

Department of Computer Science & Engineering  
University of Nevada, Reno

ARIA: Administration, Registration, and Information Assistant

K.R.E.W.  
Renee Inuma  
Wesley Kepke  
Ernest Landrito  
Kyle Lee

Instructor: Dr. Sergiu Dascalu  
Advisor: Dr. Frederick Harris (UNR CSE)  
External Advisor: Cindy Harris (NNMTA)  
March 3, 2016

## **Abstract**

The goal of ARIA is to create a WordPress plugin that will assist the Northern Nevada Music Teachers Association (NNMTA) with the registration, scheduling, and organization of their music competitions. This project is meaningful because it will significantly reduce the amount of overhead that NNMTA personnel spend in regards to the preparation of their music competitions. Moreover, this plugin will automate tasks and alleviate errors that would otherwise take NNMTA members hours to complete and validate. Once implemented, the NNMTA will have a robust and systematic manner of hosting music competition for years to come. Furthermore, ARIA will be constructed in a generalizable format such that the plugin can be used in situations other than music competitions, thus exposing ARIA's functionality to other sectors.

## **Requirements**

### **Requirements Workshop Summary**

A requirements workshop for ARIA's specification took place with Dr. Harris (UNR CSE), Mrs. Cindy Harris (NNMTA), and all of ARIA's developers on December 21, 2015 at the Harris residence. During this requirements workshop, which lasted approximately one hour, the developers listened intently to the specifications outlined by Mrs. Harris and provided feedback on what seemed achievable given the required completion date. Since Dr. Harris is serving as the technical advisor for ARIA, he also added input regarding the development aspects of ARIA.

Having all six affiliated members at the brainstorming session allowed for many preliminary requirement ambiguities to be resolved and also solidified the requirements desired by the NNMTA. For example, some of the ideas brought up during the brainstorming session were what the required fields for student and teacher registrations were going to be, the functional requirements for the chairman, the requirements necessary for scheduling, and the competition documents that needed to be created.

During the analysis session, the team worked to organize and prioritize the requirements discussed during the brainstorming session. Some of the requirements suggested during the brainstorming session were redundant or very similar, so they were combined or eliminated. For example, it is necessary to upload multiple documents, such as music data or teacher information, to the NNMTA server. These

were initially separated into multiple requirements, but the team combined them into one "file upload" requirement. Additionally, the team began prioritizing requirements based on their necessity and deadlines. For example, the lower division music competition will take place in March, so completing tasks for this competition had the highest priority. Mrs. Harris was most concerned about the process of registration and automating the scheduling for this competition.

The most important finding during this requirements workshop was the shift from a focus on the entire website to a more in-depth service for music festival registration and management. Additionally, other important findings included the need to automate as much of the process as possible, the need to use the Gravity Forms plugin to integrate with ARIA, and the restraints on scheduling for the competitions.

### Software Requirements

	Type	Requirement
	Non-functional (NR_01)	<ul style="list-style-type: none"> <li>● Compatible with all major browsers:               <ul style="list-style-type: none"> <li>○ Since a large volume of users is anticipated, ARIA will need to be compatible with the spectrum of browsers that are being utilized by users. While ARIA’s developers do not anticipate this being a major issue, it is indeed a non-functional requirement that must be adhered to.</li> </ul> </li> </ul>
	Non-functional (NR_02)	<ul style="list-style-type: none"> <li>● Developed on the WordPress platform:               <ul style="list-style-type: none"> <li>○ The NNMTA has requested to remain on WordPress; therefore ARIA will be developed using the PHP scripting language (WordPress requires PHP). Additionally, some functionality of ARIA requires client-side processing. To accomplish this, JavaScript will be used to make backend requests and retrieve necessary data.</li> </ul> </li> </ul>
	Non-functional (NR_03)	<ul style="list-style-type: none"> <li>● Reasonable response time:               <ul style="list-style-type: none"> <li>○ All operations completed by ARIA should be achieved with a reasonable response time, especially the student and teacher online registration forms.</li> </ul> </li> </ul>
	Non-functional (NR_04)	<ul style="list-style-type: none"> <li>● Utilization of Gravity Forms WordPress plugin:               <ul style="list-style-type: none"> <li>○ Creation of registration forms must use the Gravity Forms plugin for form and field creation.</li> </ul> </li> </ul>

	Type	Requirement
L E V E L 1	Functional (FR_01)	<ul style="list-style-type: none"> <li>● Music uploading/downloading: <ul style="list-style-type: none"> <li>○ ARIA will allow the festival chairman to upload and download NNMTA music. To upload music, the festival chairman will upload a CSV file containing all NNMTA music. ARIA will then process this data and add it to a centralized music database that can be accessed from many points throughout the plugin. If the festival chairman needs a CSV file of all the music, ARIA will generate this file for him/her. Also, if the festival chairman needs to modify the current selection of music, all he/she needs to do is download the CSV file, make changes to this file, and reupload it.</li> </ul> </li> </ul>
	Functional (FR_02)	<ul style="list-style-type: none"> <li>● Student and teacher registration: <ul style="list-style-type: none"> <li>○ ARIA will allow both students and teachers to register for a competition. When a festival chairman creates a competition, two new registration forms will be created, one for the students and another for the teachers. The students will begin by entering their data for a music competition. Once this information is submitted, the teacher will receive, via email, a link to a form that has many of the fields already filled out. The data that is filled out will be specific to a student. The teacher then completes a student's registration by adding some additional information, such as what piece the student will be performing and under what skill level the student is categorized as. The teacher will then submit this form and a student will have been successfully registered for a competition.</li> </ul> </li> </ul>
	Functional (FR_03)	<ul style="list-style-type: none"> <li>● Scheduling of students and teachers: <ul style="list-style-type: none"> <li>○ ARIA will perform scheduling of all students on a first-come, first-served basis. Moreover, ARIA will schedule students according to the criteria that the festival chairman defined when creating a competition, such as the amount of students that are allowed per time slot of judging and the duration of the aforementioned time slots. Scheduling is currently one of the most cumbersome activities that NNMTA personnel</li> </ul> </li> </ul>

