Project Phase 1 Deliverables: Fall 2018

Revised: 9/3/2018

As you have probably discerned from other sources, the first deliverable deals with the Analysis and Specification portions of the project. In principle, this means that you need to submit the following things: 'team###_p1_ifd.pdf', 'team###_p1_eer.pdf', and 'team###_p1_report.pdf'

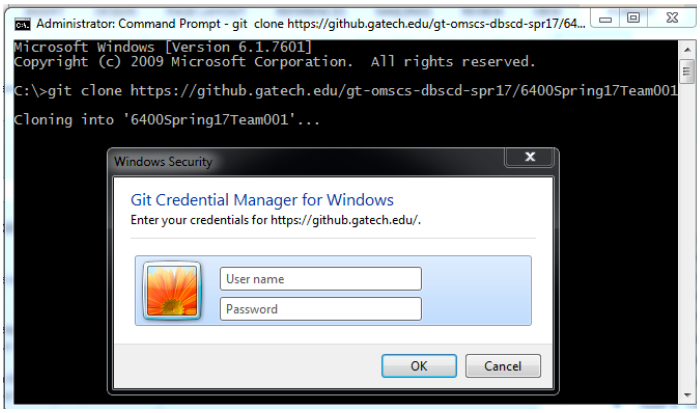
- Information Flow Diagram (IFD) **(10%)**
- Enhanced Entity-Relationship (EER) Diagram **(40%)**
- Data Types **(5%)**
- Business Logic Constraints (logic which cannot be reflected in the EER model) **(5%)**
- Task Decomposition (TD): rules of thumb/oval diagrams- both single and decomposed **(10%)**
- Abstract Code (AC) pseudo-code: input validation/error handling, how to navigate between tasks **(30%)**

Essentially, anything Leo does in the lectures on Analysis and Design for the GTOnline project you should do for your analysis and design of this project. There is also a sample submission available on Canvas that will help guide you in creating your submission. The most weight for project grading is on the EER and AC, so make sure to put as much detail and effort into those items as possible.

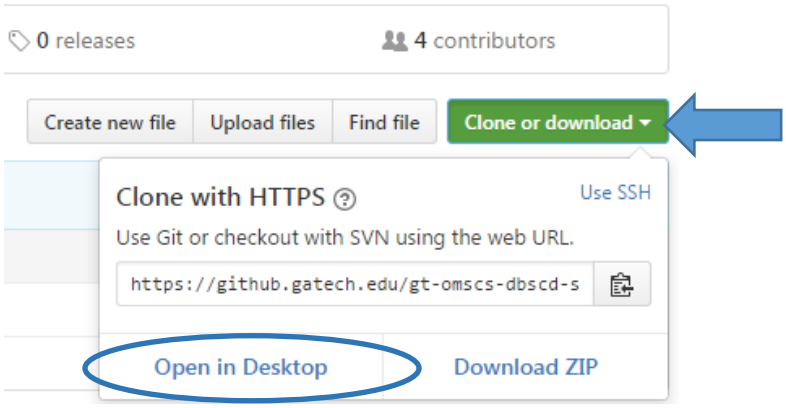
GitHub Enterprise Account:

All teams must use their official assigned GitHub Enterprise account for team projects. (Note for OMSA students: your team numbers are only two digits, remember to use your two-digit team number.)
Login using your GaTech credentials to: https://github.gatech.edu/login
Your repository has been created for you with all your team members as collaborators.
Find your teams repository (for this example, we'll assume you are team 1, replace 001 with your team number):
URL: https://github.gatech.edu/gt-omscs-dbscd-fall18/6400Fall18Team001
Now you need to clone this empty repo onto your workstation using either command line or GitHub Desktop:

```
git clone https://github.gatech.edu/gt-omscs-dbscd-fall18/6400Fall18Team001
```



Login with GaTech LDAP credentials



https://desktop.github.com/

Phase 1 Deliverables:

In your team's GitHub repo, create a folder named "Phase 1" with three PDF files:

- team#_p1_ifd.pdf
- team#_p1_eer.pdf
- team#_p1_report.pdf

(Replace the # with your team number, so if you are Team001, files would be "*team001_p1_ifd.pdf*"…)
The report file should contain your data formatting, constraints and task composition with abstract code.
There may be a grade deduction for any files not named correctly or if we cannot clearly determine which work should be graded in your repository because you did not put them in a "Phase 1" folder.

