1 Release Information

Release	*1*
Leader	*Makenna Halvensleben*

Give a one to two sentence description of new functionality provided by this release.

This release includes the implementation of our data objects and basic functionality of GUI.

2 Progress

Stories and spike solutions worked on during this release				
Issue #	**Labels**	**Description**	**% Impl.**	**Notes, plan for completion**
#1	*Story*	*As a user, I need to view my gpa, so that I know how I'm doing in school.*	*0%*	*Ran out of time to properly implement and lacked backend data that would be necessary. Will implement when the backend is developed and historical GPA data is available.*
#2	*Story*	*As a user, I need to enter how many hours I studied on a subject, so that I can track my study habits.*	*50%*	*We have implemented the backend but we have not implemented the front end part so the user cannot see it yet.*
#3	*Story*	*As a user, I need to be able to set up an account.*	*50%*	*We have to create the main page and connect it to the database calls but we have the backend set up and not the front end quite yet.*
#4	*Story*	*As a user, I need to be able to enter my grades for individual classes, so that I can track my grades.*	*50%*	*We have the frontend part but we don't have any backend implemented for adding grades.*
#5	*Story*	*As a user, I need to be able to enter assignments and exams, so that I can plan my study time for them.*	*50%*	*We have the frontend part completed but not the backend part quite yet, we*

				are working on the database side of that.
#9	*Story*	*As a user, I need the ability to add classes, so that I can get study goals for these classes.*	*50%*	*Again the frontend is done it is just connecting the right calls to the right database queries in order to assure we have added classes and can remove them from the tables in the database.*
#17	*Story*	*As a user, I need the ability to remove classes, so that I can get rid of classes I have dropped.*	*25%*	*We have part of the backend completed its just connecting it with the database part that we need to fix.*
#18	*Task*	*Create a Graph class*	*100%*	
#19	*Task*	*Create a GPAGraph class*	*0%*	*Ran out of time to properly implement and lacked backend data that would be necessary. Will implement when the backend is developed and historical GPA data is available.*
#20	*Task*	*Create a StudyLog class*	*100%*	
#21	*Task*	*Create a StudyLogGraph class*	*100%*	
#22	*Task*	*Create an Account class*	*100%*	
#23	*Task*	*Create an SQL Lite Database*	*100%*	
#24	*Task*	*Create EnterGrade method for assignments*	*100%*	
#25	*Task*	*Create CalculateGrade method for courses*	*100%*	
#26	*Task*	*Crate GradeGraph class*	*100%*	
#27	*Task*	*Crate Assignment class*	*100%*	
#28	*Task*	*Create AddAssignment method for courses*	*100%*	
#29	*Task*	*Create a Course class*	*100%*	
#30	*Task*	*Create AddCourse method for accounts*	*100%*	
#31	*Task*	*Create a RemoveCourse method for accounts*	*100%*	
#32	*Spike*	*Learn C# Form Controls*	*100%*	
#33	*Task*	*Implement logic to control forms*	*100%*	
#34	*Spike*	*Testing Connecting database to C# Console Application*	*100%*	
#35	*Spike*	*Windows Form Graphing*	*100%*	
#36	*Task*	*Create Template Form*	*100%*	
#37	*Task*	*Implement MVC for Graphing on CourseInfoForm*	*100%*	

Accomplishments by engineer		
Team Member	**Total Time (hrs)**	**Stories/tasks implemented**
Nathan	*10*	*#9, #17, #29, #30, #31*
Andrew V.	*10.5*	*#2, #5, #20, #27, #28*
Andrew H.	*11*	*#3, #22, #23, #34*
Peter	*22*	*#1, #2, #4, #18, #19, #21, #26, #35, #36, #37*
Makenna	*11*	*#4, #24, #25, #32, #33*
TEAM TOTAL	***64.5***	

Document updates	
Document	**Changes**
studdyBuddy_iteration1.jpg	*Update class diagram to implement 3-tier architecture.*
Iteration planning record-1.docx	*Updated assigned tasks to match new class names.*