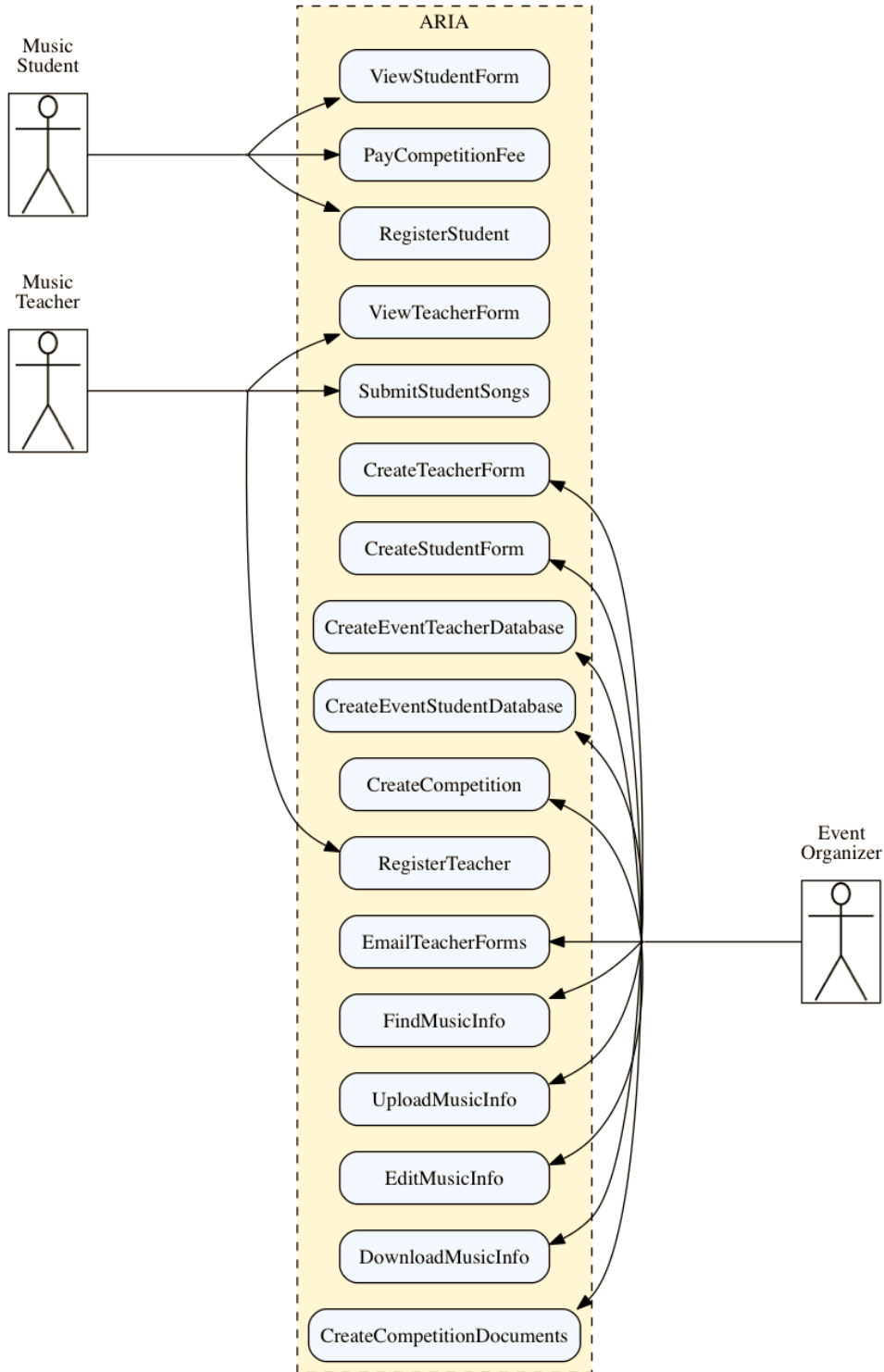
		<p>must undergo; therefore, accomplishing this functional requirement is imperative.</p>
	Functional (FR_04)	<ul style="list-style-type: none"> <li>● Production of competition documents: <ul style="list-style-type: none"> <li>○ ARIA will produce competition documents, such as which students and teachers are allocated to a competition room at a designated time. This functional requirement is an extension of the scheduling functional requirement and is yet another tedious activity that is currently bothersome to the NNMTA. Generation of these documents will assuredly mitigate problems that occur on competition days.</li> </ul> </li> </ul>
L E V E L  2	Functional (FR_04)	<ul style="list-style-type: none"> <li>● Generation of competition statistics: <ul style="list-style-type: none"> <li>○ ARIA should generate statistics information for competitions. This would include, for example, the number of students that are participating under a particular skill level, the amount of teachers that volunteered, and even the amount of superior ratings achieved by students. This information is somewhat utilized for future competitions so producing easy-to-read documents would be of great use to the NNMTA.</li> </ul> </li> </ul>
	Functional (FR_05)	<ul style="list-style-type: none"> <li>● Support for multiple competitions: <ul style="list-style-type: none"> <li>○ ARIA should be capable of supporting multiple competitions that are occurring simultaneously. That is, students and teachers should be able to sign up for a specific competition in which the dates of the competition overlap with another competition and not have the registration data mixed.</li> </ul> </li> </ul>
L E V E L  3	Functional (FR_06)	<ul style="list-style-type: none"> <li>● Optimize database queries for music uploading: <ul style="list-style-type: none"> <li>● When the festival chairman for a competition needs to upload music, they must wait for a considerable amount of time because there is 1183 songs that need to be uploaded to the central music database. Currently, ARIA will process all 1183 songs from a CSV file and will perform 1183 SQL queries. This is a brute-force approach, but modifying the SQL query code would require modification to another plugin, called GravityForms, that ARIA relies on.</li> </ul> </li> </ul>