IMPORTANT NOTE: ***NO CREDIT*** for that portion of the submission **(-40%** p1 grade) if your EER is auto-generated by any tool such as MySQL workbench, uses crows-feet notation, or deviates from the notation Leo uses in the lectures or the notation used in the textbook. Tools available to create your own EER include: draw.io, LucidCharts, among others.

How You Should Work:

Now, as to how you should work: The point of this course is to learn all phases of relational database development. As a result, we strongly prefer that teams work collaboratively, not dividedly. By which we mean you should not say "OK, I'll take the EER diagram, you go write the IFD, John can write up the Task Decomposition and Jane can take the formatting and constraints." Rather, **each member should each work on all parts of the project**, all reviewing, all having input, all fully understanding each part of the design. That way all students learn all aspects of the project to help solidify the skills needed to be successful on the exams.

1. Create a directory called "Phase 1" in the team GitHub repository we created for your team (do this by committing any file in the directory called "Phase 1" to GitHub).
2. As you work on phase 1, continue to commit and push the files to GT's GitHub enterprise server.
3. Please review the sample submission that is posted in Files on Canvas. Your deliverables should be similar to those in the sample submission.

How To Submit

1. If you use branching, make sure to move all work onto your master branch in GitHub. If you provide a commit that is not from the master branch, there may be a penalty on your grade.
2. Find the commit ID of the final commit (you are encouraged to do all of your file sharing through GitHub) using any git tool you choose (a simple "git log" will do it). The commit ID is a SHA hash such as: "40da96c63a0a82d71e531cf137b6c95e282f9289"
3. One team member should submit your team number, followed by a comma, and then the commit ID in the submission text for the Project Phase 1 assignment in Canvas.
   So, if I was on Team001 and we were submitting commit *0b0134604ced2182b8b11d7b299ac59dc70ed722*, someone would submit this:
   "*001, 0b0134604ced2182b8b11d7b299ac59dc70ed722*"
   (Please submit a full commit ID. Shortened commit ID submissions may receive a penalty. OMSA teams, please remember that your team number is only **two** digits when submitting a commit.)

Note: any team member will be able to submit the commit ID to be graded. Please make sure every team member agrees on the correct commit ID before submitting it for grading. New work committed after the deadline **will not be graded**. If a team member submits a new commit ID, and other team members request a different commit to be graded, the entire team will face a deduction and we will randomly select one of the submitted commit IDs to grade. If no commit ID is submitted, there will be a penalty on your grade for failing to submit on time.

Regrade Requests: For any project regrade requests, please offer specific evidence as to why points should be returned. "Because the grade is lower than we expected" is not sufficient reason for honoring a grade change. Be sure to include "*regrade request*" in the title of your Piazza post and itemized list, so the instructional team do not confuse your regrade request with a question seeking clarification only. **Regrade requests will usually only be accepted for 1 week after feedback is returned on Canvas**. If teams seek to return points later in the semester they will not be considered as the instructional team need to move forward with the next phase of grading.

Preparing for Future Phases

In this phase, you should not be concerned with implementation details such as what technology stack you want to build your project on. We DO want you to plan ahead so that your team can be prepared for the last phase, so we are providing these details now NOT so that you choose a stack while working on this phase, but in order to make an informed decision when the time comes to make that choice.

We suggest using *AMP or *APP, implementing in MySQL or PostgreSQL and PHP v7, but you are by no means limited to those. You are welcome to implement your system in any language or platform you wish (Python, Java, Ruby, .Net) subject to the caveat that you must **write your own SQL** and implement your own logic on the database. Use of ORM or ORM-like tools such as hibernate and others which handle your database interaction (and even your schema design) are strictly prohibited. **Note**: **SQLite or MS Access is not allowed**. A standalone, full featured *relational* DBMS is required for the project: teams are free to use PostgreSQL, MySQL, MS SQL Server, or even Oracle if desired. (Non-relational noSQL: Hadoop, Cassandra, MongoDB, etc. are not allowed. "Built-in" or application-hosted databases should not be used. Non-relational databases that utilize SQL as an interface are also not allowed.)

As always, if you have questions, please post on Piazza to your "Team+Instructors" group.