

CHAPTER I

INTRODUCTION

1.1 Rationale

In today's rapidly evolving technological environment, advances in web-based applications have drastically changed various industries, including education. Integrating technology into educational strategies provides innovative solutions for enrollment and mentorship processes. These enhancements provide effective communication, increase accessibility, and provide an efficient management system to meet user needs [1]. The music education sector has also benefited significantly from technological advancements. Web-based platforms and applications have revolutionized how music is taught, learned, and experienced. Students can now access music lessons and mentorship remotely, overcoming geographical barriers. Studies show that integrating technology into music education improves engagement, supports diverse learning styles, and enhances the overall learning experience [2]. Furthermore, e-learning environments provide flexibility in scheduling lessons and tracking progress, ensuring that students receive tailored guidance based on their unique needs and goals. As music schools worldwide embrace these advancements, the opportunity to connect aspiring musicians with experienced mentors has never been more accessible. By leveraging technology, music schools can create an inclusive environment where students from diverse backgrounds, including those in remote

areas, can thrive. Herliz Music Studio and Instruments has always been at the forefront of nurturing musical talent. To continue its legacy of excellence, integrating a web-based application to streamline enrollment and mentorship processes will ensure that the studio remains competitive while addressing the needs of its growing community of students and mentors.

Located at JA Clarin, Tagbilaran City, Bohol, Herliz Music Studio and Instruments has been a cornerstone of music education since its establishment in 1998 and has received prestigious awards such as “Best Music School in Bohol” (Year 2012), “Excellence in Music Education” (Year 2017) and “Music Studio of outstanding recognition (Year 2019).” Herliz has educated more than 3,000 students (Year 2000 to 2022), more than 1,000 of whom have successfully completed their musical education. Each year, Herliz Music Studio operates with two seasons (April to July and September to December), each lasting four months, and classes are held Monday to Friday from 8 a.m. to 12 a.m., which forms the core schedule around which the enrollment and mentorship processes are currently organized. The studio caters to a variety of music disciplines, including Violin, Piano, Acoustic Guitar, Electric Guitar, Drums, and Pop Singing. In 2019, the founder's daughter took over the leadership of Herliz. Under her leadership, the studio continues to maintain its tradition of excellence while also exploring modern methods to improve the enrollment and mentoring process check-in, which requires students or parents to physically visit the studio or check in via their Facebook page. Mentors at Herliz who are full- time employees are responsible for providing structured, curriculum-based instruction, lessons, and sessions, as

well as tracking student progress. In contrast, freelancers offer one-off, specialized sessions tailored to specific skills or genres, providing students with a broader range of learning opportunities. Mentoring sessions are managed manually with limited tools for streamlining planning, communication and monitoring progress. These traditional processes, while effective in the past, now present challenges in terms of availability, efficiency, and scalability, especially for students from remote areas or those with busy schedules. This project aims to solve these problems by developing a web-based system that integrates registration and mentoring functions. The system aims to simplify the registration process for new and returning students while providing mentors with a platform to schedule appointments, communicate with students and track their progress. Through technology, the studio can expand its reach and ensure a more efficient and accessible experience for all users.

Despite its success, as evidenced by awards and the significant number of students it has nurtured (over 3,000 enrolled and 1,000 graduates), Herliz Music Studio faces challenges in its current processes. The reliance on in-person registration requires students or parents to physically visit the studio, which limits accessibility for those who live in remote areas or have tight schedules. While the studio also accommodates inquiries through its Facebook page, there are occasional delays in responding due to the high volume of messages, which may cause inconvenience for potential students. For example, responses may take hours or even days during busy periods. While these methods have served their purpose for years, they are no longer sufficient to meet the demands of an

expanding student base and evolving customer expectations. Additionally, the process of finding qualified mentors and matching them with students involves several manual steps, such as assessing mentor qualifications and manually pairing them based on instrument specialization or other criteria. While it is essential to ensure that mentors are well-qualified, the current approach becomes time-consuming as the number of students increases. For instance, during peak enrollment periods, administrators often struggle to balance the need for personalized pairings with the growing demand for lessons. Mentorship at Herliz is conducted primarily through in-person lessons, which follow a synchronous learning model—mentors and students meet at scheduled times for one-on-one or group sessions. However, as the studio expands, there is a growing need for hybrid learning solutions to cater to diverse schedules and locations. The proposed system will support both synchronous and asynchronous (hybrid) lessons. Students can attend live online sessions with their mentors or access pre-recorded videos and exclusive learning materials created by the mentors of Herliz Music Studio. This hybrid approach allows flexibility while maintaining the studio's high standards for music education. To address these challenges, the proposed ConnectingNotes web application will integrate enrollment and mentorship functionalities into a single, accessible platform. The system will streamline enrollment by allowing students to register online, select their preferred mentors based on availability and specialization, and receive immediate confirmation. It will also provide tools for mentors to manage their schedules, communicate with students, and upload exclusive video lessons and other learning materials. By

automating these processes, the system ensures a more efficient and accessible experience for both students and mentors, ultimately enabling Herliz to cater to an even larger audience without compromising quality.

With the advancement of today's technology, the development of ConnectingNotes is proposed to address the challenges currently faced by Herliz in its enrollment and mentorship processes. The proposed system is a web-based application that provides innovative solutions. This platform will offer features such as online registration, mentor application, profile management, session booking, messaging systems, rating systems, hybrid lessons (combining synchronous and asynchronous learning), progress monitoring, and issuing achievements (badges) and certificates. The hybrid lesson model will provide flexibility for both mentors and students. Synchronous lessons will involve live, interactive sessions conducted via video conferencing tools, allowing real-time feedback and mentoring. Asynchronous lessons, on the other hand, will include pre-recorded video tutorials and instructional materials exclusively created and curated by Herliz Music Studio. These materials will be accessible through a secure, centralized repository within the system. Mentors will have the capability to upload these resources directly to the platform, ensuring that students have consistent access to quality, studio-approved learning materials. The inefficiency of the current manual processes is supported by historical data. Over the past five years, Herliz has experienced an increase in enrollment inquiries, particularly during peak periods such as summer and holiday seasons. The existing Facebook messaging-based system often results in delayed responses due to the high volume of

inquiries, causing potential students to wait for hours or even days. Furthermore, matching students with mentors based on instrument specialization, schedule, and other criteria is a time-consuming task that relies on manual coordination by administrators. This issue has led to missed opportunities to accommodate more students effectively. By integrating features such as real-time session booking and automated mentor-student matching, the system will streamline these processes and reduce delays. The hybrid lesson model will further expand accessibility, enabling students from remote areas or with busy schedules to participate in Herliz's music programs without the need for frequent physical attendance. Through these functionalities, ConnectingNotes aims to create a simple, centralized, and efficient system that enhances both enrollment and mentorship, aligning with the studio's vision of providing accessible, high-quality music education.

The proposed solution will not only simplify the enrollment process, but also improve mentor-student pairing, strengthen communication, and provide a structured learning path tailored to individual student needs. Over the years, Herliz has built a legacy of success, educating over 3,000 students since its inception in 1998, with over 1,000 students successfully completing their music training. This historical data supports the reputation of Herliz as the premier music studio in Bohol. To ensure the development of a system that truly responds to the studio's needs, data were gathered through interviews and informal discussions with the current administrator, full-time mentors, secretaries, and selected students of Herliz Music Studio. Prior to the data gathering, proper consent was requested

verbally and via email to ensure ethical compliance. Furthermore, the system's development will prioritize the confidentiality and privacy of all user data. Sensitive information collected during registration, mentoring sessions, and communication logs will be securely stored and only accessible to authorized personnel. The platform will implement standard security practices such as encrypted data transmission, password-protected access, and role-based permissions to protect personal information and uphold data privacy. Mentor-student pairing in the proposed system will be automated using a matching algorithm that takes into account specific criteria such as instrument specialization, skill level, preferred schedule, and location (for hybrid settings). This ensures that students are matched with mentors who best match their educational goals and availability, simplifying what is currently a manual and time-consuming process. To improve communication, the system will integrate a built-in messaging feature to enable seamless and direct communication between students, mentors and administrators. This feature eliminates dependency on external platforms such as Facebook Messenger and ensures timely and secure exchanges. Notifications and reminders of upcoming sessions, payments or notices will also be sent through the system to keep all users informed. The structured learning path offered by the platform is designed to adapt to student progress. The system will allow mentors to assess students' skills, monitor their achievements and recommend next steps on their educational journey.

This adaptive approach ensures personalized learning experiences, allowing students to progress at their own pace while meeting predefined

milestones. The structured path will include predefined course modules, exclusive instructional videos created by Herliz, and tailored exercises that align with the student's current level and goals. Ultimately, ConnectingNotes will help Herliz maintain its legacy of success by making music education more accessible, personalized, and efficient for all users, while continuing to nurture a new generation of musical talent.

1.2 Objectives of the Study

The primary goal of this study is to design and develop the ConnectingNotes web application to streamline and enhance the efficiency of the enrollment, mentorship, and learning processes at Herliz Music Studio and Instruments, facilitating a more accessible and effective experience for students, mentors, and administrators.

Specifically, the study aims to:

1. Design a user-friendly, web-based application that integrates the enrollment and mentorship processes, enabling online registration, mentor-student pairing, and session scheduling for students and mentors at Herliz Music Studio.
2. Develop a secure platform that facilitates both synchronous and asynchronous learning, enabling students to access live and pre-recorded lessons from mentors, as well as track their progress and achievements.

3. Automate the mentor-student matching process using an algorithm that factors in criteria such as instrument specialization, skill level, preferred schedules, and location to ensure optimal pairings and efficient scheduling.
4. Enhance communication and user experience by incorporating real-time messaging, notifications, and reminders for students, mentors, and administrators, ensuring seamless interactions and timely updates throughout the enrollment and learning processes.

1.3 Scope and Limitations of the Study

The proposed study focuses on enhancing the enrollment and mentorship processes at Herliz Music Studio and Instruments. The research will be localized to the operations of Herliz, situated in JA Clarin, Tagbilaran City, Bohol, and will encompass various stakeholders, including students, parents, mentors, the secretary, and administrators. This study aims to evaluate how the web-based system improves efficiency and accessibility. It will not replace the existing in-person registration process but will provide an online alternative for registration. Walk-in customers will still require some level of manual intervention by the secretary for specific in-person services or actions.

The study will also exclude certain operational activities that are outside the scope of the system itself. For instance, the organization of in-person events such as recitals and performances will not be considered within the study as they are

managed offline. Additionally, advertising activities conducted by Herliz to promote its services are not part of this study, as the focus is on optimizing the system for the enrollment and mentorship processes rather than marketing.

The scope of the study will be limited to verifying the functionality and usability of the proposed system, ensuring compatibility across different browsers, testing the security measures to protect user data, and assessing content accuracy. The study will not include 1-on-1 live sessions or manual feedback collection via face-to-face interactions, as these aspects fall outside the system's functionality.

Regarding data privacy and confidentiality, the study will adhere strictly to the Data Privacy Act of 2012. All personal data collected from students, parents, and mentors will be securely stored and will only be accessible to authorized personnel. Data will be handled with the utmost confidentiality, ensuring that no unauthorized party will have access to any sensitive information. Participants will be fully informed of their rights and the handling of their data through consent forms before participating in the system's usage.

To ensure that users' privacy is maintained throughout the study, the team will implement robust security measures such as encrypted data storage and secure access protocols. Additionally, any data breaches or unethical practices detected during the study will be addressed promptly to uphold the integrity and trust associated with Herliz Music Studio's operations.

Data for this study were gathered through interviews and consultations with key stakeholders at Herliz Music Studio, including the administrators, mentors, and

students. These participants were selected for their direct involvement with the enrollment and mentorship processes. Additionally, feedback from parents was collected to understand their perspectives on the online registration system. Observations of current manual processes and their limitations were also made to inform system design.

Before conducting the data-gathering activities, informed consent was obtained from all participants to ensure compliance with privacy and ethical standards. The research team outlined interview questions and shared them in advance with participants to ensure transparency and consistency. The objective and benefits of the study were communicated clearly to all participants to ensure full understanding and voluntary participation.

The focus of the study will be to evaluate and test the features, usability, and effectiveness of the system in meeting Herliz's enrollment and mentorship needs, ensuring it aligns with the studio's current goals and technological capabilities.

CHAPTER II

REVIEW OF RELATED LITERATURE

This chapter features both national and foreign literature. To ensure a more precise study, the researchers collected this literature from online journal sites. By reviewing this literature, the researchers gained a wider perspective and a deeper understanding of the gathered topics.

The study of Jamshidi and Marghitu is a web-based platform designed to facilitate remote music education by offering tools for music learning, sharing materials, and conducting virtual classes. Their platform emphasizes on making music teaching and learning more accessible, engaging, and fun for students through multimedia integration, providing an immersive audiovisual environment. [3]. The study of Jamshidi and Marghitu shows some notable features, such as score following, a technology that tracks a student's performance against a predefined score in real time, which is crucial for music accompaniment. A simplified score-following system can provide visual and auditory feedback to students during practice, enhancing their self-assessment. This feature is often paired with pitch tracking, which uses tools like the CREPE pitch detection algorithm to deliver precise real-time feedback. Their Music learning platform features different learning modes, such as Wait Mode, Slow Mode, and Loop Mode, on their virtual music sheets and videos. Allowing learners to adapt the lesson tempo, focus on difficult sections, and practice iteratively. Such innovations

can help students from ConnectingNotes refine their skills as they practice. However, their system does not have features like ConnectingNotes has to offer, such as a mentor management module that tracks student progress and allows personalized feedback. It also offers a session booking system that prevents scheduling conflicts, a messaging system for direct communication between students and mentors, and automated achievement issuance based on completed milestones. Additionally, it integrates a rating and review system to gather feedback on mentors and sessions, providing a more interactive and personalized learning experience. These features make ConnectingNotes a more robust and student-centered platform. Still, there are features from their study that can be utilized, integrating features like metronomes or interactive music sheets can enhance learning sessions. Adding such tools to ConnectingNotes would help students practice effectively between mentor sessions, boosting their progress.

Askeroth explored online learning innovations such as digital badges and Massive Open Online Courses (MOOCs). Integrating digital badges and MOOCs into educational platforms has revolutionized the learning landscape, offering scalable, accessible, and motivational frameworks for diverse learners [4]. Digital badges provide verifiable credentials for specific skills or achievements, incorporating gamification elements such as level progression, points, and leaderboards to track progress. MOOCs use asynchronous course delivery, combining video lessons, interactive discussions, and automated assessments for self-paced learning. These assessments utilize question banks, randomized quizzes, adaptive assessments, and automated grading tools with natural

language processing (NLP) and pattern recognition to evaluate performance and provide instant feedback. MOOCs typically rely on Learning Management Systems (e.g., Moodle), cloud infrastructures for scalability, and content delivery networks (CDNs) for efficient video streaming. In ConnectingNotes, badges can be issued based on specific milestones like mastering a musical technique or completing mentorship hours. MOOC features that can also be implemented include video tutorials hosted on scalable cloud services like Amazon Web Services (AWS), adaptive streaming to adjust video quality based on network conditions, and peer reviews using assessment plugins. RESTful APIs would allow badge and course data integration with external platforms like LinkedIn. These features would enable an efficient, scalable, data-driven platform for tracking and validating student progress in ConnectingNotes.

Legaspi developed a web-based course scheduling system that uses the Greedy Algorithm to automate time-slot allocation, room assignments, and faculty scheduling [5]. The greedy Algorithm works by finding the earliest available time slot, assigning appropriate rooms, selecting qualified faculty, and ensuring a conflict-free schedule through iterative optimization. From this, the Greedy Algorithm can automate the allocation of mentorship sessions by assigning mentors to time slots based on immediate availability and predefined factors such as mentor workload, student preferences, and session duration. By implementing this algorithm, the system can automatically resolve conflicts, optimize session utilization, and reduce manual scheduling errors, ensuring smooth and efficient operation. For example, when a student books a session, the system can

automatically search for an available mentor within specified time constraints and assign the session to avoid conflicts. Their implementation of Manual and Override Scheduling allows flexibility when automated schedules are not feasible. In ConnectingNotes, this feature can empower administrators or mentors to manually adjust schedules if needed. Their Copy-Previous-Schedule functionality can simplify recurring mentorship sessions by replicating previous session structures, reducing repetitive scheduling tasks. By adopting these technical approaches and system features, ConnectingNotes can deliver an efficient scheduling system, optimizing mentorship session management while minimizing conflicts and manual interventions.

The study "College Vocal Music Teaching Design Based on Internet Platform" introduces a five-tier system architecture comprising an infrastructure layer, application interface layer, application layer, user layer, and a cloud curriculum platform [6]. The infrastructure layer uses a B/S (Browser/Server) model for backend processes, while the application interface layer supports dynamic web pages and database integration. The application layer handles system maintenance and user management, and the user layer offers functionalities like information management and access control. Lastly, the cloud curriculum platform hosts courses and materials, modernizing vocal music education. Their system also employs technologies such as MySQL, Linux, Resin servers, and dynamic query execution for efficient database operations, alongside advanced algorithms like entropy-based indexing for data search and learning activity management. To adopt these features in ConnectingNotes, the modular

five-tier architecture can enhance scalability and clarity in managing components. For instance, using a B/S computing model can optimize backend processes in an online mentoring system by centralizing tasks like user authentication, session management, and data analysis on the server. This reduces client-side load, ensures scalability, and simplifies maintenance and updates. It also improves data security by keeping sensitive information on the server while employing a solid database that can support user registration, course management, and progress tracking. The application layer can integrate logic rules for data analysis to enhance student progress tracking, mentor matching, and learning activity management by utilizing Conditional rules that can automate actions like awarding certificates based on session completion and scores. Their use of Aggregation helps compile student progress data to identify performance trends, such as which courses have high completion rates or where students struggle the most. This helps administrators refine course structures. Their Anomaly detection logic rule Flags students who haven't logged in or submitted progress for a set period (e.g., 7 days). The Classification logic rule categorizes students and mentors for personalized recommendations and provides segmented access for students, mentors, and administrators, ensuring data privacy and smooth platform navigation.

Effective music pathways for instrumentalists at beginner, intermediate, and advanced levels involve tailored instructional methodologies, improvisation techniques, and composition opportunities. These pathways are designed to enhance musicianship and encourage creativity across levels [7]. The study of

Douglas, T., Owens. Shows Effective Pathways for Instrumental Music Education across Proficiency Levels. For beginners, utilizing repetitive sequencing steps aids in mastering fundamental concepts such as chord arrangements and scales, particularly for string instruments like guitars. Listening to examples while learning jazz styles further supports the development of basic rhythmic and harmonic understanding. At the intermediate level, students benefit from improvisation and composition projects that encourage the exploration of melodic and rhythmic interpretations, thus enhancing their creative skills. Additionally, introducing expanded harmonic ideas, including modes and advanced chord progressions, allows students to delve deeper into musical exploration and comprehension. For advanced learners, exposure to diverse musical styles and contemporary techniques fosters experimentation and innovation in composition, enabling them to push creative boundaries. To integrate these instructional methodologies into ConnectingNotes, beginner pathways can feature step-by-step tutorials on foundational skills like scales and chords, supported by pre-recorded examples of styles such as jazz rhythms. A progress tracker can reward completed lessons with badges. Intermediate pathways can include improvisation exercises and lessons on modes and chord progressions, allowing students to apply these concepts in their compositions and share work with mentors or peers for feedback. Advanced pathways can offer project-based learning, such as composing full pieces or exploring diverse musical styles through masterclasses. Additional flexible features can include customizable learning paths, mentor-guided adjustments, and adaptive assessments to recommend personalized next steps.

