CSC 648/848 Software Engineering – Spring 2018
Milestone 0

01/29/18

Anthony John Souza, Class CTO, SFSU
***ajsouza@mail.sfsu.edu* and slack: @ajsouza25.**

Dragutin Petkovic, Class CEO, SFSU
Petkovic@sfsu.edu

1.Introduction

The purpose of this individual milestone you will be working on with your team is to help you set up and learn how to use the development infrastructure that your team will be using this semester to develop its final presentation website. This infrastructure consists of: a) *deployment server and b) tools/stack* for delivering the final running application. The deployment server needs to be running on a remote host. ***Localhost hosting will not be accepted.***

This environment and setup is close to what you will be required to know to be successful SW engineer, so M0 is a great experience to improve your marketable skills. While some of the work is to be done *individually*, you must work together with your team to complete the task. Note that your learning will comprise: a) individual usage of tools; and b) most important: how to use the tools in team setting (e.g. code merge etc.).

In this class we will have **class CTO** who will deal with all technical issues (Anthony) and **class CEO** who deals with schedules etc. (Prof. Petkovic).,

M0 carries 10 grade points. Information about grading can be found at the end of the document. Again, each team member must complete his/her M0 tasks but to accomplish team each team member must also work with the rest of the team. **This means that you can help each other, ask for help, work in pairs and are in fact encouraged to do this to not only help with M0 but also build the teamwork.**

This semester M0 and class IT involves brand-new infrastructure designed to give teams more **freedom but also responsibility in setup**. Hence, there may be problems. Please contact class CTO Anthony ajsouza@mail.sfsu.edu if you encounter any problems that you cannot solve yourself or with your team.

Your first task is to read through the entire document. Then, follow the steps starting with Section 2 "n Setting up GitHub" . You will also give access to instructors, so they can monitor your work.

In section 3 we then describe actual M0 task where you are first to select deployment server and SW stack, then install them, and finally create a joint team web page. This WWW page can be used in your final project too (e.g. ABOUT page on your final team project WWW site) . Section 4 describes M0 grading.

The work on M0 shall commence only after the teams have been formed and basic accounts set up – you will get e-mail on this from Class CTO Anthony

Again, M0 is a great vehicle to help you learn a very important part of SW engineering work which is setup, use and maintenance of professional development and deployment infrastructure.

Note that we use *group* or *team* interchangeably.

Please read this document carefully and follow all the required steps.

You must deliver M0 on schedule. Any delays must be approved by class CTO Anthony and asked for via e-mail at latest 24 H before the M0 deadline.

2. Setting up GitHub

2.1 GitHub Team (Private) Repo

The purpose of this part of the exercise is for your team to set up the team private GitHub repository that is going to be used for storing your team's project (SW, documentation, formal milestone deliverables etc.) and be accessible only to the team members and instructors. Only one member needs to set up the private repository.

As a first step, all team members need to have their own GitHub account. This is mandatory, NO EXCEPTIONS - see step 1 below (you need to have your own github these days anyway).

Creating GitHub Account (if you do not have one).

1. If needed, create your own GitHub account. If you already have an account, you can skip this step.

a. When creating the GitHub account, select that you will have public repos. DO NOT SELECT private repos, or you will be asked to enter credit card information.

Creating Team private repo.

1. Select one team member to create the private repo.
2. Selected team member from step 1 uses the following link to create the repo: https://classroom.github.com/a/uVpy6OF5
3. Same team member ADDS ALL MEMBERS of the team to the private repo. (**For each not-invited team member, 1 point will be deducted from M0 Grade for the whole team in order to promote teamwork.)**
    a. Simply inviting the team member is not enough. They need to accept the invite. Non-confirmed invites will get the same penalty as no invite for those who do not follow up.

Team members are strongly **encouraged to** practice creation of branches and code merge, which turned out to be occurring problems with previous teams. Please consult on-line resources for **GitHub** best practices on this topic.

3. M0 tasks

This section describes how to develop actual M0 tasks, after you have completed all the steps in Section 2.

