Test Plan Document

for

Extension Tracker

Prepared by
Anum Qureshi
Anna Malyevac
Chris Watt

3/20/2019

1. Introduction

1.1 Purpose

The purpose of this document is to outline the use and the purpose of the Timebank Extension Tracker for Dr. Zeitz, and to outline the objectives, approach, and scope for testing the software. This is so that the team assigned to test the Extension Tracker will have a detailed knowledge base of how the how the project is supposed to work, what we have implemented to accomplish the goals, and how it can be assured to be working properly.

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

Section 2, the project description, will overview all of the features and uses of the system. The system overview portion of this section will provide this information in a broad sense, while the client and user characteristics will narrow down the scope of the assignment. The functional requirements and general constraints should give a more technical overview of the features to be included in the assignment. Section 3, the test plan, will provide the testing team with detailed strategies to test the software, as well as consequences of potential barriers while testing, and a schedule provided for the testing plan.

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 CS UMW 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. Test Plan

This section describes the overall testing strategy and the project management issues that are required to properly execute effective tests.

3.1 Testing strategy

The testing strategy will be a systems test. As our website has a lot of back end scripts that need to run properly, so most of the testing will be to make sure that the scripts interact with the database correctly and that the database cannot be manipulated outside the scope of what the professor allows.

In order to start the testing, one of the testers should be given administrator capabilities, so that they can upload fake students and fake assignments to be sure that they work properly. The administrator also needs to check to make sure that they can add individual assignments as needed, view all assignments by class, and edit assignments properly. After the other testers have used time bank days in varying quantities, the admin should try to view how many have been used per assignments, and the remaining amount per class. The administrator should also check to be sure that they can change their password.

Following that, one of the other testers should try to test the security of the login by attempting an SQL injection on the login and any other capabilities that interact with the database through text input, such as the make individual assignments and the edit assignment features. One tester should focus on testing the student capabilities by logging in and testing to see if they can use their time bank days, and no more than the three. They should also test that they can view all their assignments.

After all student testing and administrator testing has been done, the last thing the administrator should test is that they can delete all students and assignments. They will be able to check that they have done this by looking at the view classes and view assignments, which afterwards should be blank pages.

3.2 Testing resources and staffing

On testing day we will add a fake administrator user and password for a tester to use on our website. All they will have to do is go to our website to login and begin. For the testing of the student accounts and the assignments we will provide sample .csv files that the fake administrator will use to populate the database for student logins and assignments.

3.3 Test work products

Due to the reliance on the UMW Computer Science server as requested by the client, any type of problem resulting from the unreliability of the server, or putting it over capacity, cannot be handled by the team and must be brought up with the server's administrator.

Usability testing will be noted and reported to Dr. Zeitz with the tester's suggestions. Ultimately, the decisions on the UI will be left up to the opinion of the client, with the difficulty of implementing the tester's suggestions being taken into account.

Stability testing will be dependent on the capacity reliably needed for a given feature. For example, if the website cannot handle having 10 databases of student classes, this is a non-issue, because Dr. Zeitz as requested to delete classes at the end of the semester, and she would never have 10 classes of students. However, if the system cannot support past a certain number of assignments, this is more of a problem because the number of assignments needs to be a large. If any of this is dependent entirely on the reliability of the CS server, then it is out of our hands.

The quickness and responsiveness of each feature will be tested to see if uniform. If one particular feature of page is particularly slow, we will analyze the code to see if there is a fault in our implementation, and if it can be refactored for efficiency. Speed dependent on the status of the UMW CS Server will also be taken into consideration.

Issues found with security features of our system will have the utmost priority in solving them, to our ability. The most important security feature of our website is the integrity of our databases, and that they can only be modified and accessed by the appropriate parties.

