TCSS 360C: Software Development
and Quality Assurance Techniques - Winter 2017
http://faculty.washington.edu/wlloyd/courses/tcss360

Institute of Technology
University of Washington – Tacoma
Instructor: Wes Lloyd

Project Phase 1 – User Stories and Initial Services Development

Coffee Finder – Coffee Shop Review System

Due Date: Monday February 13th, 2017 @ 11:59 pm, tentative

Objective

Project phase I will involve reviewing the mock client interview to develop a set of user stories for your coffee finder application/services. The mock client interview script is available here:

http://faculty.washington.edu/wlloyd/courses/tcss360/project/clientInterview_phase1.pdf

Your user stories will describe the features of the system that will be built based on the customer's vision. To support user stories, a number of tasks will be involved. Tasks will describe what coding needs to take place to provide particular features.

The ultimate goal of the first agile lifecycle is to translate the customer requirements into user stories, tasks, and then build initial web services. In this phase, you'll meet as a team to write the user stories, identify associated tasks, assign points to the stories, and decide which user stories to deliver at the end of phase 1. As this is an agile project, and we are using an agile process, the team has leeway to define the user stories to provide for the project phases, and to scope stories appropriately. It is expected that teams will have user stories which may not be implemented by the end of the quarter. The group should consider what tasks are best done collectively, and which tasks can be assigned. If all tasks are done collectively through group consensus expect productivity to be lower than if tasks are distributed with different team members taking ownership of particular pieces. You are open to create whatever team hierarchy as needed to facilitate the project. It is recommended that someone creates a meeting agenda, takes notes, and ensures that meetings don't last forever, and that things remain productive.

Project Phases

For purposes of future planning, the overall plan of our agile phases for TCSS 360 are as follows:

Phase	Primary Objective	Major Milestones
Phase 1	Initial Services Development	Develop initial REST webservices to support Basic CRUD operations. In phase 1, data persistence is not required. Data is created, read, updated, and deleted entirely in memory, but when the web application is shutdown or redeployed, data isn't required to persist. Using agile project management scope appropriately and deliver some available services and data by the end of phase I.

		A simple WEB GUI should be implemented in Phase I to provide read-only views of raw data in the system based on HTTP GET calls. Basic queries can provide JSON output for POST requests. Data can be stored in memory.
Phase 2	Data persistence, and data sharing	Phase 2 goals include: (1) persisting data to a database, (2) developing data sharing services which allow data to be shared amongst the seven group projects in TCSS 360C, and (3) further implementation of REST use cases including more sophisticated data queries.
Phase 3	Interactive GUI	The primary goal of phase 3 is to develop a rudimentary interactive GUI which allows users to submit new data (new coffee shops, reviews), lookup existing data, edit existing data, and delete data. The goal will be to have a semi-working system by quarter's end.

Once user stories are implemented in phase 1, development tasks can be assigned amongst team members and initial prototype implementation of services can begin. The ultimate goal of the first agile lifecycle is to translate the customer requirements into user stories, and to begin implementation of prototype services. For SOA/SaaS architecture, each service should be independently implemented and decoupled from others. This supports easy decomposition of implementation amongst developers. For example, one developer could work on coffee shop reviews CRUD, while another works on the general coffee shop description CRUD.

Showcase Feature

Given the limited amount of time in the quarter to build a complete system, each team will meet and decide, based on team interests and skills, to develop one high quality showcase feature for the coffee finder application. For example, in the customer interviews geocoding can enable an address to be translated to a lat/long coordinate. Once lat/long data is captured for coffee shops, this can enable geospatial searches, as well as a map based displayed of where coffee shops are located. As this is an agile software project, each team can prioritize and define which features can be delivered given the available time. It may not be practical for every team to provide a map based interface or geospatial search. **However, it *is* expected that each team will provide a "showcase feature" based on talents and interests of the group.** Implementation of a showcase feature is required by the end of the quarter. It is recommended to confer with the instructor and get approval of an agreed upon showcase feature. This will both help to ensure the feature is sufficient and scoped appropriately. The showcase feature can be something from the client interview, or an entirely new idea the team comes up with. At the end of phase I, the team should provide a short written description on the proposed showcase feature.

Development Tools

For the project, it is required to use (1) PivotalTracker for user story creation and task tracking, (2) GitHub for source code management and project wiki, and (3) Heroku for web application hosting. In the coming weeks a public shared PostgreSQL database with geospatial support (via POSTGIS) will be provided for each team to store data in a relational database. The use of this publicly provided PostgreSQL database, however, is not required. The team may elect to use any technology for data persistence. (Teams can even implement their own data persistence method – as long as it works!)

Development tool and technology examples will be provided throughout the class (e.g. REST/JSON services in Java, Java UI client development, PostgreSQL data persistence). Teams, however, may elect to use any language, tool, and/or technology as tool selection is ultimately up to the team's discretion.

Project Status Reports (10% of the TCSS 360C course grade)

On the TCSS 360C syllabus approximately 10% of the course grade is derived from project status reports. To receive this 10%, each project team will develop a wiki on its project GitHub website. The wiki should include at a minimum meeting minutes for at least three meetings, one for each project phase. Additional meeting minutes, meeting agendas, and project artifacts should be added to the wiki. Adequate documentation will ensure that all group members receive the full 10% for "project status reports" as part of the TCSS 360C course grade. In addition to the wiki, project content in Pivotal Tracker will also be considered. The project status reports grade will be determined at the end of the quarter by reviewing all project documentation from each team's project wiki page and Pivotal tracker website.