The study provided by Matevž, P., Franc K., Peter S., and Matija M., focused on the development and three-month assessment of the Troubadour Platform, an open-source music theory ear training tool [8]. Key features were identified through teacher interviews to improve the platform's functionality, leading to the implementation of interactive elements such as class management tools, recurring homework assignments, and challenges. Initial research demonstrated significant improvements in student performance with the platform, though the short-term nature of previous experiments raised concerns about potential novelty bias. To address this, a comprehensive three-month evaluation was conducted, gathering data through student questionnaires and platform analytics. Previous research indicated significant improvements in student performance while using the platform. Additionally, the study analyzed students' self-reported practice habits and compared these with the data collected from the platform. The results show the need for flexible and adaptable learning tools that can complement traditional in-class education, providing students with continuous learning opportunities regardless of the setting. By integrating the insights gained from this evaluation, the Troubadour platform can further refine its features to better support music education across multiple learning environments. Furthermore, these findings can be applied to other online music education platforms such as ConnectingNotes by incorporating key features like personalized learning pathways, real-time feedback mechanisms, and a comprehensive evaluation framework. By leveraging the insights from this study, ConnectingNotes can create a versatile and effective learning environment that

addresses current educational needs and continuously improves based on user feedback and data analysis.

The study of Yuna Chen and Abdul Rahman Safian has the changing context of online learning in vocal music education, giving particular emphasis to interventions that can be supported using multimedia resources to enhance access, flexibility, and engagement [9]. Traditional music educators have faced challenges adapting to digital platforms, emphasizing the need for digital literacy and appropriate infrastructures. The study suggests that while online learning offers significant advantages, integrating digital tools with traditional methods is essential. Future research should be directed towards the long-term impact of online learning on skill development and creative growth, as well as toward the optimized platforms for real-time musical collaboration, advocating for a harmonized approach that blends technology with foundational instructional practices. Such findings are highly pivotal for our study on ConnectingNotes. By making use of insights on accessibility and engagement, addressing potential challenges, taking up a balanced blend of digital and traditional methods, and exploring future research directions, ConnectingNotes can become a more robust and effective platform for online music education. By implementing these strategies, including multimedia resources for engaging content and a user-friendly interface to accommodate varied digital literacy levels, ConnectingNotes will be sure to stand out as a comprehensive tool for educators and students alike. Additionally, tracking long-term student progress and collecting data on the effectiveness of the platform will help us refine and improve our tools

continuously. Exploring real-time musical collaboration features such as synchronized practice sessions, virtual jam sessions, and seamless integration with existing music software will improve the interactive and community aspects of music learning on ConnectingNotes.

The Mentor Management System is an innovative Java-based Database Management System (DBMS) web application designed to enhance mentor-mentee interactions and administrative oversight in academic institutions [10]. Addressing the growing demand for efficient mentorship collaborations, the system aims to streamline connections between mentors and mentees while optimizing the overall mentorship process for students, mentors, and administrative personnel. The Mentor Management System integrates several technical features that can help enhance ConnectingNotes by improving automation, communication, and user management. A leave management system in MMS automates mentor absence tracking using relational databases and a dynamic calendar interface, where leave requests trigger status changes like mentor availability in ConnectingNotes and substitute mentor assignments through scheduling algorithms. Implementing this in ConnectingNotes would improve session continuity by updating mentor availability and notifying impacted users automatically. This feature is particularly useful when a mentor is unable to attend a scheduled in-person session. Their batch assignment functionality utilizes batch processing and role-based access control (RBAC) to efficiently assign multiple students to mentors or groups, which can be adapted for managing ensemble classes or collaborative learning paths in ConnectingNotes. Lastly, MMS's real-

time notifications are powered by event-driven messaging services and push notification APIs such as FCM or OneSignal, which could be directly relevant for ConnectingNotes to manage mentor reminders and student session alerts, ensuring timely reminders for sessions, progress updates, and deadlines. By integrating these technical elements, ConnectingNotes can enhance its mentorship framework with improved automation, efficient resource allocation, and proactive communication tools.

The study by Andrade presents a web platform designed to support the Mentoring Academy, leveraging technologies like ASP.NET Core for backend development, MySQL for database management, and Angular 6 for the front end [11]. The system implements Single Page Applications (SPAs) architecture, which provides dynamic content updates without refreshing the entire page. Key functionalities include manual and automatic mentor-mentee matching using rules like location and language, integration of graphs to visualize mentoring statistics, and automated email notifications for meeting scheduling and confirmations. Their study introduces features that can enhance ConnectingNotes, such as an automated mentor-mentee matching system using predefined rules based on skill level, subject focus, schedule compatibility, language preference, and student learning goals for optimized pairings. Implementing a similar algorithm that considers expertise, learning styles, and availability would improve efficiency. Their system's dynamic data visualization offers insights into progress and engagement. A progress dashboard in ConnectingNotes would allow users to track milestones, session completion rates, and performance trends over time.

Additionally, automated email notifications for scheduling and reminders could be upgraded in ConnectingNotes with push notifications via APIs like Twilio or Firebase for real-time updates. Lastly, adopting Single Page Application (SPA) architecture with React.js or Angular would enhance user experience with faster, smoother interactions.

CHAPTER III

METHODOLOGY

3.1 Analysis

This section discusses how users interact with the ConnectingNotes system, presenting an overview of its functionalities and modules. The analysis is accomplished through the use case diagram and use case narrative. The use case diagram visually represents the relationships between actors and the system, identifying their respective actions that trigger system processes. The use case narrative provides a detailed explanation of these processes, ensuring clarity on system workflows.

This analysis is accomplished through the utilization of the system use case diagram and use case narrative. The use case diagram presents the relationship between the actors and the system, therefore, actors and their respective actions which trigger the processes of the system are identified. The system involves four (4) actors, the Administrator, the Secretary, the Mentors, and the Students, and there are five (5) processes identified in the use case diagram (refer to figure 3.1.1): Manage User Accounts, Manage Mentors, Manage Mentoring Sessions, Manage Achievements, and Generate Reports.

The Manage User Account process involves all the actors on the said platform. The Administrator, who is the owner of Herliz Music Studio, is allowed to create the Secretary's accounts, view and update their own account. The

Secretary is allowed to create the walk in clients' accounts, view and update user account's registration status, including updating their own account details. The Students must create an account in order for them to register into Herliz Music Studio through the system and they are also allowed to view, update personal details, and password on their own account. The Mentors must create an account in order for them to apply in Herliz Music Studio through the system where they can choose a user type either "Mentor" or "Freelancer" and they are also allowed to view, update personal details, and password on their own account.

The Manage Mentors involves the Administrator, the Secretary, and the Mentors. The Administrator oversees the entire application process for mentors and freelancers. This includes reviewing applications, setting a date for the interview by messaging each other through the platform, conducting interviews either online or onsite, depending on the applicant's accessibility, and making the final decision to approve or reject the applicant. Accepted applicants will be notified via email and will proceed with the onboarding process, where they will be granted system access and introduced to the studio's policies, teaching guidelines, and administrative procedures. Rejected applicants will also receive email notifications, and their accounts will be automatically deleted from the system within 24 hours to ensure data privacy and security. Throughout the process, the Administrator tracks the progress of applications and communicates with applicants via email. The Secretary assists the Administrator by verifying documents, tracking application progress, organizing communication, and setting an interview. The Secretary is allowed to filter applications by status and date. The Mentors are

responsible for providing structured, curriculum-based instructions, lessons, and sessions, while the freelancers conduct one-off sessions such as seminars or workshops. In order to apply a position in Herliz Music Studio, they must submit their application with the necessary documentation and portfolio details, through the system. They can track the status of their applications in real-time and receive notifications through email about the updates of their application.

The Manage Mentoring Sessions involves all the actors on the said platform. The Administrator manages online students registration, including registration periods, required documentation, and fee structures. The Administrator also has an overview of all sessions and access to session logs that automatically record attendance, session dates, and topics. The Secretary can also manage online and walk-in registration, which includes the verifying of submitted documents such as filled up form, birth certificate, and parents' consent that will be available online and offline and processing required payments, full payment or 50% down payment. The submitted documents submitted by walk-in clients will be encoded into the system by the Secretary, and the documents will be stored into the system. The secretary facilitates assigning students to mentors or courses based on preferences, availability, and Herliz Music Studio's policies. Assignments are criteria-based but can also allow manual intervention for special cases. The system provides the Secretary and Administrator with a dashboard that tracks registration statuses, payment confirmations, and document submissions. The Students can register through the system by submitting the documents needed, with Mentors based on their availability, view session details, receive

reminders, and provide feedback. If students miss a session, they must submit an excuse letter through the system although the students can catch up on lessons they missed since all lessons will be posted through the system. The system offers a guided registration interface with step-by-step instructions, update of successful registration, and integration with payment gateways such as Credit Card, PayPal, GCash for processing fees. The Mentors manage their course offerings and availability for new student registrations since they are only 20 students max per session. They can view and approve student registration requests for their sessions, ensuring they can manage their time effectively. These are pre-scheduled events managed by the Secretary. The system provides mentors with a calendar view of their sessions and students' view to see who and how many students they have, allowing them to manage their schedule efficiently. The system supports hybrid learning, combining synchronous tools like video conferencing and asynchronous resources such as recorded lessons and sheet music. A communication feature includes in-system messaging, email notifications, and calendar integration for effective coordination among users. The mentoring sessions cater to a variety of music disciplines, including Violin, Piano, Acoustic Guitar, Electric Guitar, Drums, and Pop Singing, ensuring a comprehensive music education experience for students. Herliz Music Studio operates with two seasons from April to July and September to December, providing structured learning schedules within these periods.

Manage Achievements involves the Administrator, the Mentors, and the Students. The Administrator oversees the issuance of achievements and

certificates, ensuring that the system supports the process effectively. While the Mentors are responsible for setting, creating, and managing the specific achievement types that students can earn, the Administrator will manage the technical aspects of the system, such as generating certificates and integrating digital signatures to maintain authenticity that will be given on the day of the recital. The Administrator ensures that achievements are awarded based on the criteria where they can view the achievements/badges set by mentors and what the students earned, providing transparency and a clear path for students to follow to earn additional recognition and motivation. The Mentors are responsible in issuing achievements to students based on their progress and performance. They can notify their students what they can accomplish through their progress and award badges and achievements for specific accomplishments. The mentors can track student progress and generate digital certificates with digital signature, which will be stored in the student's profile. The Students can view their achievements through their personal dashboards, where they can also view their progress and any badges or certificates they have earned. In every achievement that the students can achieve on their progress, it will automatically update in their achievement records that will be displayed in the student's dashboard. Students can download their certificates directly from the platform.

Generate Reports involves the Secretary and the Administrator. The Secretary is responsible for generating reports per session classified to two seasons per year, four months per season, Monday to Friday classes from 8am to 11am, related to various aspects of the system such as registration statistics,

earnings paid by clients, non-continuing students, and courses or services taken. The generated reports will be shown in a graph/chart. The Administrator is provided with a dashboard to view the reports that are generated by the Secretary, and the Administrator can select what he/she wants to view.

To summarize, in the proposed system, the Administrator will have the capability to:

- Create (including the Secretary's account), view, and update their own account.
- Validate, view, approve, or reject Mentors and Freelancers applications, and manage their workflow.
- Manage online registration and registration settings
- Access and monitor session logs
- Oversees the entirety of achievement issuance and generates certificates with digital signatures for authenticity.
- View generated reports by the Secretary (registration statistics, earnings, non-continuing students, and courses or services taken)

The Secretary will have the capability to:

- Add, view, and update user accounts (especially for walk-in clients). Assists in organizing all mentors' and freelancers' applications
- Verify, track, and process registration payments, including payment plans (full payment or 50% down payment).

- View schedule of mentors and students
- Generate reports (registration statistics, earnings, non-continuing students, and courses or services taken)

The Students will have the capability to:

- Create, view, and update their own account
- Register for courses, select mentors, schedule sessions, and process payments.
- Book and manage sessions. Also provide feedback
- View, download/share, and receive updates about their achievements

The Mentors will have the capability to:

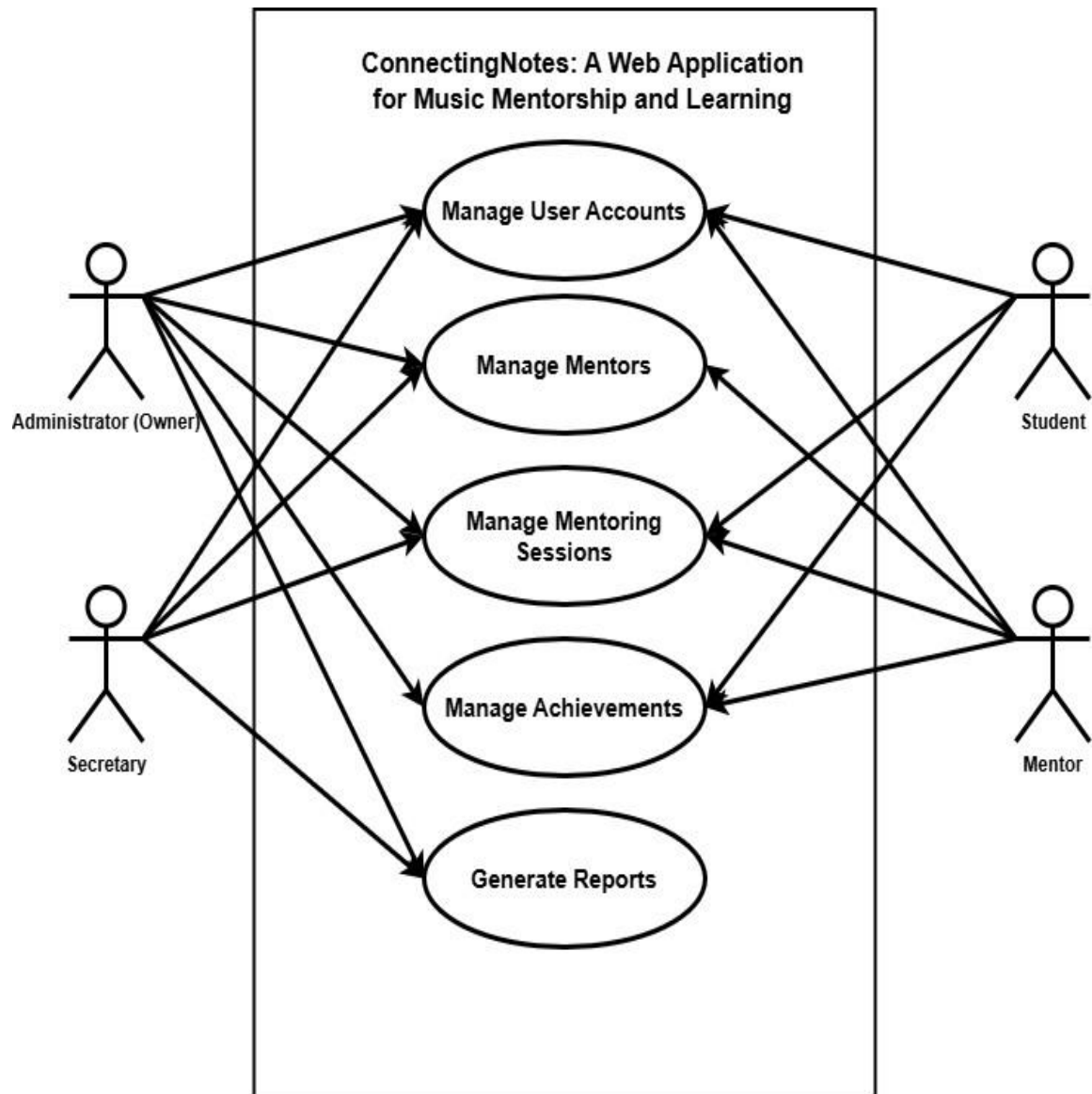
- Create, view, and update their own account
- Submit documents for application, track, and receive notifications regarding the application.
- Manage their course offerings and availability for student registrations
- Manage their session schedules, confirm student bookings, conduct sessions, and provide feedback to students.
- Conduct and view one-off sessions, such as seminars or workshops.
- Award students with badges/achievements, track progress, and generate certificates (with digital signature).

Overall Capabilities:

- Seamless account creation and verification.
- Comprehensive management of applications, registrations, mentoring sessions, achievements, and reports.

- Hybrid learning capabilities for flexibility and accessibility.
- Integrated communication features for streamlined collaboration among users.

3.1.1 Use Case Diagram



Use Case Diagram for ConnectingNotes: A Web Application for Music Mentorship and Learning

3.1.2 Use Case Narrative

3.1.2.1. Manage User Accounts (Administrator)

| | |
|--|--|
| Use Case Name | Manage User Accounts |
| Actor | Administrator |
| Pre-condition | User default account exists. User account records must exist in the database. |
| Description | Allows the administrator to view users and create an account for the secretary. The administrator can also update their own account. |
| Typical Course of Event | |
| Actor Action | System response |
| Step 1: Click the "Account" button. | Step 2: Displays dashboard "Create New Secretary" button, "Profile" button, "User" dropdown and extracts all users from the database. |
| Step 3: Click a button/dropdown. | <p>Step 4.1: If the "Create New Secretary" button is clicked, displays a form containing "Submit" button. Go to Step 5.</p> <p>Step 4.2: If the "User" dropdown is clicked, it displays lists "All", "Student", "Mentor", "Freelancer" and "Secretary" buttons. Go to Step 7.</p> <p>Step 4.3: If the "Profile" button is clicked, extracts profile from the database and displays profile information page with edit button. Go to Step 11.</p> |
| Step 5: Fill out the form and click a button. | Step 6.1: If the "Submit" button is clicked, validates the form. If valid and complete, displays a message: "New Secretary Added Successfully." End of Process. |

| | |
|--|--|
| <p>Step 7: Click a button on the “User” dropdown.</p> | <p>Step 8.1: If the “All” button is clicked, Extract all users from the database and display the list of all users.</p> <p>Step 8.2: If the “Student” button is clicked, Extract all students from the database and display the list of all students.</p> <p>Step 8.3: If the “Mentor” button is clicked, Extract all mentors from the database and display the list of all mentors.</p> <p>Step 8.4: If the “Freelancer” button is clicked, Extract all freelancers from the database and display the list of all freelancers.</p> <p>Step 8.5: If the “Secretary” button is clicked, Extract all secretaries from the database and display the list of all secretaries.</p> |
| <p>Step 9: Click a button.</p> | <p>Step 10: If a user is clicked in a list on all Step 8, extract users profile and display profile information page. End of Process.</p> |
| <p>Step 11: Update data and click the “Submit” button.</p> | <p>Step 12: Validates the profile form. If valid and complete, displays a message: “User Profile has been updated!”. End of Process.</p> |
| <p>Alternative Paths:</p> <p>Alternative Path for Step 6</p> <ul style="list-style-type: none"> • If the user did not fill up the form, it displays an error message “Please fill out this field.” Go back to Step 5. <p>Alternative Path for Step 12</p> <ul style="list-style-type: none"> • If the user did not fill up the form, it displays an error message “Please fill out this field.” Go back to Step 11. | |

