Student Planner
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. With the student planner, it can be.

1. Introduction

College is hard, every student knows this. There are so many assignments 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. With the student planner, all of this is made simple. The student planner will help students stay organized in a much easier way. Students will be able too add classes to there schedule so they can see there weekly class routine. Students will also be able to add assignments and it's due date for each course, along with some short notes about the assignment.

This application will help students keep track of all of their college 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 a college student myself, I do my best to stay organized. I tried hard at first to keep my physical student planner up to date with my assignments, but that didn't work out very well. I found an application to use as my planner, but It does not have all the functionality I desired.

1.2. Challenges

The first difficulty of the project will come from the familiarity with VS 2017 and WPF applications. We have never worked with them before and will need to learn along the way. Another challenge will be figuring out how to have the application interact with the date and time on the system. That will effect the schedule that needs to be shown and the list of due dates. It also will be a challenge to work with the calendar on WPF and to get it to do what we want.

2. Scope

We require that the students be able too add classes, assignments for each class, due dates, and can see a weekly class schedule. We would feel accomplished after completing these tasks. 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 application would be more helpful to students. Implementing a profile, cloud based system would be done at a much later date.

2.1. Requirements

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 set times for each day.
- Users need to be able to add assignments for each class – assignments have due dates, and can be altered after they are created.
- Users needs to be able to view a list off assignments – this helps the user see the due dates off assignments in chronological order.
- Users needs to be able to see professor info – professor info can be added such as email, office hours, etc.

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

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.

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 info 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.

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 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.

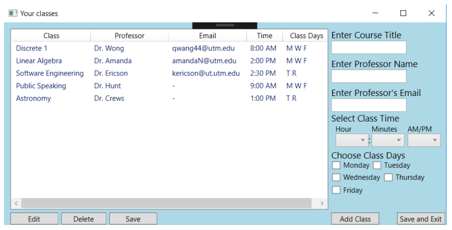
Use Case Number : 6
Use Case Name: Add assignment via calendar.
Description: User clicks on a day on the calendar to view assignments due that day, or asks if they want to create one.

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.

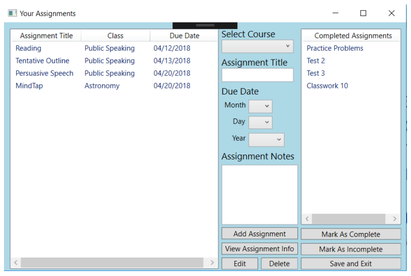
Termination Outcome: Assignments window is opened so the user can add an assignment to the desired date.

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.

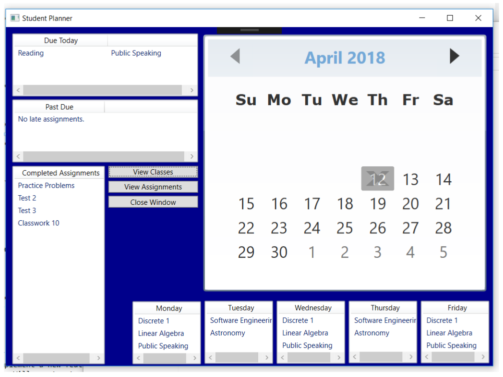

Your classes

Class	Professor	Email	Time	Class Days
Discrete 1	Dr. Wong	qwang44@utm.edu	8:00 AM	M W F
Linear Algebra	Dr. Amanda	amandaN@utm.edu	2:00 PM	M W F
Software Engineering	Dr. Ericson	kericson@ut.utm.edu	2:30 PM	T R
Public Speaking	Dr. Hunt	-	9:00 AM	M W F
Astronomy	Dr. Crews	-	1:00 PM	T R


Your Assignments

Assignment Title	Class	Due Date
Reading	Public Speaking	04/12/2018
Tentative Outline	Public Speaking	04/13/2018
Persuasive Speech	Public Speaking	04/20/2018
MindTap	Astronomy	04/20/2018

Completed Assignments
Practice Problems
Test 2
Test 3
Classwork 10


Student Planner

Due Today	
Reading	Public Speaking

Past Due
No late assignments.

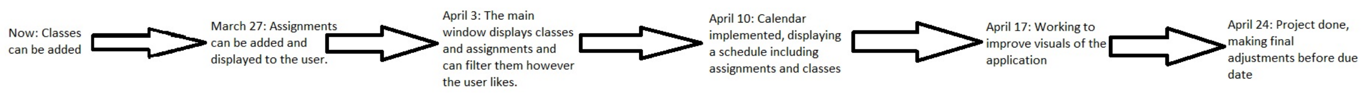
Completed Assignments
Practice Problems
Test 2
Test 3
Classwork 10

April 2018

Su	Mo	Tu	We	Th	Fr	Sa
				12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5

Monday	Tuesday	Wednesday	Thursday	Friday
Discrete 1	Software Engineerir	Discrete 1	Software Engineerir	Discrete 1
Linear Algebra	Astronomy	Linear Algebra	Astronomy	Linear Algebra
Public Speaking		Public Speaking		Public Speaking

3. Project Timeline


Now: Classes can be added

March 27: Assignments can be added and displayed to the user.

April 3: The main window displays classes and assignments and can filter them however the user likes.

April 10: Calendar implemented, displaying a schedule including assignments and classes

April 17: Working to improve visuals of the application

April 24: Project done, making final adjustments before due date

4. Project Structure

At first, this will be a little empty (it will need to be filled in by the time you turn in your final report). This is your chance to discuss all of your design decisions (consider this the README's big brother).

4.1. UML Outline

Show the full structure of your program. Make sure to keep on updating this section as your project evolves (you often start out with one plan, but end up modifying things as you move along). As a note, while Dia fails miserably at generating pdfs (probably my fault), I have had much success with png files. Make sure to wrap your images in a `figure` environment, and to reference with the `ref` command. For example, see Figure **??**.

4.2. Design Patterns Used

Make sure to actually use at least 2 design patterns from this class. This is not normally part of such documentation, but largely just specific to this class – I want to see you use the patterns!

5. Results

This section will start out a little vague, but it should grow as your project evolves. With each deliverable you hand in, give me a final summary of where your project stands. By the end, this should be a reflective section discussing how many of your original goals you managed to attain/how many desired use cases you implemented/how many extra features you added.

5.1. Future Work

Where are you going next with your project? For early deliverables, what are your next steps? (HINT: you will typically want to look back at your timeline and evaluate: did you meet your expected goals? Are you ahead of schedule? Did you decide to shift gears and implement a new feature?) By the end, what do you plan on doing with this project? Will you try to sell it? Set it on fire? Link to it on your resume and forget it exists?

References

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.