Deliverable One

CSCI 401 Project

|  |  |
| --- | --- |
| Yueh-Hsun Lin (Nelson) | yuehhsul@usc.edu |
| Mian Lu | mianlu@usc.edu |
| Yining Huang | yininghu@usc.edu |
| James Tseng | tsengj@usc.edu |

**Deliverable 1**

**Date:** 31st Aug, 2017

**Team members:** James Tseng, Yueh-Hsun (Nelson) Lin, Yining Huang, Mian Lu

**Topic:** Scheduling and Project Timeline

**Meeting Schedule:**

Tuesday 8-10:30AM: Student developers and client meeting

Tuesday 8-10PM: Student developers meeting

Thursday 4:20-4:30PM: Student developers and professor meeting

More student’s meetings will be scheduled as necessary

**Development Framework:** Agile (Scrum)

**Past Timeline:**

|  |  |  |
| --- | --- | --- |
| **Date** | **Purpose** | **Description** |
| 8/25/17 | Team meeting | Initial discussion on project expectations and preparation of questions for the elicitation process of requirements. |
| 8/29/17 | Client meeting | Introduction of project and stakeholders and discussion on technologies that will be used for the projects. Got instructions on how to do initial setup |
| 8/29/17 | Team meeting | Dissemination of information, and division of labor |

**Tentative Timeline:**

|  |  |  |
| --- | --- | --- |
| **Date** | **Purpose** | **Description** |
| 8/31/17 | Professor meeting | Status update, presenting deliverable 1 and status reports, addressing any questions and concerns |
| 9/3/17 | Team meeting | Consolidation of proof of concept, and preparation for client meeting, this will be held online |
| 9/5/17 | Client meeting | Demonstration of proof of concept, discussion of ideas, and addressing potential problems |
| 9/5/17 | Team meeting | Division of labor, finalizing ideas and preparing to start coding |
| 9/7/17 | Professor meeting | Status update, and delivering status report |

**Additional comments:** This project does not currently have set deadlines and timelines in place. Therefore, we will work according to our client’s needs and then develop a comprehensive timeline when we get approval for our proof of concept.

**CSCI 401 Project Meeting Minutes**

Time: 2-3:10PM

Date: Aug 25th 2017

Location: JFF 325

Attendance: James Tseng, Yining Huang, Mian Lu, Nelson Lin

Agenda:

* JIRA setup
* Project research → LA County Assessor's office website
* Agile/Scrum
* Formulating questions
* Draft Ideas/Proposals
* Skills and Tools required

Summary:

What is their job?

Big Picture:

The Los Angeles County Assessor is an elected official governed by the California Constitution and the rules adopted by the State of California Board of Equalization. The Office of the Assessor is responsible for valuing more than 2.6 million secured and unsecured property assessments, valued at over $1.2 trillion, including approximately: 1.8 million single-family residences and condos; 249,000 commercial-industrial properties; 247,000 residential/rental income parcels; and 277,000 business equipment and fixture assessments.

Our responsibility:

Question we want to ask:

**From Professor Miller’s sheet:**

- Can you explain the project a little bit?

- Is there existing code that we will be modifying or extending?

- What language/platform/technologies are used?

- Is there a programming team with whom we should communicate?

- How much documentation is expected at the end of the project?

- Is there a specific software methodology that we should follow (i.e. scrum, XP, waterfall, spiral)?

- We have to provide something to you every two weeks as a requirement for the class. Would

you like more frequent status updates?

- Are there any specific design restrictions or decisions that we should know about?

- Is there a version control or code repository already set up?

- What features in the project are most important to you?

- What form of communication do you prefer?

- Can we set a time and location (in-person, Skype, phone, etc.) for our next meeting?

**Our own questions:**

**Database:**

* What and how many databases do you have?
* Do we need to create a new instance or build on our own?
* Are there any security issues with the raw data?
* Do you have a specific SQL database/platform you prefer?
* What are your scalability expectations? Input/Output/Access
* What data/types do you want to store?
* How long should the information be stored?
* Compatibility to other software? I.e. Excel, PDF, Web
* Permission/privilege to access/modify data?
* Which language do you prefer?