3.1.2.2. Manage User Accounts (Secretary)

| | |
|---|--|
| Use Case Name | Manage User Accounts |
| Actor | Secretary |
| Pre-condition | User account records must exist in the database students, mentors, and freelancers must be registered in Herliz Music Studio through the system. |
| Description | Allows the secretary to view users and create an account for walk-in clients. The secretary can also update their own account (profile information and password). |
| Typical Course of Event | |
| Actor Action | System response |
| Step 1: Click a button. | Step 2: Dashboard displays "Create New Client" button, "Profile" button, "User" dropdown and extracts all users from the database. |
| Step 3: Click a button/dropdown. | <p>Step 4.1: If the "Create New Client" button is clicked, a form displays with the "Submit" button. Go to Step 5.</p> <p>Step 4.2: If the "User" dropdown is clicked, it displays lists "All", "Student", "Mentor", and "Freelancer" buttons. Go to Step 7.</p> <p>Step 4.3: If the "Profile" button is clicked, it extracts the profile from the database and displays the profile information page with the "Edit" button. Go to Step 11.</p> |
| Step 5: Fill up the form and click a button. | Step 6.1: If the "Submit" button is clicked, validates the form. If valid and complete, displays a message: "New Client Added Successfully." End of Process. |
| Step 7: Click a button on the "User" dropdown. | Step 8.1: If the "All" button is clicked, Extract all users from the database and display the list of all users. |

| | |
|--|--|
| | <p>Step 8.2: If the “Student” button is clicked, Extract all students from the database and display the list of all students.</p> <p>Step 8.3: If the “Mentor” button is clicked, Extract all mentors from the database and display the list of all mentors.</p> <p>Step 8.4: If the “Freelancer” button is clicked, Extract all freelancers from the database and display the list of all freelancers.</p> |
| Step 9: Click a button. | Step 10: If a user is clicked in each list Step 8.1 , Step 8.2 , Step 8.3 , and Step 8.4 . extract users profile and display profile information page. End of Process. |
| Step 11: Fill up the profile form and click the “Submit” button. | Step 12: Validates the profile form. If valid and complete, displays a message: “User Profile has been updated!” End of Process. |
| <p>Alternative Paths:</p> <p>Alternative Path for Step 6.1</p> <ul style="list-style-type: none"> • If the user did not fill up the form, it displays an error message “Please fill out this field.” Go back to Step 5. <p>Alternative Path for Step 12</p> <ul style="list-style-type: none"> • If the user did not fill up the form, it displays an error message “Please fill out this field.” Go back to Step 11. | |

3.1.2.3. Manage User Accounts (Students)

| | |
|----------------------|--|
| Use Case Name | Manage User Accounts |
| Actor | Students |
| Pre-condition | <p>The students must create an account through the system for them to login.</p> <p>User account records must exist in the database.</p> |

| | |
|---|--|
| Description | Allows the students to view their account and create an account on their own. The students can also update their own account. |
| Typical Course of Event | |
| Actor Action | System response |
| Step 1: Click a button. | Step 2: Displays dashboard "Create New Account" dropdown and "Login" dropdown. |
| Step 3: Click a button. | <p>Step 4.1: If the "Login" dropdown button is clicked, display the list of "Login Student" and "Login Mentor" button. Go to Step 5.</p> <p>Step 4.2: If the "Sign Up New Account" dropdown is clicked, it displays the list "Sign Up Student" and "Sign Up Mentor" button. Step 15.</p> |
| Step 5: Click a button. | Step 6.1: If the "Login Student" button is clicked, a login form will display. |
| Step 7: Fill up the form and click a button. | Step 8.1: If the "login" button is clicked, the system validates and extracts the student account from the database. |
| Step 9: Click the "Login" button. | Step 10: The system extracted users data from the database and displayed the student profile account with the edit button. |
| Step 11: Click the "Edit" button. | Step 12: The system extracts the student profile information from the database and displays the form with the submit button. |
| Step 13: Fill up the form and Click submit button. | Step 14: Validates the profile form. If valid and complete, displays a message: "User Profile has been updated!". End of Process. |
| Step 15: Click a button. | Step 16: If the "Sign Up Student" button is clicked, a create new account form will display. |
| Step 17: Fill up the form and click a button. | Step 18: If the "Sign" button is clicked, the system validates the sign up form. If valid and complete, display the verification page with the "Submit" button. The system sends verification code via email. |

| | |
|--|--|
| Step 19: Input and click submit button. | Step 20: Validates the code. If valid and complete, display a message: "New student account created successfully." End of Process. |
| <p>Alternative Paths:</p> <p>Alternative Path for Step 8.1</p> <ul style="list-style-type: none"> If the user did not fill up the form, it displays an error message "Please fill out this field." Go back to Step 7. <p>Alternative Path for Step 8.2</p> <ul style="list-style-type: none"> If the user did not fill up the form, it displays an error message "Please fill out this field." Go back to Step 7. <p>Alternative Path for Step 14</p> <ul style="list-style-type: none"> If the user did not fill up the form, it displays an error message "Please fill out this field." Go back to Step 13. <p>Alternative Path for Step 20</p> <ul style="list-style-type: none"> If the user did not fill up the form, it displays an error message "Please fill out this field." Go back to Step 19. If invalid displays a message "Verification Code Invalid!." Go back to Step 15. | |

3.1.2.4. Manage User Accounts (Mentors)

| | |
|--------------------------------|---|
| Use Case Name | Manage User Accounts |
| Actor | Mentors |
| Pre-condition | <p>The mentors must create an account through the system for them to login.</p> <p>User account records must exist in the database.</p> |
| Description | Allows the mentors to view their account and create an account on their own. The mentors can also update their own account. |
| Typical Course of Event | |
| Actor Action | System response |

| | |
|---|--|
| Step 1: Click a button. | Step 2: Displays dashboard “Sign Up New Account” dropdown and “Login” dropdown. |
| Step 3: Click a button. | <p>Step 4.1: If the "Login" dropdown button is clicked, display the list of “Login Student” and “Login Mentor” button. Go to Step 5.</p> <p>Step 4.2: If the “Sign Up New Account” dropdown is clicked, it displays the list “Sign Up Student” and “Sign Up Mentor” button. Step 15.</p> |
| Step 5: Click a button. | Step 6: If the “Login Mentor” button is clicked, a login form will display. |
| Step 7: Fill up the form and click a button. | Step 8: If the “login” button is clicked, the system validates and extracts the student account from the database. |
| Step 9: Click the “Login” button. | Step 10: The system extracted users data from the database and displayed the student profile account with the edit button. |
| Step 11: Click the “Edit” button. | Step 12: The system extracts the student profile information from the database and displays the form with the submit button. |
| Step 13: Fill up the form and Click submit button. | Step 14: Validates the profile form. If valid and complete, displays a message: “User Profile has been updated!”. End of Process. |
| Step 15: Click a button. | Step 16: If the “Sign Up Mentor” button is clicked, a create new account form will display. |
| Step 17: Fill up the form and click a button. | Step 18: If the “Sign Up” button is clicked, the system validates the sign up form. If valid and complete, display the verification page with the “Submit” button. The system sends verification code via email. |

| | |
|--|--|
| Step 19: Input and click submit button. | Step 20: Validates the code. If valid and complete, display a message: "New student account created successfully." End of Process. |
| <p>Alternative Paths:</p> <p>Alternative Path for Step 8.1</p> <ul style="list-style-type: none"> • If the user did not fill up the form, it displays an error message "Please fill out this field." Go back to Step 7. <p>Alternative Path for Step 8.2</p> <ul style="list-style-type: none"> • If the user did not fill up the form, it displays an error message "Please fill out this field." Go back to Step 7. <p>Alternative Path for Step 14</p> <ul style="list-style-type: none"> • If the user did not fill up the form, it displays an error message "Please fill out this field." Go back to Step 13. <p>Alternative Path for Step 20</p> <ul style="list-style-type: none"> • If the user did not fill up the form, it displays an error message "Please fill out this field." Go back to Step 19. • If invalid displays a message "Verification Code Invalid!." Go back to Step 15. | |

3.1.2.5. Manage Mentors (Administrator)

| | |
|--------------------------------|--|
| Use Case Name | Manage Mentors |
| Actor | Administrator |
| Pre-condition | Applications must exist in the database. User account records must exist in the database. |
| Description | The Administrator can review, approve, or reject applications, and manage onboarding. |
| Typical Course of Event | |
| Actor Action | System response |

| | |
|---|---|
| <p>Step 1: Click the “Manage Applications” button.</p> | <p>Step 2: Extracts all application data from the database, and displays a list of applicants and their status e.g. “Approved”, “Rejected”, and “Interview Scheduled” with a “View Applicant” button on each applicant on the list.</p> |
| <p>Step 3: Click the “View Applicant” button.</p> | <p>Step 4: Displays the information and documents of a specific applicant with an “Approve” button, and a “Reject” button. Go to step 5.</p> |
| <p>Step 5: Click a button.</p> | <p>Step 6.1: If the “Approve” button is clicked, Notifies the applicant of his/her approval through the applicant's email, and displays the "Manage Onboarding" button within the approved applicant's profile. Go to step 7.</p> <p>Step 6.2: If the “Reject” button is clicked, Notifies the applicant of his/her rejection through the applicant's email. The rejected applicant's data will be archived in the database. Go back to Step 2.</p> |
| <p>Step 7: Click the “Manage Onboarding” button.</p> | <p>Step 8: A window will appear where the administrator can provide the onboarding module/documents such as contracts, guidelines, and platform usage tutorials. If the applicant has already accepted and signed the onboarding module/documents, The administrator will be notified and a “Confirm Account” and “Provide Access” button will appear. Go to Step 9.</p> |
| <p>Step 9: Click a button.</p> | <p>Step 10: If the “Confirm Account” button is clicked, a message will display saying “Account Confirmed”, then the account of the accepted applicant will be saved in the database. End of process.</p> <p>Step 10.1: If the “Provide Access” button is clicked, the accepted applicant will be provided full access to their account where they can manage their account, courses, and session scheduling. End of process.</p> |

Alternative Paths:**Alternative path for Step 2**

- If data is not found, display the message “ No records found.” **End of Process.**

Alternative path for Step 6.1

- If the applicant is already approved, display the message “Applicant Already Approved”. **Go back to Step 4.**
- If the applicant is already rejected, display the message “Applicant Already Rejected”, then it will be deleted from the database. **Go back to Step 4.**

3.1.2.6. Manage Mentors (Secretary)

| | |
|--|---|
| Use Case Name | Manage Mentors |
| Actor | Secretary |
| Pre-condition | Applications must exist in the database. User account records must exist in the database. |
| Description | The Secretary can verify the initial document verification and track application progress of the Administrator. |
| Typical Course of Event | |
| Actor Action | System response |
| Step 1: Click the “Manage Applications” button. | Step 2: Extracts all application data from the database, and displays a list of applicants and their status e.g. “Accepted”, “Viewed”, “Not Viewed”, and “Withdrawn” with a “View Applicant” button on each applicant on the list and a “Filter” button. Applications labeled as “Accepted” will have a “Done” button. |
| Step 3: Click a button. | Step 4.1: If the “View Applicant” button is clicked, it displays the information and documents of a specific applicant and has a “Send” and “Reject” button. Go to Step 5. |

| | |
|--|--|
| | <p>Step 4.2: If the “Filter” button is clicked, the list will be sorted by status e.g. Accepted, Viewed, Not Viewed, and Withdrawn applications. End of Process.</p> <p>Step 4.3: If the “Done” button is clicked, it will be removed from the lists of applicants and saved in the database. End of Process.</p> |
| Step 5: Click a button. | <p>Step 6.1: If the “Send” button is clicked, The verified application will be forwarded to the administrator. A Form will appear with a “Set Schedule” button that requires date and time inputs that schedules an interview with the applicant by inputting the dates in the displayed form. Go to Step 7.</p> <p>Step 6.2: If the “Reject” button is clicked, Notifies the applicant of his/her rejection through the applicant's email. The rejected applicant's data will be archived in the database. Go back to Step 2.</p> |
| Step 7: Input date and time and click the “Set Schedule” button. | Step 8: Validates input, if valid, updates the applicant's status to "Interview Scheduled" and notify the Administrator and Applicant of the upcoming interview via email and an in-system notification. Go back to Step 2. |
| <p>Alternative Paths:</p> <p>Alternative path for Step 2</p> <ul style="list-style-type: none"> • If data is not found, display the message “No records found.” End of Process. <p>Alternative path for Step 8.</p> <ul style="list-style-type: none"> • If invalid, display the message “Invalid inputs.” Go back to Step 6.1. | |

3.1.2.7. Manage Mentors (Mentors/Freelancers)

| | |
|----------------------|---|
| Use Case Name | Manage Mentors |
| Actor | Mentors/Freelancers |
| Pre-condition | The mentors/freelancers can track the status of their applications in real-time |

| | |
|---|---|
| | <p>"Applications" must exist in the database</p> <p>User account records must exist in the database.</p> |
| Description | The mentors/freelancers can send applications, track the status of the progress of their application, and receive notifications through email. |
| Typical Course of Event | |
| Actor Action | System response |
| Step 1: Click a button. | Step 2: Manage application interface will be displayed with the "Apply Mentorship" button |
| Step 3: Click the "Apply Mentorship" button. | Step 4: Applicants will be given the option to choose and click a role either "Full Time Mentor" or "Freelancer". |
| Step 5: Click the "Full Time Mentor" button. | Step 6: A window will appear where they are required to submit their scanned Resume/Curriculum Vitae (CV) through the system with a "Back" and "Next" button. |
| Step 7: Click the "Next" button. | Step 8: A window will appear where they are required to submit their scanned Professional Regulation Commission (PRC) License with a "Back" and "Next" button. |
| Step 9: Click the "Next" button. | Step 10: A window will appear where they are required to submit their scanned medical certificate with a "Back" and "Next" button. |
| Step 11: Click the "Next" button. | Step 12: A window will appear where they are required to submit their scanned NBI clearance with a "Back" and "Next" button. |
| Step 13: Click the "Next" button. | Step 14: A window will appear where they are required to submit their Portfolio e.g. performance videos, lesson plans, teaching materials, and work samples with a "Submit" and "Back" button. |
| Step 15: Click the "Submit" button. | Step 16: A confirmation window with a "Back" button will be displayed, summarizing the documents submitted with the message 'Review Documents' and a "Confirm" button. |

| | |
|---|--|
| <p>Step 17: Click the "Confirm" button.</p> | <p>Step 18: Displays the message "Application has been successfully submitted". Records the documents in the database. Updates the application status to "Under Review". Notifies the applicant via email that confirms the submission and outlines the next steps in the review process. End of Process.</p> |
| <p>Step 19: Click anywhere.</p> | <p>Step 20: Closes the opened window and displays the Manage application interface showing the status of the application e.g. 'Under Review', 'Accepted', 'Rejected', and 'Withdrawn' including a "Withdraw Application" button. If status is 'Accepted', A window will appear containing a message 'Complete the onboarding module' with a "Contracts" button, "Guidelines" button, and "Tutorials" button.</p> |
| <p>Step 21: Click a button.</p> | <p>Step 22.1: If the "Contracts" button is clicked, a window with "Back" and "Attach Signature" buttons will appear displaying the contracts. The signed contract will be recorded and saved in the database. Go to Step 23.</p> <p>Step 22.2: If the "Guidelines" button is clicked, a window with a 'back' button will appear displaying the guidelines Including a "Done" button. Go to Step 27.</p> <p>Step 22.3: If the "Tutorials" button is clicked, a window with a "back" button will appear displaying the platform usage tutorials including a "Done" button. Go to Step 29.</p> <p>Step 22.4: If the applicant chooses to withdraw their application, they click the "Withdraw Application" button. Go to Step 31.</p> |
| <p>Step 23: Read Contracts and Click the "Attach Signature" button.</p> | <p>Step 24: A form will appear containing a "Save & Send" button. Go to Step 25.</p> |
| <p>Step 25: Attach a signature and click the "Save and Send" button.</p> | <p>Step 26: Validates the form. If valid and complete, displays a message: "Contract successfully submitted!". End of Process.</p> |

| | |
|---|---|
| Step 27: Reads the guidelines and click "Done" button | Step 28: Confirms the guidelines have been read and Closes the opened window. End of Process. |
| Step 29: Watch videos and click the "Done" button. | Step 30: Confirms platform usage tutorials are completed and Closes the opened window. |
| Step 31: Completes the reading/viewing of Contracts, Guidelines, and Tutorials. | Step 32: Mentor will be granted full access to their account in terms of managing their own accounts, courses, sessions, and achievements issuance. End of Process. |
| Step 33: Click the "Withdraw Application" button. | Step 34: A confirmation window appears with the message: "Are you sure you want to withdraw your application?" with "Yes" and "No" buttons. |
| Step 35: Click the "Yes" button. | Step 36: Displays a success message: "Your application has been successfully withdrawn." Updates the application status to "Withdrawn", and sends an email notification to the applicant confirming the withdrawal. End of Process. |
| <p>Alternative Paths:</p> <p>Alternative path for Step 20</p> <ul style="list-style-type: none"> • If Status is 'Rejected', display the message "Sorry you have been rejected." End of Process. • If Status is 'Under Review', display the message "Your application is being reviewed." End of Process. • If Status is 'Withdrawn', display the message "You have withdrawn your application." End of Process. <p>Alternative Path for Step 32</p> <ul style="list-style-type: none"> • If the "No" button is clicked, it closes the confirmation window and goes back to the "Manage Application" interface. Go back to Step 20. <p>Alternative path for Step 26</p> <ul style="list-style-type: none"> • If invalid, display the message "Attach a Signature". Go back to Step 25. | |

3.1.2.8. Manage Mentoring Sessions (Administrator)

| | |
|--|---|
| Use Case Name | Manage Mentoring Sessions |
| Actor | Administrator |
| Pre-condition | User accounts must exist in the database. |
| Description | The administrator can manage online registration, and the registration settings such as registration periods, required documentation and fee structures. |
| Typical Course of Event | |
| Actor Action | System response |
| Step 1: Click the "Manage Registration" button. | Step 2: Dashboard displays "All Students", "Registration of Students" buttons, "History" button, "search" bar with current sessions lists and upcoming sessions lists below. |
| Step 3: Click a button. | <p>Step 4.1: If the "All Students" button is clicked, a window will be displayed containing all the registered students. Go to Step 5.</p> <p>Step 4.2: If the "Registration of Students" button is clicked, a window will be displayed containing all pending registration of students. Go to Step 7.</p> <p>Step 4.3 If a session is clicked on current and upcoming sessions lists, display the session information and students. End of Process.</p> <p>Step 4.4. If the "History" button is clicked, displays list of previous sessions with "search bar" and "date filter" button. Go to Step 17.</p> <p>Step 4.5. If "search bar" is clicked, input session ID with enter button. Go to Step 19.</p> |
| Step 5: Click a specific student. | Step 6: Specific student's information will be displayed. End of Process. |

| | |
|--|--|
| Step 7: Click a specific student. | Step 8: Filled up form (personal information of the student, parental/guardian information, musical course(s) of interest, preferred mentor(s), signature, and date of submission) will be displayed for the administrator to verify it. Go to step 9. |
| Step 9: After verifying, click the "Next" button. | Step 10: Scanned birth certificate of the student will be displayed for the administrator to verify. Go to Step 11. |
| Step 11: After verifying, click the "Next" button. | Step 12: Scanned and signed Parent's consent will be displayed for the administrator to verify. Go to Step 13. |
| Step 13: After verifying, click the "Next" button. | Step 14: Proof of payment of the student will be displayed with the "Confirm" button Go to Step 15. |
| Step 15: After verifying, click the "Confirm" button. | <p>Step 16.1: A message will display "Registration Successful" indicating that the registration is successful and all documents are already saved/recorded in the database. End of Process.</p> <p>Step 16.2: The system will notify the registered student regarding their successful registration. The registered student will be displayed in the "All Students" dashboard indicating that their information is already saved/recorded in the database and will have full access to all the features of the platform. End of Process.</p> |
| Step 17. Click a button. | <p>Step 18.1. If a session is clicked on previous sessions lists, display the session information and students. End of Process.</p> <p>Step 18.2. If "search bar" is clicked, input session ID with enter button. Go to Step 19.</p> <p>Step 18.3. If "date filter" button is clicked, input range date with "okay" button. Go to Step 21.</p> |
| Step 19. Input session ID and click "Enter" button. | Step 20. Validate the session ID, display the matched session ID and display a message: "Session found.". End of Process. |
| Step 21. Fill up the range date and | Step 22. Display all sessions with the range date, display a message: "Date filter is done." End of Process. |

| | | |
|--|--------|--|
| click button. | "Okay" | |
| <p>Alternative Paths:</p> <p>Alternative path for Step 16.2</p> <ul style="list-style-type: none"> The administrator can also message through the system or email the registered student regarding their registration. End of Process. <p>Alternative path for Step 20</p> <ul style="list-style-type: none"> If the session ID is invalid, display a message: "Session does not exist.". End of Process. <p>Alternative path for Step 22</p> <ul style="list-style-type: none"> If the date range is invalid, display a message: "Date range syntax is not correct." Go to Step 21. | | |

