Software Requirements Specification

for

Bass Logic

Prepared by Gregory Vasquez

February 3, 2023

Table of the Contents

1. Introduction

1.1 Purpose

The purpose of this document is to define requirements for the website 'Bass Logic' with its initial release of version 1.0. The intended audience for this documentation are all stakeholders that will be a part of implementing these requirements.

1.2 Scope

The product to be produced is the Bass Logic website version 1.0. This application is intended for students who wish to improve their skills in marching bass drum and percussion. This is also intended to connect students with highly skilled instructors in order to schedule private lessons. The application will be accessible to any device with internet access. Future iterations of the website may include developing mobile applications rather than just the web based application.

1.3Definitions, acronyms, and abbreviations

Acronyms	
SRS	*Software Requirements Specification*
IT	*Information Technology*
BL	*Bass Logic*
HTTP	*HyperText Transmission Protocol*
TCP	*Transmission Control Protocol*
DCI	*Drums Corps International*

CSS	*Cascading Style Sheets*
HTML	*HyperText Markup Language*
JS	*JavaScript*
FAQ	*Frequently Asked Questions*

1.4 Overview

Below, there are multiple sections in this document which detail information about the product and specify requirements:

- Section 2 “Overall Description” discusses what the product features, who the application is intended for and assumptions.
- Section 3 “Use-Case Diagram and Descriptions (UCs)” shows a Use-Case diagram along with describing each Use-Case
- Section 4 “External Interface Requirements” explains how the system interacts with other systems
- Section 5 “Other Nonfunctional Requirements” lists requirements based on the quality attributes of the system
- Section 6 “Other Requirements” describes any additional requirements that are not listed in the other sections

2. Overall Description

2.1 Product Perspective

This product is a new website aimed at helping students interested in marching percussion, especially marching bass drum, acquire the skills needed to perform at the highest level.

2.2 Product Features (FEs)

FE-1: Able to view music sheet material

FE-2: Able to view other educational resources

FE-3: Able to create account

FE-4: Able to schedule private lesson

FE-5: Able to cancel or reschedule lesson

2.3 User Classes and Characteristics

USR-1: Student

Students will be the primary class when creating the application. Students will be the primary consumer that will be utilizing this application for educational resources and scheduling private lessons.

USR-2: Non-registered user

If a user does not wish to make an account or request private lessons then non-registered users will still have access to some of the education material. Content will be very basic and limited.

2.4 Operating Environment

OE-1: Student's User Webpage

In order for Bass Logic to be accessible a student, instructor or non-registered user will need to have a working PC with internet connection. The application will be utilized through Chrome, Edge or Firefox web browsers.

OE-2: Bass Logic Application

This is the main system that students, instructors, and unregistered users alike can access educational content regarding marching bass drum along with scheduling private lessons with their desired instructor.

OE-3: Scheduling System

Once a student creates an account on the Bass Logic website they will have access in order to schedule private lessons with an instructor of their choosing.

OE-4: Registered User Database

Once a student creates an account with Bass Logic their account with be stored in a database in order for the student to be able to log in and make changes or cancel the scheduled private lesson(s).

OE-5: Administrator Dashboard

This will be the environment that allows changes to accounts or overrides account information. At the moment this will be strictly managed by Gregory Vasquez until future iterations of the application are improved upon.

