Project Plan Document

for

Extension Tracker

Prepared by

Anum Qureshi
Anna Malyevac
Chris Watt

2/19/2019

1. Introduction

1.1 Purpose

This document is designed to provide a comprehensive description of the system requirements for Extension Tracker, the online tracker for time-bank-day spending in classes taught by Dr. Zeitz. Contents of this document will be easily understandable by the client, Dr. Zeitz, as well as students enrolled in CPSC430 Software Engineering at the University of Mary Washington.

1.2 Scope

Extension Tracker will be an online system for tracking the usage of time-bank days by students in Dr. Zeitz's classes. Dr. Zeitz will be able to view time-bank-day usage for each student to ensure that she does not miss the usage of a time bank day. Students will be able to register for her classes and spend time bank days using a webpage as opposed to having to personally contact Dr. Zeitz.

1.3 References

Dr. Zeitz has asked us to reference her previous syllabi and ideas she used in the past for the extension tracker.

1.4 Overview of the remainder of the document

This document has been organized to give an overview of Extension Tracker, provide some background on the client, highlight requirements specified by the client, and detail the technical processes of the project's implementation, as well as the breakdown and scheduling for individual components of the project.

2. Project Description

2.1 System overview

Each semester at UMW, Dr. Zeitz teaches multiple computer science classes. In each of these classes, she provides each student with three “time-bank days.” A time-bank day acts as a 24-hour extension on the due date of a selected assignment. The purpose of providing students with time-bank days is to allow them to extend the due dates of assignments if they are crunched for time. This project will allow for both the Dr. Zeitz and her students to keep track of how many extension days they have used. The Extension Tracker will be a website hosted on the UMW CS server that UMW students from Dr. Zeitz’s classes will register and login at. Students will have access to a simple page that will show their remaining time-bank days and allow them to use one or more. Dr. Zeitz will have a more comprehensive view that will allow her to look at the used time-bank days and remaining time-bank days of her students. The Extension Tracker will make providing time-bank days to students easier for Dr. Zeitz and prevent discrepancies between her and her students.

2.2 Client characteristics

Dr. Zeitz is a computer science professor at the University of Mary Washington. During her past few years of teaching, Dr. Zeitz has found it difficult to keep track of time-bank days for every student. She reported that students do not always accurately remember their remaining time-bank days either. Furthermore, Dr. Zeitz would prefer to not receive any email or notification whenever a student wants to use a time-bank day, which is the current method. If a class of 30 students used all of their time-bank days for the semester one at a time, that would results in 90 emails in Dr. Zeitz’s inbox. The Extension Tracker will make it easy for her view the usage of her student’s time-bank days.

2.3 User characteristics

The users of the Extension Tracker will be Dr. Zeitz and her students. Dr. Zeitz will be the only admin-level user of the website, while her students will be regular, non-privileged users. Students will only have access to their own accounts. They will not be able to view other student’s account information or directly communicate with Dr. Zeitz. Also, given the nature of Extension Tracker, students may be in a rush to login and use a time-bank day, which should be taken into consideration during development. Finally, students are expected to have the required knowledge and skills to navigate the Extension Tracker website with minimal difficulty.

2.4 Functional requirements

Both Dr. Zeitz, the only privileged user, and her students, the non-privileged users, will be required to login to the Extension Tracker website each time they want to use it. Students will be able to do a few things once they are logged in: view their remaining time-bank days, use one or more time-bank days (if they have at least one time-bank day remaining), register for a class, and view the assignments for a class. Once students are done using the Extension Tracker, they can logout and close out of the website. For Dr. Zeitz, there will be more functionality available from the website. At the beginning of the semester, she will be able to create classes that students can register to. Assignments for those classes can be created and deleted. Upon logging in, Dr. Zeitz will have a dashboard showing all of the classes she is currently teaching. Once a class is selected, she will be able to view remaining time-bank data for that class by student and assignment. Other admin-only functionalities include giving students time-bank days and deleting student accounts. Once Dr. Zeitz is done using the Extension Tracker, she will be able to logout and close the website.