Task 1: Selecting Server/Platform Provider and Deploying Your Team Web Application

The purpose of this task is for you as a team to do the following:

- Select a *Server/Platform provider*
  Some possible choices are below:
    - Amazon AWS
    - Good Compute Engine
    - Heroku
- Select *a Software Stack*. An example would be a LAMP Stack. This includes: a) the OS choice; b) web server choice; c) database choice; d) and server-side language choice, and e) any other important technologies needed your Software Stack.
- The server-side language for your web application is **strongly recommended** to be one of the following:
    a. Python
    b. JavaScript
    c. Ruby
    d. PHP

You may use another server-side language that is not listed above, but it **MUST get written email approval from CTO Anthony** to even be considered. This approval is separate from your Software Stack approval. Using one of the four languages listed does not require any special approval.

When deciding on the technology to be used in your Software Stack, besides functionality (a common factor to start from), you should keep a few additional factors in mind when comparing technologies side by side. Some key factors may be:

- How often the framework is updated?
- How well is the framework supported?
- How mature is the framework?
- How well is the documentation?
- How good is the community using this tool?
- What features do is offer compare to other frameworks
- And very important: How easy is it to use/learn for all your team members given specific class schedules?

Note: your choices must be made such as not to incur any costs, which is possible today given many free offerings on the market.

Task 2: Getting Server/Platform and Software Stack from Task 1 Approved:

The task of selection has to be done very soon after formal start of M0 (TBD e.g. within a week).

When your team has decided on a software stack, **team lead** needs to email the class CTO Anthony (cc'ing the class CEO Prof. Petkovic as well) detailing your software stack. The email **must** have the following format:

- Receivers: Class CTO and CEO
- Subject: CSC 648 848 Spring 2018 Section N Team M
  - **(N is section 01 or 02; M is team number)**
- List the Technologies used in your software stack. This includes the following:
  - Server Host, Instance size (CPU and RAM)
  - Operating System and Version Number
  - Database and Version Number
  - Webserver and Version Number
  - Server-Side language and Version Number
  - Also list any technologies or packages you will need that you think is important. (you can exclude things like git and ssh). There is no wrong

answer here, just list what you can. The more the better, it'll help me determine how sound your software stack is.

MAKE SURE TO FOLLOW EMAIL FORMAT. MOST IMPORTANTLY USE THE CORRECT SUBJECT HEADING. ANY EMAILS SENT WITH WRONG SUBJECT HEADING WILL BE EITHER RETURNED OR MISSED(IGNORED).

A sample email has been given in the Appendix of this document. Please do your best to follow this format. This will ensure a speedy response. If the class CTO has any questions about your software stack, make sure that your team replies promptly. Delayed responses can delay Software Stack approval.

Class CTO Anthony will send formal checkpoint request to solicit your e-mail on completion of task 2 in order to ensure this task is completed on time.

Once your software stack has been approved via e-mail from Anthony you must begin installing and configuring your server and completion of M0 immediately.

Task 3: Installing and Configuring Approved Software Stack.

After your server/platform provider and software stack have been approved by e-mail from Anthony it is now time to begin installing everything.

There will be no detailed instructions given for this as there are too many possible configurations. But you may follow these simple steps:

1. Start server Instance with your server host
2. SSH/log into your server
3. Update your server
4. Install Webserver
5. Install DB
6. Install Server-Side language
7. Install remaining needed packages
8. Make any needed configurations.

Some notable Stack installation instructions:

- AWS w/ Node
- LAMP Stack AWS EC2
- LAMP Google Cloud Platform

This may not be the most detailed set of instructions be it gives you an order of items to work through. If you run into any issues there is help available. You may ask anyone in your team, your class, and class CTO Anthony.

THE CLASS CTO MUST BE GIVEN ROOT ACCESS TO THE TEAM SERVER (SSH ACCOUNT) AND DATABASE (DB USER WITH FULL PRIVELEGES). THERE IS NO EXCEPTION TO THIS.

Task 4: Create a Team Website and ABOUT page

The purpose of this part of the exercise is to get you to work individually to create your own webpage within your team's framework context, and then to work with your team to join these pages together using your teams GitHub account and chosen framework into a single site. We recommend that you in fact create ABOUT WWW page which introduces the team members. This can then be part of your final application and is great for your portfolio.

Everyone in your team should clone the team's private GitHub repository into his/her individual shell account or onto your local computer, and within the chosen team framework create a page that at a minimum displays their name and their or some other image (if you are comfortable it is good idea to use your own photo which is useful for your ABOUT page and portfolio – photo is of course optional). Please make sure that you have the right to use any posted photo. This work must be completed by individual student and then pushed to the team's private GitHub repository. The file(s) created/added should then be merged into the team's private GitHub repository. **The website must be deployed onto the team's remote server setup as in Task 3.**

Make sure your individual test environment is identical to the one on the server, see end of this section.