	Functional (FR_07)	<ul style="list-style-type: none"><li>● Interact with additional APIs to enhance features:<ul style="list-style-type: none"><li>○ There are several areas in the plugin that could presumably be enhanced by integrating third-party APIs. For example, the Google Maps API could be used in conjunction with the competition location so that students/teachers could have a visual representation of where a competition is taking place.</li></ul></li></ul>
--	-----------------------	---

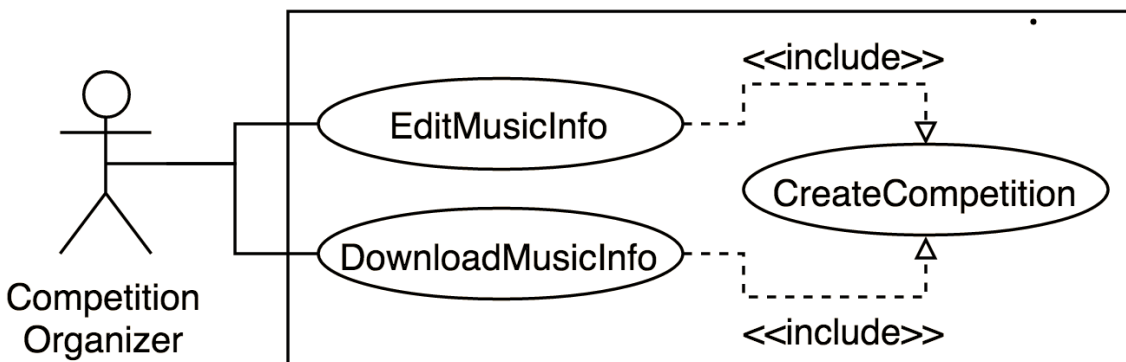
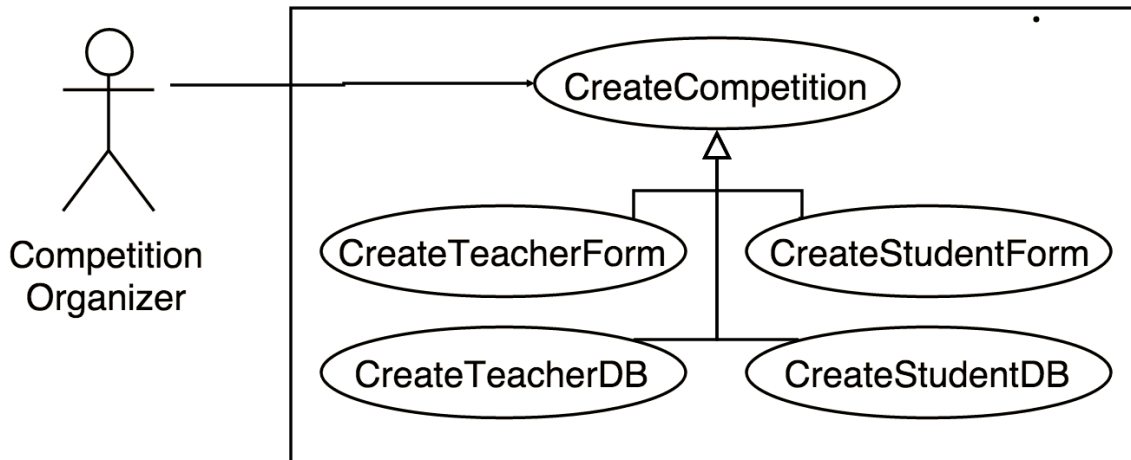
Note: Level 1 Priority requirements correspond to the requirements which the team will complete by April 2016. Level 2 Priority requirements are requirements which the team will attempt to implement if time permits. Level 3 requirements will likely not be implemented by the end of the Spring 2016 semester, but would be a good advancement for the project.

# Use Case Modeling

## Use Case Diagram



## Advanced Use Case Diagrams



## Use Case Descriptions

Note: Use cases displayed in *italics* are detailed in the Detailed Use Case section.