2.5 General constraints

Dr Zeitz has asked that the Extension Tracker be hosted on the CSUMW server, so that it may be streamlined within the University as well as free of cost to maintain. She has also requested that the front end be written in HTML, so that implementation correlates with knowledge learned in her Databases class, and easily updated.

3. Project Schedule

3.1 Approach

Constructing the code for the website will be broken up into two parts, frontend development and backend development. We will break up our team into one person dedicated to the backend scripts and functions, one person dedicated to frontend web development, and one person who will move back and forth to help integrate the two together and help where needed. As frontend development gets closer to finishing then the team will focus more on backend together.

The backend person will first work on developing an initial .sql file that will create test administrator account for logging in on the website and tables for student accounts and assignments. Following that a script will be made to parse a .csv into the database. The .csv will contain the student roster. The script will populate a table in the database with student accounts including, unique usernames, hashed passwords, and initial time bank days. After that a script will be created to parse a different .csv into a table in the database that will contain the assignments for the class and their due dates. Following that the backend person will work on a series of functions that allow the administrator account to update and change student accounts as needed from the website. Those functions will include the ability to drop assignments, change due dates, and add new assignments, the ability to drop all students from the database or individual students, add new students or upload a new .csv of students. Lastly a series of functions will be created for the student accounts. These functions will give the students the ability to use time bank days, change their passwords, and agree to the clients terms of service.

The frontend person will first be responsible for the initial website page design for administrator access and designing how an administrator would use key features such as viewing the student roster, being able to upload student .csv files, uploading an assignment .csv files. After that the frontend person will create the web page designs for the student access, how they view their unused time bank days, how they use the time bank days, and their account management pages.

The person that is neither the frontend or backend person will be responsible for helping either person when they need help with something that has a higher priority than what the other person is working on. They will also be responsible for helping integrate the scripts to the web pages and will provide advice on how to design the web pages in order to best use the scripts and functions that will be created.

3.2 Milestones and Deliverables

Import CSV option:

Students roster will be populated in the database by means of importing an CSV file. The file should be parsed into the SQL database and should be formatted to be incorporated into the front-end view.

Terms of Service:

Students will be presented with either a text box or html page to view the terms of service for the extension tracker, and have a widget to accept the terms.

Professor Account Overview/privileges:

The administrator, the professor, will have complete access to a project overview to be set up by her, where she may see how many students have used a time bank day for the following assignment. She will also have complete access to the student database, as well as the time bank days allotted and student account creation.

Student Account Overview/privileges:

Students will have a restricted view of the administrator’s view with restricted privileges. The students will be presented with their individual time bank count and each project that they can or have used time bank days on. They will be given the option to use time bank days for current projects, subtracting from the overall count.

Student Account management:

Students will have an option within the student account overview to manage and change their password, which will previously be pre-generated by a randomized system.

Student account deletion:

The administrator will have the privilege to delete student accounts at will, via an option integrated into the professor's overview.

3.3 Work Breakdown Structure

Professor Account Overview/privileges

First priority, estimated time 30-40 hours

The first task to be completed will be the administrative privileges and complete access to the designed system. This will be the bulk of the project, as it incorporates both front end and back end development, and as the administrator needs access to all key features of the website, it involves the completion of all key components. This will be our top priority to complete first, because all subsequent milestones will be a variation of or work from this step.

Import CSV

Second priority, estimated time 10-15 hours

The CSV import system is highly important to further aspects of the system, as it populates the bulk of the dataset that will be worked from. This step should be worked on simultaneously, along with the administrative privileges, so that when incorporated together a realistic depic tion of the system from the admin perspective is complete.