3 Issues

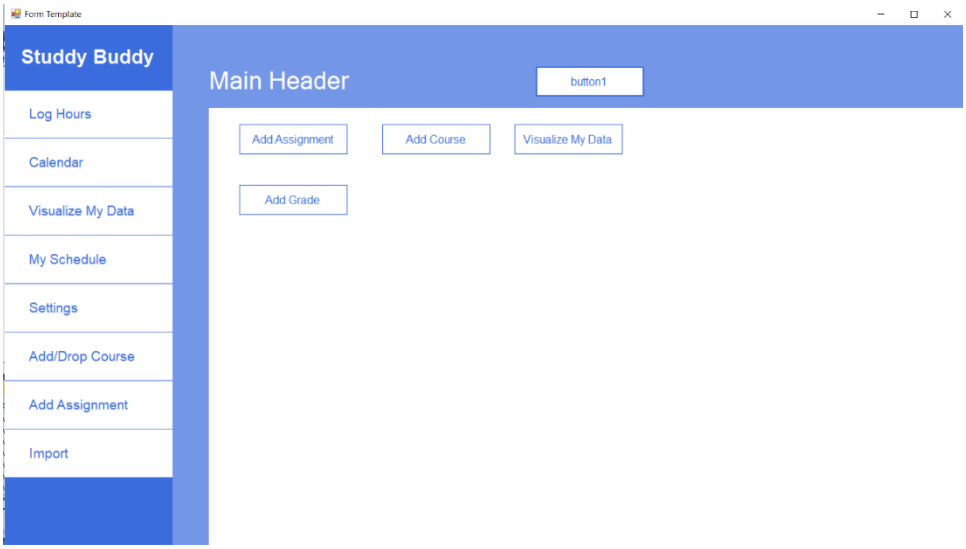
Document anything that is not decided yet and needs to be resolved at some point.

Issue Number	Discovery Date	Resolution Date (Est.–Act.)	Responsible Person	Description (Prob / Resolution)
1	*3/07/2022*	*3/14/2022*	*Andrew Haddad*	*Database is not saving tables correctly, need a fix for this because that is what our application depends on.*
2	*3/07/2022*	*3/14/2022*	*Peter Humphreys*	*Issues with Git. Several files do not*

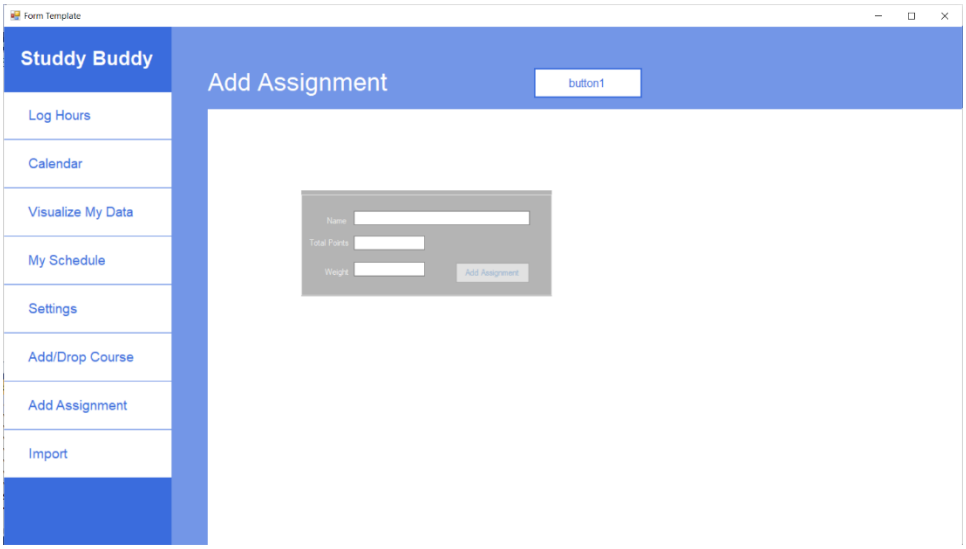
4 Screen Shots

Insert screen shots of functionality added for this release along with screen shots from spike solutions. Add short descriptions of all screen shots along with a short list of features illustrated by each. The goal is to convince your instructor that you completed the work documented above.

