X Web: Proposal Worksheet

Preston Lee

Contents

1 Abstract

X Web is a self-defined team project that uses the combined brain power of your team to define, design, build, test and demo a meaningful web application development project. This worksheet will take you through the initial team formation and project planning steps. **This a proposal that must be "approved" by the instructor prior to development.**

2 Build a team.

Self assemble yourselves into teams of your choosing. Team requirements:

1. Two people. *If the class has an odd number of people, only one team of three will be allowed.*
2. At least one person with previous "hands-on" database experience.
3. Team captain must be the person with least database experience.

2.1 Pick some cool names.

Every project needs a cool name, as well as a captain. Fill in the blanks:

Team Captain:
Project Code Name:

2.2 Gather member information.

Fill in the following table with your team information

Table 1: Team member information.

Number	Name	ASURite	Previous Database Course	Email
(e.g.)	Alice Anderson	aanderson	CST 123	aanderson
#1				
#2				
#3				

© Preston Lee 2011. All rights reserved.

3 Scope out the project.

3.1 Define the problem.

Define a one-paragraph problem statement that you will attempt to solve. In another paragraph, define the core use case you will be supporting. (Tip: Pick an *interesting* problem that you think you and your team can realistically solve in about two months time using.) Keep in mind that your final "production" deliverable is Open Source, and will be deployed to the web using the Rails webapp framework on top of the MySQL RDBMS.

3.2 Demonstrate your knowledge.

For this proposal to be approved, your project **must** prove your knowledge of course material by demonstrating *all* of the following:

Timeliness Real-world projects can't wait until the last minute, and neither should you. The development of your project should be paced throughout the time you are given, **not** crammed into the last week.

Technologies Your must use the Rails web application framework on top of the MySQL RDBMS.

Dynamic The application must be dynamic, and accept input from the user as required. "Static" sites are **not** ok.

© Preston Lee 2011. All rights reserved.

Visual Content must be displayed using HTML on at least Firefox 4. You may use HTML5, JavaScript and CSS if you so chose, but it is not required. (**Exception**: If your app is intended to provide “headless” web services–such as video encoding, number crunching etc.–you are not required to put an HTML “face” on your app.)

Output Programs must provide correct, real-time output based on the current database state.

Exception Handling Any/All user, disk, network etc. I/O must be “solid”: written with thorough exception handling practices to provide a reasonable level of application robustness. Use begin/else/recuse blocks and checked/unchecked exception handling where appropriate. (**Bonus**: Make use of a 3rd-party API. Maps, geolocation etc. is all fair game!)

Code Documentation Code with documentation on usage and programmer thinking is a magnitude more valuable than code without. All custom code must be thoroughly comment in JavaDoc format. (**Bonus**: Provide “howto” documentation on all reusable classes.)

Automated Test Cases Provide a suite of ”unit tests” that allow you to quickly run regression tests against your code base. You must also include “negative” test cases: code which intentionally calls functions with invalid input to verify that the code fails the way it is supposed to. For example, passing null, invalid numbers etc. to functions expecting “correct” input should do something reasonable. We will be using an automated test mechanism included with the Rails framework.

3.3 Project summary.

In 2-3 paragraphs, summarize the purpose, input, behavior, and expected output of your application. Get creative!

© Preston Lee 2011. All rights reserved.

3.4 Target user(s).

In 1-2 paragraphs, describe the intended users of your application. (If you're writing a game, for example, is it for kids or adults? ...fun or educational purposes? ...paying customers or free access?)

3.5 Milestones.

The team will need to hit all of the following milestones for full credit.

Thursday, February 10th Proposals due.

Thursday, March 10th Checkpoint. Your team (led by the captain) will walk me through your progress thus far. By this point I expect you to have a good start on the project objectives as well as a good working relationship (and process) with the team.

Thursday, March 31st Demo and presentation. Your team–lead by the captain–will deliver a well rehearsed project presentation and application demonstration in front of the class. Your application may be running on your local machine.

© Preston Lee 2011. All rights reserved.

4 Define success.

All team members will receive the same grade for the project. Additionally, the team captain will deliver a live 10-15 minute demo in front of the class prior to grading.

4.1 Grading breakdown.

Total points possible: 36. (Tip: This is a lot, so take the project seriously.)

20% In-class demo. *Be prepared.* Be professional. Quality of content, delivery and project demo will be considered. In other words, make sure your project doesn't "blow up" in front of the class! You will have projector access to show slides, code, and other additional resources you deem fit.

20% Peer review.

30% Given by instructor, based on the previously defined criteria.

30% Defined by your team, below.

4.2 Custom grading criteria.

Define at least **four** specific ways your project should be graded. (For example: *"Correctly parses uploaded spreadsheet"*, or *"Renders MineSweeper board correctly."*) These must be reasonably challenging criteria for a team of two people given over a month of development time. Trivial criteria such as *"Application accepts invalid user input without throwing an exception."* will be rejected.

© Preston Lee 2011. All rights reserved.

© Preston Lee 2011. All rights reserved.