Student Account Overview/privileges

Third priority, estimated time 15-20 hours

The goal of the project is to be a tool for students to use, so the student-side of the front end should be prioritized. The student account overview will incorporate similar, but restricted access to the first priority, admin overview.

Student Account Deletion

Fourth priority, estimated time 2-4 hours

At this point, the core aspects of the system are completed, and the following tools incorporate the administrative overview and the database from CSV, so the estimated time of completion is significantly lower. The process of deleting an account boils down to dropping a table in the database, taking not much time.

Student Login Management

Fifth priority, estimated time 4-5 hours

This holds a similar priority to the the account deletion, however ranked slightly less because it's less of a core functionality than removing accounts. Its implementation is only slightly more complex, as it involves merely changing a value in the password section of the database.

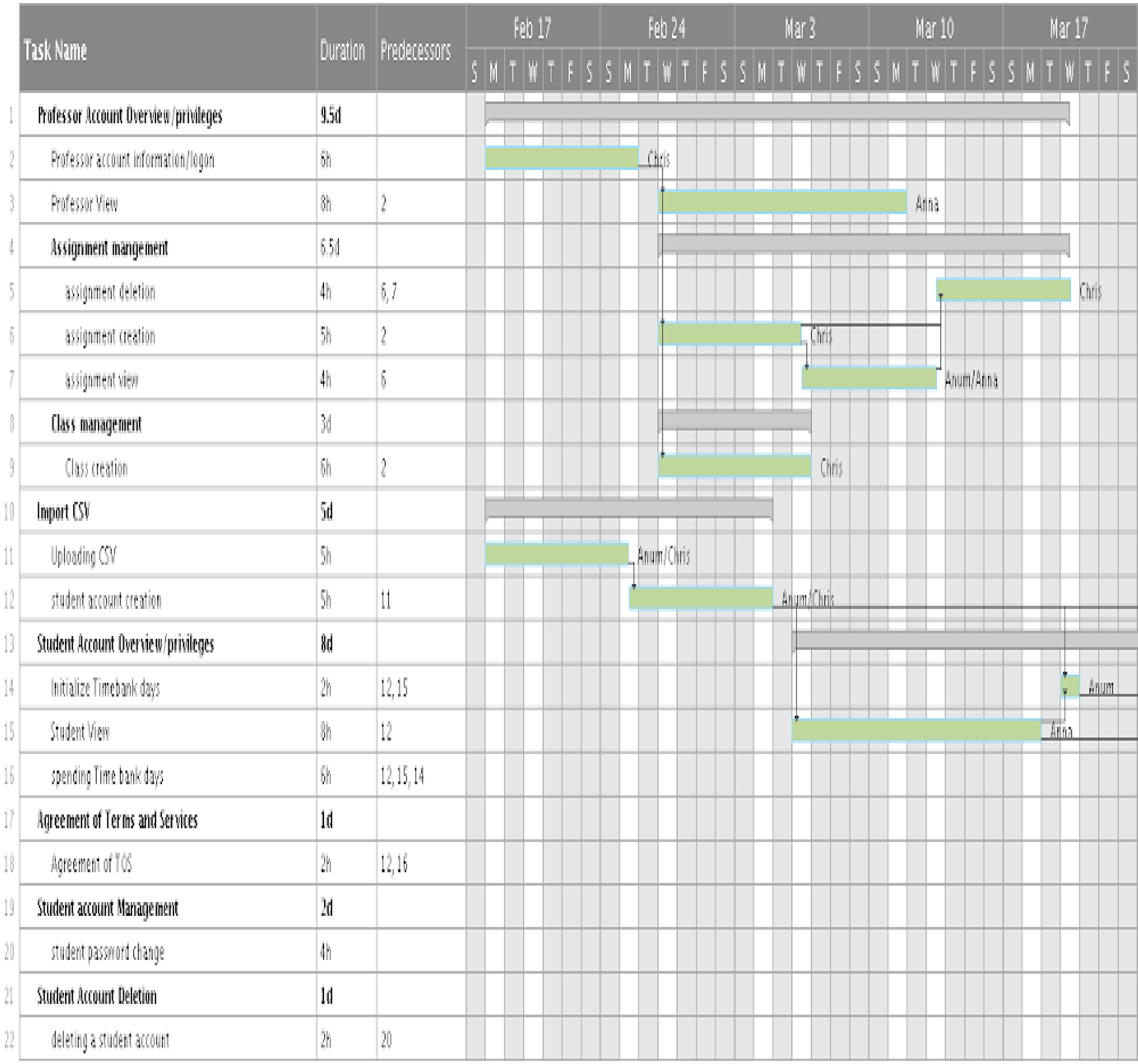
Terms of Service Agreement

Sixth priority, estimated time 1-2 hours

