Project Software Requirements Specification (SRS)

Zach Dwyer
Sam Kline
Jay Patel
Changhee Son

Iowa State University

November 2017

TABLE OF CONTENTS

I. Project Drivers: (5 pts.)

1. Purpose of the Project

(a) Background and (b) Goals (one sentence).

The purpose and goal of this project is to implement an easy, intuitive way for beginner or advanced users to learn how to play an instrument at a decreased cost and increased convenience by using an inexpensive and useful AR application.

2. Stakeholders

one-two sentences covering 2a & 2d from Appendix A.

2a: Client

The Software Solutions Company has requested our team create and implement an augmented reality device to allow beginner, intermediate and master musicians to play a fake physical reality instrument hearing sounds that augment reality sounds from our augmented reality device. Our clients will be responsible for approving our cost of production, how many types of instrument props we produce, what size of database we provide to our users, how many additional features we implement after product release, and how long we continue providing technical support and software updates to users.

2d: The Hands-on users of the product

Content:

The potential users of this product would include:

- **Beginner:**
  - User role: Someone who has never played an instrument before
  - Subject-matter experience: Novice.
  - Technological experience: Journeyman (all users should have some experience with an AR device).
  - Other characteristics: Willing to learn more about using AR. technology.
- **Intermediate:**
  - User role: Someone who has some experience with playing an instrument.
  - Subject-matter experience: Novice/journeyman.
  - Technological experience: Journeyman.

Other characteristics: Willing to learn more about using AR technology, reasonable education on playing one or multiple instruments.

- Expert:

User role: Someone who has experience with playing one or multiple instruments.
Subject-matter experience: Journeyman/master.
Technological experience: Journeyman.
Other characteristics: Willing to learn more about using AR technology, significant education on playing one or multiple instruments.

- Artist:

User role: Someone who wants to compose music and share with others.
Subject-matter experience: Master
Technological experience: Journeyman.
Other characteristics: Willing to learn more about using AR technology, willing to take time to record songs, educated about music composition.

- Teacher:

User role: Someone who wants to help less skilled musicians learn.
Subject-matter experience: Master.
Technological experience: Journeyman.
Other characteristics: Willing to learn more about using AR. technology, willing to take time to make tutorials and assign them to students, educated and/or experienced in teaching.

II. Project Constraints (5 pts.)

3. Mandated Constraints

Content: We will have several constraints for our users when it comes to our augmented reality application. For one our users will not be able to manually jump from a beginner level and then on to a intermediate level without the approval of the teacher.

Motivation: This will allow users who are not doing so well within their difficulty level to not cheat themselves and believe they are in a higher difficulty level.

Examples:
Description: The product shall use our instrument sensor.
Rationale: The instrument sensor is what will allow our users to connect their fake instrument replica with our augmented reality device.
Fit Criteria: Once the fake instrument replica is connected to our augmented reality
device via our instrument sensor the user can begin their particular instrument.

Description: Our product shall be much more affordable than going out and purchasing an actual real instrument.
Rationale: Potential users who do not have enough money to purchase a real instrument will have the ability to play our fake instrument replica and hearing augmented reality sounds through the augmented reality device.
Fit Criteria: The product shall be no more than 100 dollars for our fake instrument replica, instrument sensor and our software to install on the augmented reality device.

Description: Our product shall be connected to a secure network with adequate wifi speed.
Rationale: This will allow for the best connection when retrieving your teachers instructed song tutorials to complete.
Fit Criteria: This product shall use the TCP port 65535 for maximum internet speed.

Considerations:
All of the specific components for our application include our instrument sensor, fake instrument replica and our software to be downloaded on any augmented reality device.

Form:
All the actors that we have an ability to either check their own progress or others depending on their user type. The results will be displayed via graph chart or a Gantt chart which will make the visualization better and simpler.

4. Naming Conventions & Terminology

Define any abbreviations or special terms that developers may not understand. Don't need to duplicate data in #7 below.

AR device: This is our actual visual augmented device such as goggles, that we will be using for this project. It will have a display, camera and sound system. It will interact with our system as well as instrument prop with sensor.

Artist: User type. This type of user can record and upload songs, as well as view how many times their songs has been used by teachers and students.

Instrument Prop: Simple shape that the user will hold and the AR device will project displays onto. AR device will project things like: textures that look similar to a real instrument, tips for how to play a song, notes of current song, progress through current song.

Sensor: Pressure sensors that are placed on the instrument prop. The user will select what type of instrument they want to play through the AR device. The sensors will report user input to our application which will provide the display and audio appropriate for the selected instrument.

Song: Composition that has been uploaded by an artist can be accessed by other users. Songs can be compiled into tutorials by teachers. Artists can play and record the song using our AR device.

Student: User type. This user takes tutorials that Teachers have assigned them. Students can view their tutorial results on their profile page, which would include % of notes hit, ect.

Teacher: User type. This user takes songs that an Artist has uploaded and assigned to students. Teachers can view, through their profile page, the tutorial results of all students that they have assigned tutorials to.

Tutorial: Collection of songs that have been selected by the teacher. These are then assigned by teachers to students.

5. Relevant Facts & Assumptions, as applicable

Make sure that your (5c in App. A) are accurate and up-to-date.

Relevant Facts: Any AR device with full battery that is compatible with our software will be able to run it for at least 4 hours without recharging.