3.1.2.9. Manage Mentoring Sessions (Secretary)

| | |
|--|--|
| Use Case Name | Manage Mentoring Sessions |
| Actor | Secretary |
| Pre-condition | User accounts must exist in the database. |
| Description | The administrator is the one who will manage the walk-in registration and process the payments of the clients (students or parents). |
| Typical Course of Event | |
| Actor Action | System response |
| Step 1: Click the "Manage Registration" button. | Step 2: Dashboard displays "All Students", "Registration of Students", "Walk-In Registration" buttons, "Freelancer" button, "History" button, "search" bar with current sessions lists and upcoming sessions lists below. |

| | |
|---|---|
| <p>Step 3: Click a button.</p> | <p>Step 4.1: If the “All Students” button is clicked, a window will be displayed containing all the registered students. Go to Step 5.</p> <p>Step 4.2: If the “Registration of Students” button is clicked, a window will be displayed containing all pending registration of students. Go to Step 7.</p> <p>Step 4.3: If the “Walk-In Registration” button is clicked, a window will be displayed containing all the steps for registration. Go to Step 17.</p> <p>Step 4.4: If a session is clicked on current and upcoming sessions lists, display the session information and students. End of Process.</p> <p>Step 4.5: If “History” button is clicked, displays list of previous sessions with “search bar” and “date filter” button. Go to Step 29.</p> <p>Step 4.6: If “search bar” is clicked, input session ID with enter button. Go to Step 31.</p> <p>Step 4.7: If “Freelancer” button is clicked, extract freelancers data and display lists of freelancers with “search bar”. Go to Step 35.</p> |
| <p>Step 5: Click a specific student.</p> | <p>Step 6: Specific student’s information will be displayed. End of Process.</p> |
| <p>Step 7: Click a specific student.</p> | <p>Step 8: Filled up form with proof payments will be displayed for the administrator to verify it. Go to step 9.</p> |
| <p>Step 9: After verifying, click the “Next” button.</p> | <p>Step 10: Scanned birth certificate of the student will be displayed for the administrator to verify it. Go to Step 11.</p> |
| <p>Step 11: After verifying, click “Next” button</p> | <p>Step 12: Scanned and signed Parent’s consent will be displayed for the administrator to verify it. Go to Step 13.</p> |
| <p>Step 13: After verifying, click “Next” button</p> | <p>Step 14: Proof of payment of the student will be displayed. Go to Step 15.</p> |

| | |
|--|---|
| Step 15: After verifying, click "Confirm" button | Step 16: A message will display "Registration Successful" indicating that the registration is successful and all documents are already saved/recorded in the database. The system will notify the student regarding the successful registration. End of Process. |
| Step 17: Login the walk-in client's account | Step 18: Manage registration interface is displayed containing the "Register Now" button. Go to Step 19. |
| Step 19: Click "Register Now" button | Step 20: A form will be displayed, Payment option buttons: Full Payment and Down Payment; Instruments options: Piano, Acoustic Guitar, Electric Guitar, Drums, Violin, and Pop Sing; and Mentors options. Go to Step 21. |
| Step 21: Encode and click "next" button | Step 22: A window will be displayed containing the "Attach File" button, where the scanned birth certificate will be attached and submitted. Go to Step 23. |
| Step 23: Attach file and click "next" button | Step 24: A window will be displayed containing the "Attach File" button, where the scanned and signed Parent's consent will be attached and submitted. Go to Step 25. |
| Step 25: Attach file and click "next" button | Step 26: A window will be displayed containing the "Attach File" button, where the payment slip can also be screenshot or bank slip will be attached and submitted. Go to Step 27. |
| Step 27: Click "Confirm" button | Step 28: A message will display "Registration Successful" indicating that the registration is successful and all documents are already saved/recorded in the database. The system will notify the student regarding the successful registration. End of Process. |
| Step 29: Click a button. | <p>Step 30.1. If a session is clicked on previous sessions lists, display the session information and students. End of Process.</p> <p>Step 30.2. If "date filter" button is clicked, input the range date with the "Okay" button. Go to Step 33.</p> |
| Step 31. Input session ID and click "Enter" button. | Step 32. Validate the session ID, display the matched session ID and display a message: "Session found.". End of Process. |

| | |
|---|---|
| Step 33. Fill up the range date and click "Okay" button. | Step 34. Display all sessions with the range date, display a message: "Date filter is done." End of Process. |
| Step 35. Click a freelancer. | Step 36. Display freelancer profile information and schedule with "assign session" button. |
| Step 37. Click "Assign Session" button. | Step 38. Displays session schedule record form selection, with search bar and "assign" button. |
| Step 39. Select assign sessions and click "Assign" button. | Step 40. Validates the assigned schedule session form. If valid and complete, display a message: "Freelancer successfully assigned.". End of Process. |
| <p>Alternative Paths:</p> <p>Alternative path for Step 32</p> <ul style="list-style-type: none"> • If session ID is invalid, display a message: "Session does not exist.". End of Process. <p>Alternative path for Step 34</p> <ul style="list-style-type: none"> • If "search bar" is clicked, Go back to Step 7. • If the date range is invalid, display a message: "Date range syntax is not correct." Go back to Step 9. | |

3.1.2.10 Manage Mentoring Sessions (Students)

| | |
|--------------------------------|---|
| Use Case Name | Manage Mentoring Sessions |
| Actor | Students |
| Pre-condition | User accounts must exist in the database. |
| Description | The students can submit all the required documents and registration fees. |
| Typical Course of Event | |
| Actor Action | System response |

| | |
|---|---|
| Step 1: Click "Register Now" button | Step 2: A form will be displayed including Payment option buttons: Full Payment and Down Payment; Instruments options: Piano, Acoustic Guitar, Electric Guitar, Drums, Violin, and Pop Sing; and Mentors options. Go to Step 3. |
| Step 3 Fill-up the Registration form and click "next" button | Step 4: A window will be displayed containing the "Attach File" button, where the scanned birth certificate will be attached and submitted. Go to Step 5. |
| Step 5: Attach file and click "next" button | Step 6: A window will be displayed containing the "Attach File" button, where the scanned and signed Parent's consent will be attached and submitted. Go to Step 7. |
| Step 7: Attach file and click "next" button | Step 8: A window will be displayed containing the "Attach File" button, where the payment slip (can also be screenshot or bank slip) will be attached and submitted. Go to Step 9. |
| Step 9: Click "Confirm" button | Step 10: A message will be displayed "Registration Submitted". End of Process. |
| Step 11: Click the enrolled class. | Step 12: Displays the session class instrument enrolled with the "Home" icon button, "Calendar" icon button on the left side. Dashboard "Stream" button, "Classwork" button, and "Students" button on the upper. |
| Step 13: Click a button. | <p>Step 14.1: If the "Home" button is clicked, redirected to the classroom dashboard homepage. End of Process.</p> <p>Step 14.2: If the "Calendar" button is clicked, display the mentor schedule calendars and "Book Session" button. Go to Step 15.</p> <p>Step 14.3: If the "Stream" button is clicked, display all the announcements and session materials with "View" button. "View all" button for upcoming work sessions, "Write Text" section with "Post" button, and "Add Comments" each session materials/posts with "Sent" button. Go to Step 17.</p> <p>Step 14.4: If the "Classwork" button is clicked, display all the session materials with "View" button. Go to Step 19.</p> <p>Step 14.5: If the "Students" button is clicked, display the mentors name and students enrolled in the session. End of Process.</p> |

| | |
|--|---|
| <p>Step 15: Select a day and click “Book Session” button.</p> | <p>Step 16: Validates the request. If valid, sends the booking request to the mentor, displays a message: “Book Request Sent.” End of Process.</p> |
| <p>Step 17: Click a button.</p> | <p>Step 18.1: If the “View” button session material is clicked, display the session material page with/without “Attached file”, “Add File” button, “Mark as Done” button, and “Add comments” button with “Send” button. Go to Step 19.</p> <p>Step 18.2: If the “View all” button is clicked, display all the assigned list of session materials with “View” button under the dashboard “Assigned” button, “Missing” button, and “Done” button. Go to Step 23.</p> <p>Step 18.3: If the “Write Text” section is clicked, display the text section to write a post/announcement with “Attach file” button and “Post” icon. Go to Step 29.</p> <p>Step 18.4: If the “Add comments” button is clicked, display the text section to write comments with “Post” icon. Go to Step 31.</p> |
| <p>Step 19: Click a session material “View” button.</p> | <p>Step 20: Validates the session material. If valid, display the session material page with/without “Attached file”, “Add File” button, “Mark as Done” button, and “Add comments” button with “Send” button.</p> |
| <p>Step 21: Click a button.</p> | <p>Step 22.1: If the “Attached file” is clicked, open the file. End of Process.</p> <p>Step 22.2: If the “Add File” button is clicked, display the desktop file explorer with “Open” button then “Submit” button. Go to Step 23.</p> <p>Step 22.3: If the “Done Session” button is clicked, validates the material and sends it to the mentor. Display a form window for rating and feedback of the session with “Submit” button. Go to Step 27.</p> <p>Step 22.4: If the “Add comments” button is clicked, display the text section to write comments with “Post” icon. Go to Step 31.</p> |

| | |
|---|---|
| Step 23: Upload file and click "Submit" button. | Step 24: Validates the added file. If valid, send it to the mentor. Display a message: "Submitted!". End of Process. |
| Step 25: Click a button. | <p>Step 26.1: If the "Assigned" button is clicked, display all the assigned session materials with "View" button. Go back to Step 19.</p> <p>Step 26.2: If the "Missing" button is clicked, display all the missing session materials with "View" button. Go back to Step 19.</p> <p>Step 26.3: If the "Done" button is clicked, display all the done session materials with score and "View" button. Go back to Step 19.</p> |
| Step 27: Rate, write feedback and click "Submit" button. | Step 28: Validates the feedback form. If valid, display a message: "Thank you for your feedback.". End of Process. |
| Step 29: Write a post and click "Post" icon. | Step 30: The system updates the stream and posts the new post. Display a message: "Your post is posted!". End of Process. |
| Step 31: Write a comment and click "Post" icon. | Step 32: The system updates and displays the comments of the session material/post. Display a message: "The comment is already posted!". End of Process. |
| <p>Alternative Paths:</p> <p>Alternative path for Step 3</p> <ul style="list-style-type: none"> If the filled up form is incomplete, an error message will be displayed "Fill up all the required sections". Go back to Step 2. <p>Alternative path for Step 5</p> <ul style="list-style-type: none"> If the "Next" button is clicked without attaching a file, an error message will be displayed "Please attach a file". Go back to Step 4. <p>Alternative path for Step 7</p> <ul style="list-style-type: none"> If the "Next" button is clicked without attaching a file, an error message will be displayed "Please attach a file". Go back to Step 6. <p>Alternative path for Step 9</p> <ul style="list-style-type: none"> If the "Confirm" button is clicked without attaching a file, an error message will be displayed "Please attach a file". Go back to Step 8. | |

Alternative Path for Step 16

- If the user chooses a schedule that has a full slot, display a message: “Booking Session Request is not allowed!”. **End of Process.**

Alternative Path for Step 20

- If session material is scored, display the score above the “Added File” button. **End of Process.**

Alternative Path for Step 24

- If invalid, display a message: “Invalid File!”. **End of Process.**

Alternative Path for Step 26.1

- If there is no assigned session material, display a message: “Nothing on To-do list right now.” **End of Process.**

3.1.2.11 Manage Mentoring Sessions (Mentors)

| | |
|--|--|
| Use Case Name | Manage Sessions |
| Actor | Mentors |
| Pre-condition | The mentors must be an official mentor in Herliz Music Studio. User accounts must exist in the database |
| Description | The Mentors are responsible for managing their session schedules, including setting availability, confirming student bookings, conducting sessions, and providing feedback (comments and ratings). |
| Typical Course of Event | |
| Actor Action | System response |
| Step 1: Click “Manage Sessions” button. | Step 2: Displays all classes of the mentors and “Create class” button. |
| Step 3: Click a specific class. | Step 4.1: If a specific class is clicked, display information about the class with the “Home” icon button and “Calendar” |

| | |
|---|--|
| | <p>icon button on the left side. Dashboard “Stream” button, “Classwork” button, “Students” button, and “Grade” button, “Booking Requests” button and “Attendance” button.</p> <p>Step 4.2: If the “Create class” button is clicked, display a form for the new session classroom. Go to Step 33.</p> |
| Step 5: Click a button. | <p>Step 6.1: If the “Home” button is clicked, redirected to the homepage of the classroom dashboard. End of Process.</p> <p>Step 6.2: If the “Calendar” button is clicked, displays the full view of the calendar, where the mentor can click a specific session or classwork for further information, and Post sessions schedule with “Done” button. Go to Step 7.</p> <p>Step 6.3: If the “Stream” button is clicked, display all the announcements and session materials with the “View” button. “View all” button for upcoming work sessions, “Write Text” section with “Post” button, and “Add Comments” each session materials/posts with “Sent” button. Go to Step 9.</p> <p>Step 6.4: If the “Classwork” button is clicked, display all the session materials with “View” button and “Create” button. Go to Step 17.</p> <p>Step 4.5: If the “Students” button is clicked, display the mentors name and students enrolled in the session with the “Invite” button for students. Go to Step 21.</p> <p>Step 4.6: If the “Booking Request” button is clicked, display all the requests of the students with “Accept” button, “Decline” button, and “Schedule” button. Go to Step 25.</p> <p>Step 4.7: If the “Attendance” button is clicked, display the attendance form with a checkbox: “Check” icon and “Wrong” icon, and “Done” button. Go to Step 31.</p> |
| Step 7: Click a session and click “Done” button. | Step 8: Update the session and display a message: “Session Done!” End of Process. |
| Step 9: Click a button. | Step 10.1: If the “View all” button is clicked, display all the assigned list of session materials with “View” button under |

| | |
|--|---|
| | <p>the dashboard “Assigned” button, “Missing” button, and “Done” button. Go to Step 11.</p> <p>Step 10.2: If the “Write Text” section is clicked, display the text section to write a post/announcement with “Attach file” button and “Post” button. Go to Step 13.</p> <p>Step 10.3: If the “Add comments” button is clicked, display the text section to write comments with “Send” button. Go to Step 15.</p> |
| Step 11: Click a button. | <p>Step 12.1: If the “Assigned” button is clicked, display all the assigned session materials. End of Process.</p> <p>Step 12.2: If the “Missing” button is clicked, display all the missing session materials. End of Process.</p> <p>Step 12.3: If the “Done” button is clicked, display all the done session materials with score. End of Process.</p> |
| Step 13: Write a post and click “Post” button. | Step 14: The system updates the stream and posts the new post. Display a message: “Your post is posted!”. End of Process. |
| Step 15: Write a comment and click “Send” button. | Step 16: The system updates the comments of a session materials/post and posts the new added comment. Display a message: “Your comment is added!”. End of Process. |
| Step 17: Click a button. | <p>Step 18.1: If a session material “View” button is clicked, Go back to Step 10.1.</p> <p>Step 18.2: If the “Create” button is clicked, display the create session form: set criteria and attach files with “Post Material” button.</p> |
| Step 19: Fill up the form and click “Post Material” button. | Step 20: Validate the form and update the stream. Display a message: “Posted!”. End of Process. |
| Step 21: Click a button. | Step 22: If the “Invite Student” button is clicked, display the invite form with “Send” button. |
| Step 23: Fill up the form and click “Send” button. | Step 24: Validates the form. If valid, display a message: “Invite Sent.” and send the invitation to the respective user. End of Process. |
| Step 25: Click a button. | Step 26.1: If the “Accept” button is clicked, the system validates the request, updates the mentor’s schedule and |

| | |
|---|--|
| | <p>sends the student a message: "Your request has been accepted.". Go back to Step 7 .</p> <p>Step 26.2: If the "Decline" button is clicked, the system rejects the request and sends the student a message: "Your request has been declined." to the student. End of Process.</p> <p>Step 26.3: If the "Schedule" button is clicked, display the mentor's chart schedule with "Select" button and "Available" button.</p> |
| Step 27: Select day and click "Available" button. | <p>Step 28.1: Validates and updates the mentor's chart schedule for session booking requests. Display a message: "Your schedule has been updated!". End of Process.</p> <p>Step 28.2: If the selected day is selected, the "Available" button changes to the "Unavailable" button.</p> |
| Step 29: Click "Unavailable" button. | Step 30: Update the mentor's chart calendar schedule and cancel all the booked reservations. Display a message: "Your schedule has been updated!". Send the student a message: "Your booked session has been canceled, due to the mentor's schedule." End of Process. |
| Step 31: Check the lists and click "Done" button. | Step 32: Updates the attendance form and displays a message: "Attendance has been saved!". End of Process. |
| Step 33: Fill up the form and click confirm. | Step 34: The system validates the form then redirects and displays the new class in the homepage. End of Process. |
| <p>Alternative Paths:</p> <p>Alternative Path for Step 20</p> <ul style="list-style-type: none"> • If empty, display a message: "Please fill up this field.". Go back to Step 19. <p>Alternative Path for Step 24</p> <ul style="list-style-type: none"> • If empty, display a message: "Please fill up this field.". Go back to Step 23. <p>Alternative Path for Step 28.1</p> | |

- If empty, display a message: “Please fill up the remaining field.”. **Go back to Step 27.**

Alternative Path for Step 32

- If empty, display a message: “Please fill up the remaining field.”. **Go back to Step 31.**

Alternative Path for Step 34

- If empty, display a message: “Please fill up the remaining field.”. **Go back to Step 33.**
- If invalid, display a message: “Invalid inputs”. **Go back to Step 33.**

3.1.2.12. Manage Achievements (Administrator)

| | |
|---|---|
| Use Case Name | Manage Achievements Issuance |
| Actor | Administrator |
| Pre-condition | <p>Achievements records must exist.</p> <p>User accounts must exist.</p> <p>Extract all data achievement data from the database including Mentor, Student, and Requests of Certificates.</p> |
| Description | The Administrator oversees the issuance of achievements and certificates. Also manage the technical aspects of the system, such as generating certificates and integrating digital signatures to maintain authenticity. |
| Typical Course of Event | |
| Actor Action | System response |
| Step 1: Click the “Achievement Issuance” button. | Step 2: Dashboard displays “Mentor” button, “Student” button, “Generate Certificate” and “Requests” button. |

| | |
|---|---|
| <p>Step 3: Clicks a button</p> | <p>Step 4.1: If the “Mentor” button is clicked, displaying lists of Mentors a button. Go to Step 5.</p> <p>Step 4.2: If the “Student” button is clicked, displaying lists of Students a button. Go to Step 7.</p> <p>Step 4.3: If the “Request” button is clicked, displaying lists of request certificates is a button. Go to Step 15.</p> <p>Step 4.4: If the “Generate Certificate” button is clicked, it displays a certificate form with the “Submit” button. Go to Step 17.</p> |
| <p>Step 5: Click a mentor.</p> | <p>Step 6: If a list is clicked, displays mentor profile and set achievements. End of process.</p> |
| <p>Step 7: Click a student.</p> | <p>Step 8: If a list is clicked, displays student profile, achievements progress and back button.</p> |
| <p>Step 9: Click a button.</p> | <p>Step 10: If an “achievement” is clicked, display a certificate and attach a signature button. Go to Step 11.</p> |
| <p>Step 11: Click a button.</p> | <p>Step 12: If “attach signature” button is clicked, the certificate form displays the “Save & Send” button. Go to Step 13.</p> |
| <p>Step 13: Attaches signature and clicks the “Save and Send” button.</p> | <p>Step 14: Validates the form. If valid and complete, displays a message: “Certificate successfully submitted!”. End of Process.</p> |
| <p>Step 15: Click a certificate.</p> | <p>Step 16: If a certificate is clicked, display the certificate form with the edit button.</p> |
| <p>Step 17: Fill up the form, attach a signature and click the “Submit” button.</p> | <p>Step 18: Validates the form. If valid and complete, displays a message: “Certificate successfully submitted!” End of Process.</p> |
| <p>Alternative Paths:</p> <p>Alt Path for Step 14</p> <ul style="list-style-type: none"> • If empty/invalid, displays a message: “Please fill out this field.” Go to Step 13. • If no signature is attached, display a message “Please input signature.” Go to Step 13. | |

