSC 471	3	4.0	View	Edit	Delete

The following window will pop-up.



Name: CPSC 471

Year:

Semester: 4

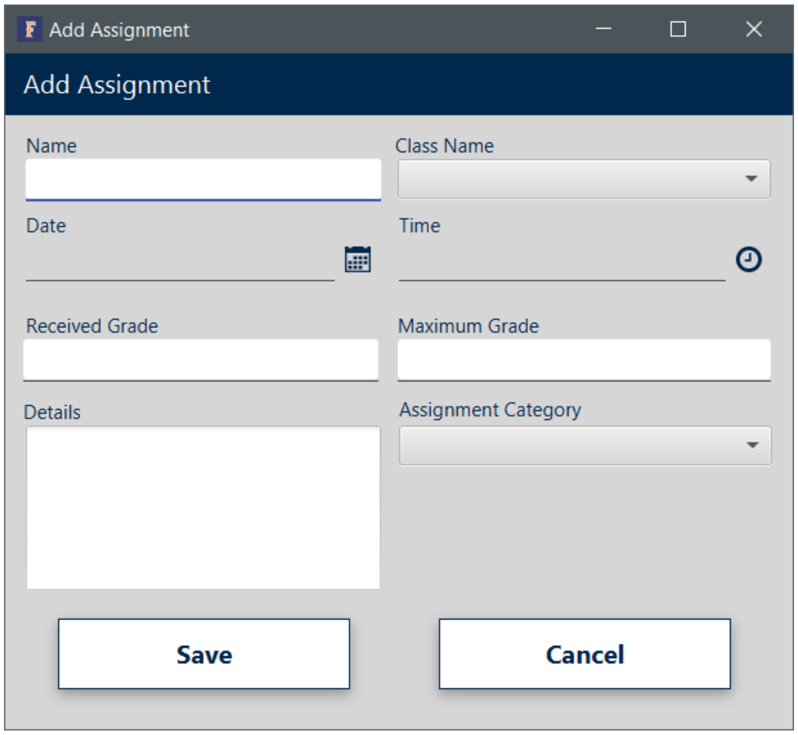
Time From: 1:00 PM

Time To: 2:15 PM

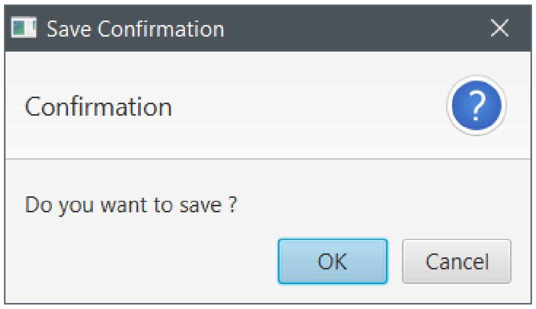
Units: 3

Days of the Week: Monday, [✓ Tuesday], Wednesday, [✓ Thursday], Friday, Saturday, Sunday

Next, click on "Add assignment."

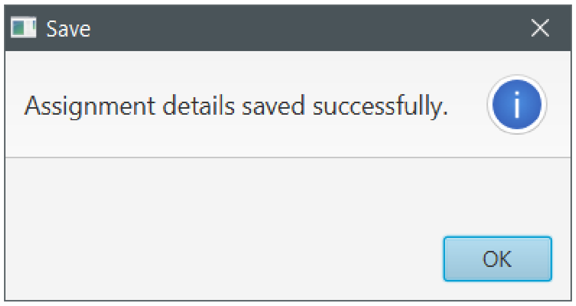


Populate each field with the necessary information. Then click save. You will see the following screen.



Do you want to save ?