Cross-Team Collaboration

Each team will elect a member to participate in one or more cross-team meetings to be held near the end of phase I and early in phase II. In the cross teams meetings, a common GET data API will be defined. The generic GET API will allow data to be retrieved in bulk from each team's coffee finder web application. Each project will then invoke the GET API of all other team's web applications to share data. By sharing data across project teams using a common GET API, all TCSS 360C projects will benefit from having more data in their application. The cross team collaboration will agree on which data to share, and the API to share it. The cross-team meeting will specify a common API for: (1) coffee shop data sharing, and (2) coffee shop review sharing. Each team will then implement this common API to share coffee shop data (1). Coffee shop review data sharing (2) will be spec'd, but the implementation is optional. Data sharing tasks should be included as a user story or stories with associated tasks in Pivotal Tracker.

Phase I Deliverables (20% of the TCSS 360C course grade)

Phase I will produce user stories, tasks, and low-fi UI sketches to support agile project development across all project phases. User stories should be written so that entire stories can be scheduled into phases I, II, and III. User stories will identify tasks which are scoped into phase I (initial services), phase II (data persistence), and phase III (UI development). The team will implement user stories in Phase I to provide an initial prototype implementation of CRUD data services for coffee shop description tracking and coffee shop reviews.

The phase I implementation should complete at least a prototype implementation of at least 2 user stories and associated tasks. If the prototype is not fully functional, there should be evidence of good progress towards completion. Other user stories can be in development or not yet started by the end of phase I. The team will deploy and host an initial set of prototype REST web services on Heroku. The team will provide JSON objects and URLs as test cases to demonstrate basic CRUD operations. Incomplete services should be "stubbed" out pending future implementation. The initial services should catalog at least 5 to 10 coffee shops. JSON objects should be created, and a REST API should support creating these in-memory objects.

Grading Rubric:

The following rubric will be used to grade the phase I project delivery. Scored out of 100 points. (100/100)=100%

User Story Definition: 50 points

User stories should answer who, why, what questions. They should be testable (specific and measureable) and assigned a point ranking. Stories / tasks should be assigned to individual team members. Phase I should include user stories spanning all three planned phases of the project (I, II, and III). Not all user stories must be provided by the end of phase I. It is expected that user stories will evolve until the end of the project. User stories should be entered into a shared Pivotal Tracker project. The project should be set as a public project. All team members should be added as well as the instructor (wlloyd@uw.edu) and grader (henrylai4@gmail.com).

Initial Prototype Services: 40 points

Phase I will include the initial implementation of at least a few CRUD services for the coffee finder application. The team should include test cases which demonstrate what is operational at the end of phase I. This should include test scripts and associated example JSON objects to demonstrate the CRUD API. Adequate documentation **must be** provided to enable the instructor and grader to assess the state of the project at the end of phase I. This documentation should be on the team's github wiki page and include a brief written description of active service endpoints (URLs). It is important to have some functionality and test cases with a description of how to test what is available. Imagine that you are providing these deliverables to the customer for review. The system should have data that describes at least 5 to 10 coffee shops. It should be possible to inspect this data live in the Heroku web app deployment. (http GET)

Code should be committed to github, and a phase I deployment made to Heroku for testing/grading.

Showcase Feature Proposal: 5 points

The team should meet and discuss a showcase feature to implement by the end of the quarter. The team should develop a short written description describing the showcase feature. The feature should be described on the group's github wiki page. User stories and tasks should be added to Pivotal Tracker at the end of phase I, or later to track the feature development.

Effort Reports: 5 points

EACH group member should provide an **effort report** to describe each member's contribution for phase I. **Effort reports** must be submitted INDEPENDENTLY and in confidence (i.e. not shared) by each team member via canvas. Effort reports are not used to directly numerically weight assignment grades.

Effort reports should be submitted in confidence to Canvas as a PDF file named: "effort_report.pdf". Google Docs and recent versions of MS Word provide the ability to save or export a document in PDF format.

Provide 1 (low effort) to 10 (high effort) rankings for: research, design, coding, testing, user stories, management. Effort scores should add up to 10 across the team for each category.

Management includes activities such as keeping meeting minutes, creating meeting agendas, assigning work to team members, facilitating group coordination and communication. User stories includes creating, writing, and administering user stories in pivotal tracker. Research includes investigating possible solutions to design and coding challenges. Design, coding, and testing should be self explanatory.

Here is an example **effort report** for a three-person team
(written from the point of view of Jane Smith):

Effort Report

1. John Doe

Research	2
Design	3
Coding	3.3
Testing	3.3
User stories	5
Management	8

2. Fred Jones

Research	2
Design	3
Coding	3.3
Testing	3.3
User stories	2
Management	1

3. Jane Smith

Research	6
Design	4
Coding	3.3
Testing	3.3
User stories	3
Management	1

Team members may not share their **effort reports**, but should submit them independently in Canvas as a PDF file. Failure of any members to submit their **effort reports** will result in all members receiving NO GRADE on the project deliverable… (*considered late until all are submitted*)

What to Submit

For project phase I, a single group member will submit a URL to the group's github wiki. The group should create a wiki on github which describes:

- The name of the group project / team
- A list of the group member names
- The group's Pivotal Tracker project URL with user stories. Be sure to add the grader and instructor emails to the Pivotal Tracker project, and to make the project public.
- Documentation describing available service endpoints on heroku (URLs please) as well as links to JSON objects and test scripts for testing.

Be sure that the github wiki is public and that the instructor and grader are able to access your repository to inspect code and documentation. We should be able to test your project deployment based on the information described on the wiki page.

All group members should submit to Canvas an "effort_report.pdf".
Google docs or office 365 can be used to easily create a PDF file.