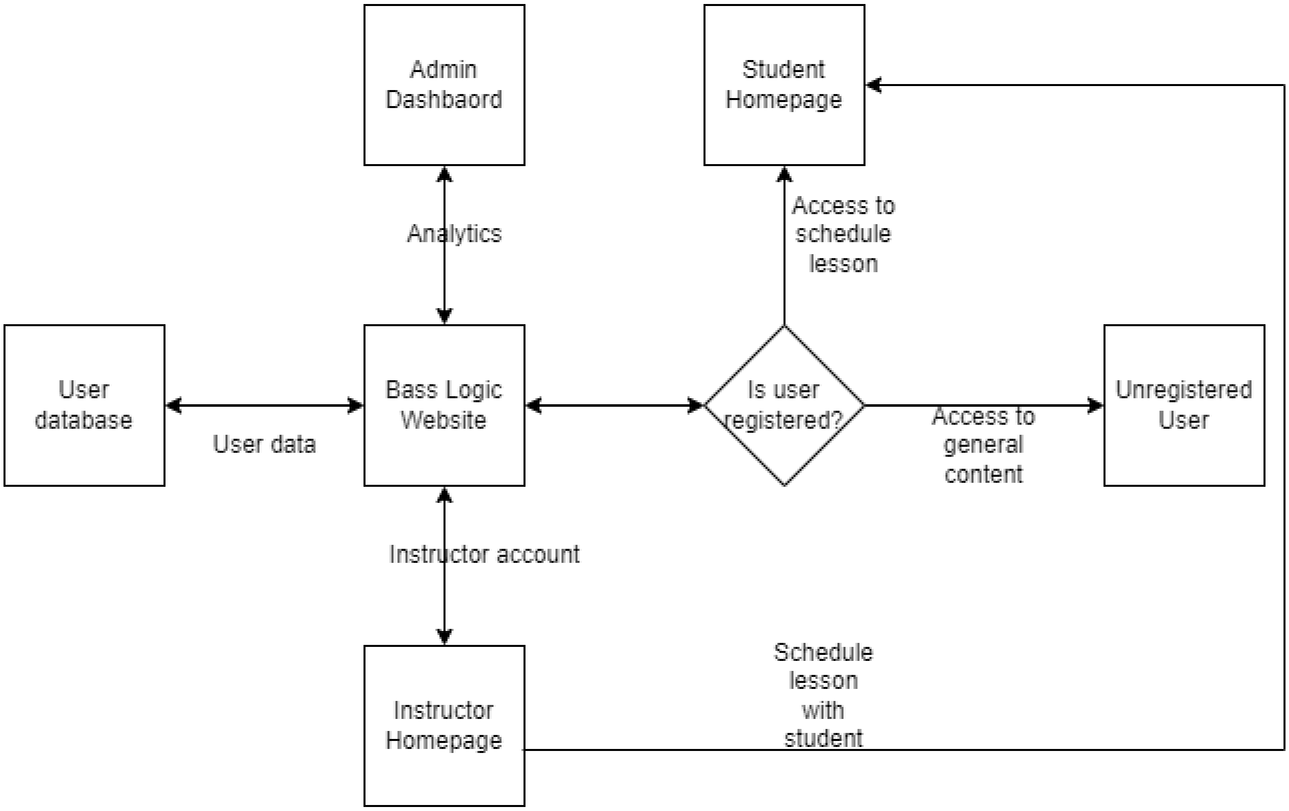


Figure 1. Ecosystem Map for Bass Logic

2.5 Design and Implementation Constraints

DIC-1: Application must be web based that can support HTML, CSS, JavaScript

2.6 User Documentation

The following documentation will be delivered along with the software:

UD-1: The bass logic website will include all resources in order to support the application including alerts to guide users in the correct direction in order to know how to properly sign up, schedule lessons, modify lessons, and cancel lessons.

2.7 Assumptions and Dependencies

AS-1: Application will be hosted on the WWW

AS-2: Website will work on modern browser; Chrome, Firefox, Edge

AS-3: User will have access to a device that has access to the internet

DE-1: If student wishes to schedule a lesson they will need a valid email in order to create an account

3. Use-Case Diagram and Descriptions

3.1 Use-Case Diagram

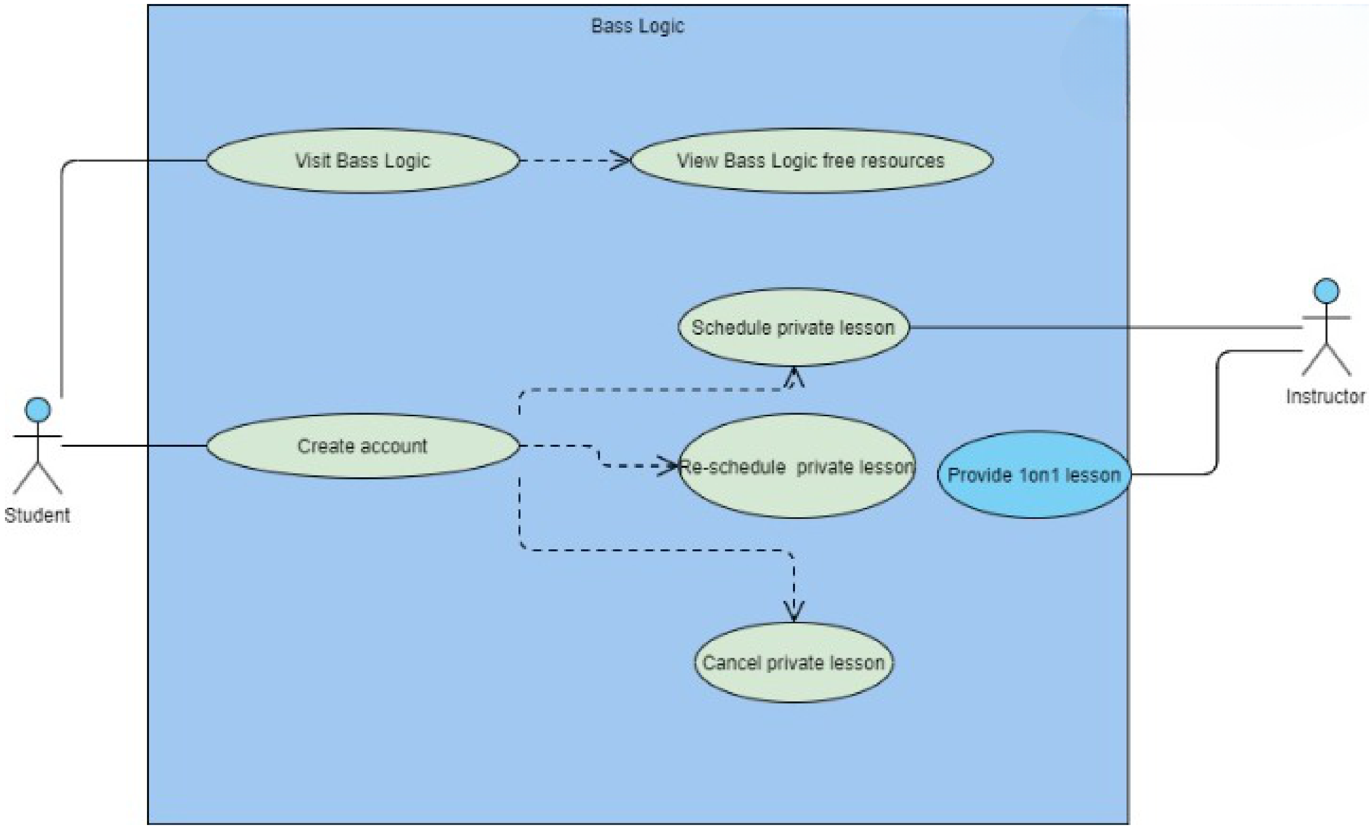


Figure 2 Use-case diagram of the Bass Logic.

