Group Project

Due: Beginning of May, 2019

Last Update: April 11, 2019

1. Introduction

In this semester, students in Senior Seminar will be working on their capstone project in eight teams of three and three team of two. Each team will design, implement, deploy, and demonstrate a web-based application for an (imaginary) company or institution to facilitate its employees to complete the online forms for authority to travel, report travel expenses after their trips, and get reimbursed if he/she follows all the established travel procedures and policies.

2. Teams

Please find your teammates and fill up Senior Seminar Group Project Teams by the class time on January 31, 2019, when all teams should be formed. I will then populate the following table correspondingly.

Team #	Team Name	Team Members
A	Ruby Miners	Caleb Watson, Chase Perry, and Robert Brown
B	Team Rick	Kayla Rivera and Jamanne Anderson
C	Ruby Charlie Foxtrots	James Newton, Michael Montgomery, and Robert De La Cruz II
D	jmj.rails	Montrez Cox, Johnnie Oldfield, and Joanne Wardell
E	Cra-ra	Marco Colasito, Wei Dong, and Wei Dong
F		
G	Minecraft Steve(s)	Christian Gerhardt, Sean Passmore, and Tyler Angelier
H	Team H	Alexander Amundson and Mohammed Qishlan
I	Anonymous	Amanda Seasholtz, Savon Jackson, and Jose Delgado
J	Ruby Group	Jasmin Miravete, Kendall Herron, and Alexis Leon
K	group.tbt	Tristan Walk, Taylor Tate, and Bugun Choi
L	Thugs R Us	Lewayne Jamison and Justin Deloach

3. Project Spec

3.1 User types

Services provided by your application are accessible solely to those users with an account. There are at least four types of accounts:

- Employees
- Budget approvers (in various branches/departments/teams/offices...)
- Payment manager

And finally, of course, there is a special type of user account:

- System admin, who is basically you, the developers of this app. System admin is ultimately responsible for the technical configurations and maintenance of the entire app.

3.2 Two steps for Travel Reimbursement

3.2.1 Travel Authorization

- Before traveling, *employee* completes and submits an online "Travel Authorization Form".

  In addition to containing necessary information of the planned trip such as destination, purpose, duration, etc., such a form also contains estimated and itemized expenses such as flight, hotel, meals, etc., and information of all sources (e.g. account string) for supporting his/her trip.

- Upon receiving the form, the relevant *Budget Approver(s)* will approve or deny it. The form will be returned back the employee for resubmission should any Budget Approver deny it.

- The form will be forwarded to the *Payment Manager* once it has received all approvals.

> It is the employee's responsibility to ensure that the Travel Authorization Form is completely approved prior to travel.

3.2.2 Completing an Expense Report

- After returning from a trip, employee completes and submits the "Expense Report" for reimbursement.

  All expenses should be itemized by type of expense and date of expense.

  All expenses should be associated with required receipts and documentation.

  Note: it is quite common for one source to cover multiple expense items and/or one expense item to be jointly covered by multiple sources.

- Upon receiving the report, the relevant *Budget Approver(s)* will approve or deny it. The report will be returned back the employee for resubmission should any Budget Approver deny it.

- The report will be forwarded to the *Payment Manager* once it has received all approvals.

  The Payment Manager processes Expense Report for a trip only if the trip was authorized before it took place and makes the ultimate decision after processing it. If he/she approves it, the employee will be reimbursed. Should more information/justification/update be needed, the report will be returned back to the employee for resubmission.

3.3 Data Analysis

The app should collect and store all travel-related data so as to allow users (i.e. employees, budget approvers, and payment manager) to perform data analysis on such data. E.g. the most expensive trip, number of approved/denied expense reports in the past year, etc.

3.4 Development Platforms

Ruby on Rails in the backend.

User Interface (UI) for at least one type of users (i.e. employees, budget approvers, or payment manager) should be implemented/componentized with ReactJS. Such UI includes all web pages that are used by the type of user you choose.

3.5 Notes

Apparently and understandably, the project spec outlined above is non-technical, open-ended, and vague. Also, for the better or worse, as the project progresses, some of the project requirements might be even changed (which by the way happens a lot in the real world). It is then your responsibility to "*technologicalize*" the project spec, adapt to its changes, implement and deliver a software product while meeting the academic standard appropriate to this capstone class.

4. Project Management

Mandatorily, you should use Github to version-control and manage your project.

Warning: when working on your code and making commits, please make sure that **every team member has a fair share of the workload** since Github will automatically keep track of it as a measurement of each team member's contributions to the entire project. Optionally, you could also explore to use tools like Github slack or Pivotal Tracker for the better management of your project.

5. Project Plan (VERY Tentative)

January – Project is kicked off. Teams are formed. Github repos are created.

Then, in general, we will develop this project in three phases.

Phase I (Early February ~ Early March) – Each team wireframes their application with use cases/page flows using *Balsamiq*.

> It is very generous for balsamiq.com to provide me and my students a "trial license" (fully-functional, but has an end-date) to use their product on home and school computers. I will share the license with you later.

Phase II (Mid March ~ Early April) – Work on the server-side/backend of the application with Ruby on Rails. You should focus on completing majority if not all of server-side functionalities of your project.

Phase III (Early April ~ Late April) – Work on prettifying the client-side/frontend of the application with

Bootstrap and componentizing the UI for at least one type of users (i.e. all web pages used by that type of user) as SPAs with React.js and React Router for better user experiences.

The project will be concluded with formal project demos tentatively scheduled in the first several days in May.