3.4 Test record keeping

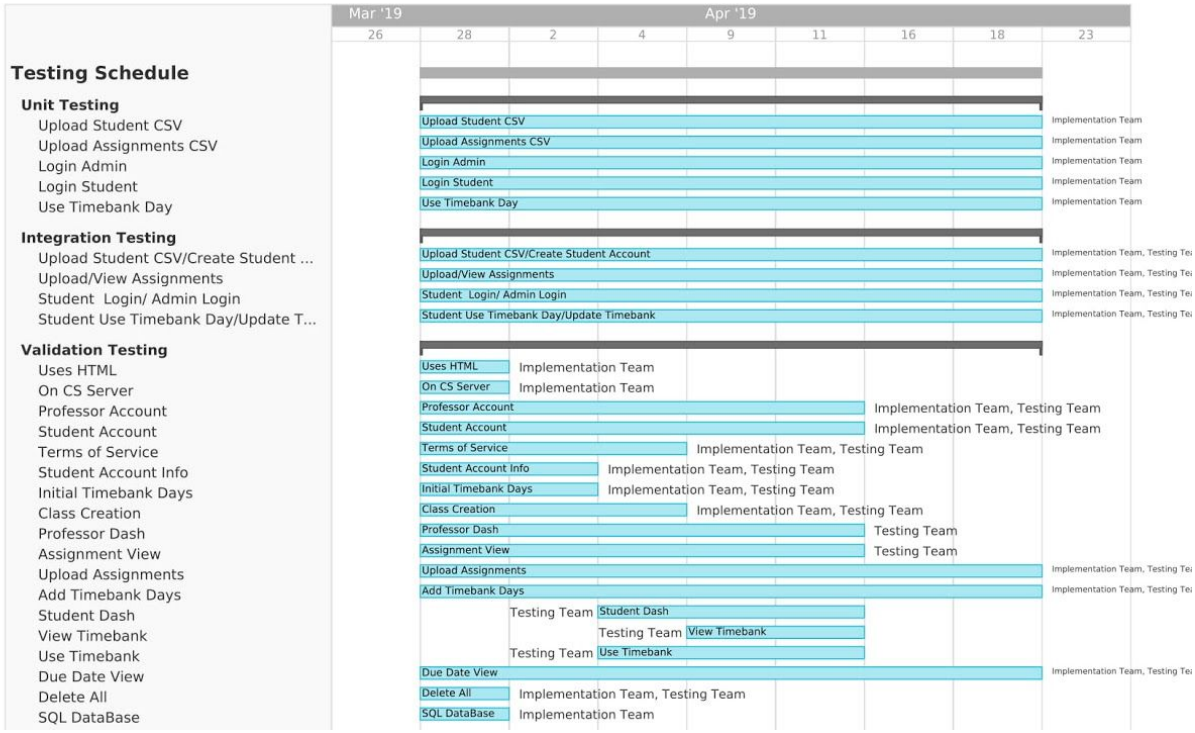
Any bugs or failures reported will include the following:

1. What file associated accessed did the bug occur on?
2. What subsection of the website did the bug occur on (admin view, student view, etc)?
3. What did you do to cause this bug? Please describe.

4. What happened as a result of the bug? Please describe.

The purpose of question one is to narrow down what function of our website caused the bug. Because the website is broken down over several dozen files, it is important to know which file specifically has coding errors. Question two's inclusion is due to the overlap between several features on the student view and the admin view; knowing specifically which user encountered the error will also help us narrow down the file and the cause of the bug. Questions three and four will help us pick out which part of the code is causing the error, after we have narrowed down the file and subsection, so we can see what feature is not working as intended, either by coding error or lack of inclusion of a feature or check.

3.5 Test schedule



Each Section is broken down into milestones and the validation testing is based on the requirements document. Most of the testing is done by the implementation team, the team has been testing the usability of the software as they progress through the project schedule. Unit Testing is done by the implementation team because they test the feature as they write the feature. Integration Testing is done by both teams because the testing team may have features or interactions they presume to happen that don't and that feedback is valuable for user bases. In validation testing, some of the features such as viewing are left only to the testing team because they will see it from the user side. The rest of the testing is done by both teams to ensure optimal testing.