3.2 Use-Case Descriptions

3.2.1 Visit the Bass Logic Website

Use Case ID:	UC-1		
Use Case Name:	Visit the Bass logic Website		
Related Features	FE-1, FE-2, FE-3, FE-4, FE-5		
Created By:	Gregory Vasquez	**Last Update By:**	
Date Created:	02/01/2023	**Last Revision Date:**	
Actors	Student		
Description:	The student visits the Bass logic website and views all the contents within it		
Trigger:	The student decides that they want to learn more about bass drumming and visits the website		
Preconditions:	PRE-1. Student requires device that has access to the internet		
Postconditions:	Post-1. Student receives new knowledge to apply to playing bass drum		
Normal Flow:	1. Student navigates to web browser on local machine 2. Student searches https://learnbasslogic.com/ 3. Student views all free content available for mastering marching bass drum		
Alternative Flows: [Alternative Flow 1 – No hardware device]	In step one of the normal flow if student does not have access to a device that has internet capabilities		
Alternative Flows: [Alternative Flow 2 – Maintenance]	In step two of the normal flow if a student searches the website but the website is down for maintenance the website will display a message to all users attempting to visit the website and provide a message on how long the site will be down for maintenance.		
Frequency of Use:	Multiple times a week, with a minimum of 3 times per week.		
Special Requirements:	Web application with need to handle traffic while multiple users		
Assumptions:	Website will be accessible 24/7		

3.2.2 Create Student Account

Use Case ID:	UC-2		
Use Case Name:	Create Student Account		
Related Features	FE-3, FE-4, FE-5		
Created By:	Gregory Vasquez		
Date Created:	02/01/2023		
Actors	Student		
Description:	The student visits the Bass logic website and creates a student account		
Trigger:	The student would like to schedule a private lesson		
Preconditions:	PRE-1. Student will need a valid email address in order to sign up		
Postconditions:	Post-1. Once signing up student is now able to schedule private lessons		
Normal Flow:	1. Students navigates to Bass Logic website 2. Student selects 'Sign Up' option 3. Students fills form for 'Sign Up' 4. Student successfully signs up to the Bass Logic website		
Alternative Flows: [Alternative Flow 1 – No hardware device]	In step one of the normal flow if student does not have access to a device that has internet capabilities		
Alternative Flows: [Alternative Flow 2 – Maintenance]	In step one of the normal flow if a student searches the website but the website is down for maintenance the website will display a message to all users attempting to visit the website and provide a message on how long the site will be down for maintenance.		
Alternative Flows: [Alternative Flow 3 – Email]	In step three if the student does not enter a valid email, the form will prompt an error message in order to correctly enter a valid email.		
Frequency of Use:	One time at initial sign up		
Special Requirements:	When signing up the website should automatically sign the user in as the current user.		
Assumptions:	Student will have a valid email		

3.2.3 Schedule Private Drum Lesson

Use Case ID:	UC-3		
Use Case Name:	Schedule Private Drum Lesson		
Related Features	FE-3, FE-4		
Created By:	Gregory Vasquez		
Date Created:	02/01/2023		
Actors	Student		
Description:	The student visits the Bass logic website and schedules a one on one drum lesson		
Trigger:	The student would like to schedule a private lesson		
Preconditions:	PRE-1. Student has a Bass Logic account		
Postconditions:	Post-1. Students receive confirmation of lessons scheduled.		
Normal Flow:	1. Student visits Bass Logic website 2. Student logs in to Bass Logic account 3. Student schedules a drum lesson 4. Student receives confirmation of scheduled lesson		
Alternative Flows: **[Alternative Flow 2 – Maintenance]**	In step one of the normal flow if a student searches the website but the website is down for maintenance the website will display a message to all users attempting to visit the website and provide a message on how long the site will be down for maintenance.		
Alternative Flows: **[Alternative Flow 2 – No Account]**	In step two if the student attempts to log in and does not have an account, the Bass Logic website will prompt student to create an account.		
Alternative Flows: **[Alternative Flow 2 –No available time]**	If a student selects a day and/or instructor that is not available, the Bass Logic website will prompt the student to select a different day and/or instructor. .		
Frequency of Use:	Multiple times a month, with a minimum of 3 times per month.		
Special Requirements:	Students will be able to see scheduled lessons once logging into the student dashboard.		
Assumptions:	Students will want to schedule a lesson.		

