The Homework App
for UTM CSCI 352 Spring 2017

Fate Hardin and Wade Wakefield

Abstract

Every student needs to be organized to stay on top of things, and staying organized is not always that easy. The Homework App will make it easiet for all students to keep up with their assignments and stay organized.

1. Introduction

School can be very stressful to keep up with. There are so many assignments that are due on so many different days. Keeping a calendar notebook can be very frustrating; pages get torn out, the pages look messy, and if you write something down wrong it is annoying to fix. The Homework App will help students stay organized in a much easier way. Students will be able too add classes to their schedule so they can see their weekly class routine. Students will also be able to add assignments and its due date for each course, along with some short notes about the assignment. A calendar shows the student what days they have an assignment due.

This application will help students keep track of all of their assignments, so they can stay on top of things. Students will be less stressed knowing that all of their assignments can be seen with the click of a button, rather then flipping through the pages of a standard, physical planner.

1.1. Background

As college students ourselves, we do our best to stay organized. We have both tried hard to keep a physical student planner up to date with assignments, but that didn't work out very well. After searching through application stores for a planner, neither of us found an application that fit our needs. We realized that we could make an application that has all we need.

1.2. Challenges

One challenge was figuring out how to have the application interact with the date and time on the system. This effects the schedule that needs to be shown and the list of due dates. It was a challenge to work with the calendar on WPF. The calendar had to be resized, and it was difficult to find a way to show images on the days assignments are due.

2. Scope

We require that the students be able to add classes, assignments for each class, due dates, and see a weekly class schedule. Students will see a calendar displayed on the home screen to show due dates. Completed assignments and past due assignments are also shown on the home screen. A stretch goal would be to add, onto the home screen, a weekly schedule showing the assignments due that weeks. Creating a mobile version of the Homework App would be more helpful to students so they have can use it at any time. We would implement a profile and cloud based system at a much later date.

2.1. Requirements

The requirements are ones that we would want in a planner application. The user needs to be able to add classes to their schedule, giving the application the class name and the time it starts. We require that the user be able to see a schedule of the assignments they have, along with when they have their classes. Users need be able to mark off assignments when completed and edited if needed.

2.1.1. Functional.

- User needs to be able to add classes – classes have times, days, title, professor name and email.
- Users need to be able to add assignments for each class – assignments have due dates, titles, classes, notes.
- Users needs to be able to view a chronological list of assignments – this helps the user see the due dates of assignments in chronological order.

Use Case ID	Use Case Name	Primary Actor	Complexity	Priority
1	Add a class	Student	Hard	1
2	Add an assignment	Student	Hard	1
3	Edit classes	Student	Med	2
4	Edit Assignments	Student	Med	2
5	View Assignment Info	Student	Easy	3
6	Add Assignment via Calendar	Student	Hard	2

TABLE 1. USE CASES

- Users needs to be able to view calendar – dates with assignments due are circled, if there is a past due assignment a red "X" is shown on that date.
- User needs to be able to view completed assignments – completed assignments are shown on the home screen.
- User needs to be able to see past due assignments – past due assignments are displayed on the home screen.

2.1.2. Non-Functional.

- Mobile version of the application – for portability
- Cloud based system – a cloud system to sync up with an account allowing the app to be used on multiple platforms.
- Customization! – with more time, we could add the option to customize the background of the windows.
- Other Scheduled Items – other than assignments, users could add meetings or other appointments
- Assignment Percentage – user could set an assignment as a certain percent complete

2.2. Use Cases

Use Case Number: 1
Use Case Name: Add a class.
Description: A student will click "view classes" to see all of their classes, there they can type the info required for a class to be added and then click the "add class" button to add it completely.

1) User click the "view class" button.
2) User types info into text-boxes for classes.
3) The list displaying the user's classes is updated.

Termination Outcome: The user has added a class to their schedule.

Use Case Number: 2
Use Case Name: Add an assignment.
Description: Student will click "view assignments" to see their assignments. They can type and fill in any information needed for an assignment and can add it to their assignment list.

1) User click the "view assignments" button.
2) User types information for the assignments.
3) The list displaying the user's assignments is updated.

Termination Outcome: The user has added an assignment to their schedule.

Use Case Number : 4
Use Case Name: Edit Assignments
Description: When in the assignments window, the user can select a class to be edited, and then modify it.

1) User clicks "View Assignments" button.
2) User clicks on an assignment that is displayed.
3) User clicks "Edit" button.
4) User changes any of the details about the assignment they need to.
5) User clicks "Save Changes" button.

Termination Outcome: The chosen assignment has been edited.

Use Case Number : 5
Use Case Name: View Assignment Info
Description: User is able to view any details of any saved assignment.

1) User clicks "View Assignments" button.
2) User selects and assignment from the displayed list.

3) User clicks the “View Assignment Info” button.