Assumptions:

- Users have access to an internet network with reasonable speed that can fetch music and tutorials from our database.
- Users can easily lift objects weighing 1KG, our instrument props will vary in size and weight depending on which instrument is needed.
- Users have good enough eyesight and limb flexibility to be able to operate an instrument.
- Users have access to an AR device that our software can be used on.

III. Functional Requirements (50 pts.)

6. Scope of work:

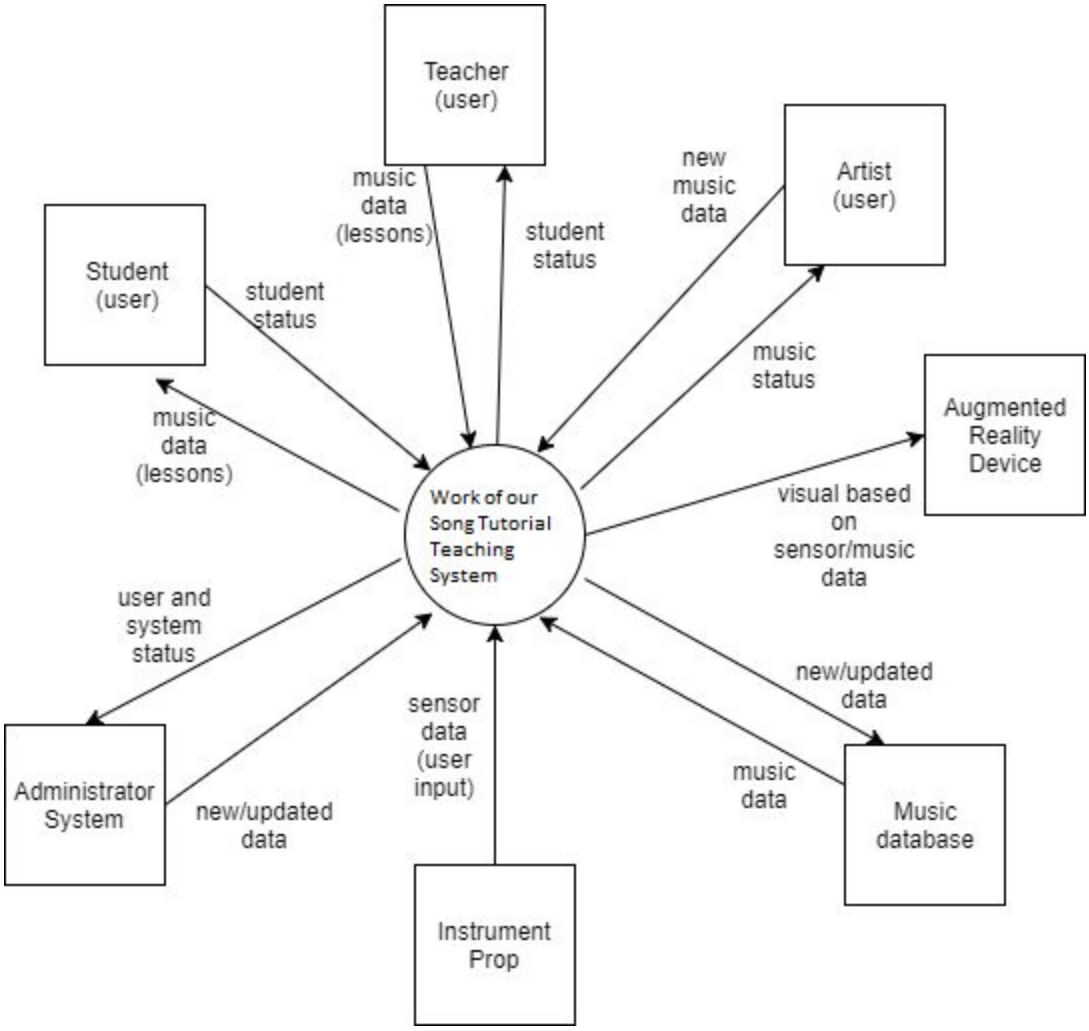
(a) Describe current situation very briefly

We have developed a detailed plan for our product including required features and a prototype for how the user interface will look for different types of users.

(b) Include your corrected & updated Context Diagram

(original was in HW#1). Don't need accompanying text, but make sure that it is consistent with #4 above.

Please see the document below:



(c) Include your corrected and updated business event list

(original was in Hw#2). Same.

1. **Identifying the events, and the input and output flows:**

Business Event List

Event Name	Input and Output	Summary of BUC
1. Booting up System	**Input**: Press the start button on the instrument prop sensor. **Output**: The instructions on how to connect the	User will need to boot the system in order to start practicing the instrument. Also, if the system boot properly, it will show the