3.2.4 Student Reschedules Lesson

Use Case ID:	UC-4		
Use Case Name:	Reschedule lesson		
Related Features	FE-3, FE-5		
Created By:	Gregory Vasquez		
Date Created:	02/01/2023		
Actors	Student		
Description:	The student visits the Bass logic website and re-schedules lesson		
Trigger:	The student would like to reschedule a private lesson		
Preconditions:	PRE-1. Student has a Bass Logic account		
Postconditions:	Post-1. Student receives confirmation of lesson scheduled.		
Normal Flow:	1. Student visits Bass Logic website 2. Student logs in to Bass Logic account 3. Student schedules a drum lesson 4. Student receives confirmation of rescheduled lesson		
Alternative Flows: **[Alternative Flow 2 – Maintenance]**	In step one of the normal flow if a student searches the website but the website is down for maintenance the website will display a message to all users attempting to visit the website and provide a message on how long the site will be down for maintenance.		
Alternative Flows: **[Alternative Flow 2 – No Account]**	In step two if the student attempts to log in and does not have an account, the Bass Logic website will prompt the student to create an account.		
Alternative Flows: **[Alternative Flow 2 –No available time]**	If a student selects a day and/or instructor that is not available, the Bass Logic website will prompt the student to select a different day and/or instructor. .		
Frequency of Use:	Once per every 2 months		
Assumptions:	Students will need to reschedule due to schedule conflicts		

3.2.5 Student Cancels Lesson

Use Case ID:	UC-5		
Use Case Name:	Cancel lesson		
Related Features	FE-3, FE-5		
Created By:	Gregory Vasquez		
Date Created:	02/01/2023		
Actors	Student		
Description:	The student visits the Bass logic website and cancels lesson		
Trigger:	The student would no longer like to have a lesson or they are unable to attend.		
Preconditions:	PRE-1. Student has a Bass Logic account		
Postconditions:	Post-1. Student receives confirmation of lesson canceled.		
Normal Flow:	1. Student visits Bass Logic website 2. Student logs in to Bass Logic account 3. Student cancels a drum lesson 4. Student receives confirmation of rescheduled lesson		
Alternative Flows: **[Alternative Flow 2 – Maintenance]**	In step one of the normal flow if a student searches the website but the website is down for maintenance the website will display a message to all users attempting to visit the website and provide a message on how long the site will be down for maintenance.		
Alternative Flows: **[Alternative Flow 2 – No Account]**	In step two if the student attempts to log in and does not have an account, the Bass Logic website will prompt the student to create an account.		
Alternative Flows: **[Alternative Flow 2 –No lessons]**	If a student does not have any pre-scheduled lessons they will be prompted that they do not have any lessons to cancel.		
Frequency of Use:	Once per every 2 months		
Assumptions:	Students will need to cancel due to unforeseen circumstances		

3.3. TABLE 1: Traceability Matrix (Features & Use-cases).

	Related Use-cases (UCs)
FE-1	UC-1
FE-2	UC-1
FE-3	UC-1,UC-2,UC-3,UC-4,UC-5
FE-4	UC-1,UC-2,UC-3
FE-5	UC-1,UC-2,UC-4,UC-5

4. External Interface Requirements

4.1 User Interfaces

UI-1: The user interface layout shall contain a navigation bar in order to easily maneuver throughout the website.

UI-2: The bass logic website shall cleary indicate where to login or create an account on the website.

UI-3: The bass logic website shall have a moving navigation bar for easy user navigation for the user.

UI-4: The bass logic website shall have a student dashboard that will allow a user to request, modify, cancel, and view any current drum lessons that have schedules.

UI-5: A student shall be able to view instructors via collapsible biographies

4.2 Hardware Interfaces

HI -1: All devices that have access to the internet shall be able to access the Bass Logic website.

HI - 2: Each device that is connected to the internet will contain network adaptors

4.3 Software Interfaces

SI-1: Netlify - The Bass Logic website shall be hosted via the web servers provided by Netlifies web hosting service.

SI-2: Database - The database system shall be in NoSQL and hosted by Google Firebase real time database.

SI-3: Web Browsers - The Bass Logic website shall be accessed by Google Chrome, Mozilla Firefox or Microsoft Edge browsers. Since the site can be accessible via different browsers, any operating system can be used.

SI-4: Programming Languages -These browsers will be using HTTP, HTML, CSS, and JavaScript to communicate.

4.4 Communications Interfaces

CI-1: Network communication will be done through HyperText Transfer(HTTP) protocol in the form of a Transmission Control Protocol (TCP).

CI-2: A three way handshake will be performed via the website and the Firebase database. The Bass Logic Website will request information on drum lesson information, firebase will perform a lookup, Bass logic website will receive that information and display the currently scheduled lessons for that specific student that is logged in.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

PR-1: Must be secure and stable.

PR-2: Loads times for resources shall be 1.5 seconds or less.

5.2 Safety Requirements

SAR-1: Users must take standard safety procedures as any other website regarding sharing any login credentials.

5.3 Security Requirements

SR-1: The Bass Logic website shall not share any user information.

SR-2: The Bass Logic website shall require a valid email address upon creating a student account.

SR-3: The Bass Logic will implement an authentication system using the Firebase Authentication libraries from https://www.gstatic.com/firebasejs/9.17.1/firebase-auth.js

SR-4: All users will have a unique username and password.

SR-5: All changes to a user account will be logged and recorded using the firebase Authentication services.

5.4 Software Quality Attributes

SQ-1: The Bass Logic Website shall be accessible on Google Chrome, Mozilla Firefox and Microsoft Edge browsers.