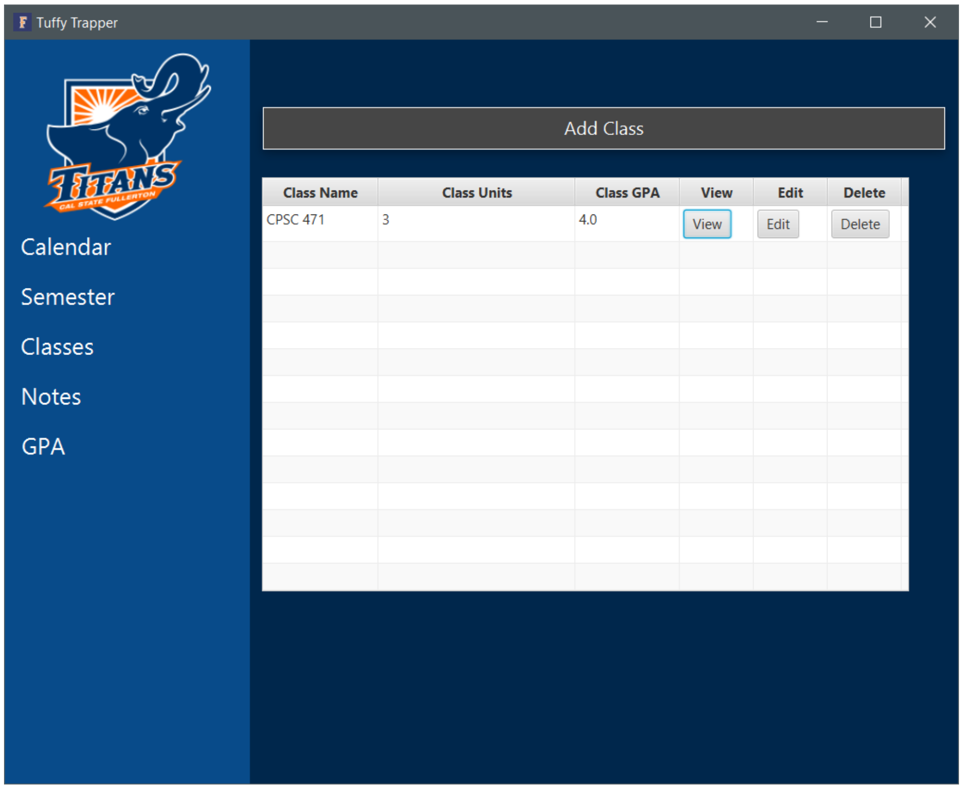
Click "OK" to confirm.



Assignment details saved successfully.

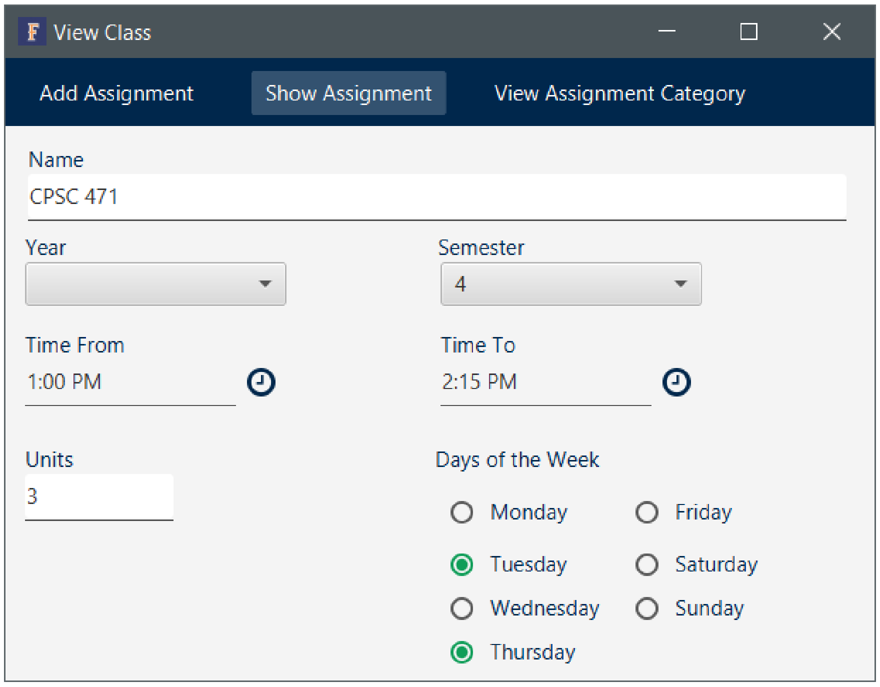
17.6 View assignments

From the main screen, click on "Classes" and then click on "View" next to the class you want to add an assignment for.