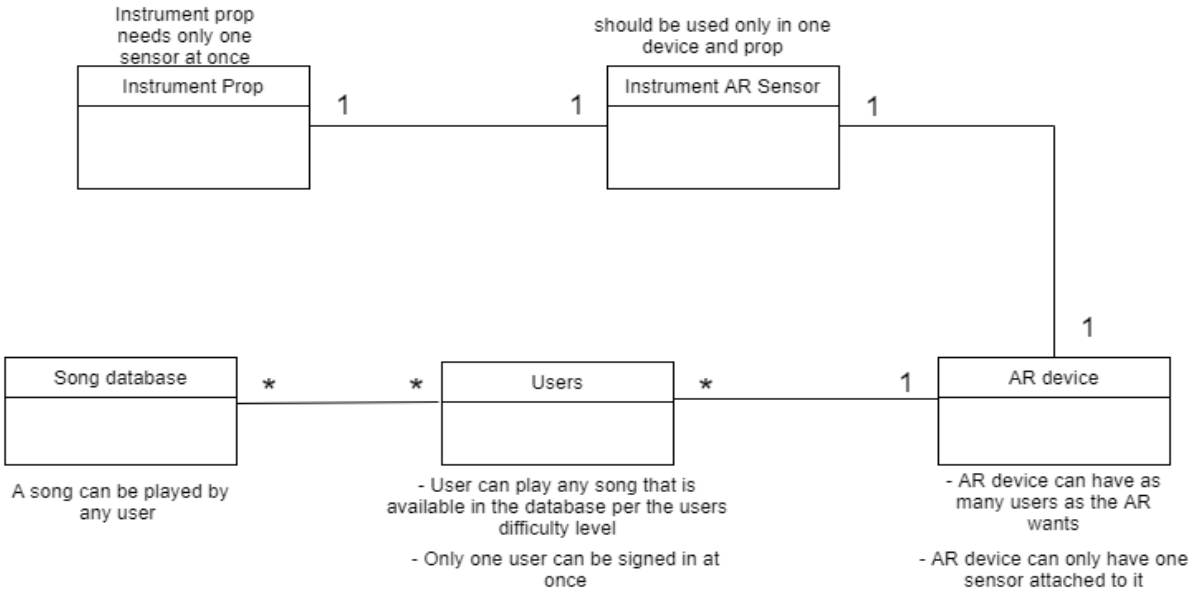
	instrument prop with our augmented reality device.	instructions of how to connect to the sensor with the instrument prop.
2. Setting instrument sensor	**Input**: The user will need to enter the type of instrument. **Output**: If detected, show the signup/login page. Else, keep detecting instrument.	Once the sensor is turned on, the user will have to input the type of instrument. If it is connected properly, it will show the signup/login page.Otherwise, it will keep trying until successful detection.
3. Signing up/Login	**Input**: Enter user credentials. **Output**: If correct, sign in successful message. Else, improper credentials message.	User will need to signup/login and determine their type of user i.e. teacher/artist/student while signing up.
4. User Chooses a Difficulty level Setting (Only for students)	**Input**: Select the level of difficulty that the student wants to learn. **Output:** Display tutorials to be learned associated with whatever difficulty level that they chose as the input to this event.	Difficulty settings are updated by the user i.e. beginner, intermediate or expert. Teachers will be able to see the students preferred difficulty level and assign tutorial to be learned associated with that difficulty level.
5. Uploading song (only Artist)	**Input:** Press upload button to upload songs. **Output**: Shows the status of uploading.	Once the artist is done making a song, artist will be able to upload song and screen will show the uploading status.
6. Assigning tutorial to students (only teachers)	**Input:** Select the songs to assign the student. **Output:** The tutorials that assigned to the student.	Assign the songs to the student based on their level of difficulty. Once the teacher finish the assigns, the screen will show the tutorial.
7. Selecting a song from tutorial (only Student)	**Input**: Select songs from the tutorials.	Student can select a song from the tutorial that student

	Output: The AR device will display selected song by user.	is willing to practice. Once the song is selected, AR device will display that song with play button.
8. Student plays tutorial	**Input:** Press the play button. **Output:** Music will be played.	By pressing the play button from the screen, user can start practicing a song from tutorial that was assigned by the teacher.
9. Student pauses/exit tutorial	**Input:** Press the pauses/exit button. **Output:** Music will be stopped.	Student can stop the tutorial during the practice by pressing the pause button or exit button if they are willing to quit.
10. Student resumes tutorial	**Input:**.Press the resume button. **Output:** Music will be resumed from the place it was paused.	Must have paused before. Student can resume from the place where it was stopped to continue.
11. Student views their profile	**Input:** Press the profile button. **Ouput:** lesson(n) results.	Can access to the profile from the navigation bar. Student can view the progress, and songs assigned to them.
12. Teacher views their profile	**Input:** Press the profile button. **Output:** Student(n) result.	Can access to the profile from the navigation bar. Teacher can see how many students the teacher has, the tutorials assigned to the student, and each student's progress.
13. Artist views their profile	**Input:** Press the profile button. **Output:** Song(n) result.	Can access to the profile from the navigation bar. Artist can view the list of songs the user made.
14. Signing out / Turning off system	**Input**: Press sign out button or turn off button **Output:** Cannot login after.	Once the practice is done student, songs are uploaded by artist and tutorials are assigned by the teacher, they all can shut down or sign out when needed.

*For input: We have underlined a word that gives an input to the product.

7. Business Data Model and Data Dictionary

Include your updated and corrected class diagram model from HW#3.