- *ViewStudentForm* (UC\_01)
  - Students will be able to view the competition form that will be used to register them for a specific competition.
- *PayCompetitionFee* (UC\_02)
  - Students will be able to pay for the competitions that they register for. The payment amount will differ based on the level of competition.
- *RegisterStudent* (UC\_03)
  - Students will have the opportunity to register for multiple competitions. Each competition has a different registration form so students can register for as many active competitions as they wish.
- *ViewTeacherForm* (UC\_04)



- Music teachers will be able to view the competition form which will be used to register them for a specific competition and will contain information about their students.
- SubmitStudentSongs (UC\_05)
  - Music teachers will be able to select the songs which their students will be performing for a specific competition. This involves selecting the song, composer, and period of a given performance piece.
- CreateTeacherForm (UC\_06)
  - The event organizer will be able to create the form which will be submitted by teachers to register for a specific competition. This form will contain specific fields corresponding to the teacher's information.
- CreateStudentForm (UC\_07)
  - The event organizer will be able to create the form which will be published to the website for students to register for a specific competition. This form will contain specific fields corresponding to the student's information.
- CreateEventTeacherDatabase (UC\_08)
  - The event organizer will be able to create the database which will contain all of the information corresponding to a specific teacher, such as their name or email, which will be populated upon form submission.
- CreateEventStudentDatabase (UC\_09)
  - The event organizer will be able to create the database which will contain all of the information corresponding to a specific student, such as their name or email, which will be populated upon form submission.
- *CreateCompetition* (UC\_10)
  - The festival chairman will be able to dynamically create a competition by filling in the necessary details, such as competition dates and configuration options.
- RegisterTeacher (UC\_11)
  - The teacher will be able to register for a competition by filling out personal information, as well as specify song selection information for his or her students.
- EmailTeacherForms (UC\_12)
  - As soon as the students register for the competition, it will email his or her music teacher letting them know that the student has completed his or her part of the registration process
- FindMusicInfo (UC\_13)
  - The event organizer will be able to search through their uploaded music information in order to find specific music pieces.
- *UploadMusicInfo* (UC\_14)

- The event organizer will be able to upload a file with all of the info about possible song selections for a specific music competition. This will be in the form of a CSV file.
- EditMusicInfo (UC\_15)
  - In collaboration with UC\_15, the festival chairman will have the ability to download the NNMTA CSV music file. The festival chairman can then edit this CSV file and re-upload the file. This will effectively update all of the NNMTA music.
- DownloadMusicInfo (UC\_16)
  - The festival chairman will have the ability to download a CSV file that contains all of the NNMTA music.
- CreateCompetitionDocuments (UC\_17)
  - The festival chairman will have the ability to generate all of the competition documents for a specific music competition. These can be judging forms, student schedule forms, and volunteer forms.

## Detailed Use Cases

<b>Use Case: ViewTeacherForm</b>	
Use Case ID	UC_o4
Actor	Teacher
Precondition(s)	Student has registered for the event.
Flow of Events	<ol style="list-style-type: none"> <li>1. The use case starts when an actor is emailed a link in order to specify student song information.</li> <li>2. The page will allow the form to render if the link that was provided is correct.</li> <li>3. If the actor has registered a student beforehand the form is prepopulated with their user information.</li> <li>4. Else the fields will not be populated and the form will be rendered.</li> </ol>
Postcondition(s)	The form for teacher registration will be rendered to the screen.

<b>Use Case: UploadMusicInfo</b>	
Use Case ID	UC_14
Actor	Organizer
Precondition(s)	<ol style="list-style-type: none"> <li>1. The Organizer is logged into wordpress.</li> </ol>
Flow of Events	<ol style="list-style-type: none"> <li>1. The actor will be prompted to select a csv containing music information.</li> <li>2. The actor will specify a csv to upload.</li> <li>3. The actor will be presented with a status bar presenting how many songs have been uploaded.</li> <li>4. The actor will be presented with confirmation that the music is uploaded.</li> </ol>
Postcondition(s)	<ol style="list-style-type: none"> <li>1. The information about music will be saved onto the database.</li> </ol>

<b>Use Case: PayCompetitionFee</b>	
Use Case ID	UC_o2
Actor	Students
Precondition(s)	1. Student Form filled out and validated.
Flow of Events	<ol style="list-style-type: none"> <li>1. The actor will input the type of their credit card.</li> <li>2. The actor will input the credit card number and expiration date.</li> <li>3. The actor will input their billing address.</li> <li>4. The actor will submit the request for payment.</li> <li>5. The actor will be notified of the status of the payment.</li> </ol>
Postcondition(s)	<ol style="list-style-type: none"> <li>1. The registration will be paid for.</li> <li>2. Money will be transferred from the Actor to the NNMTA.</li> </ol>

<b>Use Case: CreateCompetition</b>	
Use Case ID	UC_10
Actor	Competition Organizer
Precondition(s)	1. Logged in
Flow of Events	<ol style="list-style-type: none"> <li>1. The competition organizer will specify the name of the event.</li> <li>2. The competition organizer will specify the date of the event.</li> <li>3. The competition organizer will specify any divisions of the competition.</li> <li>4. The competition organizer will specify the available rooms for the competition.</li> <li>5. The competition organizer will specify the approximate time for each performance.</li> <li>6. The competition organizer will denote the description of the event.</li> <li>7. The competition organizer will submit the competition.</li> </ol>
Postcondition(s)	<ol style="list-style-type: none"> <li>1. The student and teacher forms will be created for the event.</li> <li>2. Student and teacher databases will be created for the event.</li> </ol>

## Requirement Traceability Matrix

	FR_01	FR_02	FR_03	FR_04	FR_05	FR_06	FR_07
UC_01							
UC_02							
UC_03							
UC_04							
UC_05							
UC_06							
UC_07							
UC_08							
UC_09							
UC_10							
UC_11							
UC_12							
UC_13							
UC_14							
UC_15							
UC_16							
UC_17							

	Represents a function requirement that <i>may</i> be needed for use case.
	Represents a function requirement that <i>must</i> be needed for use case.