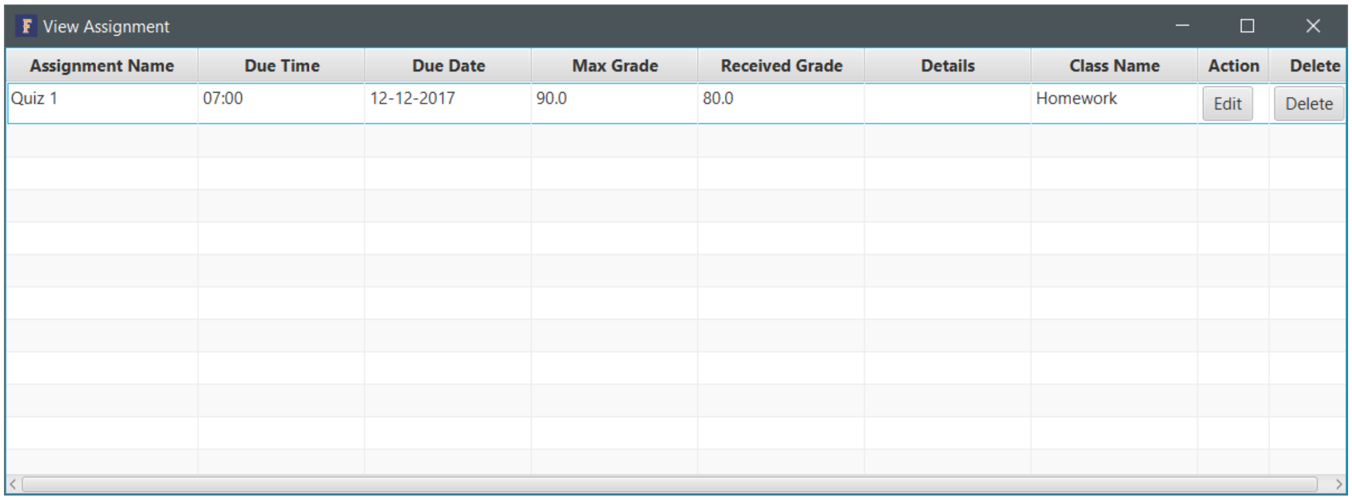
Class Name	Class Units	Class GPA	View	Edit	Delete
CPSC 471	3	4.0	View	Edit	Delete

The following window will pop-up. Then, click on "Show assignment."



Name: CPSC 471
Year:
Semester: 4
Time From: 1:00 PM
Time To: 2:15 PM
Units: 3
Days of the Week: Monday, [✓ Tuesday], Wednesday, [✓ Thursday], Friday, Saturday, Sunday

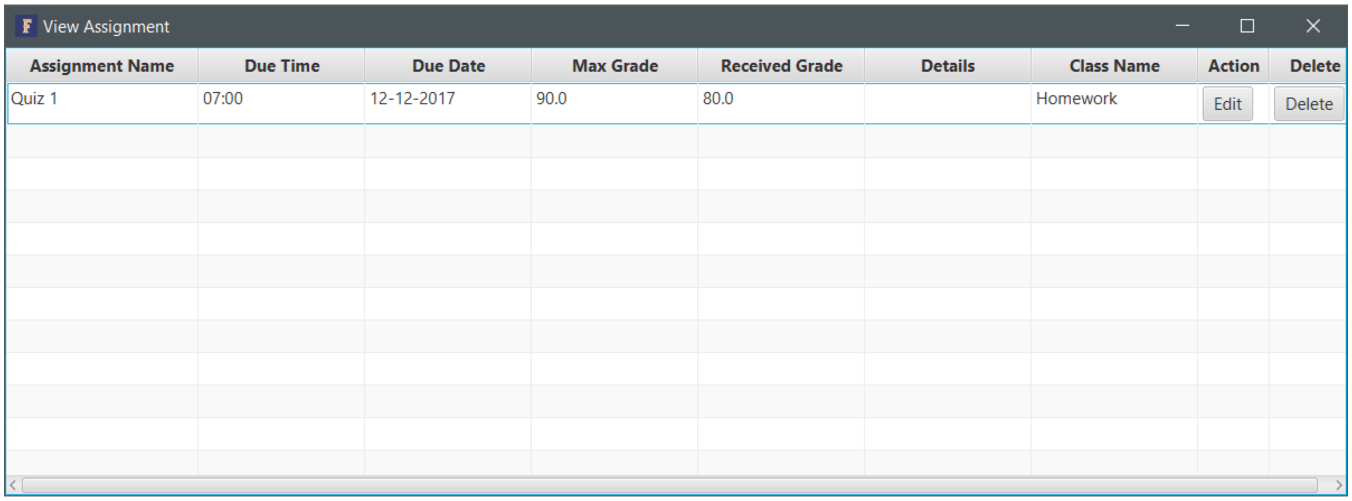
The following window will display the list of assignments.