4. Test Procedure

The test procedure details how each requirement and milestone should be tested and what kind of testing should be done for each.

Unit Tests:

Unit Tests should ensure that parts of the system work well individually.

1. Upload CSV:
    a. To test the upload CSV the tester must log into the admin account and go to the class navigation drop down and press the "Upload Class" button, to extensively test this feature different file types should be uploaded to ensure only CSV may be uploaded. The Tester should also try to upload improperly formatted CSV to ensure that the format is checked correctly. To validate that the test works the tester should move to the "View Classes" under navigation to see the class they uploaded.
2. Upload Assignments CSV:
    a. To test the upload assignments CSV the tester must log into the admin account and navigate to the assignment drop down. From here they can either press the "Upload Assignments" button or the "Make Assignment" button. Details for each type of upload will follow. To ensure that the tests are conducted properly, the Assignments table can be checked via the "View Assignments" button.
        i. "Upload Assignments": To test this feature extensively, testers should ensure only CSV may be uploaded. The Tester should also try to upload improperly formatted CSV to ensure that the format is checked correctly.
        ii. "Make Assignment": To test this feature, Testers should ensure the "initial due date" section only takes in date time formats, they should also be able to understand what the date time format is without having to guess.
3. Login Admin:
    a. To test the admin login, testers should try to log in to the admin portal with fake accounts, students accounts, and other tactics. Testers should also ensure admin login does not log into student accounts. To ensure this work the check of being logged in or not should work.
4. Login Student:
    a. To test the student login, testers should try to log in to the student portal with fake accounts, admin accounts, and other

tactics. Testers should also ensure student login does not log into admin accounts. To ensure this work the check of being logged in or not should work.

5. Use Timebank Day:
    a. To test the use of timebank days, testers should log into student accounts and then navigate to the timebank drop down and then "Use Timebank Day" button. Here the testers should ensure that they try to use more timebank days than they have and also attempt to use negative numbers. To ensure this worked or didn't the tester should look to see how many timebank days are left.

Test #	Requirement	Subsystem	Purpose	Test Case Data	Expected Results
1	Upload CSV	Uploads	To test if class creation is correct	F19_240.csv OR S19_430.csv	A class with students is made
2	Upload CSV	Uploads	To test if you can upload other files	file.txt	An error comes up that says not a CSV
3	Upload Assignments CSV	Uploads	To test if Assignment creation is correct	F19_240_Work.csv OR S19_430_Work.csv	An assignments list is made
4	Upload Assignments CSV	Uploads	To test if you can upload other files	file.txt	An error comes up that says not a CSV
5	Upload Assignments CSV	Creation	To test if an Assignment can be created	Assignment name : hello assignment Initial due date: 10/5/2018	Assignment is created

6	Upload Assignments CSV	creation	To test if an assignment with an improper due date can be uploaded	Assignment name : hello assignment Initial due date: 10-5-2018	Assignment is not created
7	Login Admin	Login	To test if admin can log in	Username: admin Password: admin	Logged in
8	Login Admin	Login	To test if fake admin can log in	Username: notadmin Password: notadmin	Not logged in
9	Login Admin	Login	To test if admin can login in student login	Username: admin Password: admin	Not logged in
10	Login Student	Login	To test if student can log in	User uploaded Student CSV and pick one	Logged in
11	Login Student	Login	To test if fake student can log in	User uploaded student csv pick one not in	Not logged in
12	Login Student	Login	To test if student can login in admin login	User uploaded Student CSV and pick one	Not logged in

				see if can log in admin	
13	Use Timebank	timebank	To test if a timebank can be used	1 day used	Days should go down and be used
14	Use Timebank	timebank	To test if a timebank can be used if there aren't enough days	6 day used	Days should not go down and not be used
15	Use Timebank	timebank	To test if a timebank can be used	-1 day used	Nothing should happen but an error