## Glossary of Terms

- Aria
  - A long song accompanying a solo voice usually in an opera. Our project is also called ARIA which stands for Administration, Registration, and Information Assistant.
- Baroque
  - A time period in Europe during 17th and 18th century that followed mannerism. Composers during this time period include Bach, Handel, and Vivaldi. This is a category to select from in the piano competition.
- Classical
  - A time period from 1750-1820 between Baroque and Romantic time periods. This is a time period to select from in the piano competition.
- Competitive / Non-competitive
  - The competition has both competitive and non-competitive formats. Competitive format means that the student will be given a rating, while the non-competitive format offers comments and critiques from the judges with no rating.
- Command Performance
  - A performance where the contestants that were rated highest in their division are allowed to perform in front of all the competitors and spectators.
- Composer
  - The person who writes music. The creator of a specific piece of music.
- Contemporary
  - The modern time period of music. This is the category for all recent compositions for students to choose songs from.
- Festival Syllabus
  - The syllabus developed by the NNMTA which contains information about a specific event including performance schedules, guest biographies, organization information, and sponsors.
- Impressionistic
  - The composers from the Impressionistic era spans from 1900-1940 and consists of composers such as Debussy, Ravel, and Griffes. This is a category to select from in the piano competition.
- Judges
  - A group of people that will be ranking the competitors by their skill level and difficulty of music. The judges for the NNMTA are composed of music teachers and college piano professors.

- Key
  - A group of notes upon which the scale is based or the specific pattern of notes that governs a song
- Lower-level
  - The Lower Level Festival is held in May that covers beginner and intermediate levels (1-4).
- Moderator
  - A type of user that has permission to oversee the website and make updates and changes.
- Music competition
  - A competition where the participants are judged based off the difficulty of music and how well they perform.
- Music festival
  - A festival oriented towards music that is sometimes presented with a theme such as musical genre, nationality or locality of musicians, or holiday.
- NNMTA
  - The Northern Nevada Music Teacher Association, a community of music teachers in Northern Nevada committed to foster professionalism, maintain a community that provides its members encouraging support, provide a program of continuing education, focused on pedagogical growth, encourage national certification, organize opportunities for student musical achievement, and encourage networking among the music community, schools, and related organizations.
- PayPal
  - A worldwide online method of payment. PayPal is used as an alternative to traditional paper methods such as checks and money orders. Users will have to use PayPal to pay for competition fees.
- Plugin
  - Programs that can easily be integrated into an existing system and used as part of your Web browser. In the case of ARIA, plugins refer to PHP plugins written for WordPress.
- Ratings
  - A type of ranking of something or someone that is based off of performance, quality, and standard. For the Reno Youth Music Festival, ratings include “Superior”, “Superior with Distinction”, “Excellent”, and “Needs Attention.”

- Recital
  - Performance of a piece of music played by a solo instrument, singer, or a small group.
- Repertoire syllabus
  - A syllabus which contains over a thousand pieces of music in which the students choose from for the Play-a-thon or the Reno Youth Music Festival. This is developed by the NNMTA board members and is included in the Festival Syllabus.
- Romantic
  - A time period from 1800-1850 characterized by emphasis on emotion and individualism. This is a time period to select from in the competition.
- Scales
  - Any set of musical notes ordered by fundamental frequency or pitch. Piano students are tested on this theory during their competitions.
- Transactions
  - An instance of making a payment in order to enter in the piano competition.
- Upper-level
  - Higher division in a competition which requires performing a piece of music that is high in difficulty.
- WordPress
  - An online, open source website creation tool written in PHP. This is the main platform that ARIA will developed under.



