

CS 3250 Team Projects and Assignment #1

Overview

You and your team have recently joined a small company that, until now, has been organizing all of their operations manually (e.g. using Excel, taking orders by phone/email, etc.). Your team's task is to help the company adopt more automated processes for all of its operations. The vision is for you to eventually implement a software that allows for real-time updating of inventory, for customers to be able to order your products online, for automated processing of customer orders, and for financial reporting on revenue/demand, etc.

Naturally, not all of this will be done at one time. You will work with your team for over the next twelve weeks, working in short "sprints" (more about sprints on Wed.). The sprints will involve working towards a [milestone](#) in each sprint. The milestone will include one or more [issues](#) that are needed. You'll use GitHub to track, and prioritize these issues, and to meet milestones (note: issues are can also be thought of as tasks or tickets). Each time there is new code added to your project you will issue a [pull-request](#) in order to have other members of your team to review your code before having merged into your master/base branch.

During the sprints you will rotate roles. Ideally each person will be the Product Manager (PM) at least one time. The remaining members of the team will function as Software Engineers, and will handle the sub-issues/tasks that are identified. When it comes to implementing code, there should be a good deal of Pairs Programming taking place.

At the end of each sprint you will present a demo of your working code to the class, and briefly summarize how well the prior two week period went, what could be improved, etc. These are known as the Demo Days (more on DD's below).

Team Kick-off Meeting (Wed)

On Wednesday (Aug 26th) you'll have roughly one-hour team meetings in your MS Teams groups to get started (I'll join each of your meetings for 10 minutes). Objectives for this first meetings are to:

1. Join MS Teams group for your team using the join code (see Team Rosters below)
2. Select a PM for the first sprint period
3. Agree on 2-3 times that you can meet as a team for brief 15-30 min "stand-ups" (stand-ups will be quick check-ins to discuss how things are going, who is working on what, what will not be done in this sprint, etc.)
4. Agree on a time for a follow-up meeting to discuss (and perhaps complete) the initial design, and assign people to specific tasks for the first sprint
5. Begin creating a software design for the first sprint (e.g. UML of classes/structures needed for a successful implementation)
6. Assign someone to create a GitHub repo for your project (maybe PM, but could be anyone)
7. Agree on a team name (can be fun but please keep names professional, e.g. "Team Turing", "Team Classy", etc.)

Assignment #1 (Due September 9th)

The milestone for the first assignment is to have a working data storage system, or data warehouse, to store the company's inventory. This can be roughly broken down into the following high-level steps.

1. Create a DB or pseudo-DB data table representing the inventory (a starting inventory file will be provided). Keep note of the following when thinking about how to implement this.
 - There will be additional data tables that will need to be created in the future
 - All of the usual CRUD operations are needed (create, read, update, delete)
 - (4) required fields are: product ID, quantity on hand, sale cost, wholesale cost, supplier
 - Thorough testing of the DB operations - can be manual to start (will later use junit or a similar tool)
2. Create a higher-level testing (i.e. systems testing) simulator that carries out both supplier and buyer actions on a daily basis as follows.
 - Supplier event - increasing inventory for given products and decreasing company assets by appropriate factor of wholesale costs
 - Buyer event - decreasing inventory for given products they purchase and increasing company assets by appropriate factor of sale cost

Some examples of milestones for future assignments as follows (more details to come towards end of first sprint).

- Thorough testing of the existing code with junit or another unit-testing tool
- Implement revenue data tables and implement reporting capabilities on weekly finances, orders, etc.
- Create a system/mechanism to allow for customers to order "online", queuing their orders appropriately and updating all relevant data
- Refine customer ordering abilities: e.g. email customers after receiving their order, implement a front-end for customers to order
- Using reporting capabilities, and a (supplied) simulation of buyer activity over a period of time, identify regions/locations where the company could open stores – including the creation and management of store inventory

Demo Days

You'll have 10-15 for the presentation. Practice ahead of time and make sure of the following:

- Your computer works in Teams (e.g. screensharing, etc.)
- You have a script for the demo and know that it works
- You know exactly what tools/etc. are needed
- You can start from scratch if need be (i.e. cloning from your GitHub repo)
- You aren't committing/pushing code at the last moment
- You will be graded on content, grammar, and presentation skills (see rubrics in Reference section)

The presentation you turn in (one per group) must include all of the following, although you don't have to present all of it in class.

- Demonstration of running program or research progress
- What you learned from the sprint/sprint retrospective
 - What went well
 - What went poorly
 - What are we going to start/stop/continue
- SDLC statistics
 - Number of issues completed in sprint, number still existing
 - Number of issues still in project overall
 - Hours each person worked
- Code statistics
 - SLoC created, delete, modified
 - SLoC for each person on team (not everyone needs committed code each sprint)
 - Number of commits/pull-requests
 - Number of commits/pull-requests for each person
 - Number of comments in pull-request reviews
 - (not required in first few sprints) Code coverage
 - (not required in first few sprints) Static code analysis – e.g. average method length, cyclomatic complexity, checkstyle, etc.
 - (not required in first few sprints) Dynamic code analysis – e.g. SpotBugs, VisualVM, SonarQube, etc.

Team Rosters

Team 1 (MS Teams join code: 0v7f190)

- Alejandro Rojas
- Alexander Sanford
- Dominic Schneider
- Jonathan Grant
- Kevin Sandoval
- John Freda

Team 2 (MS Teams join code: l3kpt4x)

- Derek Rhine
- Diptanshu Giri
- Jesse Gonzales
- Letitia Fickling
- Noel Corrales
- Shane Bowman

Team 3 (MS Teams join code: ghyt3db)

- Arturo Valenzuela Montes
- Cody Howard
- Dustin Crain
- Edom Eshete
- Reynaldo Lopez
- Shawn Ennis

Team 4 (MS Teams join code: 694nta7)

- Adam Prieto
- Muhammad Hammad
- Daniel Martinez
- David Bowling
- Malcolm Johnson
- Uriel Bello-Gray

Team 5 (MS Teams join code: 9azqkcq)

- Cesar Monzon
- Cody Barlow
- Kamron Correia
- Sarmad Tello
- Toby Sack
- Ryan Dinville

Team 6 (MS Teams join code: 0p3pwxv)

- Hector Cruz
- Busra Ozdemir
- Adam Wojdyla
- Dakota Miller
- Firew Handiso
- Riley Strong

References

GitHub References

- Milestones: <https://docs.github.com/en/github/managing-your-work-on-github/about-milestones>
- Issues: <https://docs.github.com/en/github/managing-your-work-on-github/about-issues>
- Pull-requests: <https://docs.github.com/en/github/collaborating-with-issues-and-pull-requests/about-pull-requests>
- Creating a pull-request: <https://docs.github.com/en/github/collaborating-with-issues-and-pull-requests/creating-a-pull-request#creating-the-pull-request>

Presentation Rubrics

- http://readwritethink.org/files/resources/printouts/30700_rubric.pdf
- http://www.makinglearningreal.org/pdfs/scoring_rubric.pdf
- http://www.fresnostate.edu/academics/oie/documents/oral_pres_rubric.pdf