Alt Path for Step 16

- If empty/invalid, displays a message: “Please fill out this field.” **Go to Step 13.**
- If no signature is attached, display a message “Please input signature.” **Go to Step 15.**

3.1.2.13. Manage Achievements (Mentors)

| | |
|--------------------------------|---|
| Use Case Name | Manage Achievements Issuance |
| Actor | Mentors |
| Pre-condition | Achievements records must exist. User accounts must exist. |
| Description | The mentors are responsible for setting, creating, and managing the specific achievement types that students can earn. |
| Typical Course of Event | |
| Actor Action | System response |
| Step 1: Click a button. | Step 2: Dashboard displays “Students” button, “Create Achievements” button, “Request” button and “Generate Certificate” button. |
| Step 3: Click a button. | <p>Step 4.1: If the “Student” button is clicked, extract lists of students' data. Displays lists of students with a button. Go to Step 5.</p> <p>Step 4.2: If the “Create Achievements” button is clicked, it displays an achievement form with the “post” button. Go to Step 7.</p> <p>Step 4.3: If the “Request” button is clicked, it displays lists of requests. Go to Step 9.</p> <p>Step 4.4: If the “Generate Certificate” button is clicked, display the certificate form with “submit” and “cancel” button. Go to Step 11.</p> |

| | |
|---|---|
| Step 5: Click a list. | Step 6: If a list is clicked, display student profile and achievements progress. End of Process. |
| Step 7: Fill up the form and click the “send” button. | Step 8: Validate the achievement. If valid, display a message: “Achievement successfully created!”. End of Process. |
| Step 9: Click a certificate request. | Step 10: If a list is clicked, display the certificate form with the edit button. |
| Step 11: Fill up the form, attach a signature and click the “Submit” button. | Step 12: Validates the form. If valid and complete, displays a message: “Certificate successfully submitted!” End of Process. |
| Alternative Paths: Alt Path for Step 8 <ul style="list-style-type: none"> • If empty/invalid, displays a message: “Please fill out this field.” Go to Step 7. • If no signature is attached, display a message “Please input signature.” Go to Step 7. Alt Path for Step 12 <ul style="list-style-type: none"> • If empty/invalid, displays a message: “Please fill out this field.” Go to Step 11. • If no signature is attached, display a message “Please input signature.” Go to Step 11. | |

3.1.2.14. Manage Achievements (Students)

| | |
|----------------------|---|
| Use Case Name | Manage Achievements Issuance |
| Actor | Students |
| Pre-condition | Students must be enrolled in Herliz Music Studio "Achievements" records must exist User accounts must exist in the database |
| Description | The students can view their achievements through their personal dashboards, where they can also view |

| | |
|---|---|
| | their progress and any badges or certificates they have earned. |
| Typical Course of Event | |
| Actor Action | System response |
| Step 1: Click the “Achievements” button. | Step 2: Extract all achievements. Dashboard displays a list of “Achievements/Badge” buttons. |
| Step 3: Click an achievement/badge. | <p>Step 4.1: If a complete achievement/badge is clicked, the system validates and displays the achievement progress, certificate and badge with “share” and “download” button. Go to Step 5.</p> <p>Step 4.2: If a process achievement/badge is clicked, displays the achievement/badge progress. End of Progress.</p> |
| Step 5: Click the “Share” button. | <p>Step 6.1: If the “share” button is clicked, it displays a window with an “email” button, “facebook” button, and “instagram” button. Go to Step 7.</p> <p>Step 6.2: If the “download” button is clicked, the system downloads the certificate in the device file. Display a message: “Download.” End of Process.</p> |
| Step 7: Click a button. | <p>Step 8.1: If the “email” button is clicked, the system shares and sends the certificate via email. End of Process.</p> <p>Step 8.2: If the “facebook” button is clicked, the system shares and sends the certificate via facebook post. End of Process.</p> <p>Step 8.3: If the “instagram” button is clicked, the system shares and sends the certificate via instagram post. End of Process.</p> |
| Alternative Paths: Alternative path for Step 2 <ul style="list-style-type: none"> • If data is not found, display the message “ No records found.” End of Process. | |

3.1.2.15. Generate Reports (Administrator)

| | |
|---|--|
| Use Case Name | Generate Reports |
| Actor | Administrator |
| Pre-condition | Generate Reports window is opened. Registrations, payments/ earnings and course records must exist in the database. |
| Description | Allows the administrator to view generated reports by the secretary. |
| Typical Course of Event | |
| Actor Action | System response |
| Step 1: Click a Button "Reports" | Step 2: Displays custom selection "Enrollment", "Registration", "Invoice", and "Course" buttons are displayed on the upper left. "Print" and "Date filter" buttons are displayed on the upper right. |
| Step 3: Click a Button | <p>Step 4.1: If the "Enrollment" button is clicked, displays the reports and charts. Display a message: "Enrollment.". End of Process.</p> <p>Step 4.2: If the "Registration" button is clicked, displays the reports and charts. Display a message: "Registration.". End of Process.</p> <p>Step 4.3: If the "Invoice" button is clicked, displays the reports and charts. Display a message: "Invoice.". End of Process.</p> <p>Step 4.4: If the "Course" button is clicked, displays the reports and charts. Display a message: "Courses.". End of Process.</p> |
| Step 5: Click "Print" Button | Step 6: If the "Print" button is clicked, display the report sheets and charts with another print button. |

| | |
|---|--|
| Step 7: Click the secondary “print” button. | Step 8.: If the secondary “Print” Button is clicked, display a message “Are you sure to Print the Generated Report?” for verification with the “Okay” button.. Go to Step 9. |
| Step 9: Click the “okay” button. | Step 10: Validates then sends the file and command to the hardware printer.Display a message: “Printing Report!”. End of Process, |
| Step 11: Click “Date Filter” Button | Step 12: If the “date” button is clicked, display date entries with the “OK” button. |
| Step 13: Set a date and click the “Ok” button. | Step 14: Validates the date range. If valid and complete, display a message: “Done!” and display all data with the date range. End of Process. |
| Alternative Paths: Alternative Path for Step 14 <ul style="list-style-type: none"> • If the set date is invalid, display a message: “Set Date Invalid!”. Go to Step 13. • If there is no entry, display a message: “Please select date!”. Go to Step 13. | |

3.1.2.16. Generate Reports (Secretary)

| | |
|--------------------------------|--|
| Use Case Name | Generate Reports |
| Actor | Secretary |
| Pre-condition | Generate Reports window is opened. Registrations, payments/earnings, enrollments and course records must exist in the database. |
| Description | Allows the Secretary to generate reports weekly/monthly registrations, payments/ earnings and courses. |
| Typical Course of Event | |

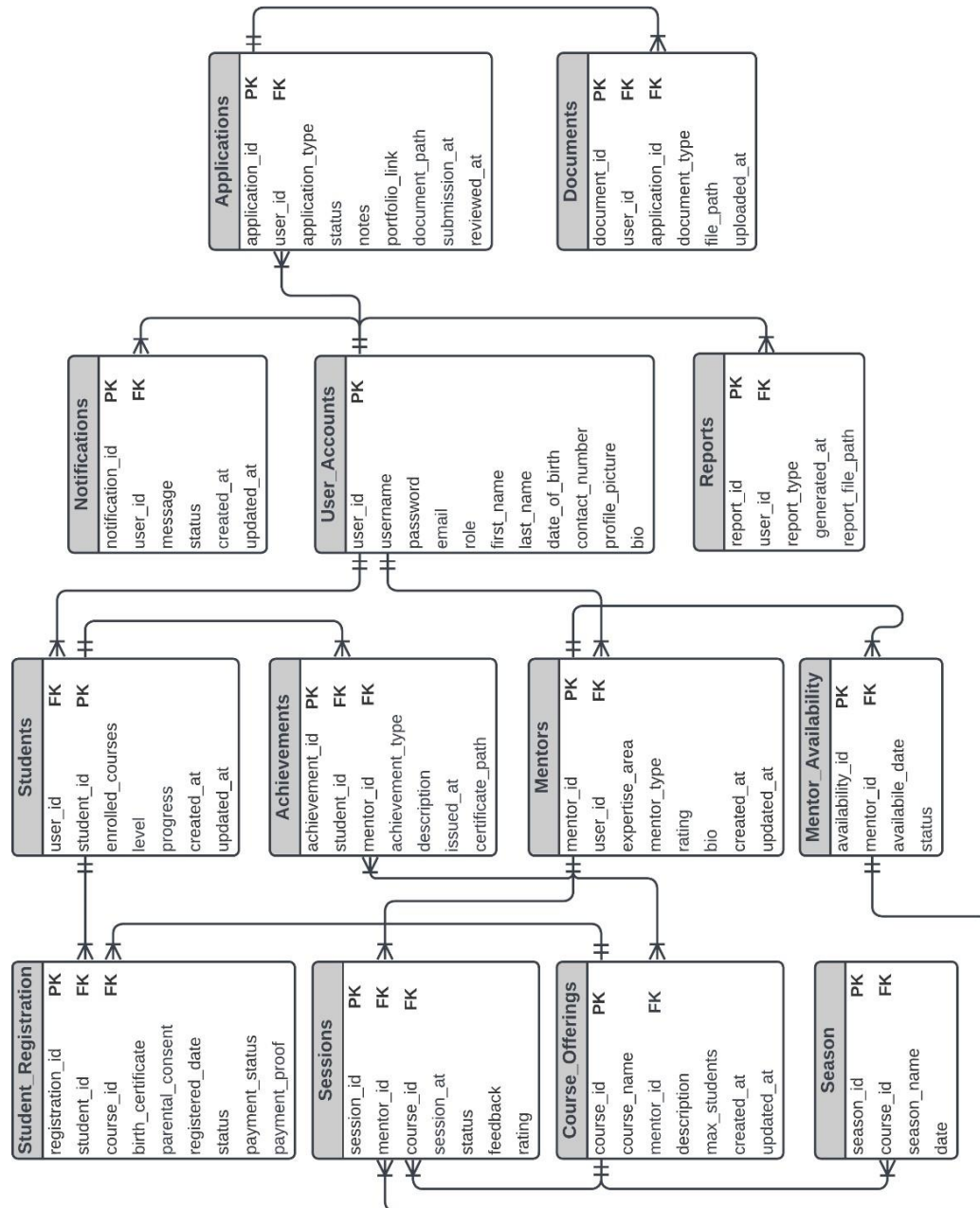
| Actor Action | System response |
|---|--|
| Step 1: Click a button | Step 2: Display custom selection “Enrollment”, “Registration”, “Invoice”, and “Course”, “Session”, “Date Filter”, and “Print” buttons are displayed on the upper window and “Create Report” displayed on the lower left. |
| Step 3: Click a button | <p>Step 4.1: If the "Enrollment" button is clicked, the window loads displaying the registration records “Finished” and “Unfinished” button. Go to Step 5.</p> <p>Step 4.2: If the "Registration" button is clicked, the window loads displaying the payment transaction records. End of Process.</p> <p>Step 4.3: If the "Invoice" button is clicked, the window loads displaying the payment transaction records. End of Process.</p> <p>Step 4.4: If the “Course” button is clicked, the window loads displaying the course records and a custom selection “ALL”, “VIOLIN”, “PIANO”, “ACOUSTIC GUITAR”, “ELECTRIC GUITAR”, “DRUMS”, and “POP SINGING” buttons. Go to Step 15.</p> |
| Step 5: Click a button | <p>Step 6.1: If the “Finished” button is clicked, window loads displaying finished registration. End of Process.</p> <p>Step 6.2: If the “Unfinished” button is clicked, window loads displaying finished registration. End of Process.</p> |
| Step 7: Click “Print” Button | Step 8: If the "Print" button is clicked, a window displays the report sheets and charts with another print button. |
| Step 9: Click the secondary “print” button | Step 10: If the secondary “Print” Button is clicked, display a message “Are you sure to Print the Generated Report?” for verification with the “Okay” button.. Go to Step 9. |
| Step 11: Click “okay” button. | Step 12: Validates then sends the file and command to the hardware printer. Display a message: “Printing Report!”. End of Process, |
| Step 13: Click “Date Filter” Button | Step 14: If the “date” button is clicked, display date entries with the OK button. |

| | |
|---|---|
| Step 15: Set a date and click the "Ok" button. | Step 16: Validates the date range. If valid and complete, display a message: "Done!" and display all data with the date range. End of Process. |
| Step 17: Click Create Report button | Step 18: Display the containing respective/selected records with "Date filter" , "Chart", "Submit" buttons. |
| Step 19: Click a button | <p>Step 20.1: Go to Step 13.</p> <p>Step 20.2: If the "Chart" button is clicked, a custom selection button displays for charts options. End of Process.</p> <p>Step 20.3: If the "Submit" button is clicked, the system will validate and store it to the database. End of Process.</p> |
| Step 21: Click a Button | <p>Step 22.1: If the "ALL" button is clicked, the system will display all the course records. Display a message: "All Records." End of Process.</p> <p>Step 22.2: If the "VIOLIN" button is clicked, the system will display all the violin records. Display a message: "Violin Records." End of Process.</p> <p>Step 22.3: If the "PIANO" button is clicked, the system will display all the piano records. Display a message: "Piano Records." End of Process.</p> <p>Step 22.4: If the "ACOUSTIC GUITAR" button is clicked, the system will display all the acoustic guitar records. Display a message: "Acoustic Guitar Records." End of Process.</p> <p>Step 22.5: If the "ELECTRIC GUITAR" button is clicked, the system will display all the electric guitar records. Display a message: "Electric Guitar Records." End of Process.</p> <p>Step 22.6: If the "DRUMS" button is clicked, the system will display all the drums records. Display a message: "Drums Records." End of Process.</p> |

| | |
|---|---|
| | Step 22.7: If the “POP SINGING” button is clicked, the system will display all the singing records. Display a message: “Pop Singing Records.” End of Process. |
| Alternative Paths: Alternative Path for Step 16.3 <ul style="list-style-type: none">• If the submitted report is invalid, a prompt message “Invalid Report!” will be displayed. Go back to Step 14. | |

3.2 Design

3.2.1. Relational Database



Relational Database for ConnectingNotes: A Web Application for Music Mentorship and Learning

3.2.2 File Structure

Table 1. Student_Registrations

This table handles student registration details.

| Field Name | Data Type | Description |
|-----------------------------|---------------|---|
| registration_id (PK) | INT (11) | Unique identifier for each registration |
| student_id (FK) | INT (11) | Foreign key linking to the student table |
| course_id (FK) | INT (11) | Foreign key linking to the Courses table |
| birth_certificate | VARCHAR(225) | File path to scanned birth certificate |
| parental_consent | VARCHAR(225) | File path to scanned parental consent form. |
| registration_date | DATE | Date when the registration was made |
| status | ENUM | Status of the registration ('Pending', 'Confirmed', 'Canceled') |
| payment_status | ENUM | Status of payment ('Unpaid', 'Paid') |
| payment_proof | VARCHAR (100) | Proof of payment or reference number for the payment |

Table 2. Sessions

This table records session schedules between students and mentors/freelancers.

| Field Name | Data Type | Description |
|------------------------|-----------|---|
| session_id (PK) | INT (11) | Unique identifier for each session |
| course_id (FK) | INT (11) | Foreign key linking to the Courses table |
| mentor_id (FK) | INT (11) | Foreign key linking to the mentor table |
| session_at | DATETIME | Date and time of the session |
| status | ENUM | Status of the session ('Scheduled', 'Completed', 'Cancelled') |
| feedback | TEXT | Feedback provided by the mentor or student |
| rating | INT (2) | Rating given for the session |

Table 3. Course Offerings

This table contains Courses.

| Field Name | Data Type | Description |
|-----------------------|---------------|--|
| course_id (PK) | INT (11) | Unique identifier for each course |
| course_name | VARCHAR (100) | Name of the course |
| description | TEXT | Detailed description of the course |
| mentor_id (FK) | INT (11) | Foreign key linking to the mentor table |
| max_students | INT (3) | Maximum number of students per course |
| created_at | TIMESTAMP | Timestamp when the course was created |
| updated_at | TIMESTAMP | Timestamp when the course was last updated |

Table 4. Season

This table contains Information of classes and sessions in each season.

| Field Name | Data Type | Description |
|-----------------------|-----------|---|
| season_id (PK) | INT (11) | Unique identifier for each Season |
| course_id (FK) | INT (11) | Foreign key linking to the Course table |
| session_name | TEXT | Name of the Season |
| date | DATE | Date when the Season was created |

Table 5. Students

This table contains the Students.

| Field Name | Data Type | Description |
|-------------------------|--------------|---|
| student_id (FK) | INT (11) | Unique identifier for each student |
| user_id (PK) | INT (11) | Foreign key linking to User_Accounts table |
| enrolled_courses | BLOB | Encrypted field storing courses enrolled by the student to protect sensitive information. |
| level | VARCHAR (50) | Level or grade of the student (e.g., beginner, advanced) |
| progress | TEXT | Text field to track student progress |
| created_at | TIMESTAMP | Timestamp when the student record was created |
| updated_at | TIMESTAMP | Timestamp when the student record was last updated |

Table 6. Achievements

This table contains achievements obtained by students.

| Field Name | Data Type | Description |
|----------------------------|---------------|--|
| achievement_id (PK) | INT (11) | Unique identifier for each achievement |
| student_id (FK) | INT (11) | Foreign key linking to the student table |
| mentor_id (FK) | INT (11) | Foreign key linking to the mentor table |
| achievement_type | VARCHAR (50) | Type of achievement earned |
| description | TEXT | Description of the achievement |
| issued_at | DATE | Date when the achievement was awarded |
| certificate_path | VARCHAR (255) | File path to the achievement certificate |

Table 7. Mentors
This table contains Mentors.

| Field Name | Data Type | Description |
|-----------------------|---------------|---|
| mentor_id (PK) | INT (11) | Unique identifier for each mentor |
| user_id (FK) | INT (11) | Foreign key linking to User_Accounts table |
| expertise_area | VARCHAR (100) | The area of expertise of the mentor (e.g., piano, violin) |
| mentor_type | ENUM | Type may be 'Full-time Mentor' and 'Freelancer' |
| rating | FLOAT | Average rating of the mentor based on feedback |
| bio | TEXT | A short biography or profile of the mentor |
| created_at | TIMESTAMP | Timestamp when the mentor record was created |
| updated_at | TIMESTAMP | Timestamp when the mentor record was last updated |

Table 8. Mentor_Availability
This table tracks mentor availability for sessions.

| Field Name | Data Type | Description |
|-----------------------------|-------------|--|
| availability_id (PK) | INT (11) | Auto-incremented availability ID (Primary key) |
| mentor_id (FK) | INT (11) | Foreign key linking to the Mentor table |
| available_date | DATETIME | Date and time slot when the mentor is available |
| status | VARCHAR(50) | Availability status, (Available, Booked, Unavailable). |

Table 9. Notifications

This table manages user system notifications regarding applications, session bookings, and achievements.