## Analysis

# ARIA: Analysis Diagram

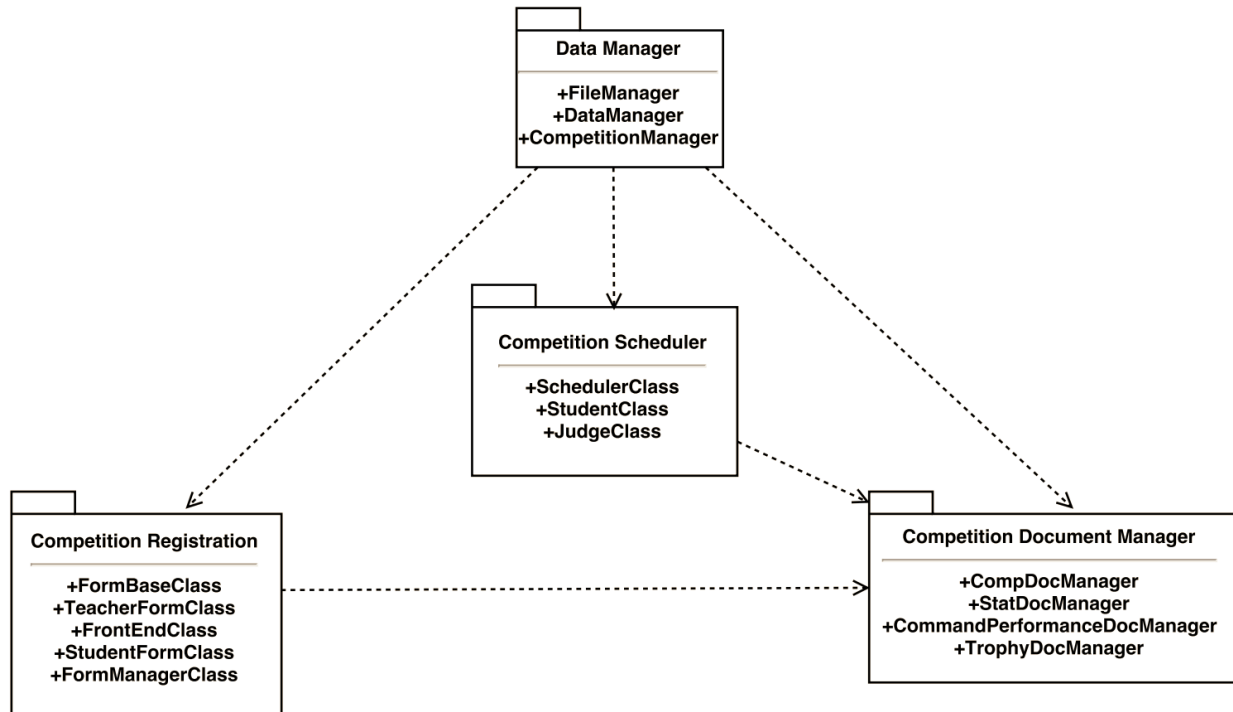
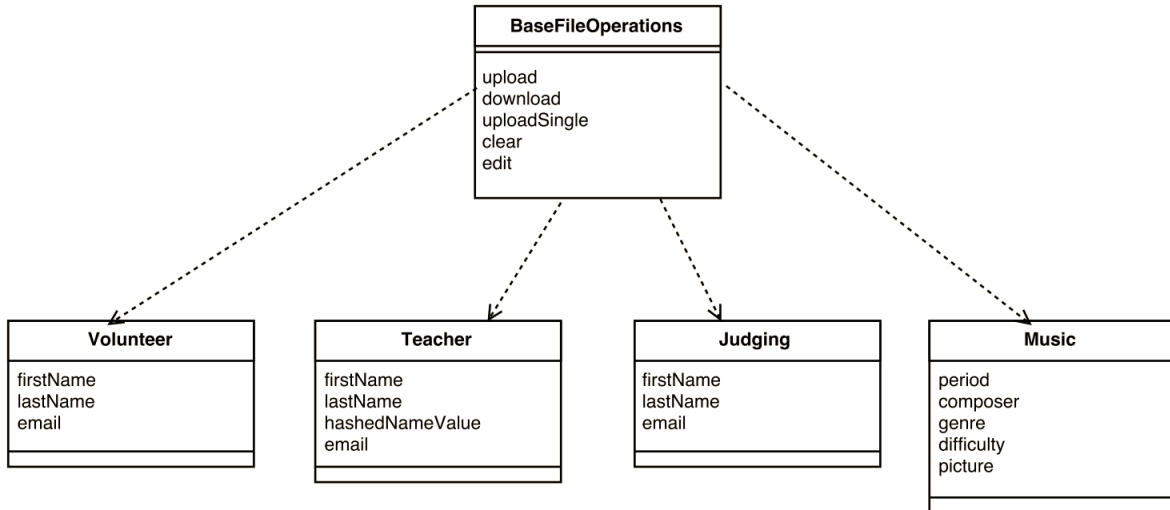


Figure 1: The following diagram represents the analysis model for ARIA. The diagram is composed of a data manager module, a competition registration module, a competition scheduler module, and a competition document manager module. These packages not only depict the high-level structure of ARIA, but also pertain to the fundamental functional requirements that are defined earlier in this document.

# Data Manager

## File Manager



## Data Manager

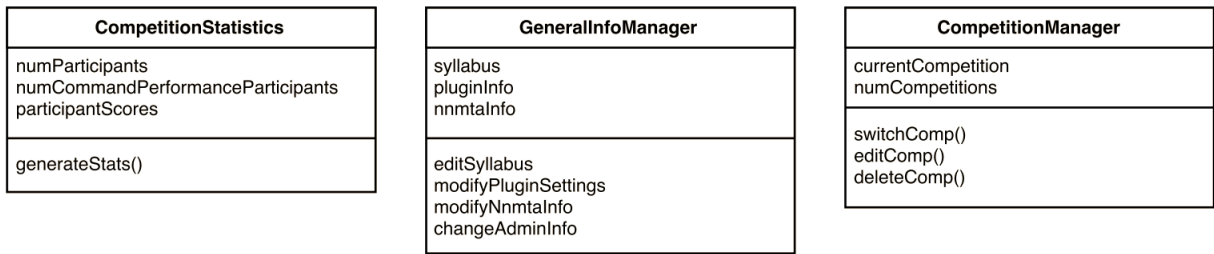


Figure 2: The following diagram represents the data manager analysis package. It's primary responsibility is the handling of ARIA's data, whether it corresponds to data accumulated from files or data that exists are part of the plugin. The file manager is further broken down into a management system for volunteers, teachers, judges, and music, all of which are input from files. The data manager, as previously mentioned, handles all other data associated with the ARIA plugin.

