Sarah, Terry/Jerry, Shannon (same project)

Introduction

- Problem
- Motivation
- Solution

Use cases

Activity diagrams

Requirements

- Critical
    - Functional
        - Storage
        - Event reports
        - Check-in
    - Nonfunctional
        - Reliable
- Recommended
    - Functional
        - Ratings
        - Categories and filters
    - Nonfunctional
        - Load bearing
- Suggested
    - Functional
        - Regional reports
    - Nonfunctional
        - Web design

Architecture

- Data-centric architecture
- Processing done by server
- Simple query actions

Demo

- Very delicate (great ui)
- The location doesn't necessarily need to be a location on campus
- Create events from alumni side
- Create events from alumni office sidet

Technologies

- Front end
- Back end

Testing procedure

- White box
    - Ran thru each use case
    - Tested with simultaneous clients
- Server testing

Obstacles encountered

- Incompatibility between technologies
- Inexperience with technologies
    - Asynchronous requests
- Small bugs

Task remaining

- Full report creation
- Automatic alumni verification
    - Alumni office will go through document about alumni to verify
- Edit event information
- Black box testing

Lessons learned

- Research compatibility of technologies
- More detailed planning
- Follow the plan
    - Meet deadlines

Feedback: very good

Marko, Data, James (Alumni Business Directory System)
Introduction
- Directory of alumni business-owners
- The listings will be organized by way of various criteria
- The system will track users and create report for alumni office

Requirements
- Functional
    - Create and post business listing
    - Verification by alumni office
    - Search by attributes
    - Edit listings
    - Full reporting of data
- Recommended
    - Deletion of listing

Design structure
- Use cases
- User activity examples (is like activity diagram)

Technologies used
- Html
- Css
- Javascript
- Node.js
- Mongodb (will transition to mysql)

Web app demo: very simple
Design architecture & rationale
Testing procedure
- Compliance
    - Page loading
    - Registrations
    - Logins
    - Approvals and rejections
- To do
    - Unit testing
    - Documenting
    - Interface
    - Robustness

Obstacles encountered
- Database configuration
- New languages
- Geographically dispersed team
- Time constraints
- Bugs

Lessons learned

- Work allocation
- Time management
- Knowledge of technology
- SE is hard

Tasks to be completed

Casey, Paul, Eli (Bug Byte)
Objective
- System requirements
- Our design
- Rationale, demo, testing

Functional requirements
- The system will allow users to authenticate themselves by logging in with username and password
- The system will present a home page dashboard to the clients, developers, and managers
- The system will allow a manager to retrieve the bug report
- The system will let the manager assign the bug to developers
- The system will let the developer indicate the status of the bug to todo, in pro
- …….

Nonfunctional requirements
- The system should pass bug report from client to manager to developer quickly
- The system should provide a simple users interface with minimal pages

Security requirements
- ……..

Design constraints
- System must run on the design center
- ……

Activity diagrams
- Client
- Manager
- Developer

Use cases
Technologies used
- Languages
    - Sql
    - Php
    - Javascript
    - Html5
    - css3
- Frameworks
    - React

Architectural design
- Client server architecture

Design rationale
- Heavy client side processing
- Database component
- Familiarity

Project demo: good ui

Testing procedure

- Black box
    - Input and expected output
- Configuration testing

Obstacles encountered

- React set up in the design center
- Efficient workflow
- Coding standards/styles (collaborate with each other)

Lessons learned

- Split up the work evenly
    - Follow timeline closely
- Work within the constraints from the beginning
- Testing locally

Feedback:

- Requirement (no need to list all of the requirement, and just list bullet point, what they have is far too long)
- Activity diagram (something is not stated clearly enough)

Karen, Dominic, Manuel (bug)
Overview
Introduction
- Scope
- Current solution
- Our solution

Requirements
- Functional
- Nonfunctional

Design constraints
Use case diagram
Activity diagram
Architectural diagram
Technologies used
- Github
- Javascript
- Firebase
- Html

Test plan
- Testing we have done
- Testing needed to be done

Project demo
Obstacles encountered
- Making our software web-based
- Utilizing a database
- Creating separate pages for the user interface
- Choosing which programming language to use

Lessons learned
- Learned to work together
- Not everyone will be available at a given time
- Review each other’s work

Tasks left to complete

Feedback:
- It’s important to have improvement compared to the first demo
-