| Field Name | Data Type | Description |
|-----------------------------|-----------|--|
| notification_id (PK) | INT (11) | Unique identifier for each notification |
| user_id (FK) | INT (11) | Foreign key linking to the Users table |
| message | TEXT | Content of the notification |
| status | ENUM | Status of the notification ('Unread', 'Read') |
| created_at | TIMESTAMP | Timestamp when the notification was created |
| updated_at | TIMESTAMP | Timestamp when the notification was last updated |

Table 10. User_Accounts

This table stores the general information of all users of the system.

| Field Name | Data Type | Description |
|---------------------|---------------|--|
| user_id (PK) | INT (11) | Unique identifier for each user |
| username | VARCHAR (50) | The username for user login must be unique |
| password | VARCHAR (255) | Encrypted password for user login |
| email | VARCHAR (100) | The user's email address must be unique |
| role | ENUM | Role of the user ('Administrator', 'Secretary', 'Mentor', 'Freelancer', 'Student') |
| first_name | VARCHAR (50) | User's first name |

| | | |
|------------------------|------------------|---|
| last_name | VARCHAR (50) | User's last name |
| date_of_birth | DATE | User's date of birth |
| contact_number | VARCHAR (15) | User's contact phone number |
| profile_picture | VARCHAR (255) | File path to the user's profile picture |
| bio | TEXT | Short biography of the user |

Table 11. Reports

This table manages system notifications for administrators to track various metrics (e.g., use activity, session participation).

| Field Name | Data Type | Description |
|-------------------------|------------------|--|
| report_id (PK) | INT (11) | Unique identifier for each report |
| user_id (FK) | INT (11) | Foreign key linking to the Users table |
| report_type | ENUM | Type of report ('Registration', 'Earnings', 'Course', 'Session') |
| generated_at | DATETIME | Date and time when the report was generated |
| report_file_path | VARCHAR (255) | File path to the generated report |

Table 12. Applications

This table contains Applications Submitted by Mentors and Freelancers.

| Field Name | Data Type | Description |
|----------------------------|---------------|---|
| application_id (PK) | INT (11) | Unique identifier for each application |
| user_id (FK) | INT (11) | Foreign key linking to the Users table |
| application_type | ENUM | Type of application ('Full-time Mentor', 'Freelancer') |
| status | ENUM | Status of the application ('Viewed', 'Not Viewed', 'Accepted', 'Rejected', 'Withdrawn') |
| submission_at | DATE | Date when the application was submitted |
| reviewed_at | INT (11) | Date when the application was reviewed |
| notes | TEXT | Review notes for the application |
| portfolio_link | VARCHAR (255) | URL link to the applicant's portfolio |
| document_path | VARCHAR (255) | File path to application documents |

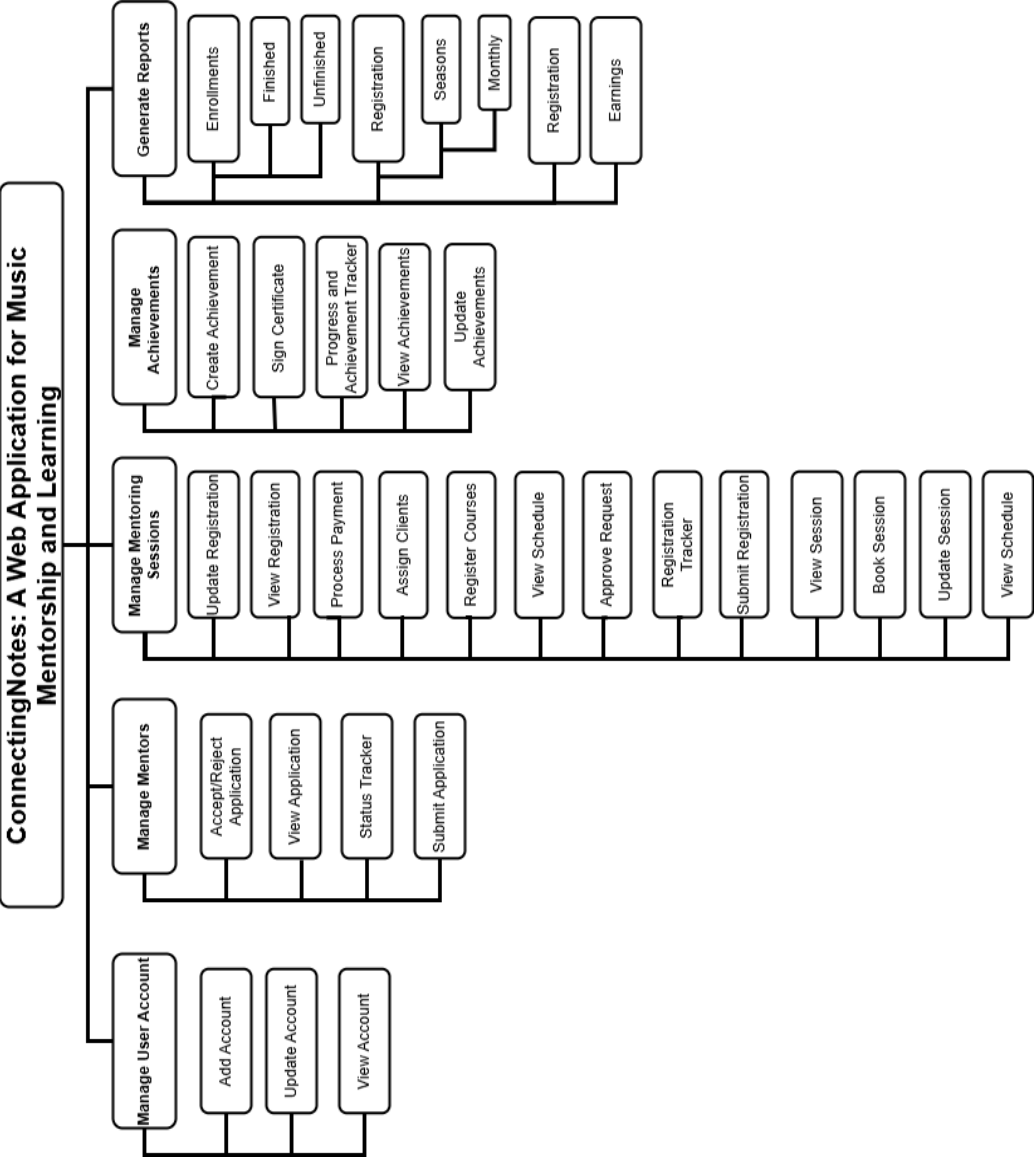
Table 13. Documents

This table contains documents submitted by applicants.

| Field Name | Data Type | Description |
|------------------------|------------------|--|
| document_id (PK) | INT (11) | Unique identifier for each document |
| application_id (FK) | INT (11) | Foreign key linking to the Applications table |
| document_type (FK) | ENUM | Type of document ('Resume', 'BIR Registration', 'NBI Clearance', etc.) |
| document_path | VARCHAR (255) | File path to the uploaded document |
| uploaded_at | TIMESTAMP | Timestamp when the document was uploaded |

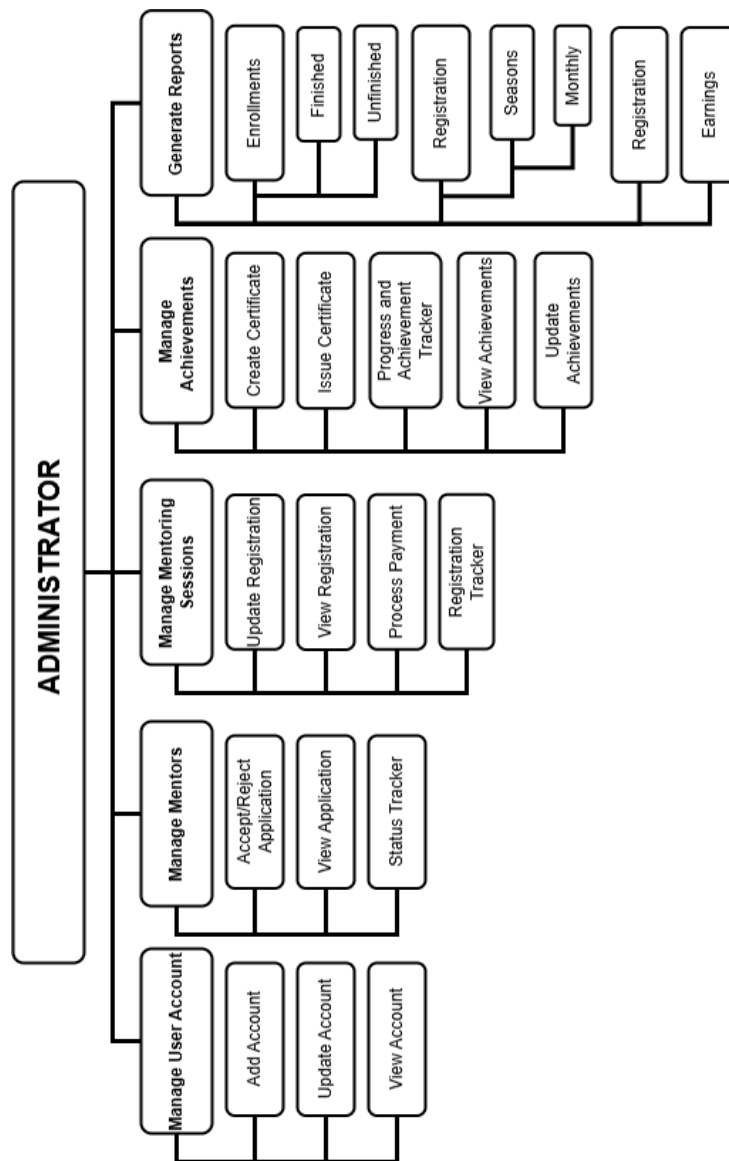
3.2.2 Program Hierarchy

3.2.2.1. Top Level



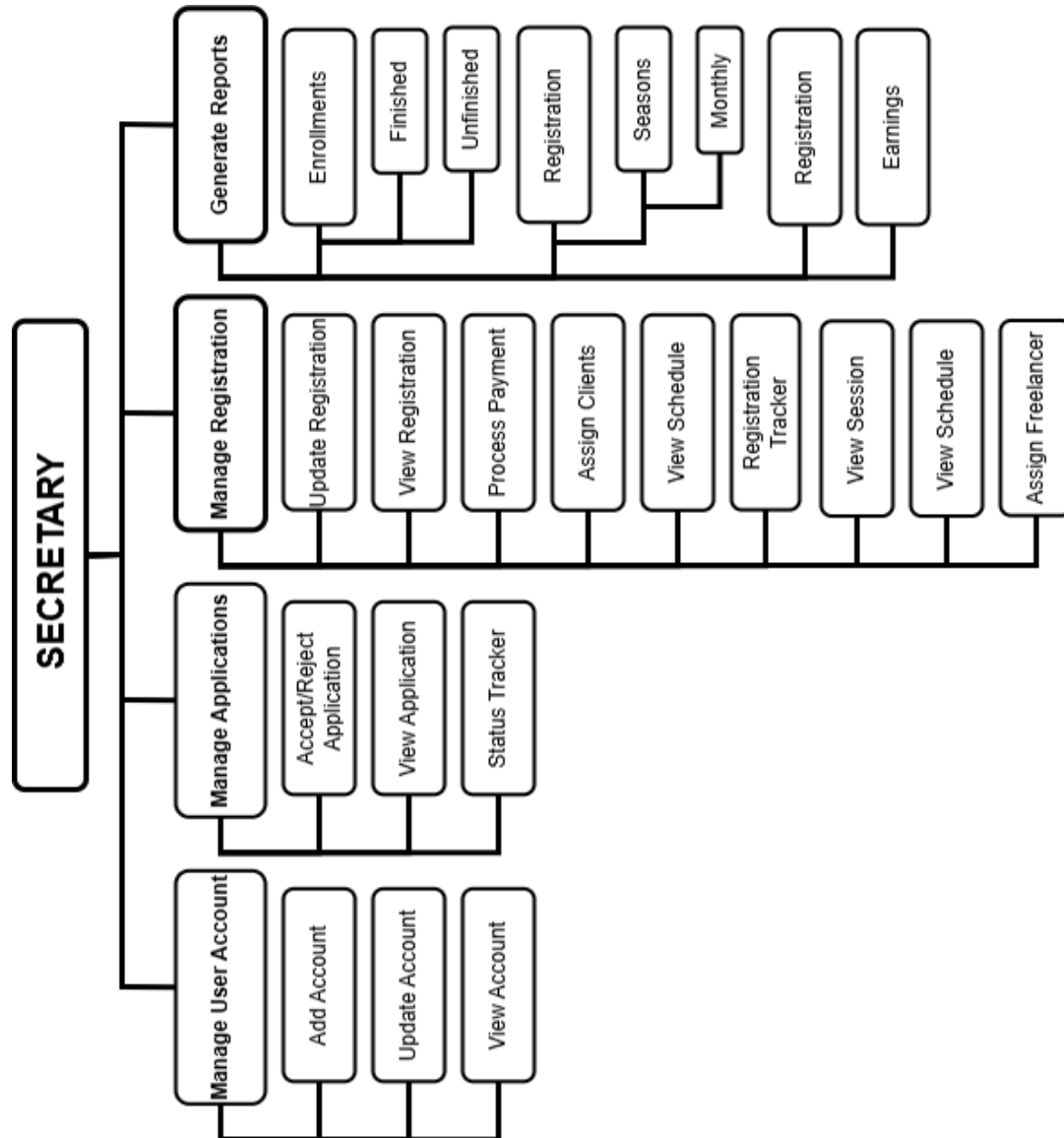
TOP LEVEL PROGRAM HIERARCHY

3.2.2.2. Administrator Level



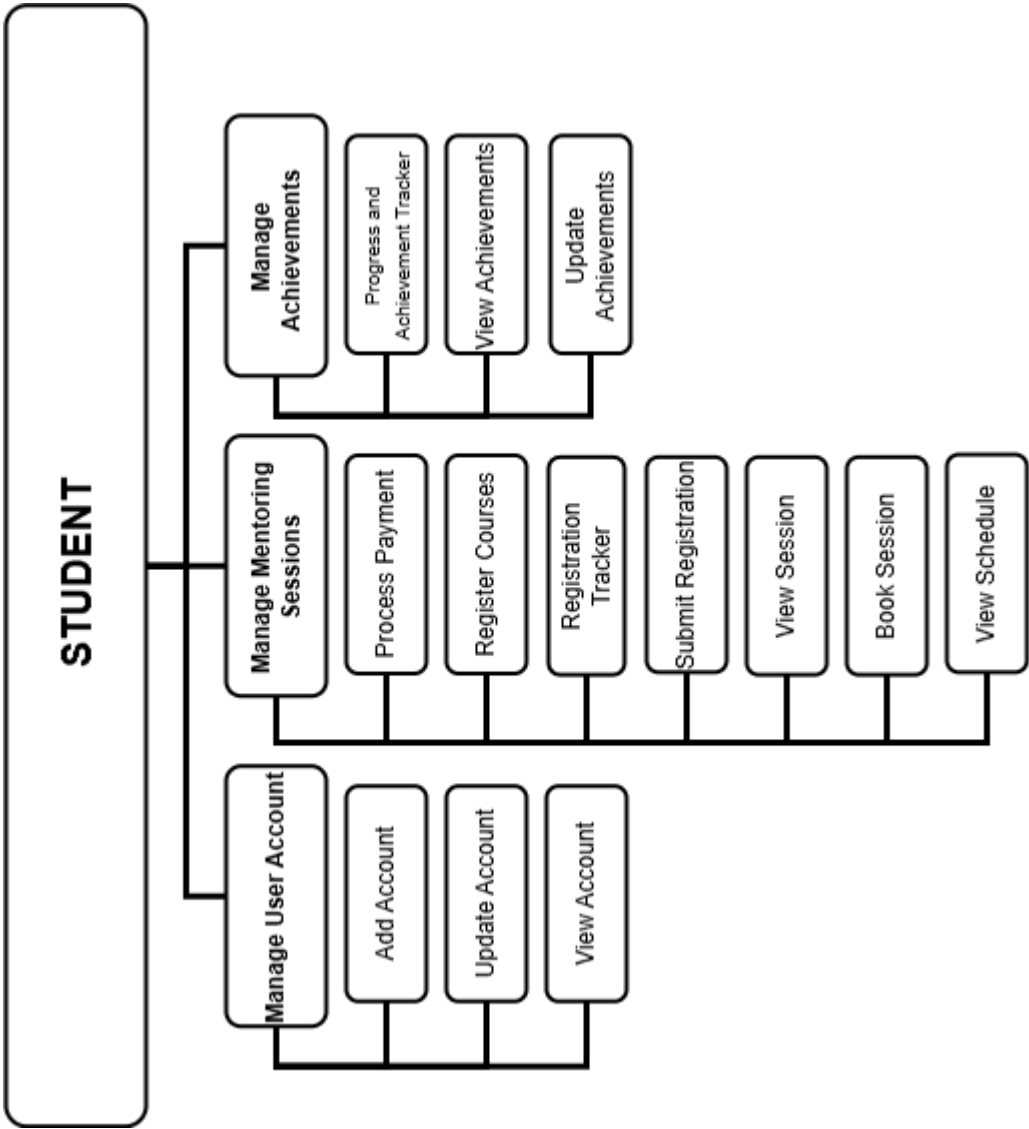
ADMINISTRATOR LEVEL PROGRAM HIERARCHY

3.2.2.3. Secretary Level



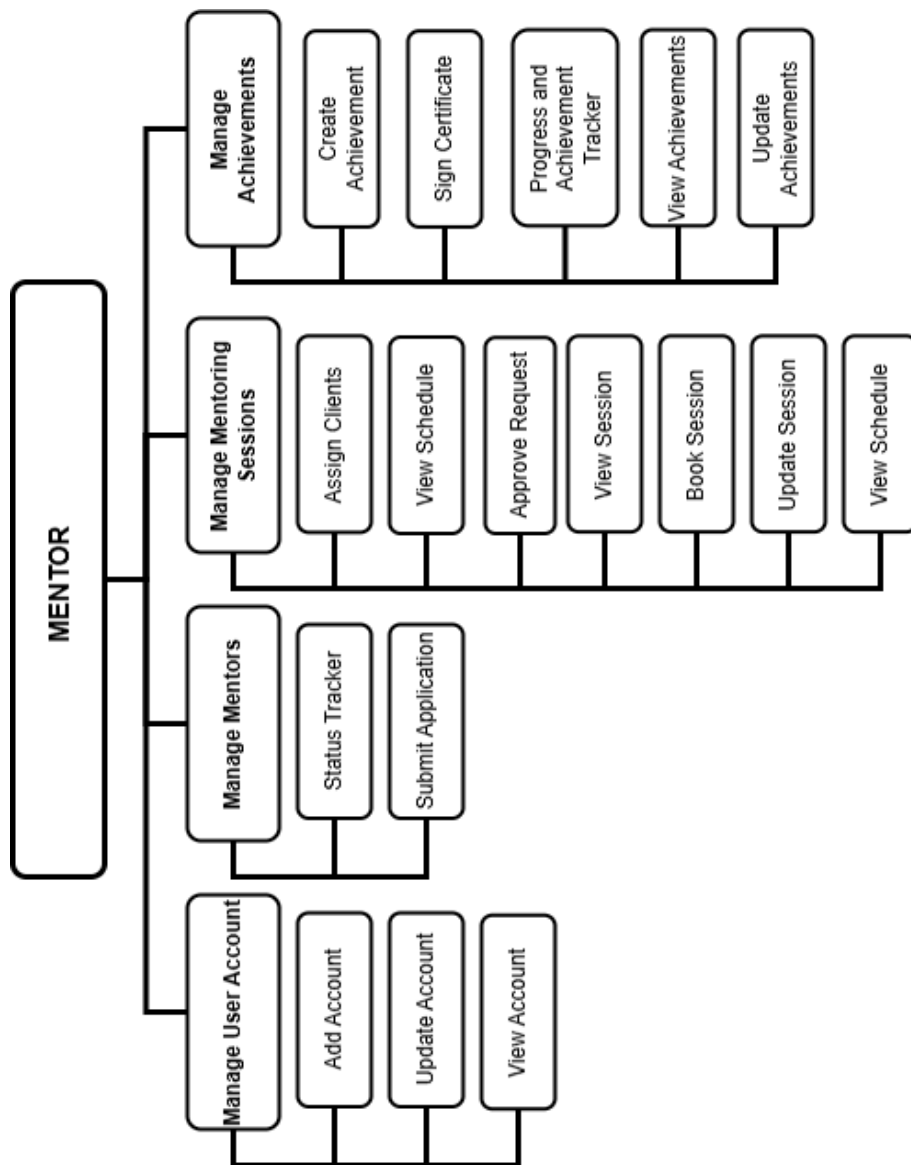
SECRETARY LEVEL PROGRAM HIERARCHY

3.2.3.4 Student Level



STUDENT LEVEL PROGRAM HIERARCHY

3.2.3.5. Mentor Level



MENTOR LEVEL PROGRAM HIERARCHY

3.3 Development and Testing

This section discusses the hardware and software that will be used during the development and testing processes of the proposed system ConnectingNotes: A Web Application for Music Mentorship and Learning.