## Competition Registration

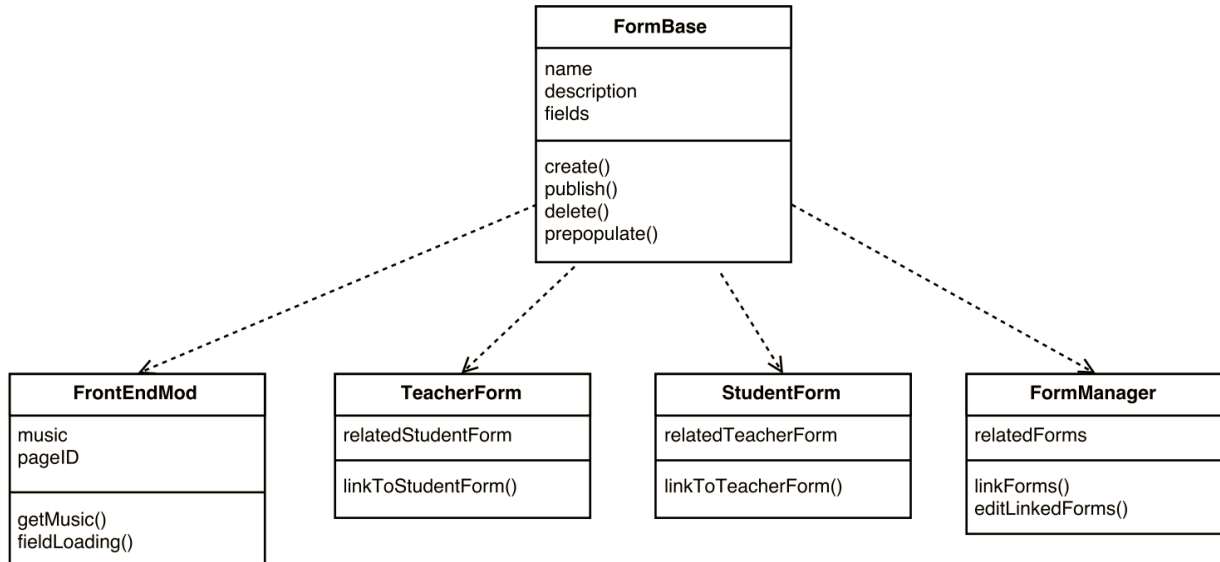


Figure 3: The following diagram represents the competition registration analysis package. The intended purpose of this analysis package is the management of all forms associated with ARIA. This includes the teacher and student forms, how these forms are displayed on the front end of the plugin (FrontEndMod) and also a backend interface for the festival chairman (FormManager) that can be used to manage any form. Each of these derives information from a base form module.

# Competition Scheduler

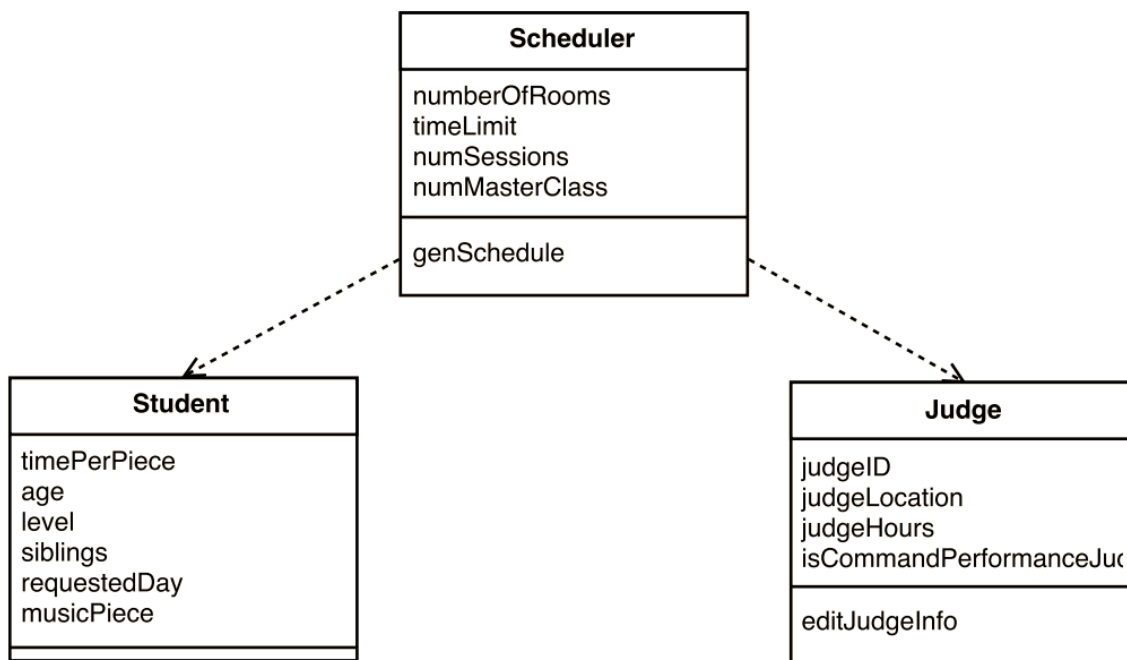


Figure 4: The following diagram represents the competition scheduler package. The modules included are scheduler, student, and judge. These modules are created so that the scheduler is able to generate the schedule for a competition which meets the requirements for both students and judges without any conflicts.