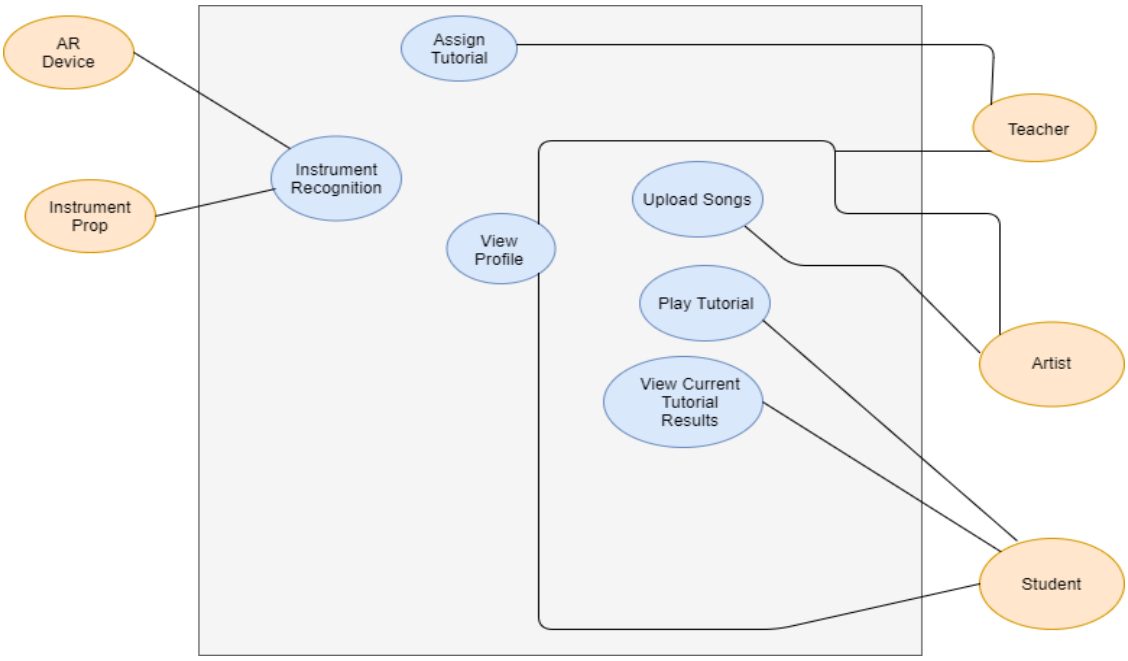


8. Scope of product:

(a) Include your corrected & updated Product Use Case Diagram

(original was in HW#2).

Please see the diagram below:



(b) skip

(c) Include your updated and corrected scenarios (originals were in HW#2)

Business Event Name: Configuration of Instrument prop through sensor and AR device.
Business Use Case Name and Number: Setting a sensor that way we can detect the device.
Trigger: Needs to turn the system on.
Precondition: The instrument prop must send a signal that is detectable by the AR device.
Interested Stakeholder: Sensor, system, AR Device.
Active StakeHolder: Student, teacher, artist, system.
Normal Case Step:

Step 1: Turn the system on in order to get the sensor started.
Step 2: Once the system is started it starts searching for a signal.
Step 3: System detects a valid signal from an instrument prop.
Step 4: System allows the user to use the application's other functionalities.
Step 5: User signs out.
Step 6: Turn system off.

Alternatives for:

Step 3.1: If the system does not detect a valid signal from the prop, offer the option to reboot the system and try connecting again or shut down.

Step 5.1: User could sign back in as a different user with different credentials.

Exceptions: If the user keeps trying and doesn't get connected to the instrument, user can quit by shutting the system off.

Outcome: They can then start practicing and view their profile after the booting is done and the sensor is detected. When they are done using the application, they can sign out and either turn the system off or sign in as another user.

Business Event Name: Artist decides to upload new songs.

Business Use Case Name and Number: Upload a song to the database as an Artist user.

Trigger: Artist starts recording and then uploads for the teacher to view them.

Preconditions: The artist must have signed up in our system before uploading songs.

Interested Stakeholders: Teacher, Artist, system, instrument prop, sensor, AR Device.

Active Stakeholders: Artist, system, instrument prop, sensor.

Normal case steps:

Step 1: Login into the system.

Step 2: Start recording input.

Step 3: Stop recording input.

Step 4: Input song information (name, difficulty)

Step 5: Confirm submission to database.

Step 6: Start recording new song (step 2)

Alternatives for:

Step 2.1: Options to pause and resume recording of input.

Step 3.1: Resume recording of input.

Step 4.1: Cancel this step and return to recording (step 3)

Step 5.1: Cancel submission and return to recording (step 3)

Step 5.2: Cancel submission and trash the recorded song.

Exceptions:

If uploading the song fails because of connection issues or file corruption, the user can keep our application open and try again later or quit by shutting the system off.

Outcome:

Once an artist has submitted their song, teachers will be able to access it to add it to tutorials. Artists will be able to see how popular their song is through their profile page.

Business Event Name: Teacher allocated tutorials to students.
Business Use Case Name and Number: Assign tutorials to the students based on their level of those 4 difficulty.
Trigger: Teacher must make tutorials for the students according to their level of difficulty.
Precondition: Artist must have songs uploaded to our system.
Interested Stakeholder: Teacher, student, system.
Active StakeHolder: Instrument prop, teacher, student, system.
Normal Case Step:

Step 1: Login into the system and get access to the system.
Step 2: Access the songs from the database uploaded by the artist.
Step 3: Once the teacher can access the songs from the database, the teacher can make tutorials using the songs based on students level of difficulty.
Step 4: Teacher assigns tutorial to student, teachers can see a student's progress on teacher's profile page.
Step 5: Start assigning more songs when the student is done with the song that was assigned before.

Alternatives for:

Step 2.1: If there are no songs in the database, she cannot assign to the student.
Step 5.1: Teacher cannot assign more than one tutorial to the student at once.

Exceptions: Before assigning the tutorials to student, teacher might want to look at the student's progress again in order to assign proper songs to practice for a student. In this case, the teachers can return to their own profile at any time.
Outcome:

Once a teacher has assigned a tutorial to a student, the student should be able view what tutorials have been assigned to them and begin a tutorial. The teacher will be able to see what tutorials they have assigned to specific students, and results of the students doing the tutorials.

