**Feasibility Study and Project Plan**

**Online Judge App**

**Version 4**

**Prepared by**

**Mohammed Abdullah Albukhari**

**Carlos A. Morales**

**Instructor:**Dr. Masoud Sadjadi

**Course:** CIS 4911

**Date:** January 2015

The MIT License (MIT)

Copyright (c) 2014 Florida International University

Permission is hereby granted, free of charge, to any person obtaining a copy  
of this software and associated documentation files (the "Software"), to deal  
in the Software without restriction, including without limitation the rights  
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell  
copies of the Software, and to permit persons to whom the Software is  
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in  
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR  
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,  
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE  
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER  
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,  
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN  
THE SOFTWARE.

Abstract

To optimize the evaluation process that takes place during the final rally of high-level projects, The Mobile Judge App has been created. Over a period of a few hours, the student will be judged by several judges. The classification process of various projects seems very inefficient, and judges Online App attempt to make the process easier to judge.

The three previous versions of the application have exceptional work performance of judges and the requirements of the students. However, since the application was used by the department of computer science. Based on user experience and current features some new functionalities have been requested. We strive to fulfill this gap by extending the current system, the desired functionality.

The new version will be adding new requested feature into the current system, as well as it will also fix some existing bugs those have been identified during user’s evaluation phase. This system will use the same kind of technologies that has been used before and will also add some new libraries wherever necessary.

Contents

[1. Introduction 4](#_Toc399712058)

[1.1 Problem Definition 4](#_Toc399712059)

[1.2 Background 4](#_Toc399712060)

[1.3 Definitions, Acronyms, and Abbreviations 5](#_Toc399712061)

[1.4 Overview of Document 5](#_Toc399712062)

[2. Feasibility Study 6](#_Toc399712063)

[2.1 Limitations and Constraints of the Current System 6](#_Toc399712064)

[2.2 Function of the New System 6](#_Toc399712065)

[2.3 High-level Definition of User Requirements 7](#_Toc399712066)

[2.4 Alternative Solutions 7](#_Toc399712067)

[2.4.1 Description of Alternatives 7](#_Toc399712068)

[2.4.2 Selection Criteria 8](#_Toc399712069)

[2.4.3 Analysis of Alternatives (refer to Appendix C – Feasibility Matrix) 8](#_Toc399712070)

[2.5 Recommendations 8](#_Toc399712071)

[3. Project Plan 9](#_Toc399712072)

[3.1 Project Organization 9](#_Toc399712073)

[3.1.1 Project Personnel Organization 9](#_Toc399712074)

[3.1.2 Hardware and Software Resources 9](#_Toc399712075)

[3.2 Work breakdown of Tasks, Milestones and Deliverables 11](#_Toc399712076)

[4. Appendix 13](#_Toc399712077)

[4.1 Appendix A – Project Schedule (Agile Approach) 13](#_Toc399712078)

# Introduction

## Problem Definition

It is always a difficult task for a professors and Instructor to grade exam or projects. If to the traditional burdens we add a live environment where multiple students have to be graded simultaneously in a timely manner then grading a senior project showcase is at minimum challenging that is why the Mobile Judge App was designed; to ease the judges’ task while keeping the excitement.

The first three versions of the Online Judge App accomplished this task beautifully but once the app has been used, it generates data, information that could be used but there’s no functionality to access it yet.

Version 4 of the Mobile Judge App focuses on the enhancement of the administrative aspects of the app. A new email module will be added to allow the creation and management of email template along with the capability to do bulk notifications that will access the historical data available in the application.

The application will allow administrators to switch terms thus allowing them to view different information pertaining to the selected term as well as execute some actions in each term that won’t affect the other terms and their data.

Administrators will now be able to accept or reject multiple grades simultaneously from a single screen and to get live statistics regarding the state of the grading by the judges. The current application platform will be expanded to support multiple roles for the same user and it will provide an interface for that user to switch between his/her roles.