3.3.1 Development

During the development, the researchers used a laptop with the following specifications: Intel(R) Celeron(R) N4020 CPU @ 1.10GHz 1.10 GHz as a processor, 4.00 GB Random Access Memory (RAM), and 230GB Solid State.

In developing the system, the researchers used a 64-bit Windows 11 Operating System together with the following softwares: Tailwind CSS version 3.4.3, a free front-end framework for responsive user interfaces, faster, and easier web development, allowing the researchers to create responsive designs for the system. HTML5 is a standard markup language for developing web pages. This was used to create the basic system design, which was compatible with all browsers and supported the web pages' interactive content. Google Chrome version 112.0.5615.49 is used to test the overall functionalities of the system. XAMPP version 8.1.12 is a simple, lightweight Apache distribution for creating a local web server for testing and deployment purposes, which contained PHP version 8.2.5, used as the back-end programming language. MariaDB is an open-source relational database management system that supported the system's back-end to improve the user experience. It was used to develop the database for the system. Visual Studio Code version 1.68.0 is used as a code editor and an

Integrated Development Environment (IDE).

The system integrates seven (7) Application Programming Interfaces (APIs) to enhance its functionality: Authentication and User Management API that handles secure user registration, login authentication (via JWT or OAuth), and role-based access control for students, mentors, freelancers, secretaries, and administrators. Document Upload and Verification API that manages the submission and verification of required mentor and freelancer documents, such as IDs, certifications, and contracts. Scheduling and Booking API that facilitates session scheduling by allowing students to book mentoring sessions, enabling mentors to manage availability, and providing automated reminders. Virtual Classroom API that supports learning management features, including assignment submissions, announcements, comments, and access to educational materials. Payment Gateway API that integrates payment processing services (such as PayPal, Stripe, or GCash) to handle student payments for mentoring sessions and generate digital receipts. Achievement and Certification API that tracks student progress, updates achievements, and generates downloadable digital certificates for completed courses. Reporting and Analytics API that generates administrative reports on session completions, student and mentor activity, and financial transactions related to bookings and payments.

3.3.2 Testing

Testing will be conducted to evaluate the system's functionalities and verify whether the expectations and requirements are met. The testing process will

involve key users of the system, including the Admin, Secretary, Students, and Mentors. The target population for this phase will consist of approximately eight (8) participants. The proposed testers will include one (1) Admin, one (1) Secretary, two (2) Students, and four (4) Mentors/Freelancers. The Administrator and Secretary will be selected due to their roles in managing user accounts, session records, and achievement issuance. Students will be chosen based on their active enrollment status and willingness to participate. One student will be a beginner, and the other a returning student, ensuring perspectives from different levels of experience. Mentors, both full-time and freelancers, will be included to provide diverse insights regarding teaching workloads and familiarity with digital tools. The researchers will employ purposive sampling, having already identified ideal participants who match the system's user roles.

Recruitment of participants will be conducted through personal visits to the Herliz Music Studio and Instruments office. The researcher will coordinate directly with the administrator and secretary to identify suitable participants who match the required system user roles. Once potential participants are identified, the researcher will personally invite them by visiting the studio, explaining the purpose of the testing, and providing printed copies of the consent forms. For students, especially minors, coordination will also be made with their parents or guardians to secure informed consent. Mentors, both full-time and freelancers, will be selected based on their current involvement in teaching at Herliz. During the visits, the researcher will ensure that each invited participant understands the nature of the testing, the voluntary nature of their involvement, their expected role in the

process, and the data confidentiality and privacy handling protocols in place. This approach will foster trust and ensure that participants are fully informed and willing to contribute meaningfully to the evaluation process.

Participants will be selected based on predefined criteria to ensure that feedback is gathered from appropriate and relevant system users. Students must be currently enrolled in Herliz Music Studio, must have access to a device with internet connectivity. One beginner and one returning student will be selected to reflect varying levels of familiarity with music learning and digital platforms, and if the student participant is a minor (under 18 years old), written consent from a parent or legal guardian will be required before participation. Mentors (Full-time and Freelancers) must be officially affiliated with Herliz Music Studio either as a full-time mentor or a registered freelancer, must have at least one active student assigned or in the process of accepting student bookings, and must be willing to participate in digital platform testing and provide constructive feedback. Secretary and Administrator must be currently assigned to their respective roles at Herliz Music Studio, and must be involved in daily administrative or operational tasks of the studio, including system-based transactions. Participants will receive a small token of appreciation, such as a certificate of participation or a gift item, in recognition of their time and contribution to the study.

Informed consent will be obtained from all participants before the start of the testing. The consent form will include a clear explanation of the testing's purpose, the tasks involved, the estimated duration, and assurances of voluntary participation and confidentiality. For minor participants, informed consent will be

co-signed by a parent or legal guardian to ensure compliance with ethical standards concerning child involvement in research.

Participants will be excluded from the testing process based on the following criteria, which go beyond simply being the opposite of the inclusion factors. These criteria will ensure the quality and relevance of feedback. Individuals who do not perform any of the user roles defined within the system (e.g., students, mentors, secretary, administrator) will be excluded, as they would not be able to provide meaningful or practical feedback based on actual system use. Candidates who express scheduling conflicts, limited availability, or an unwillingness to complete the tasks and usability assessment will be excluded to maintain consistency and fairness in the evaluation process. Individuals without access to a functional device or stable internet connection will be excluded to avoid external issues that might affect the accuracy of testing feedback. Individuals who demonstrate difficulty understanding or navigating basic system operations, not due to system usability but due to their unfamiliarity with digital tools, will be excluded. The goal will be to evaluate the system—not train users unfamiliar with technology. Individuals directly involved in the system's development (such as developers or researchers) will be excluded to prevent biased evaluations and ensure the integrity of the results.

Given that Herliz Music Studio and Instruments accepts students of varying age groups, the researcher anticipates the potential involvement of minors (students below 18 years old) in the usability testing process. Since minors are considered a vulnerable population in research ethics, appropriate measures will

be put in place to protect their rights and ensure ethical compliance once ethical clearance is obtained. Specifically, the researcher plans to coordinate directly with the students' parents or legal guardians during the recruitment phase. Before involving any minor in the testing, the researcher will conduct a personal visit to the studio to explain the purpose of the study to both the student and their guardian. A formal informed consent form will be provided and signed by the parent or guardian. Additionally, assent will be obtained from minor participants aged 13 to 17. These minors will be given an age-appropriate explanation of the study and will be asked to sign an assent form to confirm voluntary participation. The testing process for minors is planned to be conducted in a safe, non-intrusive environment, during appropriate hours, and under the supervision of the staff at Herliz Music Studio. At no point will minors be left unsupervised, and their privacy and comfort will be prioritized throughout the process. These safeguards are intended to ensure that vulnerable participants are respected, informed, and protected throughout their involvement in the research.

All participants will be properly oriented with the purpose of the system and guided on how to interact with its features. The Admin tester will be assigned to monitor system operations, check for conformity with organizational needs, and manage user roles and access levels. The Secretary tester will handle administrative tasks such as user applications, document verification, and session coordination to ensure a smooth workflow for both mentors and students. The Student testers, as the primary users of the system, will test the ease of booking sessions, accessing learning materials, and tracking progress through the

achievement system. They will also test the responsiveness of virtual classroom functionalities, including assignment submission, announcement viewing, and mentor interaction. The Mentor/Freelancer testers will play a key role in testing features related to scheduling, student interaction, and achievement issuance. They will evaluate the system's ability to manage availability, respond to student inquiries, and issue certificates for completed sessions. Additionally, they will provide feedback on how well the system supports mentorship engagement, lesson delivery, and feedback processes.

The researcher is a fourth-year Bachelor of Science in Information Technology (BSIT) student who has undergone formal training in systems analysis, software engineering, and user experience evaluation. As part of their academic curriculum, the researcher has completed case studies and software testing exercises under the guidance of experienced faculty. Furthermore, the researcher has been collaborating closely with Herliz Music Studio throughout the capstone project, gaining valuable insight into the operational workflow and user expectations of the system. These qualifications equip the researcher to conduct the testing procedures, participant interactions, and data analyses in a professional and ethical manner once ethical clearance is granted.

The study is planned to last seven (7) months, with testing expected to span two (2) weeks. Before testing begins—and only after ethical clearance is granted—the researchers will visit the participants' respective offices or locations to formally invite them and will provide Informed Consent Forms outlining the study's purpose, procedures, and their rights. The research will be localized to the operations

defined within the study's scope before any final decisions are made. The researchers will offer comprehensive information, including details about the testing tools, which participants will have the opportunity to review and discuss. A team member will guide each participant through the provided information and will address any questions. Participants will be given sufficient time to decide whether they wish to participate. Participation in the testing process will be entirely voluntary. Those who agree will sign the consent form to confirm their willingness to take part. Each participant will have twenty (20) minutes to interact with the system, followed by another twenty (20) minutes to complete functionality and usability test cases and provide feedback. If participants encounter technical issues during testing, researchers will be present to assist. Clear instructions, a stable testing environment, and backup plans will be put in place. If a participant withdraws, a replacement will be recruited. If the Administrator withdraws, special consideration will be given due to their critical role. To maintain confidentiality, all testing data will be anonymized during analysis and stored securely. Access to testing records will be restricted to the research team only. These measures are in place to ensure both system quality and participant protection.

To ensure confidentiality and privacy, all collected data will be treated as confidential and will be accessible only to authorized researchers. Participants may withdraw from testing at any time if they feel their privacy is compromised. Providing names in test cases will be optional and unnecessary. Researchers will implement strict data protection measures in compliance with the Data Privacy Act of 2012. All collected data will be anonymized to protect participants' identities.

Upon completion of the study, all data will be securely disposed of or deleted to prevent unauthorized access.

Individuals who are not affiliated with Herliz Music Studio will be excluded from participation. Children under the age of 13 who do not have parental or guardian consent will also not be eligible. Additionally, participants who are unable to use a web-based platform due to technological limitations will not be included in the study.

Two (2) testing tools will be used to assess the overall functionality and usability of the system. Test cases will guide the users through the system's functionalities.

The **User Functionality Test** involves letting the testers use the system and evaluate each feature and capability. The testers are guided by test cases that they must follow. The researchers give predicted outcomes to aid in testing. A "PASS" remark indicates that the module test was successful, but a "FAIL" remark indicates that functionality failed and must be corrected for possible improvement, as well as that the result is unsatisfactory. Participants are free to remark and offer ways to improve the system. The test concludes with the tester's name and signature over the printed name.

The **User Usability Test** allows testers to use the system and assess the learnability and user-friendliness of the system's user interface for each target user. Users will be directed by a series of Likert scale items that they will follow. The webpage's style, color scheme, learnability, navigation, and general system friendliness are all examined. A Likert scale (5 for Excellent, 4 for Very Good, 3 for

Average, 2 for Fair, and 1 for Poor) will be used to assess the tester's perceptions of the aforementioned criteria.

After completing the testing phase, the researchers will collect and analyze all the gathered data from the participants. The data collection period from the participants is expected to last for two (2) weeks. For the usability test, researchers will employ a Likert scale categorization, followed by statistical analysis using weighted mean and grand mean methodologies post-data collection. The weighted mean is particularly significant as it provides the summation of the averages of each criterion utilized in evaluating the system. To establish the range of the 5-point Likert-type scale, the difference between the maximum and minimum values ($5 - 1 = 4$) is divided by five ($4 \div 5 = 0.80$). Subsequently, the least value of the scale (1) is added to identify the maximum value within this range. To ascertain the subsequent range, the last range is incremented by 0.8. The table below illustrates the interpreted results utilizing the aforementioned scale.

| Value | Description | Range |
|-------|-------------|-------------|
| 5 | Excellent | 4.21 - 5.00 |
| 4 | Very Good | 3.41 - 4.20 |
| 3 | Average | 2.61 - 3.40 |
| 2 | Fair | 1.81 - 2.60 |
| 1 | Poor | 1.00 - 1.80 |

Table 14. *Likert Scale*

In the functionality tests, researchers will review the remarks provided by the participants. If the majority of the responses indicate "OK," it signifies that the functionality of each module meets the evaluator's standards. Conversely, if a

respondent indicates "NOT OK," it indicates that the functionality falls short of the tester's expectations, signaling a need for improvement. The comments and suggestions provided by the participants will serve as guidance for researchers to enhance the system's functionalities. To analyze the results of the usability test, the following statistical method was used:

Weighted Mean Formula:

Table 17. Summary of Testers

| Testers | Number of Participants |
|---------------|------------------------|
| HMS Admin | 1 |
| HMS Secretary | 1 |
| HMS Students | 2 |
| HMS Mentors | 4 |
| Total | 8 |

The results of the user functionality and usability tests were favorable, and all the testers expressed their satisfaction with the system's performance in each test case; the functionality test received a positive remark, indicating that the functionalities met the specified requirements. The charts below show visual representations of the outcomes for each criterion on the usability test.

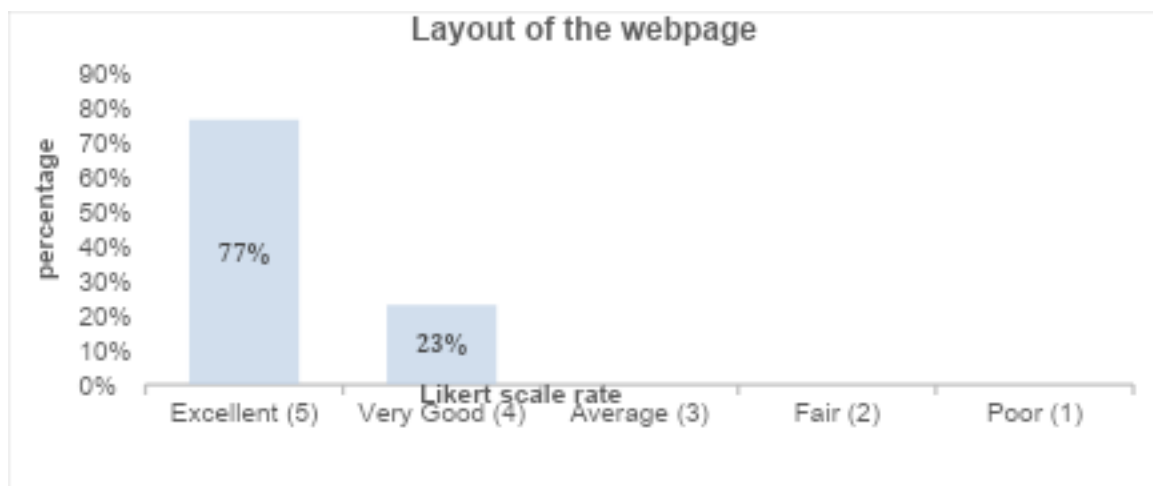


Figure 1. Testing Results for Criteria No. 1

For the first criterion (layout of the webpage), the bar graph shows that most of the respondents, 10 out of 13, gave an "Excellent" remark while the few other testers gave a "Very Good" Remark.

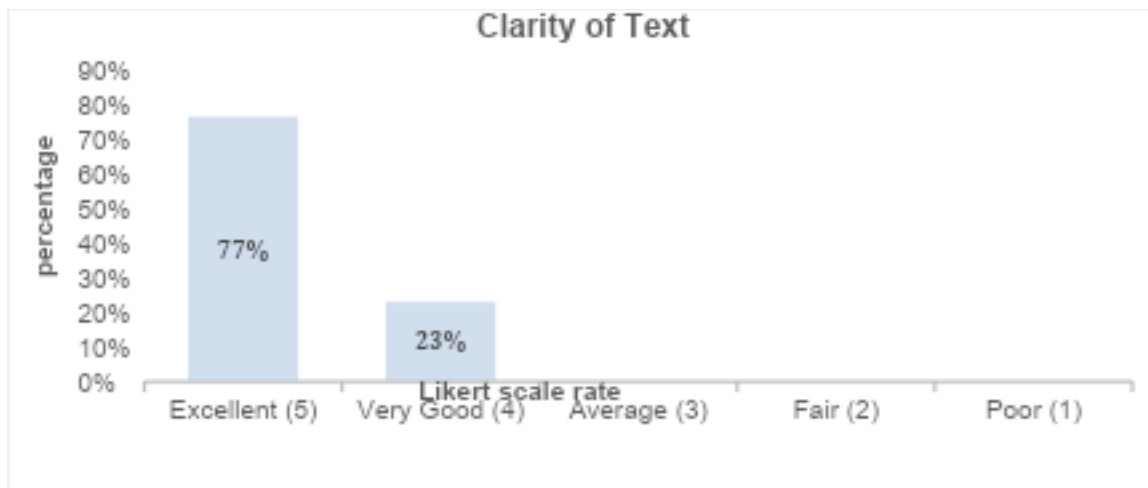


Figure 2. Testing Results for Criteria No. 2

For the second criterion (clarity of text), the bar graph shows that the majority gave an "Excellent" remark while the few other testers gave a "Very Good" rating.

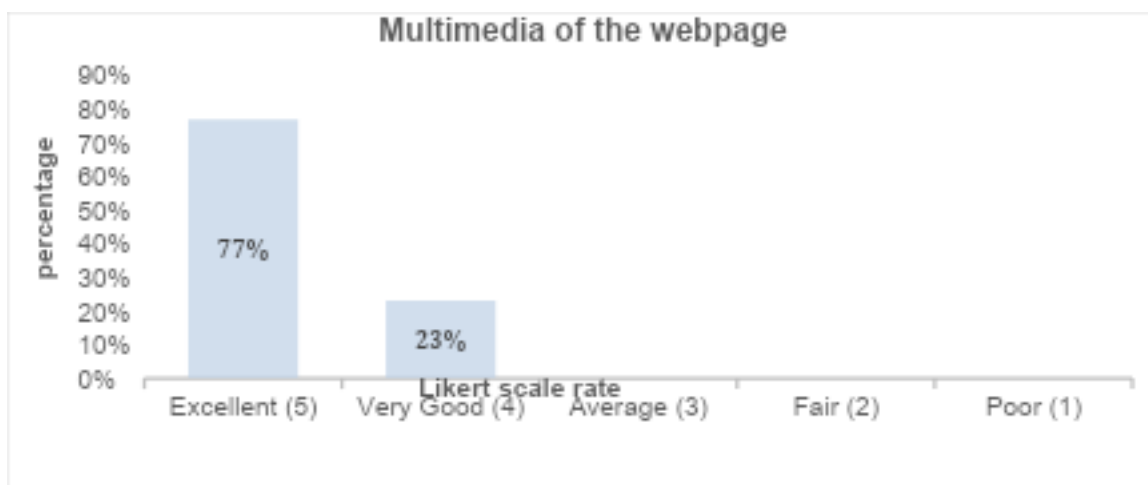


Figure 3. Testing Results for Criteria No. 3

For the third criterion (multimedia of the webpage), the bar graph shows that the majority gave an "Excellent" remark while the few other testers gave a "Very Good" rating.