SQ-2: The Bass Logic Website text and language shall all be in English.

SQ-3: The Bass Logic Website shall be accessible 24/7.

SQ-4: The Bass Logic Website shall process every function under 5 seconds or less.

SQ-5: The Bass Logic Website shall have clear indicators if an action is attempted that cannot be done.

SQ-6: The Bass Logic Website shall have alters if there is certain information missing that requires a function to correct be performed.

6. Other Requirements

OR-1: Must follow Industry Standards

7. Functional Requirements (FRs)

2.4.1 FR1: Student Visits Bass Logic Website

a. Introduction/Functionality: *When visiting the Bass Logic Website, the student shall be able to interact with all resources available on the website*

b. Traced : UC-1

c. Inputs: *Student selects the resource that they want to view*

d. Processing: *System takes inputs from mouse selection on which resources the student wishes to view.*

e. Outputs: *System provides user with correct user interface depending on the selection*

f. Error Handling : *If the website is down due to maintenance, the website will prompt the student.*

2.4.2 FR2: Create Account

a. Introduction/Functionality: *When visiting the Bass Logic Website, the student shall be able to create an account*

b. Traced : UC-1, *UC-2*

c. Inputs: *Student selects the the sign up option on the home page of the Bass Logic Website*

d. Processing: *System takes inputs from a sign up form that is filled out by Student.*

e. Outputs: *System generates account for Student and stores data in database*

f. Error Handling : *If the systems detects an invalid email is being used for sign up it will prompt the Student to enter a valid email.*

2.4.3 FR3: Create Scheduled Lesson

a. Introduction/Functionality: *When logged in to a student account, the student shall be able to schedule a private lesson with an instructor.*

b. Traced : UC-1, *UC-3*

c. Inputs: *Student fills out form based on the time and date they would like to schedule their lesson*

d. Processing: *System takes inputs from a form that is filled out by Student.*

e. Outputs: *System adds lesson to Student account. The next time the Student logs in, the system will display all current lessons scheduled.*

f. Error Handling : *If the system detects an invalid date selection the systems will prompt user to select a valid date to schedule lesson*

2.4.4 FR4: Modify Lesson

a. Introduction/Functionality: *When logged in to a student account, the student shall be able to modify a schedule for a private lesson with an instructor.*

b. Traced : UC-1, UC-3, UC-4

c. Inputs: *Student selects which lesson date they want to modify*

d. Processing: *System takes inputs by student of modification by Student*

e. Outputs: *System correctly adjust time and date based on modifications by Student*

f. Error Handling : *If the system detects an invalid date selection the system will prompt the user to select a valid date to schedule a lesson.*

2.4.5 FR5: Cancel Lesson

a. Introduction/Functionality: *When logged in to a student account, the student shall be able to cancel a scheduled lesson.*

b. Traced : UC-1, UC-5

c. Inputs: *Student selects which lesson they wish to cancel*

d. Processing: *System takes input selection by student*

e. Outputs: *System correctly cancels the lesson that was selected by the student*

f. Error Handling : *If the system detects there are no lessons to cancel in the first place, students will be prompted with the appreciate message.*

2.4.6 FR6: Sign in

a. Introduction/Functionality: *When at the Bass Logic website a Student shall be able to login to their account*

b. Traced : UC-1, UC-2

c. Inputs: *Student enters credentials on the sign in page.*

d. Processing: *System takes credentials and validates authentication*

e. Outputs: *System authenticates Student credentials and logs Student in*

f. Error Handling : *If the system detects incorrect credentials are entered it will prompt the student to check and correct credentials.*

2.4.7 FR7: Sign Out

a. Introduction/Functionality: *When logged in to a student account, the student shall be able to sign out.*

