**Software Quality Plan**

**Section 1: Product Information**

**Description:**

A web application that is intended to be implemented into an audio competition organizations existing website. Used by judges at a car audio competition event to streamline the process of recording and sorting the data from each competitor. The application is designed to take most of the manual work out of tracking the names, scores, and classes of each competitor so the judge can spend more time running the meter. This can lead to a much more efficient process which will get the competitors in and out of the competition lanes much faster.

There are a total of nine (9) pages to be implemented into an organization’s website; The ‘main’ screen, which shows a table of all previous competitions in order by date, and a button to add a new competition. The ‘input’ screen is where you input your new event and all subsequent scores during said event. The ‘summary’ screen is an overview of the current event with a table showing all scores with buttons to sort by name, class, or score. The ‘standings’ page is used for the secondary display at the audio event, this display faces the crowd and cycles through the scores for each class in the current event. The ‘winners’ screen sorts and displays the winners of each class for the judge to give out awards at the end of an event. Once an event is complete, the judge then takes the text file generated by the application and uploads it to the organization's website using the ‘upload’ page. Upon uploading the file the application will add the file to an archive and sort/update the database with the new scores and events. The ‘backup’ page is where these archived files go for auditing at the end of the audio season. All users can use the ‘lookup’ page to search for event information from any event in the archive. The ‘tutorial’ page is intended for judges to learn how to use this application in a simple step by step process.

**Intended Market:**

Audio organizations that hold events where audio cars are acoustically measured to see which is the loudest in a specific category.

**Quality Expectations:**

Depending on how big a specific audio event is, there may be a rush to get everyone through the competition lanes. So the application should not throw any fatal errors, and should be simple enough that non tech-savvy people can input values properly 95% of the time. Uptime of the application is dependent on the uptime of the organization's web platform.

**Section 3: Process Descriptions**

**Process Descriptions:**

Process descriptions are an integral part of this project. Each process must be followed to ensure that all pieces of the program are being created at the same level of quality. This also helps us manage all requests and changes to the program.

One of the most important processes will be the [change request process](https://drive.google.com/file/d/1TbK4bGcyZEaI6EAGoXhhUxn6zjKr6Cu8/view). Change requests must be submitted on the proper, formal change request form and submitted to the team for review. Every change request must be properly reviewed by the team before moving forward or throwing the request out. If the change request is accepted, it will be ranked by priority. High priority change requests will be taken care of as soon as possible, and pushed out in an immediate update. This includes issues such as security flaws. Medium and low priority change requests will go out in the next scheduled update, and will include issues such as coding errors, spelling errors, and customer suggestions. Bad change requests such as employment requests and spam will be immediately rejected and not included in the updates. When change requests are accepted, they must be documented in the change management log by a member of the review team.

[Version release management](https://drive.google.com/drive/folders/12dkQENh8G_g8069vySzE2yprxMBM8jAh) will also be important for this application. There are 5 main stages of this process: plan release, build release, test user acceptance, prepare release, and deploy release. In the first step, Plan Release, the release must be structured from start to finish and documented clearly in order to set expectations. After expectations are laid out, the Build Release stage starts. This includes designing and building the needed software for the release. Next, the built software will go through User Acceptance Testing in order to sort out bugs and other issues. All bugs and other issues must be documented, and then fixed by returning back to the Build Release stage. Once the acceptance testing stage is completed, there is the Prepare Release stage. This stage exists as an opportunity to put finishing touches onto the product. This will also be the stage that the product will be tested against minimum acceptable standards and business requirements outlined in the Release Plan. The last stage is Deploying Release. Deployment needs to include public information on why the new version is being pushed out, and how it will affect app use.

**Section 4: Quality Goals**

SoundScorez seeks to deliver high quality usability of the software application by vigorously testing its functional processes and system attributes.

|  |  |
| --- | --- |
| **Quality Attribute or Goal** | **Evaluation** |
| System Usability | A series of moderated usability tests.  Participants are required to navigate through the interface to complete several tasks  Observational focus:   * Task success * Completion speed * Body Language * Common errors |
| Responsive Design & Portability | Test application accessibility over a wide variety of devices (personal computer, mobile, etc) running various operating systems (macOS, Windows, iOS, Android, etc.)   * Proper execution * Displays properly |
| Functionality of Software Components | Automated unit tests |
| Software Testing of Overall System | Peer review   * Functional Testing * Performance Testing * Use Case Testing * Security Testing |
| Resilience of Overall System | Performance testing   * Load testing * Stress testing * Endurance testing * Spike testing |
| Adherence to Our Group’s Quality Standards | Provide and review project documents, ensure documents encapsulate the entirety of the project (processes and procedures) and conform to our group’s standards |

**Section 5: Risks and Risk Management**

|  |  |
| --- | --- |
| **Risks** | **Mitigation** |
| Incoherent coding styles | Written set of rules regarding programming style and formatting |
| Lack of pre-developed programs - Ground of build style does not come with any proven-working programs | Each object should be made to be compatible and flexible with other parts of the program so they are able to be implemented elsewhere within the program |
| Lack of communication with end-client | Regular updates sent to client |
| Implementation trouble with TermLab | Make sure the program is able to decipher and present the information exported from the TermLab program. Communicate issues with TermLab team for guidance |