One of your teammates will need to modify your framework's home page to point to all the different team member pages. The finished site must look something reasonably close to the example in Figure 1. The user 1...user 6 buttons should be replaced with buttons pointing to the individual pages labeled with the username of the person who created that page. **The website shall be deployed onto the team's group shell account**.

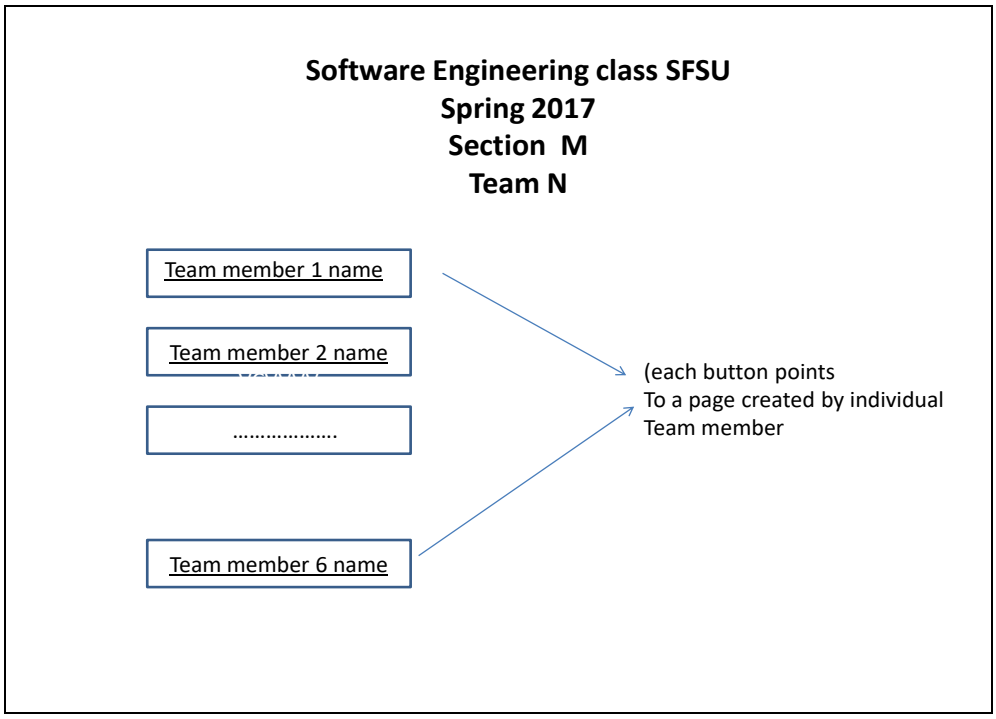


Figure 1. Example mockup of Milestone 0 team home page.

This single site must **be served from the remote server you setup in Task 3.** This is the same way you will deploy your final app, hence this is useful to make sure you learn and test it., especially how to deal with branching and code merge. Every team must understand how your team's application is deployed and managed.

NOTE on student individual test environment:
Every student should have a correct test environment on their working PC e.g. **make SURE the same steps you do to configure your deployment sever, you also do for your test environment.**. This has caused a lot of issues in the past. For example, students would setup test environments on their personal laptops, but the environment would not match what the deployment server had. If your deployment and test environment differ too much, then deployment becomes a hassle especially running the app on the deployment server.

For example, last semester the deployment server was running Node Js with NGINX webserver in front of the Node apps. Many students would have test environments on their PCs mimicking their node app on their PC but forgot to add NGINX as well. Forgetting NGINX caused the following problems:

- Apps would not even start after being pulled from GitHub
- Apps paths did not work
- Apps routes did not work.
- Apps were missing dependencies obtained from test environment.
- Apps would perform differently compared to test environment.

- And the list goes on.

4. Submission and Grading of M0

Submission of M0 for grading

All projects will be inspected and graded on the due date TBD. The key for grading will be your team's correct installation and configuration of software stack, correct setup and usage of GitHub, and the deployment of you About Me Web Application from ***YOUR CHOICE OF REMOTE SERVER (section 2).*** Emphasis for M0 grading is on correctness (not so much on UI design of the final team page) and proper usage of development tools and deployment method.