b. Traced : UC-1, UC-2

c. Inputs: *Student selects the log out option*

d. Processing: *System takes input selection by mouse coordinates*

e. Outputs: *System successfully sign out Student from account.*

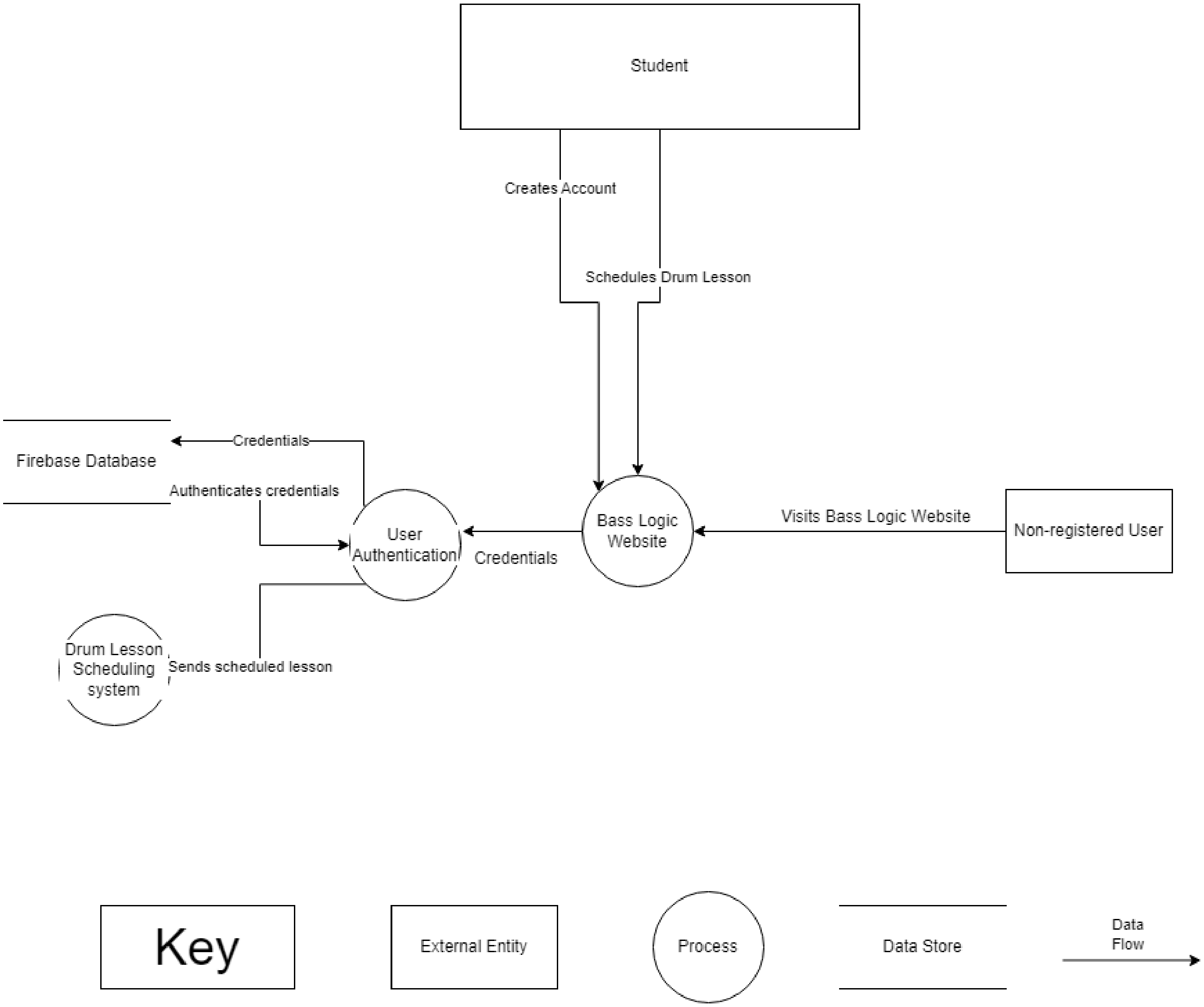
f. Error Handling : *If the system detects traffic interruption or connection to the internet is lost the system will sign the user out.*

TABLE 2: Traceability Matrix (Use-cases & FRs).

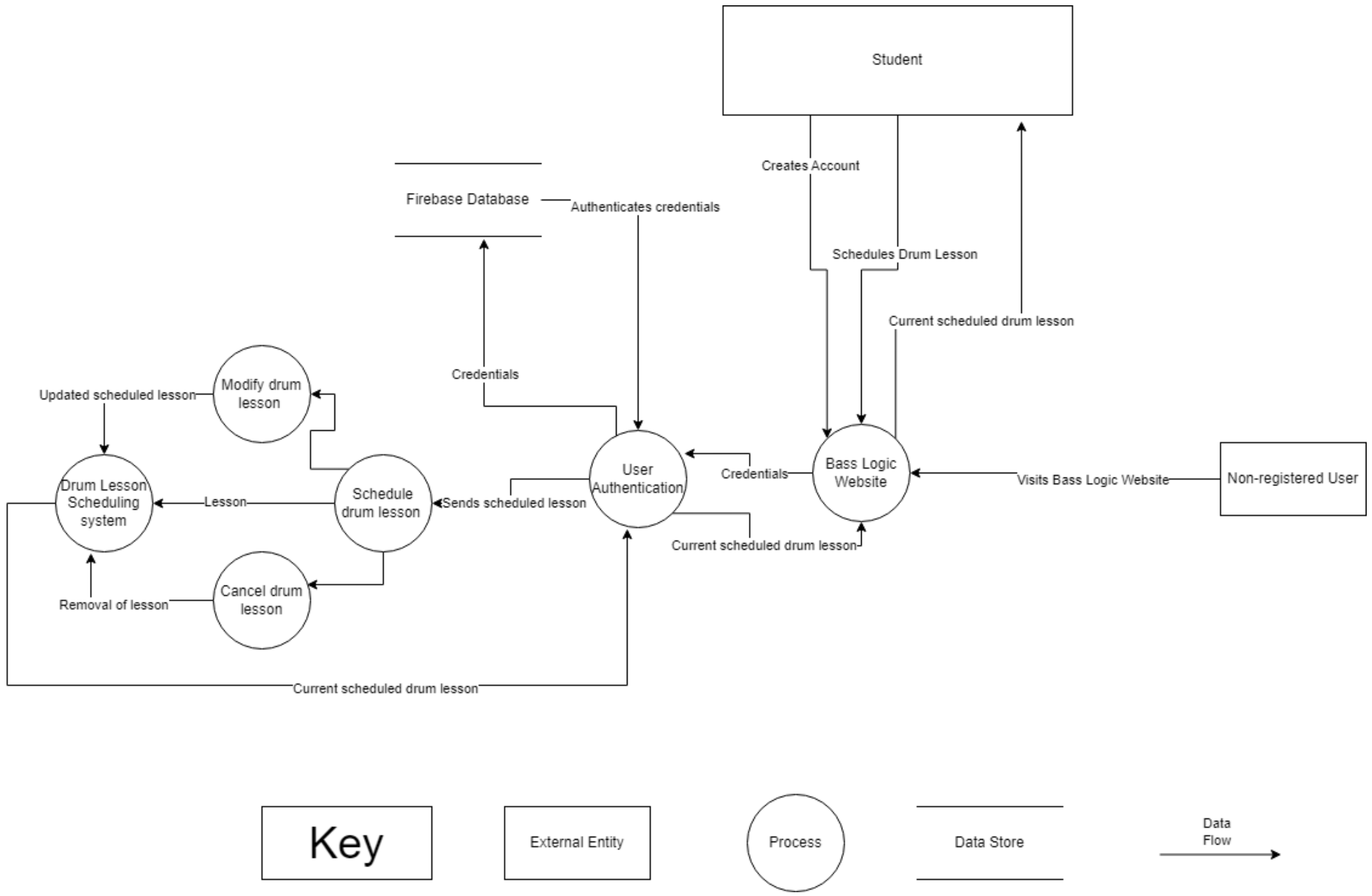
	Related FRs
UC-01	FR-1, FR-2,FR-3, FR-4,FR-5,FR-6,FR-7
UC-02	FR-1, FR-2,FR-3, FR-4,FR-5,FR-6,FR-7
UC-03	FR-1, FR-2, FR-4,FR-5
UC-04	FR-1, FR-2,FR-3,FR-4
UC-05	FR-1, FR-2,FR-3,FR-5