# CompetitionDoc Manager

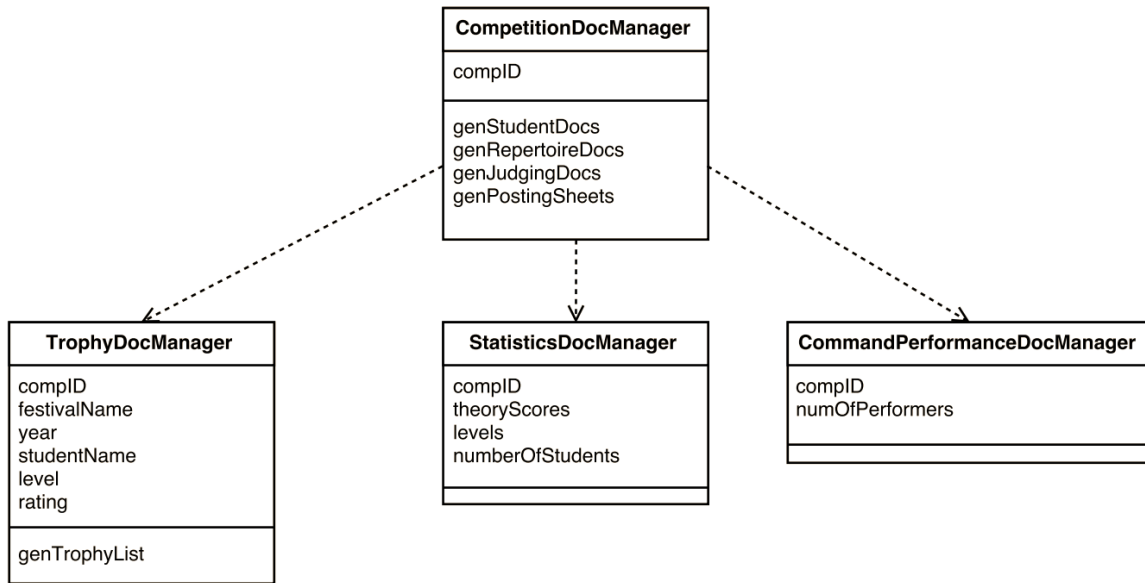


Figure 5: The following diagram represents the competition document manager package. The modules consists of a competition document manager, trophy document manager, statistic document manager, and a command performance document manager. The main functionality for each module is to hold the data needed to generate documents for the competition once the time comes.

## References

- Problem-domain book
  - Professional WordPress Plugin Development  
Brad Williams - Ozh Richard - Justin Tadlock - Wiley Pub. - 2011
- Journal papers, conference papers, and technical reports
  - A Growth Model for Form Generation
    - Rosenman, M. A., "A Growth Model for Form Generation Using a Hierarchical Evolutionary Approach." (1996). *Computer-Aided Civil and Infrastructure Engineering*, 11: 163–174. doi: 10.1111/j.1467-8667.1996.tb00320.x
  - Awards, Success and Aesthetic Quality in the Arts
    - Ginsburgh, Victor. "Awards, Success and Aesthetic Quality in the Arts" (2003). *The Journal of Economic Perspectives*, Volume 17, Number 2.
  - Automated creation of a forms-based database query interface
    - Jayapandian, Magesh and Jagadish, H. V. "Automated creation of a forms-based database query interface" (2008) *Proc. VLDB Endow.* 1, 1 (August 2008), 695-709.
  - Expressive query specification through form customization
    - Jayapandian, Magesh and Jagadish H. V. "Expressive query specification through form customization" (2008). In *Proceedings of the 11th international conference on Extending database technology: Advances in database technology (EDBT '08)*. ACM, New York, NY, USA, 416-427.
  - Consistency in piano performance evaluation
    - Wapnick, J., Flowers, P., Alegant, M., & Jasinskis, L. "Consistency in piano performance evaluation" (1993). *Journal of Research in Music Education*, 41(4), 282-292.
  - Marshall University Sonatina Festival & Competition
    - Alves, Júlio Ribeiro, "Marshall University Music Department Presents the Marshall University Sonatina Festival & Competition" (2011). *All Performances*. Book 426. [http://mds.marshall.edu/music\\_perf/426](http://mds.marshall.edu/music_perf/426)

## Contributions of Team Members

Team Member	Time (Hours)	Activity
Renee Inuma	7	<ul style="list-style-type: none"> <li>● Glossary</li> <li>● Requirements workshop summary</li> <li>● Analysis Package/Analysis class diagram</li> <li>● References</li> <li>● Use Case Descriptions</li> <li>● Traceability Matrix</li> </ul>
Wesley Kepke	7	<ul style="list-style-type: none"> <li>● Abstract</li> <li>● Software Requirements</li> <li>● Requirements workshop summary</li> <li>● Analysis Package/Analysis class diagram</li> <li>● Use Case Descriptions</li> <li>● Traceability Matrix</li> </ul>
Ernest Landrito	7	<ul style="list-style-type: none"> <li>● Use Cases Diagram</li> <li>● Detailed Use Cases</li> <li>● Requirements traceability matrix</li> </ul>
Kyle Lee	7	<ul style="list-style-type: none"> <li>● Glossary</li> <li>● Requirements workshop summary</li> <li>● Analysis Package/Analysis class diagram</li> <li>● Use Case Descriptions</li> <li>● Traceability Matrix</li> </ul>