Integration Tests:

Should test how groups of things interact

1. Upload CSV/Create Student Account:
    a. To test the upload CSV the tester must log into the admin account and go to the class navigation drop down and press the "Upload Class" button, to extensively test this feature different file types should be uploaded to ensure only CSV may be uploaded. The Tester should also try to upload improperly formatted CSV to ensure that the format is checked correctly. To validate that the test work the tester should move to the "View Classes" under navigation to see the class they uploaded.Then the tester should try to go log in to one of the created student accounts to ensure the students were made.
2. Upload Assignments/View Assignments:
    a. To test the upload assignments CSV the tester must log into the admin account and navigate to the assignment drop down. From here they can either press the "Upload Assignments" button or the "Make Assignment" button. Details for each type of upload will follow. To ensure that the tests are conducted properly, the Assignments table can be checked via the "View Assignments" button. They should ensure the assignment is created and then check if a student account can interact with it.

i. "Upload Assignments": To test this feature extensively, testers should ensure only CSV may be uploaded. The Tester should also try to upload improperly formatted CSV to ensure that the format is checked correctly.
ii. "Make Assignment": To test this feature, Testers should ensure the "initial due date" section only takes in date time formats, they should also be able to understand what the date time format is without having to guess.

3. Student Login/ Admin Login:
    a. To test the student login, testers should try to log in to the student portal with fake accounts, admin accounts, and other tactics. Testers should also ensure student login does not log into admin accounts. To ensure this work the check of being logged in or not should work. Then the tester should test the admin login, testers should try to log in to the admin portal with fake accounts, students accounts, and other tactics. Testers should also ensure admin login does not log into student accounts. To ensure this work the check of being logged in or not should work. The tester should ensure they interact how they are supposed to, and that the admin can check if a student used a timebank day.
4. Use Timebank Day/Update Number days/assignment due date:
    a. To test the use of timebank days, testers should log into student accounts and then navigate to the timebank drop down and then "Use Timebank Day" button. Here the testers should ensure that they try to use more timebank days than they have and also attempt to use negative numbers. To ensure this worked or didn't the tester should look to see how many timebank days are left. The tester then needs to go check the assignment due dates in the view assignments tab.

Test #	Requirement	Subsystem	Purpose	Test Case Data	Expected Results
1	Upload CSV	Uploads	To test if class creation is correct	F19_240.csv OR S19_430.csv	A class with students is made in the view classes area
2	Upload CSV	Uploads	To test if you can upload other files	file.txt	students/class should not be

					made
3	Upload Assignments CSV	Uploads	To test if Assignment creation is correct	F19_240_Work.csv OR S19_430_Work.csv	An assignments list is made
4	Upload Assignments CSV	Uploads	To test if you can upload other files	file.txt	No new assignments in assignments view
5	Upload Assignments CSV	Creation	To test if an Assignment can be created	Assignment name : hello assignment Initial due date: 10/5/2018	Assignment is created in assignment view
6	Upload Assignments CSV	creation	To test if an assignment with an improper due date can be uploaded	Assignment name : hello assignment Initial due date: 10-5-2018	No new assignments in assignments view
7	Login Admin	Login	To test if admin can log in	Username: admin Password: admin	Logged in and cannot navigate to student dash pages
8	Login Admin	Login	To test if fake admin can log in	Username: notadmin Password: notadmin	Not logged in
9	Login Admin	Login	To test if admin can login in student login	Username: admin Password: admin	Not logged in

10	Login Student	Login	To test if student can log in	User uploaded Student CSV and pick one	Logged in and cannot navigate to admin account dash
11	Login Student	Login	To test if fake student can log in	Pick something not in an uploaded Student CSV	Not logged in
12	Login Student	Login	To test if student can login in admin login	Try the student in the admin username and password location	Not logged in
13	Use Timebank	timebank	To test if a timebank can be used	1 day used	Days should go down and be used assignment updated in assignment view for student
14	Use Timebank	timebank	To test if a timebank can be used if there aren't enough days	6 day used	Days should not go down and not be used nothing changed in assignment view for students
15	Use	timebank	To test if a	-1 day used	Nothing

	Timebank		timebank can be used		should happen but an error

Validation Tests:

Should test if requirements are met correctly.