Termination Outcome: A message box pops up showing all the information for the selected assignment.

Use Case Number : 6

Use Case Name: Add assignment via calendar.

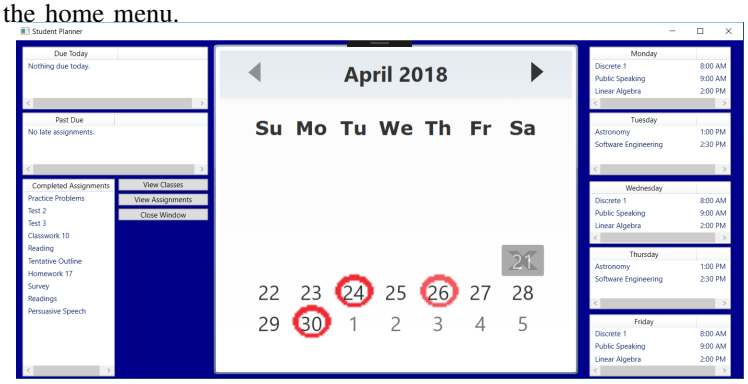
Description: User clicks on a day on the calendar to view assignments due that day, or is asked if they want to create one.

1) User clicks on a day on the calendar.
2) If there are no assignments set due for that day, the user is asked if they want to add one.
3) If the user selects yes, they are taken to the view assignments window.

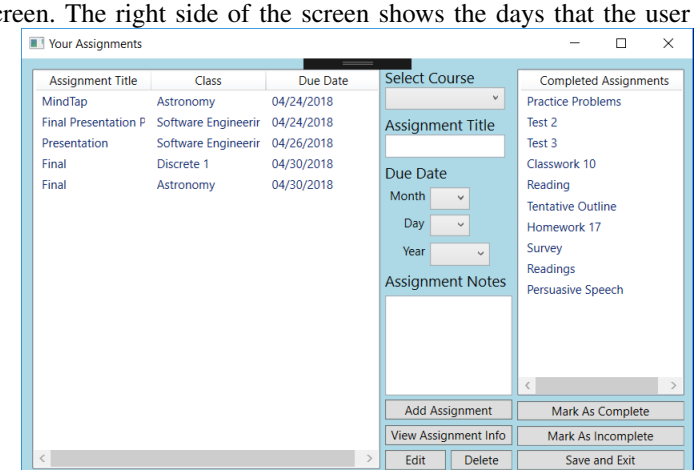
Termination Outcome: Assignments window is opened so the user can add an assignment to the desired date.

2.3. Interface Mockups

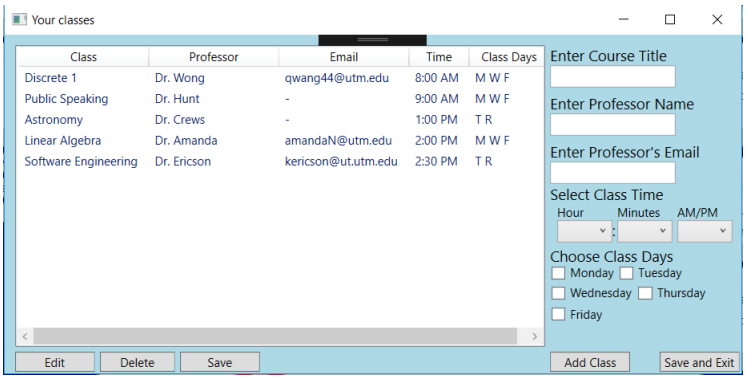
The first image is how the user adds a class. The second image is how the user will add assignments. The third image is the home menu.



This is the home screen of the application. The user is shown a calendar, where dates with due assignments will be circled. Completed assignments, past due assignments, and any assignments due today are displayed on the left side of the screen. The right side of the screen shows the days that the user has the classes they have added.

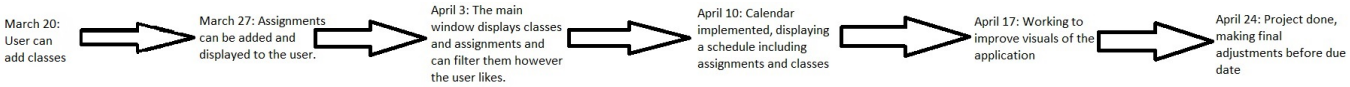


This is the window seen after the user clicks “View Assignments” on the home screen. Here the assignments are displayed in order of due date. The user can add the required information for an assignment. The user can also mark assignments as completed or incomplete in this window. Assignments can be edited and deleted.



This is the window seen after the user clicks "View Classes" on the home screen. All the information of a class is displayed to the user in the window. The user enters any required information and creates a class. Classes can be edited and deleted.