Business Event Name: Student decides to play a song from the list of tutorial songs.
Business Use Case Name and Number: Play a tutorial as a student user.
Trigger: A user presses play tutorial button in order to start off the practice.
Preconditions: The user must have our device and sign up in our system.
Interested Stakeholders: Teacher, Artist, Student, system, instrument prop, sensor, AR Device.
Active Stakeholders: Student, system, instrument prop, sensor, AR Device.
Normal case steps:

Step 1: Login into the system
Step 2: Setting the level of difficulty
Step 3: Select the tutorial provided by the teacher in order to practice a song
Step 4: System will show tutorial interface through an AR Device.
Step 5: Student will then be able to select a song from the tutorial by pressing the play button from the navigation bar.
Step 6: Complete the tutorial.
Step 7: After completion of a tutorial, students can immediately view the current tutorial results.
Step 8: If the student wants to practice again, they can go back to step 3 to 7.

Alternatives for:

Step 5.1: Student have an alternative option to either pause, exit or resume from wherever they stopped during the practice.
Step 5.2: If the student miss a node that was prompted to them on the AR Device, they will be warned with an error on their screen.
Step 6.1: Student can change their level of difficulty if they think the tutorial that was provided by the teacher is too easy or hard.
Step 7.1: If the user don't want to view the grade, they can choose to exit out of the system.
Step 8.1: If they don't wish to practice more, they may exit out of the system.

Exceptions:

If the user want to quit at any point after the tutorial has begun, they can either quit or exit out of the system anytime.

Outcome:

Once they finish the tutorial, their results will be saved that way it can be accessed by either students or teachers.

Business Event Name: User wants to access their profile.
Business Use Case Name and Number: Access user's profile.
Trigger: A user presses their profile page button.
Preconditions: User must be logged in as a Student, Teacher, or Artist, and their must be a connection to the system.
Interested Stakeholders: Teacher, Artist, Student, system, AR Device.
Active Stakeholders: Teacher, Artist, Student, system, AR Device.
Normal case steps:

Step 1: Login to the system.
Step 2: Select profile page icon.
Step 3: System determines what type of user is attempting to access information.
Step 4: System fetches appropriate information from database based on user's type and ID.
Step 5: AR device displays the information for the user.
Step 6: User browses profile page information.

Alternatives for:

Step 6.1: Close profile page.
Step 6.2: Refresh profile page.

Exceptions:

If accessing profile information fails because of a bad connection or other error, provide summary of the error and offer an option to attempt a reconnection or exit the system by shutting down.

Outcome:

When a user selects their profile page it will display the information based on the type of users (Student, Teacher, or Artist). Student should see only their results of the tutorials that they have completed. Teachers should see a list of all students that they have assigned tutorials to and their results. Artists should see a list of the songs that they have Uploaded and information about how popular their songs are.

9. Functional requirements:

(a) Include an updated & corrected version of the FRs

from HW#3 with Fit Criteria where needed (i.e., to make each FR testable).

1. Boot System **(Event Driven) Event #1**

   When the user wishes to use our instrument tutorial system our system shall boot up AR device, instrument sensor and connect to our database.

   Rationale: The system needs to turn on when user want to practice with this product.

   Fit Criteria: On boot, the system won't produce any errors and successfully connect to the prop sensor and database in under 30 seconds.

   Priority: High.

2. Setting Instrument Sensor (**Event Driven) Event #2**

   When the instrument sensor is activated our AR device shall connect to that instrument sensor to begin using our instrument tutorial system.

   Rationale: Input data needs to be received from the prop in order for it to be used as an instrument by our app.

   Fit Criteria: When connecting via the instrument sensor, the AR device is able to receive input data from the instrument prop.

   Priority: High.

3. Sign up/Sign in (Choose User type) **(Unwanted Behaviour) Event #3**

   If the user is not signed up or signed into our instrument tutorial system, then the instrument tutorial system will prompt the user to sign up or sign in.

   Rationale: Validation; Users should only be allowed to access data that is relevant to them, and not access information about other users.

   Fit Criteria: When the user is logged in succesfully to our application they will have all the proper permissions associated with their user type. (ie. Beginner, Intermediate, Expert, Artis & Teacher).

   Priority: High.

4. Uploading Songs (For Artist) **(Event Driven) Event #5**

   When the user artist attempts to upload songs to our song tutorial database the database shall store this specific song in its particular genre.

   Rationale: System should have a songs made by artist user. Our app will release with some songs but that's not enough to maintain service. So, this function is needed to update the songs and add new contents continuously.

   Fit Criteria: Songs are uploaded to the database in under 30 seconds and are retrievable by other users.

   Priority: Medium.

5. Assign Tutorials to Students (Teachers Only) (**Event Driven**) **Event #6**

   When the user teacher assigns tutorials to user students, the song tutorial database shall return the selected songs stored within our database.

   Rationale: The user student will need to be able to learn song tutorials based off of their skill set with whatever instrument they may be learning.

Fit Criteria: When a Teacher assigns a tutorial the database shall return the specific tutorial as requested by the Teacher user.
Priority: Medium.