## Background

The previous version of the app used Sencha Touch as the main framework for development of the mobile app. This approach allows targeting multiple mobile platforms with the same code base while losing some functionalities available in the traditional native apps. However the tradeoffs are worth it. The app is data driven for most use cases and the framework provides most of the desired functionalities so for this version we will continue to use **Sencha Touch** as the main development framework.

## Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| Task | A piece of job that serves as a unit of work. |
| App | A mobile application that can be installed on smartphones. |
| Alternative | A possible manner by which a given problem may be resolved. |
| PERT chart | Program Evaluation and Review Technique. A tool that is used to model the tasks and schedules associated with the completion of a project. |
| Feasibility study | Evaluating the best way for resolving a problem. |
| Sencha Touch | High-performance HTML5 mobile application framework |
| iOS | The mobile operating system offered by Apple |
| Android | The mobile operating system offered by Google |
| UML | Unified Modeling Language |
| DBMS | Relational Database Manage System |
| SQL | Structured Query Language |
| JS | Javascript Language |
| CSS | Cascading Style Sheet |

## Overview of Document

This document provides information about the feasibility study carried out before implementing the project, followed by the manner in which the project will be organized.

* Chapter 2 contains all the details associated with the feasibility study undertaken by the group.
* Chapter 3 contains the details regarding the project plan.
* Chapter 4 contains various appendices that aid in the understanding of this document.
* Chapter 5 contains references to the websites that were used in order to complete this document.

# Feasibility Study

## Limitations and Constraints of the Current System

The current version of the Online Judge app accomplishes the objectives of that version scope, but in general lacks functionality that the administrators using the app today would like to have in order to ease their jobs. In version 3 there is no mechanism to send emails to students or judges, only one role is linked to any given account, and administrators have no real-time visibility of the grades during the showcase. Also, the term is only limited to the current one in which it is being applied.

## Function of the New System

* The purpose of the new system, Mobile Judge Version 4, is to extend the previous version and provide the desired functionalities to the app administrators.
* A brand new multiple email sending module will be added that will allow filtering the historical data by students, judges, semesters and state of the judges’ responses and performing bulk email notifications.
* Keep historical record of email send successfully.
* Add or remove specific users from within application with chosen role
* Users will be able have multiple roles and switch amongst them. An administrator will have real-time feedback of the grades during the showcase with filter and graph capabilities.
* Implement a "view as" feature where the admin can view as if he was the student or judge he selects
* Add a new graph/chart which shows the average class question grade. A bar chart which has class average for Question 1, Question 2, etc
* A new feature which will alert the administrator if there is a grade submitted which is bellow a threshold (example a judge gave a student a 10/50)
* It is also going to have a brand new google authentication system as previous version has some problem with that.
* Provide information depending on which term it was initialized and only show the information when the term is selected.

## High-level Definition of User Requirements

* The user requirements for this system are very minimal. The user would be required to have either a smartphone or a tablet in order to use the application or any other device with web access where the user can access the application online.
* Second set of requirements is for each student is judge fairly and equally by using the pre-specified rubric.
* In order to assure privacy each student will only be able to view their scores after being graded, and each judge will be able to see only the student that they are assigned to.
* For security purposes both judges and students will need to register in order to have access to the application and the data available.
* The registration process will be standardized requesting basic information such as first name, last name, username, password and email address. The information gather at registration will be used for authentication when accessing one user’s specific information i.e. grades for a student.
* Fixed student’s login using Gmail (FIU email) authentication system.

## Alternative Solutions

### Description of Alternatives

* In alternative 1 to keep the Sencha Framework as the front-end interface and replace the backend implementation with a new one that will provide the desired functionalities.
* In alternative 2 to keep all existing frameworks and expand on them to integrate the new functionalities to the existing code base.

### Selection Criteria

The feasibility criteria that would be used consist of four categories: operational, technical, economic and schedule.

Operating feasibility includes functionality, meaning to what extent the admin would benefit from the email capabilities and how well the system would work. Technical feasibility includes the technology involved in the development of the functionality and the expertise in such technology required from the developer. Economic feasibility is the cost associated with the project. Lastly, the schedule feasibility is how long the system would take to be designed and implemented.