1. Uses HTML:
    a. This is tested by the implementation team by checking code to ensure there is HTML
2. On CS Server:
    a. Implementation Team checks there the code is hosted.
3. Professor Account:
    a. The professor account is like the admin account. To test the admin login, testers should try to log in to the admin portal with fake accounts, students accounts, and other tactics. Testers should also ensure admin login does not log into student accounts. To ensure this work the check of being logged in or not should work. All of the other features like upload CSV should also be checked to ensure there is a professor account.
4. Student Account:
    a. The student account needs to log in first. To test the student login, testers should try to log in to the student portal with fake accounts, admin accounts, and other tactics. Testers should also ensure student login does not log into admin accounts. To ensure this work the check of being logged in or not should work. Other features like view assignments, and use timebank day should also be tested for this.
5. Terms of Service:
    a. When a student account is created a TOS agreement should pop up before they can start using the timebank day tracker. To test this feature the tester should create a student account and then log in with that student account.
6. Student Account Info:
    a. Student account information should be accessible from the professor account. This can be seen in the Analysis tab. The tester should navigate to this tab and attempt to see different student information. The student password is not visible to admins.
7. Initial Timebank Days:

    a. When a student account is created it should default to three timebank days. This can be done the same way the upload CSV test is done, and then checked via the class view tab.
8. Class Creation:
    a. To test the upload CSV the tester must log into the admin account and go to the class navigation drop down and press the "Upload Class" button, to extensively test this feature different file types should be uploaded to ensure only CSV may be uploaded. The Tester should also try to upload improperly formatted CSV to ensure that the format is checked correctly. To validate that the test work the tester should move to the "View Classes" under navigation to see the class they uploaded.Then the tester should try to go log in to one of the created student accounts to ensure the students were made.
9. Professor Dash:
    a. To test this feature, the tester should log into a admin account and then be redirected to the admin dashboard.
10. Assignment View:
    a. To test this, testers should first log into admin accounts and ensure an assignment has been made, then testers should go and see if they can see the assignments in the professor view assignments tab. Testers should then log into a student account and attempt to see the assignments tab and ensure they can only see their classes assignment.
11. Upload Assignments:
    a. To test the upload assignments the tester must log into the admin account and navigate to the assignment drop down. From here they can either press the "Upload Assignments" button or the "Make Assignment" button. Details for each type of upload will follow. To ensure that the tests are conducted properly, the Assignments table can be checked via the "View Assignments" button. They should ensure the assignment is created and then check if a student account can interact with it.
        i. "Upload Assignments": To test this feature extensively, testers should ensure only CSV may be uploaded. The Tester should also try to upload improperly formatted CSV to ensure that the format is checked correctly.
        ii. "Make Assignment": To test this feature, Testers should ensure the "initial due date" section only takes in date time formats, they should also be able to understand what the date time format is without having to guess.
12. Add Timebank Days:
    a. To test the add timebank days feature, testers must log into admin accounts and navigate to the give timebank day tab(if there is one).

They should input which student gets a day for which class and then press submit.