Once you are done and ready for grading you must send e-mail to A. Souza with the following subject line (**you MUST follow this protocol**)

"CSC 648-848 Fall 2017 M0 Section N Team M"
(N is section 01 or 02; M is team number)

In the e-mail say that M0 is ready for grading and report on chosen Node framework, if one is selected. Anthony will then do the checking and grading

You must deliver M0 on schedule. Any delays must be approved by class CTO Anthony and asked for via e-mail at latest 24 H before the M0 deadline.

Grading of M0

This assignment is worth 10% of your class grade, as indicated in the syllabus. The grade for this assignment will be determined according to the following criteria:

Category	Points
Correctly Installed and Configured Software Stack	3
Given Class CTO root Server and Database access	2
Correct use of Git and GitHub	2
Correct team WWW page functionality, deployment and proper usage of team's Node Application for creating web page	3
Total:	10

Appendix

i. Sample Email for Software Stack Approval

Hello Class CTO

Below is a list of the technologies used in TeamNN's software stack:

Sever Host: Google Compute Engine 1vCPU 2 GB RAM
Operating System: Ubuntu 16.04 Server
Database: PostgreSQL 10.1
Web Server: NGINX 1.12.2
Server-Side Language: Python
Additional Technologies: Web Framework: Flask
IDE: PyCharm InteliJ,
Web Analytics: Google Analytics
SSL Cert: Lets Encrypt (Cert Bot)
SASS: 3.5.5

Member's Familiarity with Server-Side Language on a scale of 1 to 5,
with 5 being very familiar and 1 being never used it.

Jane: 5
Jack: 2
John: 3
Joe: 1
Jill: 4
Jake: 5

Then any extra information you think is necessary.

Best,
TeamNN

ii. Sample Email for Milestone 0 Submission

Hello Dr. Petkovic(CEO) and Anthony(CTO),

We have completed milestone one.
Below you find the links to our GitHub Repository and Web Application

Web Application URL: https://sfsuse.com

GitHub URL: https://github.com/CSC-648-SFSU/csc648-fall17-teamNN

DateTime : January 1st, 1970

Best,
TeamNN

iii. Connecting to MySQL Database via cmd line or Workbench

Connecting to MySQL Server via MySQL Workbench

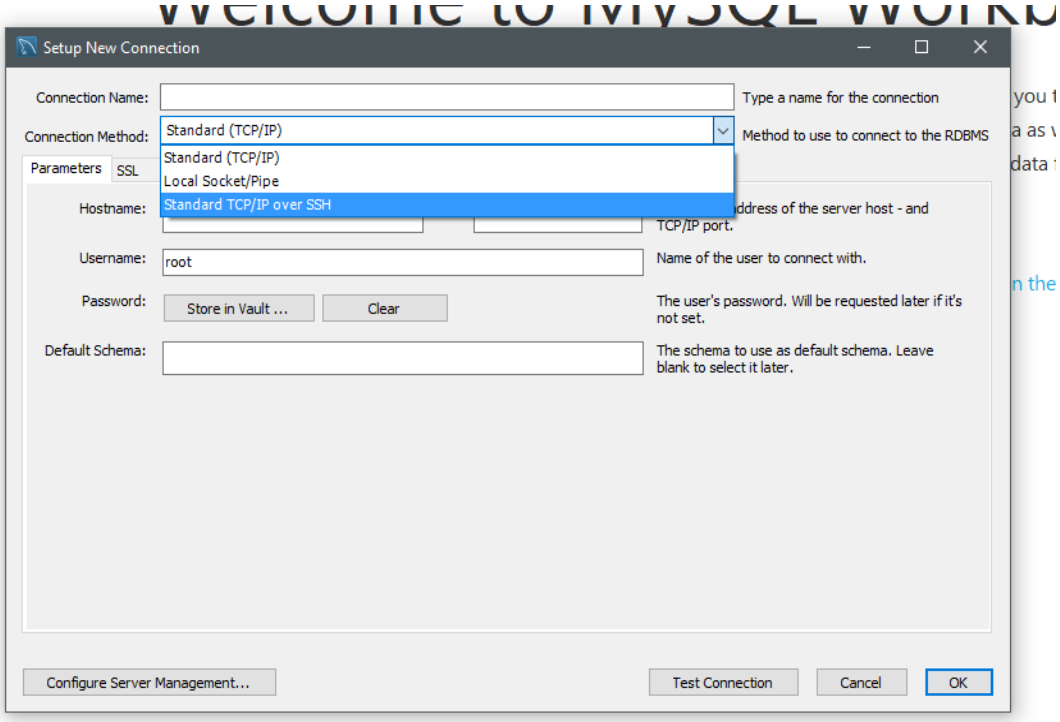
There are two ways you may access your database. One way is logging into the remote Google Compute Engine server and accessing your database via command line with the following commands:

mysql -u username -p
(where username is your database username and password is your db password)
connect db_name
(where db name is the database name you wish to connect to)