3. Project Timeline



4. Project Structure

We created an Assignment class and a Classinfo class. These classes formed objects used for everything in the application. The assignments and classes are shown in a ListView, allowing the user to be able too easily identify each part of the objects. A builder class builds the class and assignment objects when necessary. The CalenderBackground class is from Lars Pehrsson 1 , and it allowed us to circle the dates in the calnedar that assignments are due.

4.1. UML Outline

See Figure 1 and 2.

4.2. Design Patterns Used

For one of our patterns, we used the builder pattern. The builder class is an abstract class, and the info_builder class is the concrete implementation. The info_builder class builds the Classinfo object and the Assignment object. The Assignment and Classinfo objects are built when the user goes to either "View Assignments" and creates an assignment, or goes to "View Classes" and creates a class.

5. Results

Over the course of the project, we learned that it is very important to make the UI first. We had some designs at first, but did not plan everything out for it in much detail which caused us to redesign the UI a few times. Overall, we achieved everything we set out too. The way the user adds a class or assignment feels comfortable when using the application. The calendar works the way we had hoped. We found a class that was made by Lars Pehrsson, that helped with the calendar. At first, we were scared hearing about our semester project, but now we feel like we have more experience and are happy with the final product.

5.1. Future Work

If we continued to work on the project, we would want to add some sort of online cloud system. We would use a profile system to do that. However, we do not intend to continue work on the project. Fate plans to work on another WPF project as summer practice, and too increase his knowledge on the way WPF works. Wade plans to use what he has learned when working with C# to work on VR games. The future is going to be fantastic.

References

[1] Lars Pehrsson. The CalenderBackground class. (2013)[Online]. Available: https://www.codeproject.com/Tips/547627/Highlight-dates-on-a-WPF-Calendar