> Notice that during the development of your application, it is quite natural and understandable that in order to fix bugs, change data formats, enhance existing features, and/or for whatever reason, modifications (some of which might be even major) have to be made to some of the elements in your project that you have already completed before. Also, although Phase II is supposed to be backend-centric, reasonable time has to be spent on a basic but working frontend (primarily in order to test the backend).
>
> The bottom line is: the three phases above are not necessarily mutually exclusive or clearly separated.

Table below outlines a very **tentative** project plan including important dates, milestones, deliverables, and demos.

Week of	Activity	Deliverables
03/19 (spring break week)	Project development	
03/26	Project development	
04/02 (End of Phase II)	Project development	Progress demos. See here for the schedule. All are required to attend.
04/09	Project development	
04/16	Project development	Progress demos. See here for the schedule.
04/23 (End of Phase III)	Project development	
04/30	Project development and finalization	Final demos. See here for the schedule.

5. Final Project Demonstrations and Deliverables

5.1 Details of the Final Project Demos

Since the number of students in Senior Seminar this semster is unusually high, some of the final demos have to be scheduled on Wednesday 05/01.

- Four groups on Tuesday 04/30, three on Wednesday 05/01, and four groups on Thursday 05/02. See here for the schedule.

  All students are required to attend all demos on 04/30 and 05/02.

  Since the demos on 05/01 are not during our class time, except for those who are scheduled to demo, your attendance is going to be optional.

- 20 minutes per group. **Each student** should have a chance to talk during the demo.
- All projects must be running on Heroku.
- Each demo needs to be road-mapped by PowerPoint slides. The slides need to contain no more than 5 pages, 3 of which cover the following:
  - An Introduction to your project. **Somewhere on this page shows the URL of your app on Heroku.**
  - High-level design of your project where unique features are highlighted.
  - A list of demo scenarios/cases the at leaset covers what is listed in the Appendix at the end of this document.
- Majority of the time (> 90%) should be spent on the actual demo of the project. Make sure the pages implmented in React.js are highlighted.
- Suggestions
  - **Rehearse! Rehearse! Rehearse!**
  - Pre-seed DB tables with necessary data ahead of time.
  - Open multiple browser windows when your demo starts so you don't have to waste your time switching between different types of users.

5.2 Final Project Deliverables

By the time when your final demo starts,

- Submit to me a **print copy** of your slides
- Push the final version of your project to the master branch of your group repository on Github
- Deploy the final version of your project on Heroku with all DB tables appropriately created and seeded
- Zip up the following files into one zipped file and send it to me through **BlazeVIEW email**

> **One submission per group please**. If I received multiple submissions from the same group, I will assume they are identical and arbitrarily choose one to grade:

- Slides for the final demo
- A project report (up to 3 pages)

  Although I don’t have a format/template for the report, which means you have the freedom to include anything that you want me to read such that I can have more insights into your project, there are at least three things that I DO expect in your project report:

  - A section that clearly lays out each **individual** team member’s contributions to the project.
  - A section that clearly lists all the pre-seeded users' names, passwords, and types.
  - A section that lists all the commands to run before firing up the servers of your project on a local computer. It is your responsibility to ensure the commands are complete and accurate because I will literally run them when testing your project. Any command that fails to run will **substantially** hurt your grade.

6. Group Project Points Breakdown

- 5%: Revised project mockups that you pushed to the *project_design* branch on Github on 03/05.
- 5%: The progress demo made on 04/16 or 04/18.
- 80%: Functionalities of the project. **For this part, I will make adjustment (up to 30%, or equivalently 24% of the entire group project) for each individual student based on his/her contributions to the project and to the team.** Such adjustment will be based on:
  - Github insights
  - Project report
  - Training projects
- 5%: Quality of the final project demo on 04/30 or 05/01 or 05/02.
- 5%: Project report.

Appendix

In the following, please find a minumum set of test scenarios that your app is supposed to pass.

- All users have to login before using this app.
  - Naming convention: Employee names should begin with an "E"; budget approver names should begin with a "B"; and the payment manager's name should begin with a "P".
  - It is implied that accounts have been pre-created for the users.

    - It is also implied that relevant departments/units/offices have been pre-created with certain budget for travel.

- Scenario 1: A travel auth. form with *estimated costs* is submitted by an employee before his/her trip, approved by all relevant budget approvers, and eventually makes it to the payment manager's page.
- Scenario 2: A travel auth. form with *estimated costs* is rejected by one (or more) budget approver(s) due to various reasons (e.g. overbudget, inappropriate travel item, etc.). The employee fixes and resubmits the form, which gets approved by the relevant budget approvers.
- Scenario 3: An expense report with *actual costs* is submitted by an employee after his/her trip. No rearrangement of expenses (i.e. split, join, etc) is needed. The form is approved by all the relevant budget approvers, and eventually approved by the payment manager for reimbursement.
- Scenario 4: An expense report with *actual costs* is submitted by an employee after his/her trip. Rearrangement of expenses (i.e. split, join, etc) is needed. The form is approved by all the relevant budget approvers, and eventually approved by the payment manager for reimbursement.
- Scenario 5: An expense report is rejected by one (or more) budget approver(s) due to various reasons (e.g. unauthorized travel, overbudget, etc.). The employee fixes and resubmits the form, which gets approved by all budget approvers.
- Scenario 6: An expense report is rejected by the payment manager due to various reasons (e.g. missing receipts, inappropriate travel item, etc.). The employee fixes and resubmits the form, which gets re-processed and approved by relevant budget approvers. The payment manager evantually approves it for reimbursement.