### Analysis of Alternatives (refer to Appendix C – Feasibility Matrix)

* Alternative 1 is less expensive than alternative 1, but given the close ties between the database schema and the User Interface on the current version, replacing the database schema would require to rewrite a great percentage of the existing functionalities.
* Alternative 2 uses as much of the existing code base as possible. This close relation between the old and new code can bring some complexities, however it allows for an incremental development where the system evolves slowly between stable states.

## Recommendations

The recommendation for this project is alternative 2. It provides the best compromise to get the desired functionalities without affecting or having little risk of affecting the existing ones. By reusing existing code and schemas, the design prioritizes the new functionalities over refactoring existing code.

# Project Plan

## Project Organization

The project has been divided into five different sections: the Email section, the Roles section, the View as section, the Grade Monitor section and the Stats section. Each section will be done one by one by the only developer will be in charge to develop the components needed in each tier to support the new functionality. In this way the one section would be responsible for the database changes, PHP API additions, and for the Sencha views and stores in the client code.

### Project Personnel Organization

There will be one people working on the project. So all the task will be done one by one based on priority.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Position | Periods required | Key Phases |
| Mohammed Abdullah Albukhari | Website/Mobile Developer | 08/25/14 to 12/12/14 | All |
| Carlos A. Morales | Website Developer & Database Manager | 09/15/14 to 1/15/15 | All |

### Hardware and Software Resources

In order to successfully complete this project the proper hardware and software are needed. Developer member will be using his own personal laptop, having one being a PC with windows and another using a MacBook. Each computer will be required to have a copy of Sencha Architect and any database software that supports SQL.

Hardware resources:

* 1 local server
* 1 development server
* 1 integration server
* 1 Android mobile phone
* 1 iOS mobile phone
* 1 Windows phone

Software resources:

* Sencha Architect, Sencha Cmd
* IIS or Apache
* MYSQL
* SourceTree
* Git Server

## Work breakdown of Tasks, Milestones and Deliverables

This semester the group used the agile methodology. This approach is an incremental software development technique that divides the project’s time into sprints where small features have to be developed, tested and showcased before the moving to the next sprint. The chart schedule can be seen in Appendix A.

Sprint 1:

* Requirements Definition
* Developers Roles
* Initial Mockups Design
* Environment Setup

Sprint 2:

* Multiple Email Static View
* Statistics Graph/Chart Static View
* Login Sequence Modification with Gmail
* View as for Admin Static View
* Automatic Grade Check Static View
* Unit Testing

Sprint 3:

* Multiple Email PHP API Development
* Statistics Graph/Chart View
* Statistics PHP API
* Automatic Grade Check PHP API
* Unit Testing

Sprint 4:

* Multiple Email Template Database
* Statistics Judge Graph/Chart
* View as for Admin PHP API
* Feature Integration
* Unit Testing

Sprint 5:

* Multiple Email Feature Final
* View As as Admin Feature Final
* Graph/ Chart Final
* Automatic Grade check Final
* Unit Testing Integration Testing
* Deployment

Sprint 6:

* Implement term switching for Students, Judges, Invitations, and Email Filtering
* Allow Students that are loaded in from the website to have appropriate initialized term
* Easy switching between terms
* Faster website loading