6. Select difficulty Level (Only for Students) **(State Driven) Event #4**
   While the user is student, the instrument system shall allow the student user to update their difficulty to the next difficulty level. ie. beginner to intermediate to Expert.
   Rationale: User students will need to have the correct difficulty associated with their skill set in order to properly advance to higher difficulty level songs.
   Fit Criteria: While the user is a student and meets all of the requirements to advance to the next difficulty their profiles difficulty level will be changed to reflect that upgrade in difficulty.
   Priority: Low.
7. Select Song from Tutorials **(Event Driven) Event #7**
   When the user selects the song tutorials from the song tutorial database, the song tutorial database shall return that specific tutorial song to be learned.
   Rationale: Students should be able to choose what order they play the songs in a tutorial.
   Fit Criteria: When a song is selected, it will be added to the tutorial within 10 seconds and be retrievable by other users
   Priority: High.
8. Students Play Song from Tutorials **(Complex: Event & State Driven) Event #8**
   While not playing any song, if the student presses the play button, the song tutorial database shall begin to play that specifically selected song from our song database.
   Rationale: User students will need to have the ability to begin their song tutorial at any moment.
   Fit Criteria: If the student selects a song to play the database shall return that specific song to be played.
9. Students Pause/Exit Song from Tutorials **(State Driven) Event #9**
   While the user student is playing, the song tutorial database shall pause the specific song tutorial that they are currently using.
   Rationale: Students may need to stop playing the current song, and so they need the option to pause or exit.
   Fit Criteria: If a user presses the pause the button the application shall then pause the song tutorial.
   Priority: Low.
10. Students Resume Song from Tutorials **(State Driven) Event #10**
    While the user student is paused, the song tutorial database shall resume the

specific song tutorial that they are currently using.
Rationale: User students will need to have ability to play their song tutorial once more, if they ever needed to pause while attempting the tutorial.
Fit Criteria: If the student presses the play button while the song is paused the application shall then resume playing of the song.
Priority: Low.

11. Tutorial Completion by Students **(Event Driven) Event #8**
    When the user completes the entire song with the proper notes hit on time the song tutorial database shall return the analyzed data of the completed song tutorial.
    Rationale: Students need to view their results so they can see areas that need improvement. Teachers also need those results to see how students are progressing.
    Fit Criteria: If the song tutorial is complete the database shall analyze all of the properly hit notes on the right timing to produce a visual graph of the data.
    Priority: Medium.
12. Users view Profile **(Event Driven) Event #11-13**
    When the user attempts to view another user profile our user database shall return that specific users profile.
    Rationale: Users need to be able to view their personal information that has been collected by our app.
    Fit Criteria: If the user attempts to view another user's profile the database shall return that specified profile.
    Priority: Medium.
13. Signing Out **(State Driven) Event #14**
    While the user is currently signed in, the user data base shall log that specific user out of our instrument tutorial system.
    Rationale: Users need to be able to exit their account so that it is kept secure.
    Fit Criteria: If the user selects the option to log off of our application the system shall log that user out of the AR application.
    Priority: High.
14. Turning System off **(State Driven) Event #14**
    While the system is in the on state, the instrument tutorial system shall turn instrument sensor, AR device and disconnect to our database.
    Rationale: The system needs to turn off when users finish their tutorial.
    Fit Criteria: If the system is turned off in the on state the AR device shall disconnect from the instrument sensor and turn off both the AR device and the instrument sensor.

Priority: High.

Note: All these functional requirements are matched with Event Table in part 6-(c)

(b) Annotate each FR with its product use case, rationale and priority

(as appropriate, i.e., if not clear), and priority (see pp. 363-364, 382-383). Be sure to provide a key to the prioritization ranking.

Done above by combining the priority and rationale in part 9a.
Requirement priorities: High, Medium, and Low
High: Functionality required for our product to have any use, other functions build off of these.
Medium: Main features of the application that build off of high priority requirements.
Low: Extra features that enhance the usability of higher priority requirements.

IV. Nonfunctional Requirements (30 pts.)

10-17. Nonfunctional requirements:

(a) Include an updated & corrected version of the NFRs

from HW#3 with Fit Criteria where needed (i.e., to make each FR testable). Optional whether you divide according to the textbook's types, or use another reasonable grouping.

Look and Feel:
Priority: Medium.
Rationale: An unattractive user interface can distract and frustrate users from using the application.

a. The product shall have a responsive profile for each user.
    i. **Fit criteria:** Profile is updated with most current user information within 30 seconds of a change.
b. The product shall have a user interface that is attractive to its user community.
    i. **Fit criteria:** Typical users will not be distracted by the user interface design, they won't submit complaints about it.
c. The product shall have sound quality(.wav file) that is pleasing to its user community.
    i. **Fit criteria:** Sound files will be equal or higher quality than the same sound files in similar applications.

Usability and Humanity:
Priority: High.
Rationale: The product will be intuitive for any potential customer with no musical

background to begin and start learning an instrument they may have always wanted to learn.

a. The product shall be easy for people ages 8-50 to use.
    i. **Fit criteria:** Typical users age 8 and 50 will be able to navigate our application at similar speeds.
b. The product shall be used by people with no prior training.
    i. **Fit criteria:** Users with no prior training and users with prior training will be able to navigate our application at similar speeds.
c. The product shall make the users want to continue using it.
    i. **Fit criteria:** Average time between installation and uninstallation of our application will be several hours.
d. The product shall use words that are normally understandable by the user community.
    i. **Fit criteria:** All text shall be understandable by typical users ages 8-50.
e. Teachers will have the ability to assign as many songs as they feel necessary for the student to progress with their particular instrument.
f. System must not allow beginners to select difficulty levels above their skill set.
    i. **Fit criteria:** Tutorial results of beginner users will show improvement over time, which means their difficulty level is appropriate for their skill set.
g. The product shall only show details that are understandable by the user (will simplify error messages, will simplify product info, etc).
    i. **Fit criteria:** Typical users are able to understand what is causing application errors based on what error messages displays.