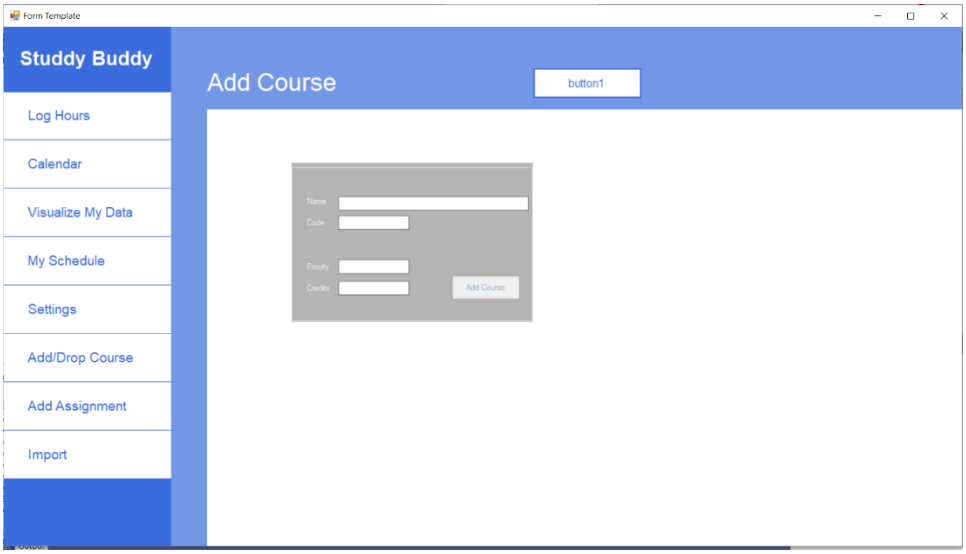
Main Header: This is the homepage of the application, where you can select pages to go to and see the different options.



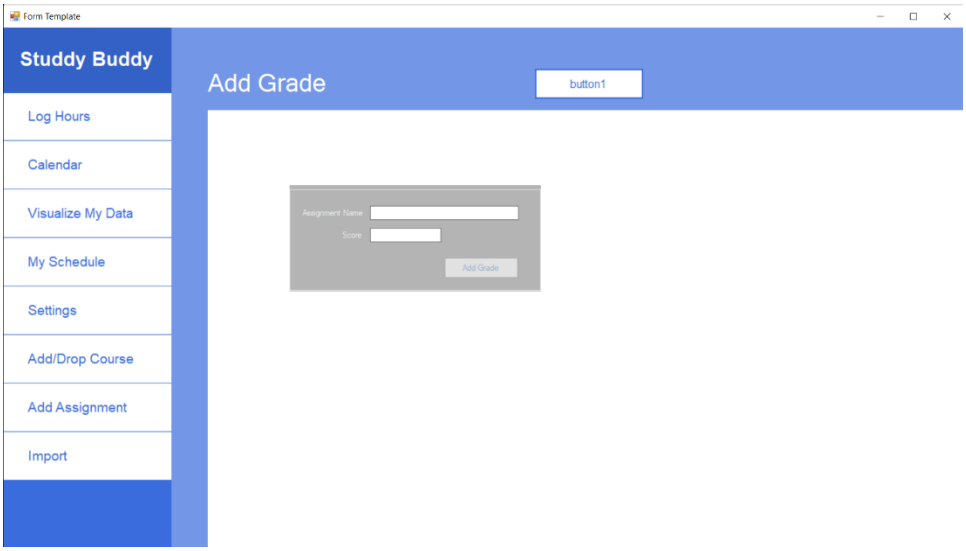
Add Assignment: In this page you can add an assignment that you may have coming up. You have to add the assignment name, the total hours you think it will take you, and the weight of the assignment in the class.