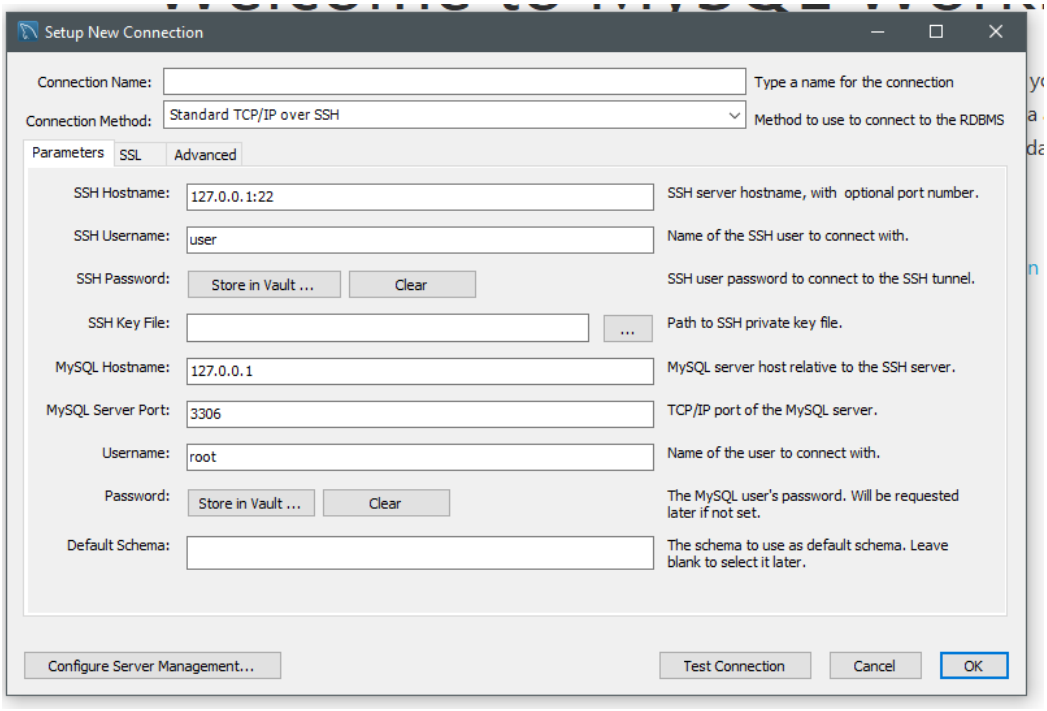
At this point you may start executing SQL commands against your connected database.

The other way is via MySQL Workbench. This is GUI application used to manage MySQL databases.

To connect to your database, you first need to create a new connection. This can be done by clicking the plus sign (+) near the text MySQL Connections. This will open a new tab. You will then select a connection type. The connection to select is "Standard TCP/IP over SSH". Figure shown below.



Next, you will need to fill in the required information for creating a new connection over SSH.



Using the above figure as an example, fill in the following information:

- Connection Name: teams_db
- SSH Hostname: your host name (could be an IP)
- SSH username: your_ssh_username
- SSH Key File: path you your key file
- MySQL Hostname: 127.0.0.1
- MySQL Server Port: 3306
- Username: database username, same as shell account name
- Password: database password, should be student id, unless changed.

Default Schema: student_username, this your database name, where username is your database account name.

Other Option is to Connect with username and passed.

This is a simple process and is the default way to connect. Therefore, the steps will not be shown. But you should note the following:

- To connect to database remotely not via SSH a port needs to be opened on the server pointed to the database. Note that this is against good security practices.
- A user account in your database needs to be made for the connection. DO NOT use root to do this, while easy and simply its lazy and goes against good security practices.