Performance:

Priority: High.

Rationale: A slow running application can cause users to become frustrated and tired of waiting for slow load times.

a. Boot system shall load all of the components of our product in under 30 seconds.
b. The response time for any user interface in our AR device shall be fast enough to avoid interrupting the user's flow of thought.
c. The time needed for input from the instrument prop to be sent to and displayed by our AR device shall be close enough to real time that any latency cannot be detected by the user.
d. The product shall check and update status of database connection every 10 seconds.
e. The product shall download selected songs from the database within 5 minutes.
f. Turning off the system and saving user data shall be completed in under 30 seconds.
g. The product shall not emit extremely bright light that could damage people's eyes.
    i. **Fit criteria:** Prolonged use of our app will not cause irregular eye damage.
h. Sensitive electronic components shall be shielded from human contact.

    i. **Fit criteria:** Typical users won't be able to access electronic components without using power tools.

i. Accuracy of tutorial results shall be accurate to two decimal places (ex: 95.27% notes hit).
j. Accuracy of popularity of artist's songs shall be accurate to two decimal places (ex: 90.47% of users liked your song).
k. The product shall achieve 99% uptime.
l. The product shall notify users of any future non availability, such as system updates or database maintenance.
m. The product shall continue to operate in local mode with limited functionality whenever it loses connection to the database.
n. The product shall operate for at least 1 day before requiring a recharge.
o. The product shall be expected to operate for a minimum of 3 years before requiring replacement.
p. The product's database and server shall be capable of servicing 1000 users and processing their uploads and downloads without producing unacceptable latency.
q. User shall not be allowed to select more than one song at a time to prevent the system from crashing from too many requests.
r. Users shall only view one tutorial at a time to allow the student to better understand and complete the tutorial.

Operational and Environmental:

Priority: Medium.

Rationale: This product shall be easy to begin to start with no prior musical background.

a. The product shall be used by someone indoors or outdoors, in dry conditions.
    i. **Fit criteria:** Useage of the product in these conditions will not change performance requirements.
b. The product shall not be loud enough to disturb people other than the user.
c. New versions of the product shall be able to access data from at least two previous versions.
d. The product shall be able to be installed and used by an untrained user without the need for separately printed instructions.
    i. **Fit criteria:** Users without printed instructions and users with printed instructions will be able to use our product at similar efficiencies.
e. Each maintenance release shall not cause previous releases and their features to malfunction.

Maintainability and Support:

Priority: Medium.

Rationale: Our product will undoubtedly have unaccounted for bugs upon our release, there will need to be external application support to fix these bugs.

a. The product shall remove songs that are duplicates of songs already in the database.
b. This product will be available in any free market society whose government approves our system.
c. If our any of our products is in need of repair we will fix the product if it is under the a year warranty.
    i. **Fit criteria:** All products that are broken and covered by warranty will be repaired or replaced, and the process recorded for validation.

Security:
Priority: Low.
Rationale: Users may want their musical skills and data kept private from other people.

a. The password should be at least 8 characters in length and must have at least one upper, lower case character, number with a special character, if the user fails to input the properly specified password, then the user shall be prompted to enter the password that meet the requirements.
b. Only the user's student, teacher, and the system administrator shall be able to access the user's information.
c. Only the song artist and system administrator shall be able to access and change details of an uploaded song.
d. Only a teacher and the system administrator shall be able to modify the teacher's tutorials.
e. The product shall prevent corrupted or malicious data from being uploaded to the database.
f. The product shall notify users of changes in its information policy and what data is being collected.
    i. **Fit criteria:** If prompted, a typical user would know about our information policy and collected data.
g. The product shall reveal private information only in compliance with the information policy.
h. User must complete every single song associated with a particular tutorial in order to receive credit.
    i. **Fit criteria:** Tutorial data correctly reflects progress based on a given amount of songs completed.

Culture and political requirement:
Priority: Low.
Rationale: Illogical discrimination will negatively impact sales of our product.

a. The product shall not be offensive to religious or ethnic groups.

    i. **Fit criteria:** At least 90% of users will not stop using or application due to offence to their religion or ethnicity.
b. The product shall allow multiple versions of the same song in different languages to be uploaded so users can switch to their preferred language if it is available.
    i. **Fit criteria:** At least 90% of users will be able to select a language that is understandable to them.

Legal:
Priority: Low.
Rationale: Legal issues will cost money

a. Information of the song must be filled out when artist upload the song such as name of artist and album name to give credit to the original artist to comply with the Copyright Act of 1976.
    i. **Fit criteria:** Artist information is easily accessible by a typical user without instructions.
b. Our instrument tutorial system shall comply with the Copyright Act of 1976 as amended and will safeguard us from any possible legal allegations of copyright being protected under this federal law (from book).
    i. **Fit criteria:** No legal allegations arise while our product is being distributed.

(b) Annotate each NFR with its rationale and priority.

(as appropriate, i.e., if not clear)

Done above by combining the priority and rationale in part 10a.
Requirement priorities: High, Medium, and Low
High: Most relevant to our product and the features we are implementing.
Medium: Relevant to our product and its features, but won't have as much impact on our customers as high priority.
Low: Not relevant to our product or will likely have little effect on our customers.

