Post/Pre-Lab X Report
COMP 4018
Prof. Alcibiades Bustillo

Objectives

1. Understand the design and implementation of a complex database-backend web app.
2. Understand the use of the E-R model for database application design.
3. Gain experience by implementing applications using layers of increasing complexity.
4. Gain further experience with web or mobile app programming, and REST APIs.
5. Learn to manage a complex software development project leveraging use of industry standard project management and software development tools.

Overview

Your task is to build an excellent development team with two members to design and develop a novel web application. Although we provide you with a set of minimal requirements we hope that your team will come up with an original design.

Your web app must use a REST API and industry standard web development frameworks. This web app could either be developed as a responsive web application or as a native iOS or Android mobile app. The app should also follow an MVC (Model-View-Controller) architecture with clean separation between client and server sides. Web development framework options are open, so we strongly encourage your team to select one of these.

Client-side Framework

- Web: AngularJS or EmberJS.
- Mobile: iOS, Android or Ionic Framework

Application Server Framework

- Flask Framework (Python)
- Play Framework (Java or Scala)

DBMS technologies

- MySQL
- MariaDB
- PostgreSQL

Sprint 0: Conceptual Design and Development Plan (weeks 1-2)

Due Date: February 15, 2019, 11:59 PM (Grade weight: 20%)

Minimal Deliverables:

- Team/App Name and Concept - Select a Project Manager (PM)
- Written **Intellectual Property Agreement** (see below)
- Selected Development Frameworks
- Complete ER Diagram
- Requirements Definition Document - Description of each operation supported by the app (2-4 pages)
- Github account - With project repository created and initial commit with the Hello World app
- Working IDE Environment on every team member workstation with Hello World app running and connected to teams GitHub repository
- Master Project Development Plan. mapped to requirements definition document
  - The team should submit a general overall project plan plus a detailed project plan for the first 2 sprints of project development. Checkpoint / happy hour will be scheduled every 2 weeks for sprint analysis and next sprint planning.
  - The report should include high-level milestones for each sprint. These milestones represent the most relevant requirements that the project need to achieve on each sprint.
  - Teams are strongly encouraged to use gantt charts and/or other project management tools for defining tasks and distributing time.

Sprint 1: App GUI Design and DBMS instance (weeks 3-4)

Due Date: March 1, 2019, 11:59 PM (Grade weight: 20%)

Minimal Deliverables:

- ER Mapping and working DBMS instance hosted somewhere on the Cloud
- App (client side) mock screenshots and/or wire frames with matching requirements and task descriptions
- Updated Gantt chart with percent completion for each task
- Detailed Project Development Plan for Sprint II
- Updated Master Project Development Plan

Sprint 2: Open (weeks 5-6)

Due Date: March 15, 2019, 11:59 PM (Grade weight: 20%)
Minimal Deliverables:

- Updated Gantt chart with percent completion for each sprint I-II tasks
- Overall project percent completion should be 50% or more
- Detailed Project Development Plan for Sprint III
- Updated Master Project Development Plan

Sprint 3: Open (weeks 7-8)

Due Date: March 29, 2019, 11:59 PM (Grade weight: 20%)
Minimal Deliverables:

- Updated Gantt chart with percent completion for each sprint I-III tasks
- Overall project percent completion should be 65% or more
- Detailed Project Development Plan for Sprint IV
- Updated general Project Development Plan

Sprint 4: Open (weeks 9-10)

Due Date: April 12, 2019, 11:59 PM (Grade weight: 20%)
Minimal Deliverables:

- Updated Gantt chart with percent completion for each sprint I-IV tasks
- Overall project percent completion should be 80
- Detailed Project Development Plan for Sprint V
- Updated general Project Development Plan

Sprint 5: Open (weeks 11-12)

Due Date: April 26, 2019, 11:59 PM (Grade weight: 20%)
Minimal Deliverables:

- Updated Gantt chart with 100% percent completion for each sprint I-V tasks
- Fully working project hosted on the Cloud

Intellectual Property Agreement

There is no reason why your class project cannot become an eventual commercial success. In fact we strongly encourage you to think and dream big. Therefore, all team members should agree **IN WRITING AND SIGN** a document describing how ownership of any intellectual property generated during the development of the project and/or future revenues generated by a business should be distributed among these team members. No project will be allowed to proceed until such an agreement is agreed and signed by all members of the team. You may keep your agreement confidential as long as you convince us that it exists in writing.

Evaluation:

- The work performed on each sprint will be evaluated independently.
- A working project at the end of the term is required in order to get any points earned during previous sprints.
- For each sprint, the PM is responsible of scheduling a checkpoint meeting no more than 2 days after the sprint due date. All team members should be prepared to attend this meeting.
- Evaluation will take strong consideration on individual commits performed during each sprint by different team members and how the team is performing according to the project master plan. Adequate progress must be shown on each sprint.

Development:

- We strongly encourage your team to use a UNIX (Linux or OSX) based environment for development. Windows users should consider installing a VM with VirtualBox or other virtualization software.
- ORMs are strictly prohibited. Manual SQL query development is required.
- Each team member should use the Github Education Pack to gain access to free development tools. Some of these are: DigitalOcean (DO) for cloud VMs, SendGrid for emails, and Bitnami for deployments. Also, its encouraged to use Heroku for easier setup and deployment.
- Each project should be publicly deployed to a staging environment (Heroku, DO, Amazon Web Services (AWS) or Microsoft Azure).

Report and Docs Submissions:

- All written documentation should be uploaded to the project Github Repo under a /Documents directory using Markdown files. PDFs, DOCs or any other format will NOT be accepted as reports.

Sprint Meetings Reports:

- On each checkpoint meeting the team is required to update the repository readme.md file with a sprint retrospective analysis answering the following questions:
  - What Worked Well?
  - What Didn't Worked Well?
  - What We Learned?
- After the Retrospective Analysis, the team needs to add the next sprint planning about what task/requirements would be completed on the next sprint.