Assignment Name	Due Time	Due Date	Max Grade	Received Grade	Details	Class Name	Action	Delete
Quiz 1	07:00	12-12-2017	90.0	80.0		Homework	Edit	Delete

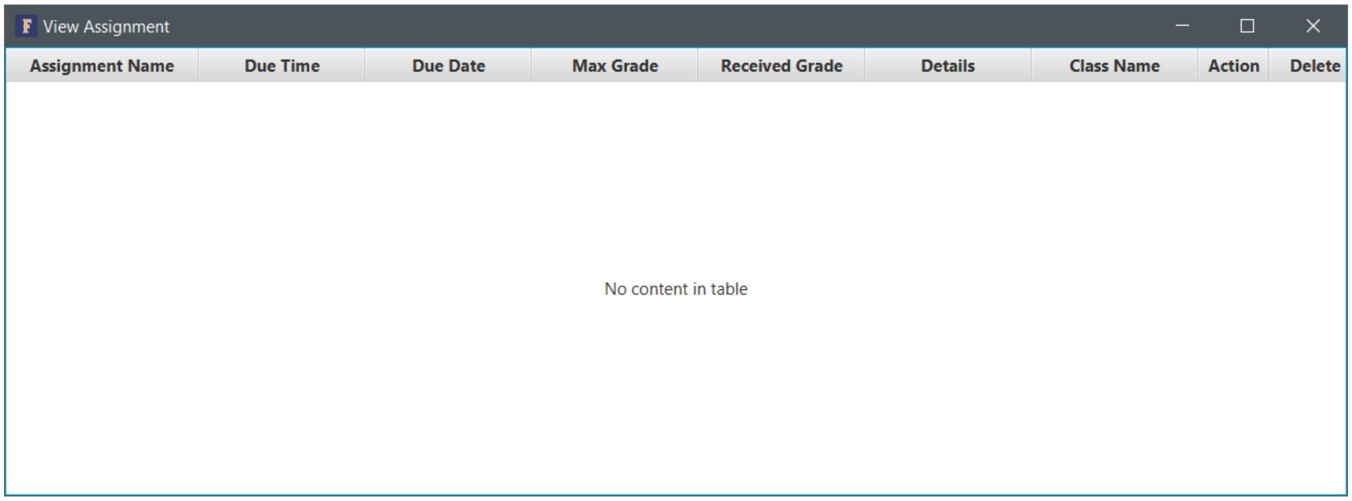
17.7 Delete assignments

From the same window, click on the "delete" button next to the assignment you want to remove.


View Assignment

Assignment Name	Due Time	Due Date	Max Grade	Received Grade	Details	Class Name	Action	Delete
Quiz 1	07:00	12-12-2017	90.0	80.0		Homework	Edit	Delete

The assignment will be removed.