8. Functional Modeling (DFDs)

8.1 Context Diagram (level 0)

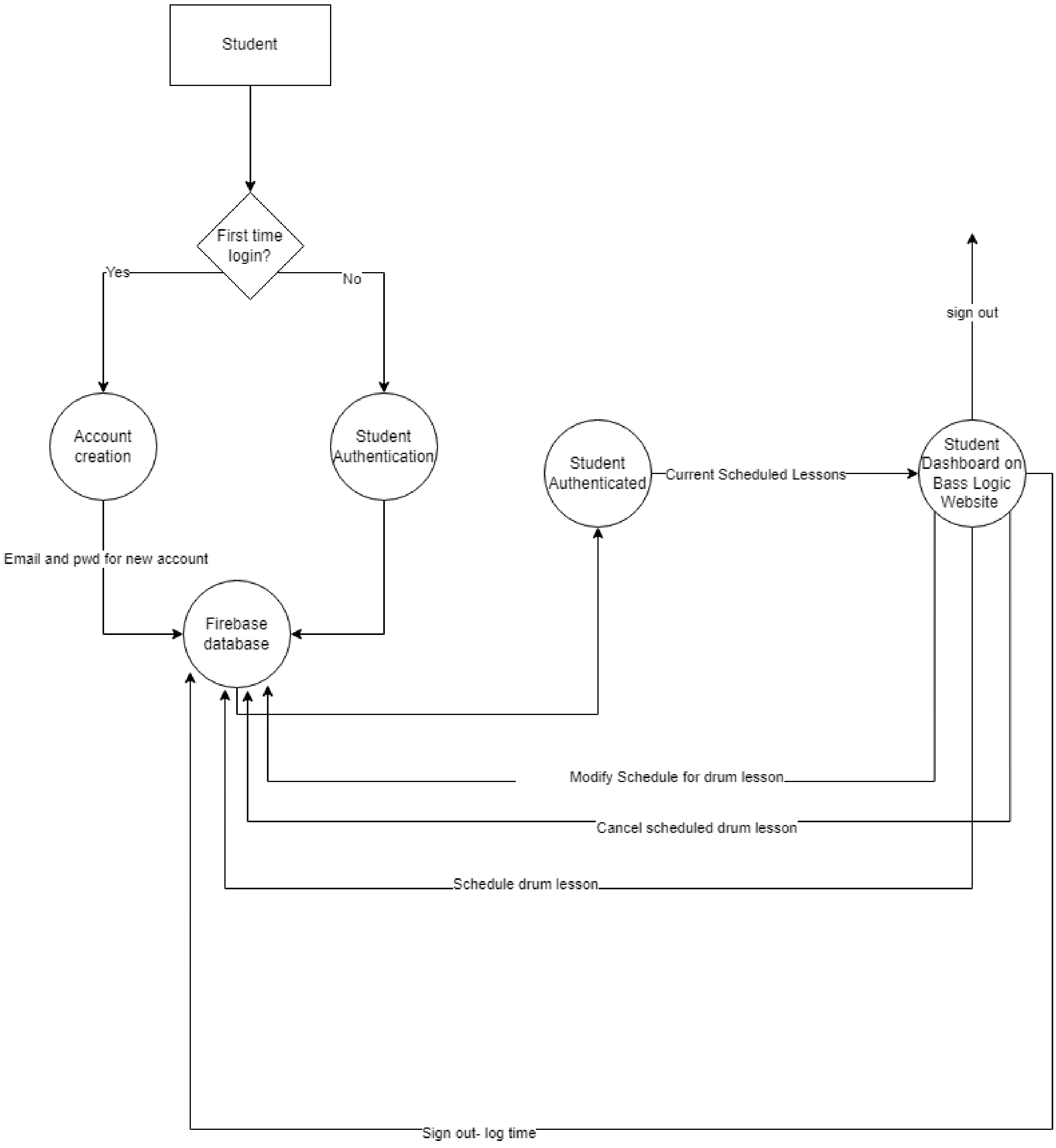


8.2 Level 1 DFD



8.3 Level 2 DFD's

8.3.1 Student login, logout and create, modify, and cancel lessons

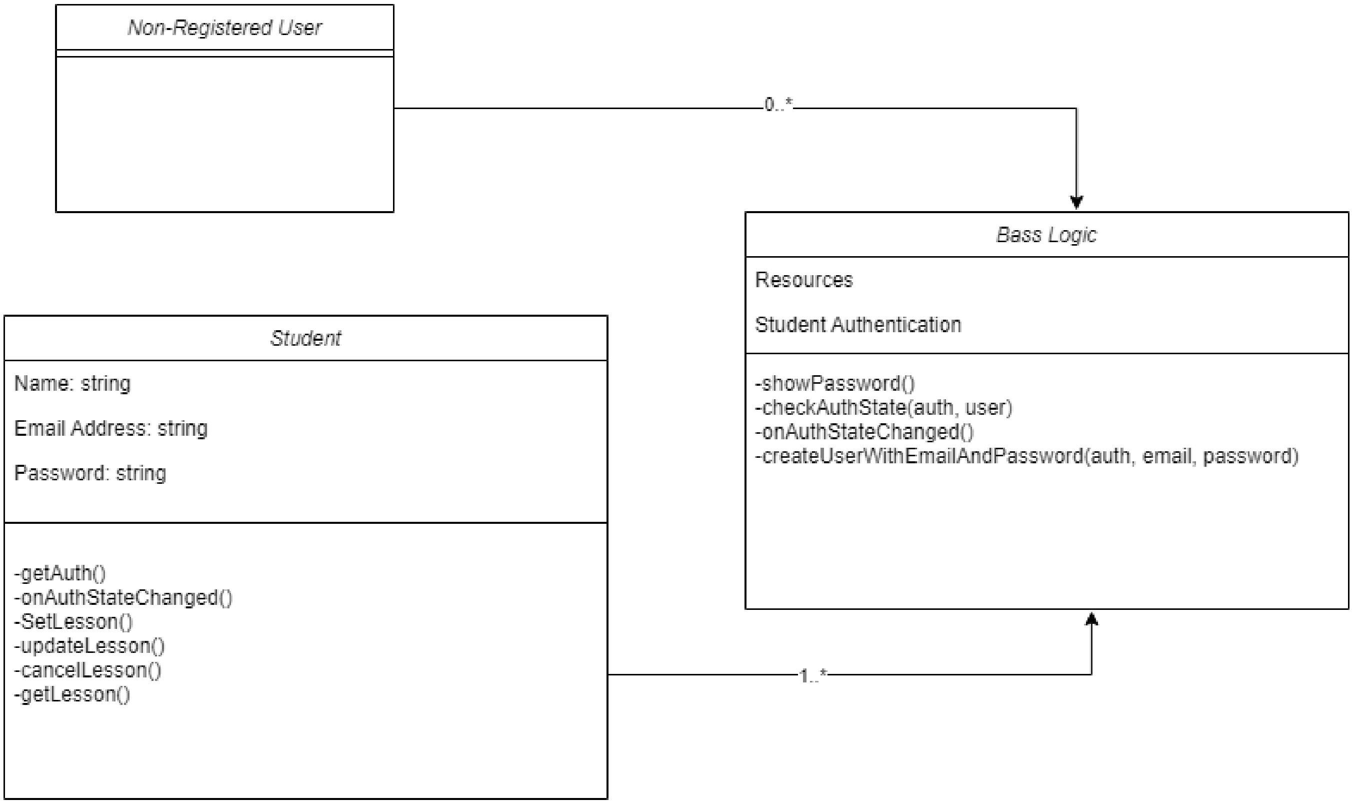

Student
First time login?
Yes
No
Account creation
Student Authentication
Email and pwd for new account
Firebase database
Student Authenticated
Current Scheduled Lessons
Student Dashboard on Bass Logic Website
sign out
Modify Schedule for drum lesson
Cancel scheduled drum lesson
Schedule drum lesson
Sign out- log time

8.4 TABLE 3: Traceability Matrix (FRs and DFD Processes)

	All Related Processes in DFD
FR1	DFD0, DFD1, DFD2
FR2	DFD0, DFD1, DFD2
FR3	DFD0, DFD1, DFD2
FR4	DFD0, DFD1, DFD2
FR5	DFD0, DFD1, DFD2
FR6	DFD0, DFD1, DFD2
FR7	DFD0, DFD1, DFD2

9. Class Analysis Modeling

9.1 Class Diagram


Non-Registered User
0..*
Bass Logic
Resources
Student Authentication
-showPassword()
-checkAuthState(auth, user)
-onAuthStateChanged()
-createUserWithEmailAndPassword(auth, email, password)
Student
Name: string
Email Address: string
Password: string
-getAuth()
-onAuthStateChanged()
-SetLesson()
-updateLesson()
-cancelLesson()
-getLesson()
1..*