Add Course: In this page you can add courses to your course list.

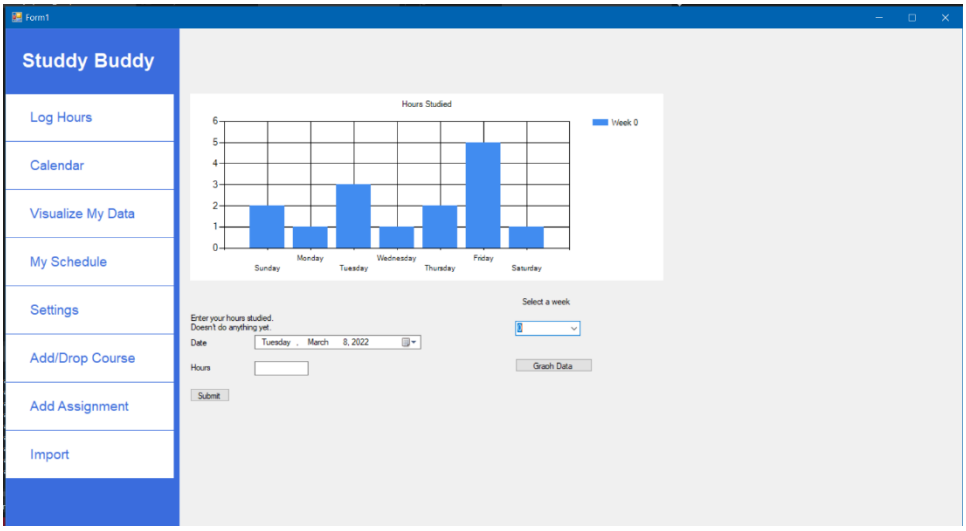


Add Grade: This adds a grade for an assignment or test/exam.

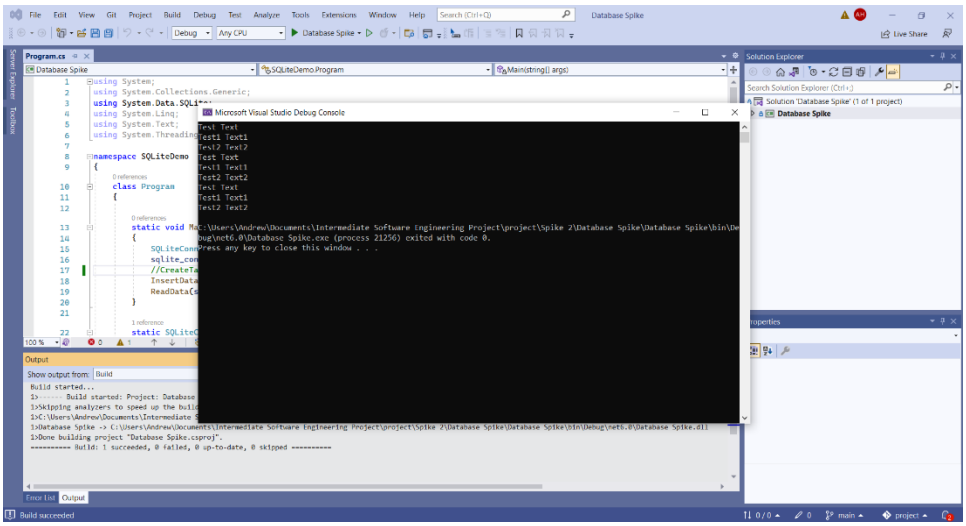


Visualize my data: This shows a visualization of your data, including how your grade has changed throughout the semester through a line graph and a column graph. Indicating how well you have been performing throughout the semester.