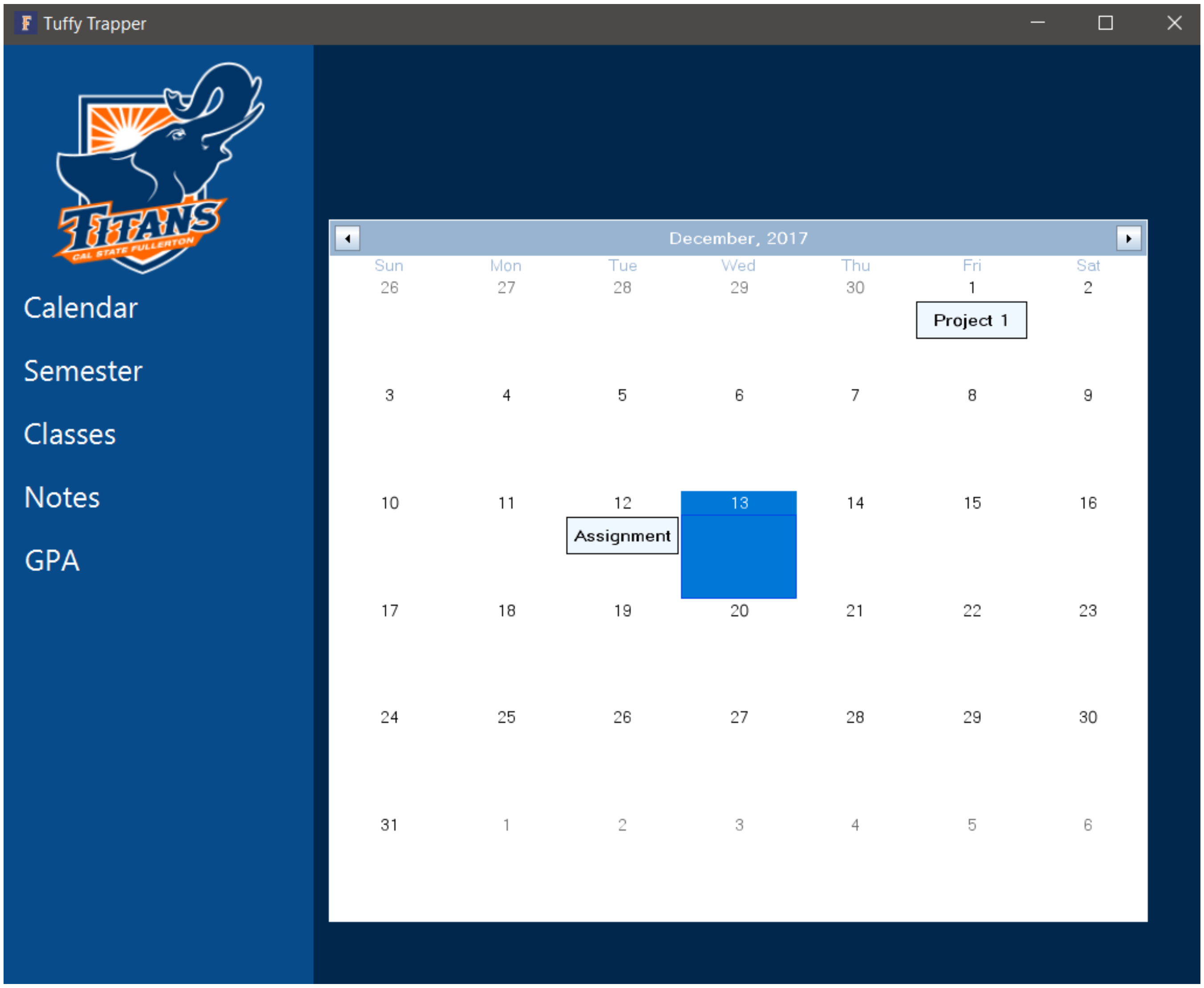

View Assignment

Assignment Name	Due Time	Due Date	Max Grade	Received Grade	Details	Class Name	Action	Delete

No content in table

17.8 Show Calendar

From the main window, click on “Calendar”



Sun	Mon	Tue	Wed	Thu	Fri	Sat
26	27	28	29	30	1 Project 1	2
3	4	5	6	7	8	9
10	11	12 Assignment	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

The calendar will show your upcoming assignments.