V. Project Issues (10 pts.)

18. Open issues

Use this section to explain any issues/concerns that have not yet been resolved and which you think the person/people who will be implementing these requirements in the future should know about or understand. Your goal is for the students who code & test from these requirements to be able to successfully & efficiently produce a product that satisfies the requirements that you've specified.

Feedback from Student User

Users, especially student users, might want to give feedback of the tutorial. There may be a problem with the tutorials or songs such as mismatch of difficulty or the sync of the songs. In this case, if the student user find this problem during practicing, student user want to let the teacher or the artist know about the problem. We have not made this function yet but it should be included in our system for interaction between all users in order to fix the any problem.

Number of props available

Different types of instruments are different shapes and sizes, so different shapes and sizes of props would be needed to simulate them. We have not yet decided on how many types of props will designed. We thought of designing one prop that could be combined with additional copies to form larger shapes for different instruments, but did not decide on one design.

Instrument Sensor Cannot Connect with AR Device

The basis of our application is built upon the connection of our AR device with our instrument prop. Our instrument prop is connected to our AR device via the instrument sensor. If there is an issue with the connection of the instrument sensor and the AR device then our users will not be able to practice their instruments. This will be an issue that we will constantly need to be tackling, even upon release there will be bugs that could possibly plague our system instrument sensor connection.

Users forget their account ID or password

The problem will happen a lot with other services as well. We have not made a function for finding users information. However, we expect it to happen. So, the function for this problem will be implemented by using email verification. People can send an email to their personal email account that they input when they first register for their account by pressing “forget account” button and they can find or change their account information via email.

19-22. Skip

23. Risks

optional (i.e., assume competence on the part of the developers).

The Problem of The Sensor

Not enough functionality may be implemented in the process of creating the sensor. The sensor is one of the most important components for receiving user input. That means the sensor is a very sensitive part. If it is not made properly, our products will not work properly.

24. Costs

Include your Function Point count (assuming we get to this topic), together with a brief explanation of what they're for (e.g., what do they have to do with Costs?), your estimated schedule according to the textbook's rule of thumb, and any doubts you have about the accuracy of that (in case this is used to hire a developer, you want to make sure you've given an accurate accounting).

Output Cost:

Element 1: Teacher

a. Data Elements on incoming data flows:
   Assign Songs, Give feedback, Sign out: **3**
b. Number of classes referred by the use case: **2**
   4 from the chart

Element 2: Student

a. Data Elements on incoming data flows:
   Sign out: **1**
b. Number of classes referred by the use case: **3**
   4 from the chart

Element 3: Artist

a. Data Elements on incoming data flows:
   Sign out, upload songs: **2**
b. Number of classes referred by the use case: **2**
   4 from the chart

Element 4: Administrator System

a. Data Elements on incoming data flows:
   Shows tutorial feedback: **1**
b. Number of classes referred by the use case: **6**
   5 from the chart

Element 5: Instrument Prop

a. Data Elements on incoming data flows:
   User data: **1**
b. Number of classes referred by the use case: **1**
   4 from the chart

Element 6: Music Database

a. Data Elements on incoming data flows:
   Song tutorial: **1**
b. Number of classes referred by the use case: **1**
   4 from the chart

Element 7: Augmented Reality Device

a. Data Elements on incoming data flows:
   Displays Interfaces: **1**
b. Number of classes referred by the use case: **1**
   4 from the chart

Using Jones principal:

$$29/150 * 29^{0.4} = 0.74/\ month$$

Input Cost:

Element 1: Teacher

a. Data Elements on outgoing data flows:
   Check grades, view profile, sign in, booting, set instrument sensor, play, pause, Resume, turn off: **9**
b. Number of classes referred by the use case: **2**
   4 from the chart

Element 2: Student

a. Data Elements on outgoing data flows:
   View profile, sign in, boting, set instrument sensor, set difficulty, view tutorials, select song, play, pause, resume, turnoff: **11**
b. Number of classes referred by the use case: **2**
   4 from the chart

Element 3: Artist

a. Data Elements on outgoing data flows:
   View profile, sign in, booting, set instrument sensor, play, pause, resume, turn off: **8**
b. Number of classes referred by the use case: **3**
   6 from the chart

Element 4: Administrator System

a. Data Elements on outgoing data flows:
   Set up profile: **1**
b. Number of classes referred by the use case: **6**
   4 from the chart

Element 5: Instrument prop

a. Data Elements on outgoing data flows:
   User data when playing instrument: **1**
b. Number of classes referred by the use case: **1**
   3 from the chart

Element 6: Music DataBase

a. Data Elements on outgoing data flows:
   Uploading the song tutorials: **1**
b. Number of classes referred by the use case: **1**
   3 from the chart

Element 7: Augmented Reality Device

a. Data Elements on outgoing data flows: Instrument sensor data: **1**
b. Number of classes referred by the use case: **1**
   3 from the chart

Using Jones principal:

$$27/150 * 29^{0.4} = 0.67/\ month$$

Instrument Prop Cost: The whole basis of our application is to have a much more affordable way for our customers to learn an instrument. Our instrument prop will be $30.00 to $50.00, roughly. This is much cheaper than an actual instrument like a violin which will run you around $100.00 for a very cheap violin instrument.
Instrument Sensor/AR display device/Software Cost: These items are the cornerstone of our application and will allow our customers to have a fully implemented AR Song Tutorial Application system. The price of instrument sensor will be $30.00 to $50.00. The price of AR display device will be $150.00. The price of Software will be $40.00. Therefore, the total price will be $240.00 to $250.00.
Maintenance/year Cost: $5.00/month subscription service.

25-27. skip