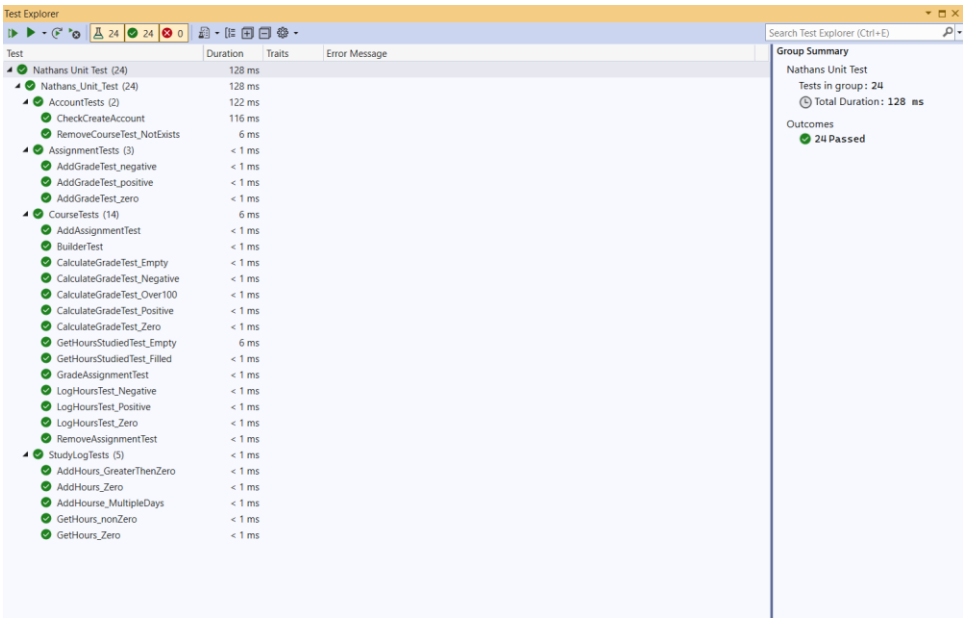
Spike 1: I created a basic graph of study log hours using the Windows form chart user control. This screenshot demonstrates the drawn graph based on the selected week.



Spike 2: I connected an SQLite database to a C# console application. This is a print out of the data, the tables have been created and it calls from the tables to print out that data.



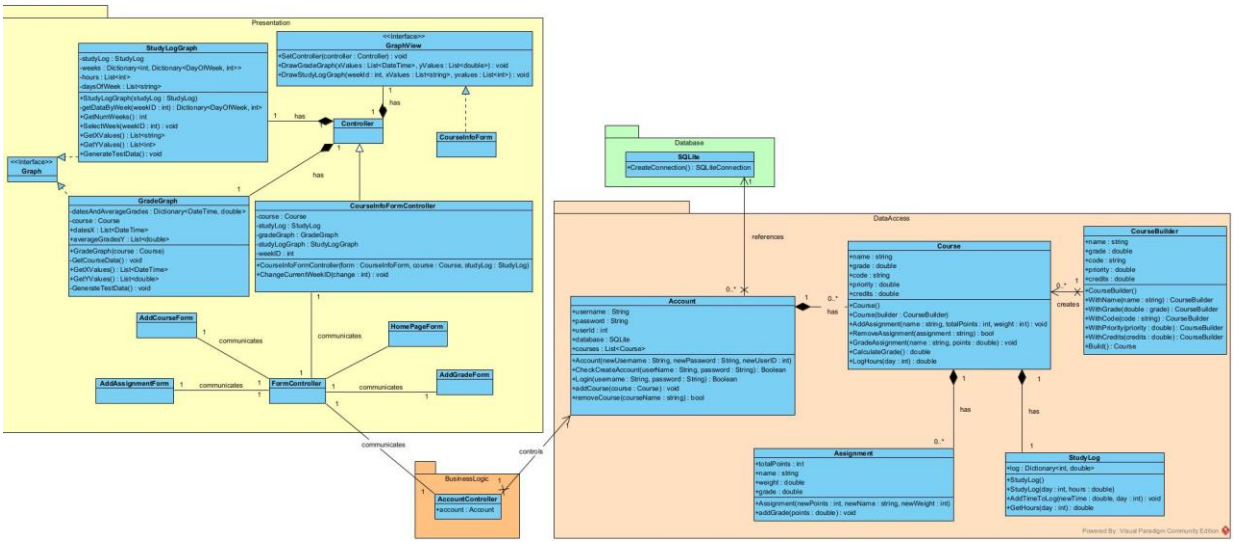
Test Cases: We have successfully run and the test cases and they have all completed successfully.