Webpage:

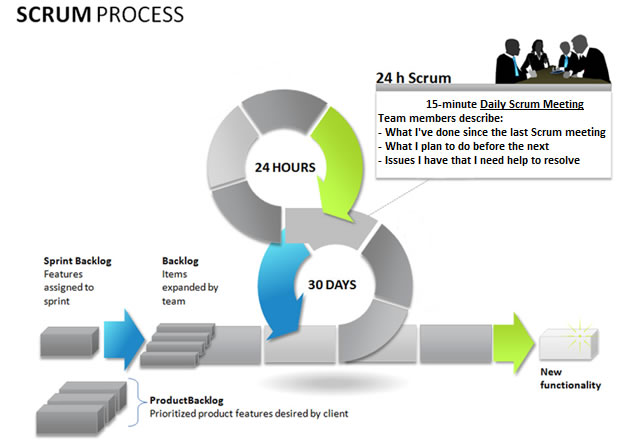
* Specific UI requirements
  + Color of buttons: borders?
  + Color of background
  + Font,size, color
* Login: manually selecting admin/user? Or automatically verifying
* Login credentials, database, APIs, etc.

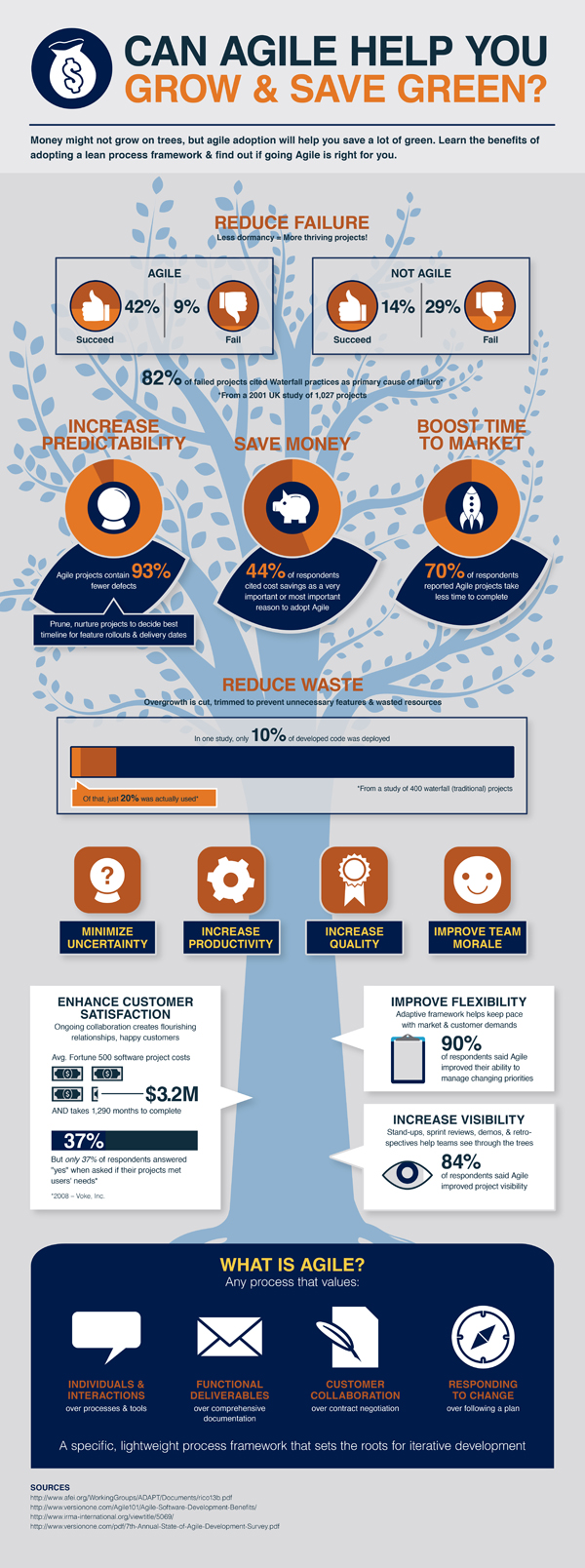
Engine:

* What language do you want

Suggested structure for project:

|  |  |  |
| --- | --- | --- |
| Back-End | Database | Front-End |
|  |  |  |





**CSCI 401 Project Meeting Minutes**

Time: 8:30-10:30PM

Date: Aug 29th 2017

Location: The Lorenzo

Attendance: James Tseng, Yining Huang, Mian Lu, Nelson Lin

Agenda:

* Summarization of today’s meeting
* Requirements for the project
* Proof of Concept
  + Data Structure/Primary Keys
  + ER Diagrams
  + Stored Variables
  + Extendibility
  + Normalization
  + Edge cases/possible problems with data
  + UI Functionality
* Division of Tasks (Due Sunday 6PM)

**Summarization:**

MOSCOW LIST

MUST HAVE

1. BOE DATA IMPORT
2. DESIGN DATA REPOSITORY
3. USER INTERFACE
   1. SCREEN TO VIEW PERSONAL TRAINING HISTORY
   2. SELF-REPORTING
   3. ADMIN SCREEN
4. REPORTING
   1. INDIVIDUAL TRAINING HISTORY
   2. AUTOMATED PDF GENERATION AND DISTRIBUTION
   3. USERS’ DASHBOARD: “QRYDATAENTRYDATABASE” QUERY
5. BROWSER COMPATIBILITY
   1. IE 10/11/CHROME/FIREFOX
   2. HTML5/CSS3

COULD HAVE

1. CARRY END CALCULATOR
2. VALIDATION

**Proof Of Concept**

**Data Structure/Primary Keys**

Employee ID and Certification No.

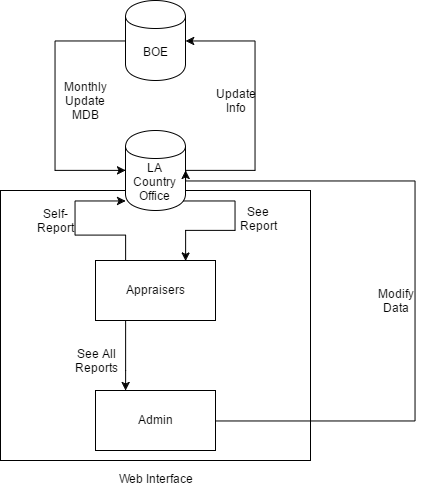
Problem we are currently facing:

* Appraisers all have Employee ID and Certification number, however, other employees may not have a certification number, therefore it can cause problems.
* Board of Equalization uses Certification number as their primary key

Possible Solutions:

* Hashing Employee ID and Certification number together to create a unique value
* Creating a lookup table to match employee ID and certification number, and the table will only be for appraisers/BOE data → Use Employee ID as primary key

**System architecture**

****

**Stored Variables**

BOE variables:

Annual Requirement:

* Last Name, First Name, Middle Name, Cert. No., County Code, County Name, Temporary Cert. Date, Perm. Cert. Date, ADV Cert. Date, Current Status, Status, Cert. Type, Fiscal Year, Earned Hours, Required Hours, Current year Balance, Prior year balance, Carry to year 1, Carry to year 2, Carry to year 3, Carry forward total

Details

* County Code, County Name, Certification No., Last Name, First Name, Middle Name, Fiscal Year, End Date, Course, Location, Grade, Hours Earned

Summary

* County Code, County Name, Fiscal Year, Last Name, First Name, Middle Name, Cert. No., Auditor, Carry Forward, Certificate Type

**Extendibility**

* Other employees (without certification number)
* Variables from LA county Assessor's office

**Normalization**

* Database normalization, or simply normalization, is the process of organizing the columns (attributes) and tables (relations) of a relational database to reduce data redundancy and improve data integrity. Normalization is also the process of simplifying the design of a database so that it achieves the optimal structure. It was first proposed by Edgar F. Codd, as an integral part of a relational model. (Wikipedia)

**Edge cases/possible problems with data and Validation**

* Bad Data (Typographical Errors)
* Duplication
* Input Field Standardization
* Employee ID doesn’t match name

**UI Functionality**

* Self-Reporting/Updating information such as classes, personal data

**Division of Tasks**

* Frontend: James Tseng and Nelson Lin
  + Frontend backend connection
  + Web UI design
  + Functionality
  + Sessions
  + PHP/CSS3/HTML5
* Backend: Mian Lu and Yining Huang
  + Frontend backend connection
  + Data Structure
  + Importing Data
  + ER Diagram
  + Microsoft Access/Microsoft SQL
  + Extendibility
  + Normalization