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

Project Phase 3 – Final Services Development & GUI

Coffee Finder – Coffee Shop Review System

Due Date: Monday March 13th, 2017 @ 6:30pm

Objective

Project phase III involves building a web GUI (**View)** to interface with your **Model** and **Controller** functionality from Phases 2 and 3 to provide operational Model-View-Controller Web services application which supports create, read, update, and delete functionality multiple core coffee shop data classes. For Phase 3 projects should persist (save) data to the PostgreSQL database (or another database or custom built persistence layer, e.g a file). Relevant user stories and features should be implemented.

The mock client interview script is available here:
http://faculty.washington.edu/wlloyd/courses/tcss360/project/clientInterview_phase1.pdf

For phase 3, user stories and tasks in PivotalTracker should be reviewed and updated to capture the final state of the project development. Where new required features have been identified, new user stories should be added. For user stories which are moving from requirements definition to implementation, tasks should be added to track key steps and activities of the implementation.

The ultimate goal of the third agile lifecycle is to build a GUI to interface with the services built in phases 1 & 2, and to wrap up data services and functionality for your entire application to provide an overall working prototype application. Phase 3 project implementations should include implementation of the common data sharing API that teams defined in the joint meeting at the beginning of phase II. Using a script file, teams should be able to call the Get APIs to acquire the list of coffee shops in JSON format. The JSON output can be saved as a local file, and used to call your create coffee shop service. There are several options for how you could handle loading other group's data into your application. All options are acceptable. The only requirements is that every coffee shop application should have at least one record from every other coffee shop application. Assuming the common API returns an array of coffee shops then there are a few possibilities for loading the shared data:

1. Develop a create coffee shops service that accept a JSON array to create an array of coffee shops. Your array-based create coffee shops service could simply send individual array elements to your existing create coffee shop service.
2. The JSON array could be parsed/separated using Linux bash scripting to make individual calls to an existing "create coffee shop" service endpoint.
3. Teams can manually edit the JSON array output returned from each team's coffee shop application to then manually call and load the shared data. This is the easiest, but the most tedious option.

4. You could create a new service which accepts a Heroku coffee shop application URL and implement a REST client in Java to automatically call the other group's projects and populate your database with their data. This is the most desirable, but potentially most complex option.

At the end of phase three, CRUD services should be complete for the majority of the core objects of the system design.

Key Tasks for Phase Three:

- Ensure your github repository is a private repository (if not already). Once the private repository is free add the instructor (userid:wjlloyd). Students are eligible for free accounts from github with unlimited free repositories. To sign up follow the instructions here:

  https://education.github.com/

- Update Pivotal Tracker to reflect the final state of the project. Mark completed tasks, etc.
- Develop a prototype GUI that enables CRUD for data objects of your Coffee Shop Application.
- Complete implementation for the majority of the core objects of your system.
-

In phase three, teams should meet to complete implementation of the system. This includes testing. The group should consider what tasks are best done collectively, and which tasks can be assigned. It is recommended to have group working sessions which are not distributed/online. If the group has a strong understanding of what tasks need to be completed, and is confident with the technology stackm then remote work can be more successful. Software development tasks should be assigned to individuals or programmer pairs. The team is responsible for creating the necessary team hierarchy to facilitate the project. It is recommended that a leader creates a meeting agenda, facilitates meetings, takes notes, and ensures meetings produce action items for the next meeting.

Project Phases

The overall plan of our agile phases for TCSS 360 are as follows:

Phase	Primary Objective	Major Milestones
Phase 1 *Complete*	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

Complete		database, (2) specification and prototype of data sharing services to allow data to be shared amongst the six group projects in TCSS 360C, and (3) complete implementation of 70-80% of the core services with primary focus on the model and controller.
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 working application by quarter's end.

Showcase Feature

The showcase feature is now extra credit. Up to 10% can be earned for the Phase III deliverable for completion of the showcase feature. Extra credit points will be assessed based on the complexity of the showcase feature. The showcase feature description, whether implemented or not, should still be described on the group's project wiki.

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. PostgreSQL is recommended for data persistence, but teams are free to choose any available database /database service as needed. The use PostgreSQL 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!)

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

On the TCSS 360C syllabus approximately 10% of the course grade is derived from project status reports. This 10% reflects the completeness of the final project documentation. To receive this 10%, each project team will develop and maintain 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.

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

Phase III will focus on the development of a GUI to provide a web interface (View) for the services built in phases 1 and 2 of the Coffee Finder application. User stories should be written/revised as needed and tasks scheduled into phase III (UI development). By the end of phase III a prototype working, but not polished GUI, should be available to interact with web services built in phases 1 and 2. It is ok if there are still loose ends as we are not developing production quality software in TCSS 360. Clear progress should be visible towards completing some GUI to interface with working web services for at least 2-3

core data classes to fulfil major user stories of the project design. Coffee shop applications should use a database or file to persist data so when a user visits the website there is available data at all times. Changes to the data should be savable. Teams should implement a GET api to share data across project groups. When the shared data API is complete, project Wiki's should be updated to reflect its availability.

Project Demonstration – Monday March 13, 6:30pm CP 108

On Monday March 13th, groups will demonstrate their completed coffee shop applications in class. Groups should prepare a 5 to 10 minute presentation that shows the available functionality of their coffee shop web applications. The project demonstration will be used to make a case for the 90 points eligible for "Overall Project Functionality" of phase III. A good presentation ***can also*** influence "Project Status Reports – 10%" for the overall course grade. Artifacts created for the final presentation (PowerPoint slides, videos, etc.) should be included on group's wiki pages. Groups may combine with a live demonstration the use of power point slides and videos to describe and demonstrate the functionality of their coffee shop applications. Presentations can simple be a live demo, or additionally include powerpoint and/or videos.

Grading Rubric:

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

Overall Project Functionality: 90 points

Phase III will include the implementation of the model, view, and controller. 90% of the Phase III deliverable is based on the final state of the project functionality. Teams should have developed a coffee shop finder application based on web services that is functional, but not polished. There should be some evidence of model, view, and controller components implemented using web services to fulfil key user stories from the application. When functionality is not available in the GUI, test scripts and JSON objects should be provided to show functionality for these pieces via the group's wiki. Adequate documentation **must be** provided on the team's wiki page to support testing of the GUI and available services. The wiki should include links to test scripts and service endpoints (URLs). The wiki page will provide deliverables "to the customer" for review.

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

Effort Reports: 10 points

EACH group member should attend class on Monday March 13th at 6:30pm in place of the final exam. In class there will be a final effort report based on that used by Dr. Tenenberg. There is no online effort report submission for phase III.

What to Submit

For project phase III, the group's github repository should be made private using the free student accounts (https://education.github.com/). Github wiki content should be moved into the new repository. The wiki collects and describes all team development activities. The wiki should include:

- The name of the group project / team
- A list of the group member names
- The group's Pivotal Tracker project URL with updated content. *(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:
    - Test scripts
    - Test data objects (JSON)
    - Endpoints where appropriate (URLs)
- Link to group's WEB GUI
- Any other project documentation:
    - Project demonstration slides, videos
    - Meeting minutes, notes

The github wiki should be public, and the instructor (userid:wjlloyd) and grader (userid:) should have access to your github repository and pivotal tracker site for code and documentation inspection.