Lowest priority task and also the simplest, involves mostly text and doesn't incorporate previous aspects of the database, and is not code-heavy.

3.4 Gannt chart

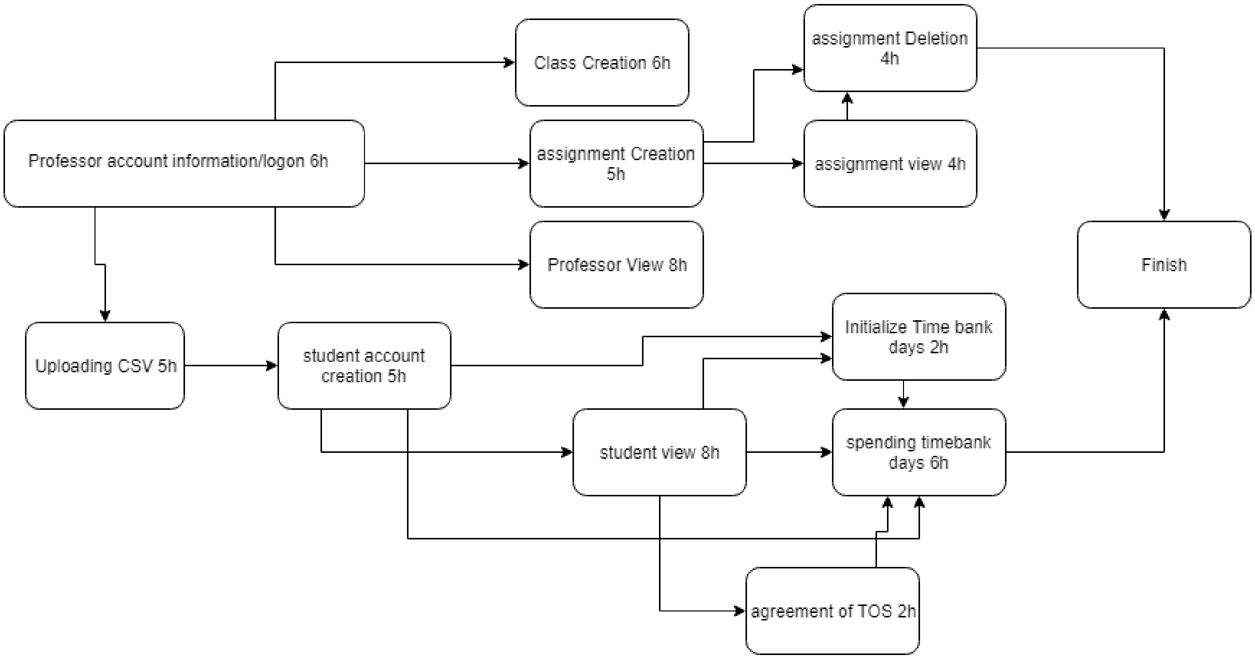
The Gannt chart above was made to keep track of dates and when tasks should be completed and milestones should be met. The chart was made with task durations and dependencies in mind, there are small black arrows going from one task to the one it is dependent on. Grey bars represent bigger over arching milestones, with their tasks directly under them in green. The work was broken down as such under the assumption that there will be work done on the project twice a week with a 4 week leeway for the main project. The tasks, Student account Management and Student Account Deletion are put close to project end, as to give a timeframe showing how early they would need to be started to properly implement, giving leeway to testing. As shown on the Gannt, each task is assigned to one member. One member will be focused on front end, the other on backend with one jumping in between as to keep the two connected and on task. Tasks may be traded between members as they see fit. The duration of each Task/Milestone were made