5 Updated Design

Insert the most up to date design here.

Updated class diagram is posted here. To see the picture better, it is added to the repository here: https://gitlab.com/uwplatt1/se3330-s22/1-fancy-team-name/project/-/blob/main/doc/studyBuddy_iteration1_classDiagram.jpg


Presentation
StudyLogGraph
-studyLog : StudyLog
-weeks : Dictionary<int, Dictionary<DayOfWeek, int>>
-hours : List<int>
-daysOfWeek : List<string>
+StudyLogGraph(studyLog : StudyLog)
-getDataByWeek(weekID : int) : Dictionary<DayOfWeek, int>
+GetNumWeeks() : int
+SelectWeek(weekID : int) : void
+GetXValues() : List<string>
+GetYValues() : List<int>
+GenerateTestData() : void
<<Interface>>
GraphView
+SetController(controller : Controller) : void
+DrawGradeGraph(xValues : List<DateTime>, yValues : List<double>) : void
+DrawStudyLogGraph(weekId : int, xValues : List<string>, yvalues : List<int>) : void
has
Controller
CourseInfoForm
<<Interface>>
Graph
GradeGraph
-datesAndAverageGrades : Dictionary<DateTime, double>
-course : Course
+datesX : List<DateTime>
+averageGradesY : List<double>
+GradeGraph(course : Course)
-GetCourseData() : void
+GetXValues() : List<DateTime>
+GetYValues() : List<double>
-GenerateTestData() : void
CourseInfoFormController
-course : Course
-studyLog : StudyLog
-gradeGraph : GradeGraph
-studyLogGraph : StudyLogGraph
-weekID : int
+CourseInfoFormController(form : CourseInfoForm, course : Course, studyLog : StudyLog)
+ChangeCurrentWeekID(change : int) : void
AddCourseForm
communicates
HomePageForm
AddAssignmentForm
FormController
AddGradeForm
communicates
controls
BusinessLogic
AccountController
+account : Account
Database
SQLite
+CreateConnection() : SQLiteConnection
references
DataAccess
Account
+username : String
+password : String
+userId : int
+database : SQLite
+courses : List<Course>
+Account(newUsername : String, newPassword : String, newUserID : int)
+CheckCreateAccount(userName : String, password : String) : Boolean
+Login(username : String, password : String) : Boolean
+addCourse(course : Course) : void
+removeCourse(courseName : string) : bool
has
Course
+name : string
+grade : double
+code : string
+priority : double
+credits : double
+Course()
+Course(builder : CourseBuilder)
+AddAssignment(name : string, totalPoints : int, weight : int) : void
+RemoveAssignment(assignment : string) : bool
+GradeAssignment(name : string, points : double) : void
+CalculateGrade() : double
+LogHours(day : int) : double
creates
CourseBuilder
+name : string
+grade : double
+code : string
+priority : double
+credits : double
+CourseBuilder()
+WithName(name : string) : CourseBuilder
+WithGrade(grade : double) : CourseBuilder
+WithCode(code : string) : CourseBuilder
+WithPriority(priority : double) : CourseBuilder
+WithCredits(credits : double) : CourseBuilder
+Build() : Course
Assignment
+totalPoints : int
+name : string
+weight : double
+grade : double
+Assignment(newPoints : int, newName : string, newWeight : int)
+addGrade(points : double) : void
StudyLog
+log : Dictionary<int, double>
+StudyLog()
+StudyLog(day : int, hours : double)
+AddTimeToLog(newTime : double, day : int) : void
+GetHours(day : int) : double