13. Student Dash:
    a. To test this feature the tester should log into a student account and ensure they can see the student navigation and features, and then ensure they cannot navigate to professor dashboard items. The tester should also ensure that the professor account cannot navigate to the student dash.
14. View Timebank:
    a. To test this feature the tester should log into a student account and ensure they can see the student navigation and features, the number of timebank days a student has left should appear on the student dash homepage.
15. Use Timebank:
    a. To test the use of timebank days, testers should log into student accounts and then navigate to the timebank drop down and then "Use Timebank Day" button. Here the testers should ensure that they try to use more timebank days than they have and also attempt to use negative numbers. To ensure this worked or didn't the tester should look to see how many timebank days are left.
16. Due Date View:
    a. To ensure this feature works, testers should log into admin accounts and navigate to the view assignments tab, here they can ensure that the due date is viewable to the professor in the assignments list. To test this feature in student view, testers should log into student accounts and then go into the student assignment view tab and see if they can see the due dates for assignments on there.
17. Delete All:
    a. This feature should be tested alongside the upload csv feature, this should only be accessible via the admin account, once there navigate to the "delete all classes" button via the navigation bar under classes. They should then input an admin username and password to delete the classes.

Test #	Requirement	Subsystem	Purpose	Test Case Data	Expected Results
1	Uses HTML	Code	To ensure the code is HTML based	none	The code is in HTML/PHP
2	Hosted on	Code	To ensure website	none	Website is

	CS server		is on CS server		on CS server
3	Professor account/professor dash	Login	To test if admin can log in	Username: admin Password: admin	Logged in and see the professor dash and cannot navigate to student dash pages
4	Professor account/professor dash	Login	To test if fake admin can log in	Username: notadmin Password: notadmin	Not logged in
5	Professor account/professor dash	Login	To test if admin can login in student login	Username: admin Password: admin	Not logged in
6	Student account/ student dash	Login	To test if student can log in	User uploaded Student CSV and pick one	Logged in and can see student dash and cannot navigate to admin account dash
7	Student account/ student dash	Login	To test if fake student can log in	User uploaded Student CSV and pick one not in the CSV	Not logged in
8	Student	Login	To test if student	User	Not logged

	account/ student dash		can login in admin login	uploaded Student CSV and pick one and to log into admin	in
9	Terms of Service	Login	To test if a first time user sees the Terms of Service and has to agree to them	agree	Logged in and can see student dash and cannot navigate to admin account dash
10	Terms of Service	Login	To test if a first time user sees the Terms of Service and has to agree to them	disagree	Not logged in
11	Student Account Info	admin	To see if the admin can see the student account info	none	Admin can navigate to and see student info in analysis and in class view. Admin cannot see passwords
12	Student Account Info	admin	To see if the student can see the student account info	none	Student cannot access any files but their own
13	Initial Timebank Days/	admin	To see if timebank days are created with the student	F19_240.csv OR S19_430.c	Class should be created and

	Class creation		account creation	sv	students should have 3 timebank days
14	Assignment View/ Upload Assignments	admin	To see if assignments can be viewed and uploaded	Assignment name : hello assignment Initial due date: 10/5/2018 OR F19_240_Work.csv OR S19_430_Work.csv	Assignments should be viewable and created
15	Add timebank days/ view timebank days	timebank	To see if timebank days can be viewed and can be added to only by admin	1 day added	Timebank days are viewable and increased
16	Use Timebank	timebank	To test if a timebank can be used	1 day used	Days should go down and be used assignment updated in assignment view for student
17	Use Timebank	timebank	To test if a timebank can be used if there aren't enough days	6 day used	Days should not go down and not be used

					nothing changed in assignment view for students
18	Use Timebank	timebank	To test if a timebank can be used	-1 day used	Nothing should happen but an error
19	Due date view	UI	To see if you can view the due date in the assignments tab	none	Should be viewable for both professors and students
20	Delete all	admin	admin can delete all	Username: admin Password: admin	All classes and assignments deleted
21	Delete all	admin	Student cannot delete all	Choose a valid student from the student CSV that has been uploaded	Page cannot be accessed.

5. Appendices

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.

5.1 Glossary of terms related to your project

5.1.1. 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.

5.1.2. 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.

5.1.3. 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.

5.1.4. 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.

5.2 Author information

Anna - 1.1, 1.4, 3.3, 3.4
Chris - 3.1, 3.2
Anum - 3.5, 4
Remainder: copy and pasted from past documents

5.3 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