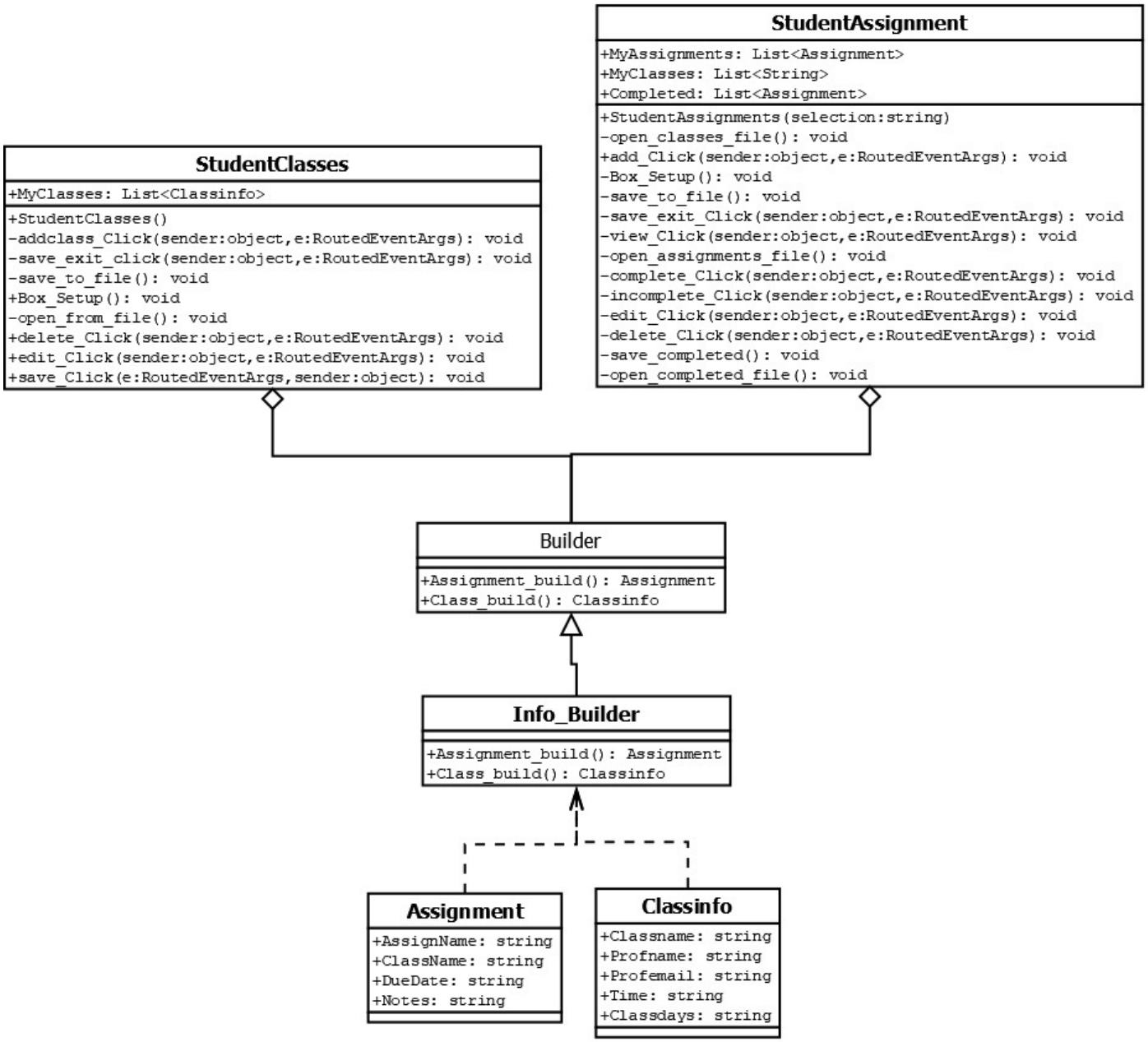


Figure 1. This diagram show the UML outline of the project.

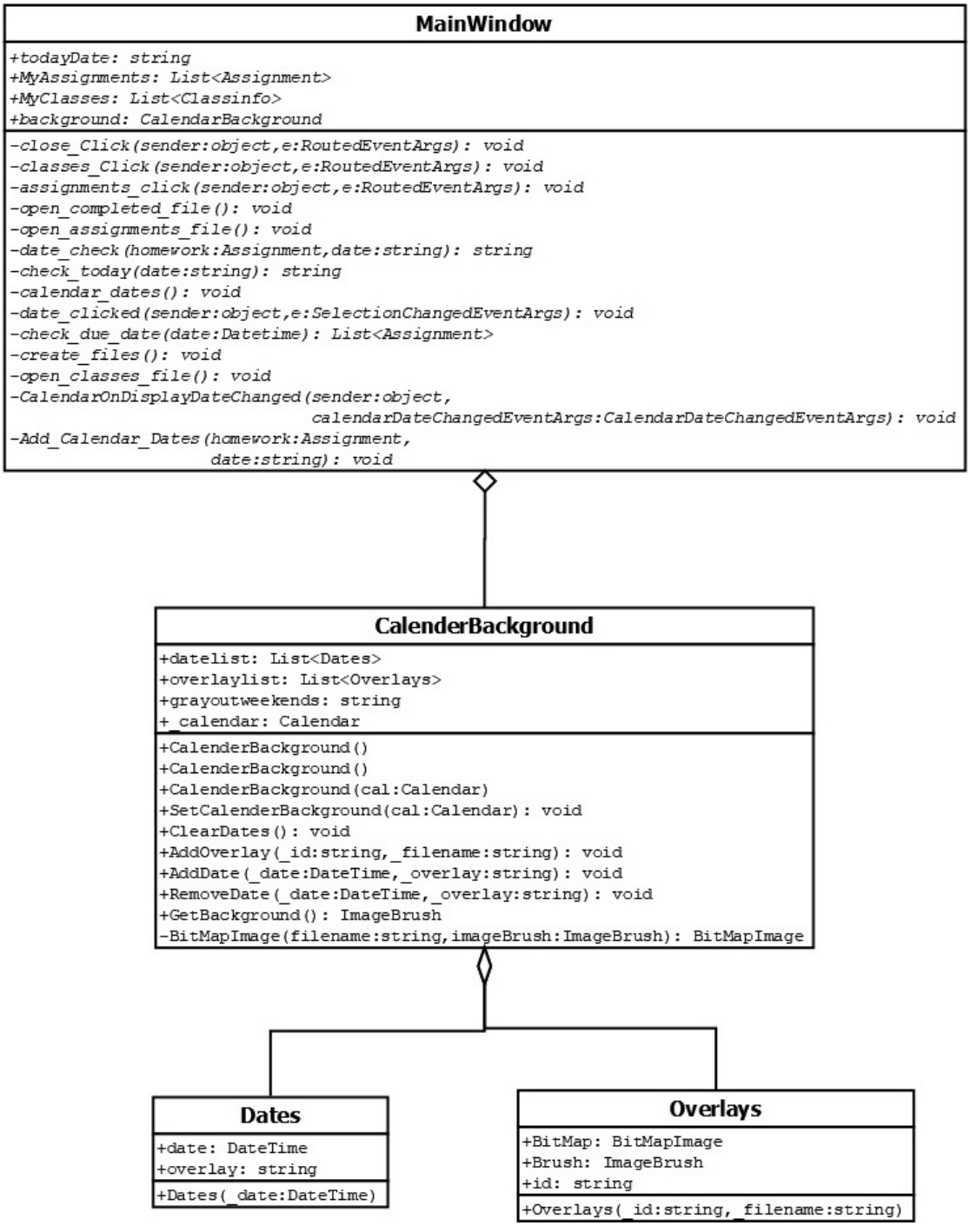


Figure 2. This diagram show the UML outline of the project.