

Project Phase 2 – Additional Services Development – Database Persistence

Coffee Finder – Coffee Shop Review System

Due Date: Wednesday March 1st, 2017 @ 11:59 pm, tentative

Objective

Project phase II involves completing **Model** and **Controller** functionality to provide operational web services for create, read, update, and delete functionality for the core coffee shop data classes. Phase 2 includes persisting (saving) data to the PostgreSQL database. 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 2, user stories and tasks in PivotalTracker should be reviewed and updated to capture the ongoing state of 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 second agile lifecycle is to continue the implementation of customer requirements from user stories and tasks to produce a more complete set of web services required for the Coffee Finder web services application.

At the end of phase two, CRUD services should be complete for the majority (70-80%) of the core objects of the system design.

Key Tasks for Phase Two:

- Update completed user stories and tasks in Pivotal Tracker
- Write and refine new tasks for Phase II/III user stories
- Rank/refine point assignments for Phase II/III user stories
- Assign development tasks to individuals or programming pairs for phase II
- Implement the majority of the core objects of the system (70-80%)
 - o Focus is on Model and Controller development, View is phase III.

In phase two, teams should meet to write new user story tasks, rank and assign points to the stories, and decide which user stories to complete by the end of phase 2. The group should consider what tasks are best done collectively, and which tasks can be assigned. 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 TCCS 360 are as follows:

Phase	Primary Objective	Major Milestones
Phase 1 <i>Complete</i>	Initial Services Development	<p>Develop initial REST webservice 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.</p> <p>Using agile project management, scope appropriately and deliver some available services and data by the end of phase I.</p> <p>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.</p>
Phase 2	Data persistence, and data sharing	<p>Phase 2 goals include: (1) persisting data to a database, (2) specification and prototype of data sharing services to allow data to be shared amongst the six group projects in TCCS 360C, and (3) complete implementation of 70-80% of the core services with primary focus on the model and controller.</p>
Phase 3	Interactive GUI	<p>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.</p>

Showcase Feature

In phase 1 a showcase feature was proposed. In phase 2 research, prototyping, and implementation should begin to develop the showcase feature with a planned completion by the end of phase 3. The showcase feature should 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 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 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.

Cross-Team Collaboration

The cross-team collaboration meeting will be held in class on Wednesday February 15th. Each team will elect a member to participate in the meeting. If needed follow-up meetings can be scheduled in phase II. In the cross-team meeting, 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. Teams will also share Heroku endpoints with each other via the project wiki pages. Each project team will then invoke the GET API of all other team’s Coffee Finder applications to share data. By sharing data across project teams using a common GET API, all teams 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) can 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 II Deliverables (20% of the TCSS 360C course grade)

Phase II will focus on user story (feature) implementation to build out the Model and Controller aspects of the Coffee Finder application. User stories should be written/revised as needed and tasks scheduled into phase II (implementation/data persistence), and phase III (UI development). By the end of phase II the majority of the Model and Controller code should be implemented to provide a headless (no GUI), but functional application. It is ok if there are still loose ends, but clear progress should be visible towards completing web services to support core functionality. Data persistence should be at least 50% functional or more. Teams should begin development of the common data sharing API to share data across project groups. When the shared data API is complete, project Wiki’s should be updated to reflect its availability.

Grading Rubric:

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

Pivotal Tracker Updates: 20 points

At the end of Phase II, teams should have updated pivotal tracker to accurately capture the current state of project development. New user stories should be created, tasks added and updated accordingly. The group should provide a link on the project wiki to the pivotal tracker website.

Model, Controller, Data Persistence Implementation: 70 points

Phase II will include the implementation of the majority (70-80%) of the model and controller functionality to develop functional, though headless (no GUI) application. Phase II will also include the implementation of data persistence. Test cases (endpoints, shell scripts, JSON objects for tests) should be provided to demonstrate operational services at the end of phase II. Adequate documentation **must be** provided on the team's wiki page to support testing of available services including links to test scripts and service endpoints (URLs). The wiki page will provide deliverables "to the customer" for review. It should be possible to inspect data using http GET endpoints.

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

Data Sharing API Definition: 5 points

At the end of phase II project teams should have met and proposed a common API for data sharing. Each team should describe this common API on their wiki page (by Phase II), and provide suitable links once the API is available (Phase III).

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 II, the group's github wiki will collect and describe 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 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:
 - o Test scripts
 - o Test data objects (JSON)
 - o Endpoints where appropriate (URLs)

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

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.