

---

# DEBATE APPLICATION REQUIREMENT SPECIFICATIONS

---

A BRIEF SPECIFICATION OF THE REQUIREMENTS, DESIGN, AND  
IMPLEMENTATION PLAN FOR A SOFTWARE ENGINEERING CLASS PROJECT

DEVELOPED BY

JENNY LI  
JOSHUA MUIR  
LIAM O'BRIEN

OWNED BY

DR. ALEX ARAVIND

*The University of  
Northern British Columbia*



FALL 2019

# Table of Contents

<b>1 Introduction</b>	<b>2</b>
<b>2 Requirements</b>	<b>3</b>
2.1 Server	3
2.2 User Interface	3
2.3 Functionality	3
2.4 Security	4
<b>3 Workflow Design</b>	<b>5</b>
<b>4 Implementation Plan</b>	<b>6</b>
4.1 Work Distribution & Task Assignment	6
4.2 Timeline	6

## **1 INTRODUCTION**

This document is a tentative requirement specification and implementation plan for our term project given to us by Dr. Aravind. Our task for this project was to create an application that would assist in high school-level judging debate competitions. The application would automate many of the processes that are currently being done manually, with pen and paper. The application would offer interactive score sheets, automatic total score calculation, and generate detailed feedback reports for each student, as well as a total ranking for teams and individual students.

## 2 REQUIREMENTS

### 2.1 SERVER

Our project will be implemented as a web application, with the use of HTML, CSS, and Javascript. We will therefore require a server to host our application. Our app will be hosted on a server owned by UNBC, courtesy of Dr. Aravind.

### 2.2 USER INTERFACE

The end goal of the project is to create an application that can be used by schools and the debate association to judge debate competitions between high school students. The judges are typically volunteers, and will likely have limited technical backgrounds. Our application will therefore require a user-friendly interface and intuitive navigation.

We would like our application to be accessible by laptop and by mobile devices (smartphones, tablets etc.). We will therefore be designing both a horizontal layout for laptop/desktop use, and a vertical layout for mobile use.

### 2.3 FUNCTIONALITY

The application will have several back-end processes to serve different purposes such as

- **Pairing teams by score brackets** - teams will be paired manually at registration, and then with each proceeding round, they will be paired with a team of equivalent score.
- **Totaling scores** - this will be done automatically after each score sheet entry, and a total score per team will be kept for overall ranking.
- **Generating individual feedback reports** - students will have a detailed report with their score in each category, for each round of debate they participated in, as well as any comments from the judges and their total score.
- **Ranking** - individual and team ranking will be implemented as per request of the product owner, so that students can see how they performed with respect to their competitors
- **Registration** - this will be done in-app by the debate coordinators. Each team will need to register with their school ID and member

names, and they will be assigned a unique team ID.

## **2.4 SECURITY**

This application is intended for use by the school district and the debate organization. Access will need to be restricted from the general public by restricting access to the server.

We will consider also having different levels of access, to allow a read-only version of the application for spectators and students to see the overall ranking and team matching for the next round.

Each competition will have a key to login to the event. The key will be generated at creation.

### **3 WORKFLOW DESIGN**

See DebateFlow.png.

## 4 IMPLEMENTATION PLAN

### 4.1 WORK DISTRIBUTION & TASK ASSIGNMENT

Our project responsibilities during the prototype period will be distributed as so:

- **Jenny** - design, documentation, prototype development (CSS, "look and feel").
- **Josh** - GUI wireframe development (HTML, basic components), and later server integration.
- **Liam** - backend development, functionality implementation (Javascript).

Once we have a working prototype that has been approved by the project owner, everyone will take part in full stack app development, each working in different areas and pages of the app. These different components include: the registration/home page, the interactive score sheet, ranking screen, and the individual reports. A final breakdown of the work distribution will be documented in the final solution document.

### 4.2 TIMELINE

- Week of Oct. 14th - Design and initial documentation done.
- Week of Oct. 21st - GUI "exoskeleton", or wireframe implemented.
- Week of Nov. 11th - Working prototype finished - all local app functionality implemented.
- Week of Nov. 25th - Server integration completed.
- Dec. 5th: Final product completed.