# Appendix

## 4.1 Appendix A – Project Schedule (Agile Approach)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Task Decription | **Duration** | **Start Date** | **Finish Date** |
| Sprint 1: |  | 4 weeks |  |  |
|  | ·         Requirements Definition |  | 9/2/2014 | 9/28/2014 |
|  | ·         Developers Roles |  | 9/2/2014 | 9/28/2014 |
|  | ·         Initial Mockups Design |  | 9/2/2014 | 9/28/2014 |
|  | ·         Environment Setup |  | 9/2/2014 | 9/28/2014 |
| Sprint 2: |  | 2 weeks |  |  |
|  | ·         Multiple Email Static View |  | 9/29/2014 | 10/12/2014 |
|  | ·         Statistics Graph/Chart Static View |  | 9/29/2014 | 10/12/2014 |
|  | ·         Login Sequence Modification with Gmail |  | 9/29/2014 | 10/12/2014 |
|  | ·         View as for Admin Static View |  | 9/29/2014 | 10/12/2014 |
|  | ·         Automatic Grade Check Static View |  | 9/29/2014 | 10/12/2014 |
|  | ·         Unit Testing |  | 9/29/2014 | 10/12/2014 |
| Sprint 3: |  | 2 weeks |  |  |
|  | ·         Multiple Email PHP API Development |  | 10/13/2014 | 10/27/2014 |
|  | ·         Statistics Graph/Chart View |  | 10/13/2014 | 10/27/2014 |
|  | ·         Statistics PHP API |  | 10/13/2014 | 10/27/2014 |
|  | ·         Automatic Grade Check PHP API |  | 10/13/2014 | 10/27/2014 |
|  | ·         Unit Testing |  | 10/13/2014 | 10/27/2014 |
| Sprint 4: |  | 3 weeks |  |  |
|  | ·         Multiple Email Template Database |  | 10/28/2014 | 11/19/2014 |
|  | ·         Statistics Judge Graph/Chart |  | 10/28/2014 | 11/19/2014 |
|  | ·         View as for Admin PHP API |  | 10/28/2014 | 11/19/2014 |
|  | ·         Feature Integration |  | 10/28/2014 | 11/19/2014 |
|  | ·         Unit Testing |  | 10/28/2014 | 11/19/2014 |
| Sprint 5: |  | 2 weeks |  |  |
|  | ·         Multiple Email Feature Final |  | 11/20/2014 | 12/1/2014 |
|  | ·         View As as Admin Feature Final |  | 11/20/2014 | 12/1/2014 |
|  | ·         Graph/ Chart Final |  | 11/20/2014 | 12/1/2014 |
|  | ·         Automatic Grade check Final |  | 11/20/2014 | 12/1/2014 |
|  | ·         Unit Testing Integration Testing |  | 11/20/2014 | 12/1/2014 |
|  | ·         Deployment |  | 11/20/2014 | 12/1/2014 |
| Sprint 6 |  | 4 weeks |  |  |
|  | Implement Student Load, Home tab number correction |  | 12/14/2014 | 12/21/2014 |
|  | Change Term when Selected, Show Available Terms |  | 12/21/2014 | 12/28/2014 |
|  | Settings, Invitations, and Grades term switch |  | 12/28/2014 | 1/4/2015 |
|  | Email Filtering shows data from term selected in settings |  | 1/4/2015 | 1/11/2015 |
|  | Faster Website and Documentation |  | 1/11/2015 | 1/15/2014 |

4.2 Appendix B – Feasibility Matrix

|  |  |  |  |
| --- | --- | --- | --- |
| Feasibility Criteria | Weight | Alternative 1 | Alternative 2 |
| Operational Feasibility | 30% |  |  |
| Functionality: To what degree does the candidate solution benefit? |  | Fully supports required  Functionality. | Fully supports required  Functionality. |
|  |  | Score: 100 | Score: 100 |
| Technical Feasibility | 30% |  |  |
| Technology: An assessment of the maturity, availability, ability to acquire, and desirability of computer technology needed to support this candidate. |  | All smartphones platforms would be supported | All smartphones platforms would be supported |
| Expertise: An assessment of the technical expertise needed to develop, operate, and maintain the candidate system. |  | Extensive knowledge on database design and medium level of Sencha Touch is required | Medium level of expertise in database and Sencha Touch are required |
|  |  | Score: 75 | Score: 90 |
| Economic Feasibility | 30% |  |  |
| Cost to Develop |  | Database Hosting: Self  Maintained/School's  Resources - 0 | Database Hosting: Self Maintained/ School's  Resources - 0 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| Payback |  | Same as 1. | Same as 1. |
| Net Present Value |  | NA | NA |
|  |  | Score: 100 | Score: 100 |
| Schedule Feasibility |  |  |  |
| Assessment of how long the solution will take to design and implement. | 10% | 4 -5 months | 3 to 4. |
|  |  | Score: 85 | Score: 90 |
| Ranking |  | 83.5 | 93 |