	Task Name	Duration	Predecessors
1	Professor Account Overview/privileges	9.5d	
2	Professor account information/logon	6h	
3	Professor View	8h	2
4	Assignment mangement	6.5d	
5	assignment deletion	4h	6, 7
6	assignment creation	5h	2
7	assignment view	4h	6
8	Class management	3d	
9	Class creation	6h	2
10	Import CSV	5d	
11	Uploading CSV	5h	
12	student account creation	5h	11
13	Student Account Overview/privileges	8d	
14	Initialize Timebank days	2h	12, 15
15	Student View	8h	12
16	spending Time bank days	6h	12, 15, 14
17	Agreement of Terms and Services	1d	
18	Agreement of TOS	2h	12, 16
19	Student account Management	2d	
20	student password change	4h	
21	Student Account Deletion	1d	
22	deleting a student account	2h	20

such that they fit the minimum of the work breakdown schedule with allowed leeway time.

3.5 Task Dependency Diagram



This diagram shows task dependencies. The first thing that needs to be started is the professor account information/logon. This will lead into the other tasks that are dependent on it. The professor account needs to have its information and be made before anything else because the professor will be the one handling all the subsequent information in the project. They will do everything from upload a csv to create classes and assignments. The professor account is the only way we will know if the professor view works. After the csv is uploaded we will have the students names and ids to create the student accounts. Once the student accounts are made we can ensure the student view is going to be made correctly and will let us keep track of what they can and cannot see or do. Once started with the view we can create the TOS so that students must agree to the TOS written by the professor before using the site. After the student view is started we can also look into initializing the timebank days, this is done automatically for each student and is set to three, the student view will help us ensure that we are correctly initializing it. The spending of timebank days is last in the student view because students should have timebank days to spend. Once the professor is in the system they can create assignments, we need to have assignments created and a professor view in progress to ensure the assignment view is correct. Only after an assignment is viewable can you delete it.the dependencies are in this order so that one thing can either be started or underway before the next is begun.

4. Appendix

This section includes all the items that don’t fit in the main sections of the document. You must include each of the following subsections.

4.1 Glossary of terms related to your project

4.1.1. Time-bank day: In case any stakeholder has not attended a class taught by Dr. Zeitz, she has 3 days given at the start of a semester to be used on any assignment (other than group projects) to add a day onto the due-date. These can be used until the last day of the semester and sometimes there are opportunities to get additional days.

4.1.2. MySQL: This is an open source relational database management system. Its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language.

4.1.3. PHP: Hypertext Preprocessor is a server-side scripting language designed for web development. Primarily implemented using C with some parts of C++ for the language.

4.1.4. Javascript: often abbreviated as JS, is a high-level, interpreted programming language that conforms to the ECMAScript specification. It is a language that is also characterized as dynamic, weakly typed, prototype-based and multi-paradigm.

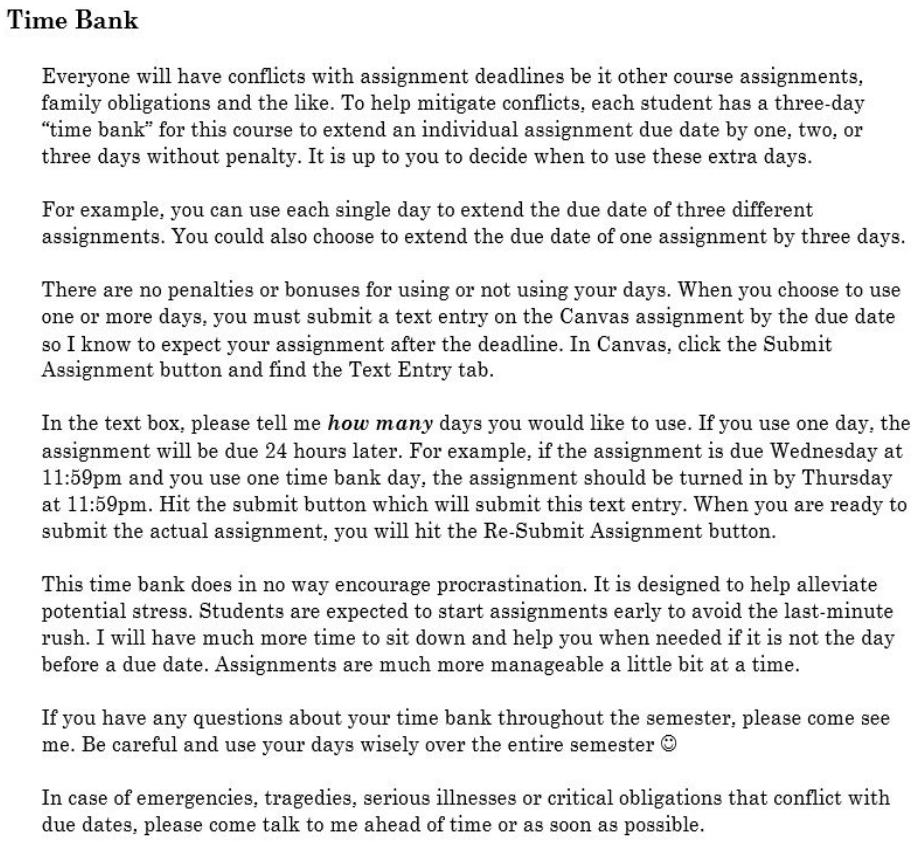
4.2 Author information

Anum - 3.4, 3.5

Anna - Title page, index, Section 1 (copy and pasted except 1.3 and 1.4), Section 2 (copy and pasted except 2.5) 4.1 (copy and pasted), 3.2, 3.3

Chris - Approach to development (3.1), 4.3

4.3 Additional documents



Time Bank

Everyone will have conflicts with assignment deadlines be it other course assignments, family obligations and the like. To help mitigate conflicts, each student has a three-day "time bank" for this course to extend an individual assignment due date by one, two, or three days without penalty. It is up to you to decide when to use these extra days.

For example, you can use each single day to extend the due date of three different assignments. You could also choose to extend the due date of one assignment by three days.

There are no penalties or bonuses for using or not using your days. When you choose to use one or more days, you must submit a text entry on the Canvas assignment by the due date so I know to expect your assignment after the deadline. In Canvas, click the Submit Assignment button and find the Text Entry tab.

In the text box, please tell me ***how many*** days you would like to use. If you use one day, the assignment will be due 24 hours later. For example, if the assignment is due Wednesday at 11:59pm and you use one time bank day, the assignment should be turned in by Thursday at 11:59pm. Hit the submit button which will submit this text entry. When you are ready to submit the actual assignment, you will hit the Re-Submit Assignment button.

This time bank does in no way encourage procrastination. It is designed to help alleviate potential stress. Students are expected to start assignments early to avoid the last-minute rush. I will have much more time to sit down and help you when needed if it is not the day before a due date. Assignments are much more manageable a little bit at a time.

If you have any questions about your time bank throughout the semester, please come see me. Be careful and use your days wisely over the entire semester ☺

In case of emergencies, tragedies, serious illnesses or critical obligations that conflict with due dates, please come talk to me ahead of time or as soon as possible.

Time bank section of 220 Syllabus provided by Dr. Zeitz