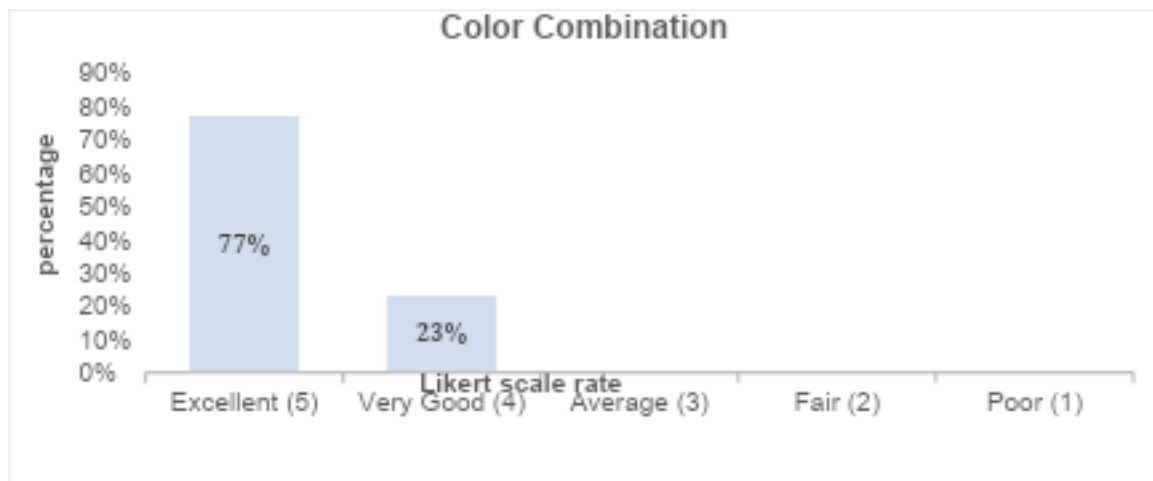


Figure 4. Testing Results for Criteria No. 4

For the fourth criterion (color combination), the bar graph shows that the majority gave an "Excellent" remark while the few other testers gave a "Very Good" rating.

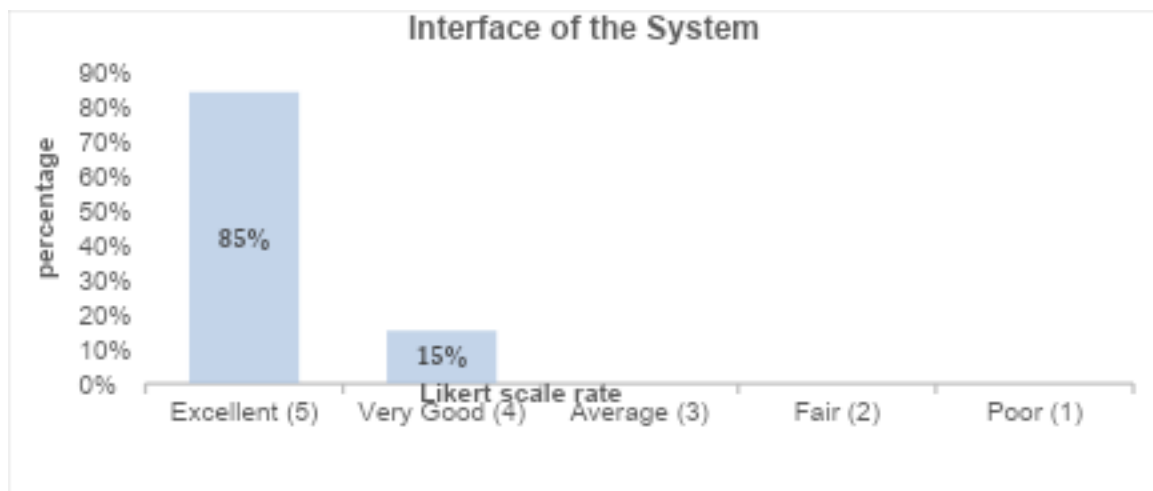


Figure 5. Testing Results for Criteria No. 5

For the fifth criterion (interface of the system), the bar graph shows that the majority gave an "Excellent" remark while the few other testers gave a "Very Good" rating.

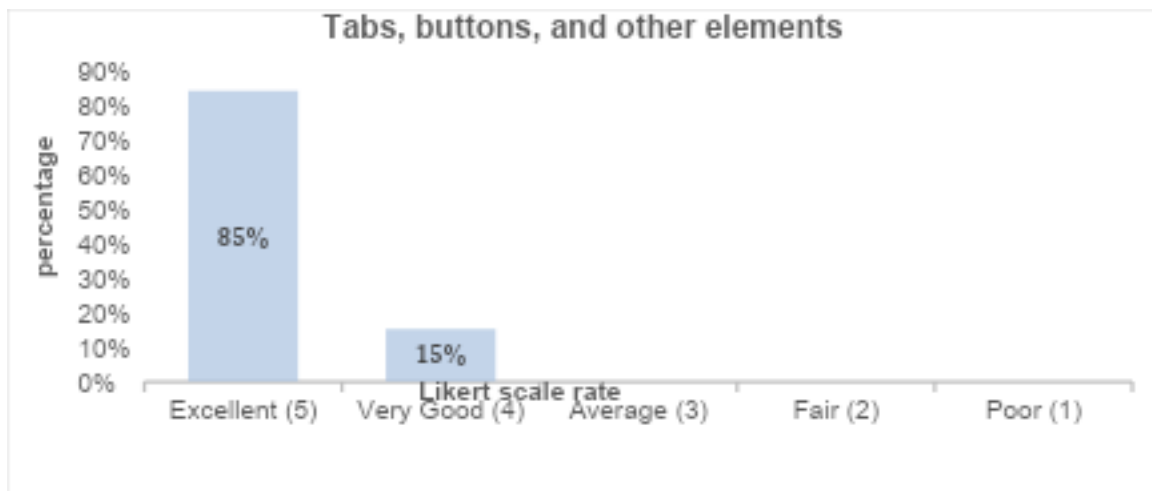


Figure 6. Testing Results for Criteria No. 6

For the sixth criterion (Tabs, buttons, and other elements used on the page are working correctly), the bar graph shows that the majority gave an "Excellent" remark. In contrast, the few other testers gave a "Very Good" rating.



Figure 7. Testing Results for Criteria No. 7

For the seventh criterion (Shifting to another page), the bar graph shows that most of the testers gave an "Excellent" remark while only one tester gave a "Very Good" rating.

Therefore, the usability test results show that most testers think the system provides an excellent user experience. Some respondents gave suggestions and comments on how to enhance the system's overall design. The system's overall design is as follows: the font size and style should be improved. Another suggestion was about the color combination used; they suggested making the color combination lighter. One respondent made another suggestion to allow downloading multimedia images.

In conclusion, this study details the development and testing of the ConnectingNotes System, focusing on hardware and software specifications, participant recruitment, testing methods, and data analysis. By thoroughly testing the system and incorporating user feedback, the researchers aim to improve its effectiveness and usability for managing enrollment and mentorship in Herliz Music Studio, ultimately benefiting its users from the system's potential to make enrollment and mentorship more efficient and convenient, ultimately improving their overall experience within the music learning environment.

Usability Testing Tool

After navigating the application, please answer the following (Check (✓) the rating that fits with your response):

5 - Excellent 4 - Very Good 3 - Average 2 - Fair 1 - Poor

Name of Evaluator: _____

Role being evaluated: HMS Admin

| Criteria | Excellent (5) | Very Good (4) | Average (3) | Fair (2) | Poor (1) |
|--|----------------------|----------------------|--------------------|-----------------|-----------------|
| Layout of the webpage | | | | | |
| Clarity of texts (labels, font size, and font styles of texts) | | | | | |
| Multimedia of the webpage (such as images) | | | | | |
| Color Combination | | | | | |
| The interface of the system is pleasant. | | | | | |
| Tabs, buttons, and other elements used on the page are working properly. | | | | | |
| Shifting to another page | | | | | |

Signature over Printed Name of Evaluator

User Functionality Test

Name of Evaluator: _____

Role being evaluated: HMS Admin

Module: Login/Logout

| Test No. | Test Case | Expected Results | Remarks |
|----------|--|---|---------|
| 1 | Admin inputs a valid email and password to the login page. | Redirects to the dashboard with the respected type of user (HMS Admin). | |
| 2 | Admin inputs an invalid email and password. | Display an error message "Invalid email or password". | |
| 3 | Admin clicks the logout button. | Redirects to the login page. | |

Module: Manage User Accounts

| Test No. | Test Case | Expected Results | Remarks |
|----------|---|---|---------|
| 1 | Access the "Account" page | The system displays the dashboard with options: "Create New Secretary" button, "Profile" button, and "User" dropdown. The system extracts all users from the database and displays them. | |
| 2 | Fill out and submit the secretary registration form | If the form is valid and complete, the system displays the message: "New Secretary Added Successfully." If the form is incomplete, the system displays an error message "Please fill out this field." and prevents | |

| | | | |
|---|--|---|--|
| | | submission until all required fields are completed. | |
| 3 | Click the "User" dropdown and select a category | <p>Clicking "All" displays a list of all users.</p> <p>Clicking "Student" displays a list of students.</p> <p>Clicking "Mentor" displays a list of mentors.</p> <p>Clicking "Secretary" displays a list of secretaries.</p> | |
| 4 | Click on a specific user in the list | The system extracts the user's profile from the database and displays their profile information page. | |
| 5 | Click the "Profile" button | The system extracts the administrator's profile from the database and displays their profile information page with an edit button. | |
| 6 | Modify and submit changes to the administrator profile | <p>If the form is valid and complete, the system displays the message: "User Profile has been updated!"</p> <p>If the form is incomplete, the system displays an error message "Please fill out this field." and prevents submission until all required fields are completed.</p> | |

| | | | |
|---|--|---|--|
| 7 | Leave required fields blank and submit the form | The system displays an error message: "Please fill out this field." and does not proceed with the submission. | |
| 8 | Leave required fields blank and submit the profile update form | The system displays an error message: "Please fill out this field." and does not proceed with the update. | |

Module: Manage Mentors

| Test No. | Test Case | Expected Results | Remarks |
|----------|--|--|---------|
| 1 | Click the "Manage Applications" button | <p>The system extracts all application data from the database and displays a list of applicants with their status (Approved, Rejected, and Interview Scheduled).</p> <p>Each applicant in the list has a "View Applicant" button.</p> <p>If no records exist, the system displays: "No records found."</p> | |
| 2 | Click the "View Applicant" button | <p>The system displays the applicant's information and submitted documents.</p> <p>The system provides options to Approve or Reject the applicant.</p> | |
| 3 | Click the "Approve" button on an applicant's profile | The applicant receives an email notification about their approval. | |

| | | | |
|---|---|--|--|
| | | <p>The system displays the “Manage Onboarding” button within the approved applicant’s profile.</p> <p>If the applicant is already approved, the system displays: “Applicant Already Approved” and does not proceed with approval.</p> | |
| 4 | Click the “Reject” button on an applicant’s profile | <p>The applicant receives an email notification about their rejection.</p> <p>The rejected applicant’s data is archived in the database.</p> <p>If the applicant is already rejected, the system displays: “Applicant Already Rejected”, then deletes the record.</p> | |
| 5 | Click the “Manage Onboarding” button | <p>The system opens a window where the administrator can upload onboarding materials (contracts, guidelines, platform usage tutorials).</p> <p>The applicant must review, accept, and sign the onboarding documents.</p> <p>The system notifies the administrator when the applicant has completed onboarding.</p> | |

| | | | |
|---|------------------------------------|---|--|
| | | The “Confirm Account” and “Provide Access” buttons appear. | |
| 6 | Click the “Confirm Account” button | <p>The system displays a message: “Account Confirmed”.</p> <p>The applicant’s account is successfully saved in the database.</p> | |
| 7 | Click the “Provide Access” button | <p>The approved applicant gains full access to their account.</p> <p>The applicant can now manage their account, courses, and session scheduling.</p> | |

Module: Manage Mentoring Sessions

| Test No. | Test Case | Expected Results | Remarks |
|----------|--|--|---------|
| 1 | Click the "Manage Registration" button | The Dashboard displayed the following: “All Students” button, “Registration of Students” button, “History” button, and Search bar with current and upcoming session lists. | |
| 2 | Click the "All Students" button | A window displays all registered students. | |
| 3 | Click the "Registration of Students" button | A window displays all pending student registrations. | |
| 4 | Click on a specific student from the "All Students" list | The system displays the student’s detailed information. | |

| | | | |
|----|--|--|--|
| 5 | Click on a specific student from the "Registration of Students" list | The system displays the filled-up form containing: Personal information, Parental/guardian details, Musical course(s) of interest, Preferred mentor(s), Signature, and date of submission. | |
| 6 | Click "Next" after verifying the registration form | The system displays the student's scanned birth certificate for verification. | |
| 7 | Click "Next" after verifying the birth certificate | The system displays the scanned and signed Parent's consent form for verification. | |
| 8 | Click "Next" after verifying the Parent's consent | The system displays the proof of payment with a "Confirm" button. | |
| 9 | Click the "Confirm" button after verifying all documents | <p>A message appears: "Registration Successful".</p> <p>The student's details are saved in the database.</p> <p>The system notifies the student about their successful registration.</p> <p>The registered student appears in the "All Students" list.</p> | |
| 10 | Click on a session from the current or upcoming sessions list | The system displays the session details and enrolled students. | |
| 11 | Click the "History" button | The system displays a list of previous sessions with: Search bar and Date filter button. | |

| | | | |
|----|--|--|--|
| 12 | Input a Session ID into the search bar and click Enter | <p>If the Session ID is valid, the system displays: "Session found."</p> <p>If the Session ID is invalid, the system displays: "Session does not exist."</p> | |
| 13 | Click the "Date Filter" button, enter a date range, and click "Okay" | <p>If the date range is valid, the system displays: "Date filter is done."</p> <p>If the date range is invalid, the system displays: "Date range syntax is not correct."</p> | |

Module: Manage Achievements

| Test No. | Test Case | Expected Results | Remarks |
|----------|--|---|---------|
| 1 | Click the "Achievement Issuance" button. | The system displays the dashboard containing the "Mentor" button, "Student" button, "Generate Certificate" button, and "Requests" button. | |
| 2 | Click the "Mentor" button to view mentor achievements. | The system displays the list of mentors with an option to manage their achievements. | |
| 3 | Click a mentor from the list. | The system displays the mentor's profile and the set achievements. | |
| 4 | Click the "Student" button to view student achievements. | The system displays the list of students with an option to manage their achievements. | |

| | | | |
|----|---|---|--|
| 5 | Click a student from the list. | The system displays the student's profile, achievement progress, and a back button. | |
| 6 | Click an "achievement" to generate a certificate. | The system displays the certificate and an attach signature button. | |
| 7 | Attach a signature and click the "Save and Send" button. | The system validates the form and, if complete, displays the message: "Certificate successfully submitted!" | |
| 8 | Click the "Requests" button to view certificate requests. | The system displays the list of requested certificates. | |
| 9 | Click a requested certificate from the list. | The system displays the certificate form with an edit button. | |
| 10 | Fill up the certificate form, attach a signature, and click "Submit." | The system validates the form and, if complete, displays the message: "Certificate successfully submitted!" | |
| 11 | Submit an incomplete or invalid certificate form. | The system displays the message: "Please fill out this field." and returns to the form. | |
| 12 | Submit a certificate form without attaching a signature. | The system displays the message: "Please input signature." and returns to the form. | |

Module: Generate Reports

| Test No. | Test Case | Expected Results | Remarks |
|----------|-----------------------------|--|---------|
| 1 | Click the "Reports" button. | The system displays the report window with the "Enrollment," | |

| | | | |
|----|--|--|--|
| | | “Registration,” “Invoice,” and “Course” buttons on the upper left, and the “Print” and “Date Filter” buttons on the upper right. | |
| 2 | Click the “Enrollment” button. | The system displays the enrollment reports and charts with the message: “Enrollment.” | |
| 3 | Click the “Registration” button. | The system displays the registration reports and charts with the message: “Registration.” | |
| 4 | Click the “Invoice” button. | The system displays the invoice reports and charts with the message: “Invoice.” | |
| 5 | Click the “Course” button. | The system displays the course reports and charts with the message: “Courses.” | |
| 6 | Click the “Print” button. | The system displays the report sheets and charts along with a secondary print button. | |
| 7 | Click the secondary “Print” button. | The system displays a verification message: “Are you sure to Print the Generated Report?” with an “Okay” button. | |
| 8 | Click the “Okay” button in the print confirmation. | The system validates the report, sends it to the hardware printer, and displays the message: “Printing Report!” | |
| 9 | Click the “Date Filter” button. | The system displays date entry fields and an “OK” button. | |
| 10 | Set a valid date range and click “OK.” | The system validates the date range, displays all | |

| | | | |
|----|---|--|--|
| | | data within the selected range, and shows the message: "Done!" | |
| 11 | Set an invalid date range and click "OK." | The system displays the message: "Set Date Invalid!" and returns to the date selection step. | |
| 12 | Click "OK" without selecting a date. | The system displays the message: "Please select date!" and returns to the date selection step. | |

Signature over Printed Name

Usability Testing Tool

After navigating the application, please answer the following (Check (✓) the rating that fits with your response):

5 - Excellent 4 - Very Good 3 - Average 2 - Fair 1 - Poor

Name of Evaluator: _____

Role being evaluated: HMS Secretary

| Criteria | Excellent (5) | Very Good (4) | Average (3) | Fair (2) | Poor (1) |
|--|----------------------|----------------------|--------------------|-----------------|-----------------|
| Layout of the webpage | | | | | |
| Clarity of texts (labels, font size, and font styles of texts) | | | | | |
| Multimedia of the webpage (such as images) | | | | | |
| Color Combination | | | | | |
| The interface of the system is pleasant. | | | | | |
| Tabs, buttons, and other elements used on the page are working properly. | | | | | |
| Shifting to another page | | | | | |

Signature over Printed Name of Evaluator

User Functionality Test

Name of Evaluator: _____

Role being evaluated: HMS Secretary

Module: Login/Logout

| Test No. | Test Case | Expected Results | Remarks |
|----------|--|---|---------|
| 1 | Secretary inputs a valid email and password to the login page. | Redirects to the dashboard with the respected type of user (HMS Secretary). | |
| 2 | Secretary inputs an invalid email and password. | Display an error message "Invalid email or password". | |
| 3 | Secretary clicks the logout button. | Redirects to the login page. | |

Module: Manage User Accounts

| Test No. | Test Case | Expected Results | Remarks |
|----------|---|--|---------|
| 1 | Click the "Create New Client" button. | The system displays a form with the "Submit" button. | |
| 2 | Fill up the form with valid details and click "Submit." | The system validates the form and displays the message: "New Client Added Successfully." | |
| 3 | Click "Submit" without filling out the form. | The system displays an error message: "Please fill out this field." and returns to the form. | |
| 4 | Click the "User" dropdown. | The system displays the options: "All," "Student," | |

| | | | |
|----|---|--|--|
| | | "Mentor," and "Freelancer." | |
| 5 | Click the "All" button in the User dropdown. | The system extracts all users from the database and displays the list of all users. | |
| 6 | Click the "Student" button in the User dropdown. | The system extracts all students from the database and displays the list of all students. | |
| 7 | Click the "Mentor" button in the User dropdown. | The system extracts all mentors from the database and displays the list of all mentors. | |
| 8 | Click the "Freelancer" button in the User dropdown. | The system extracts all freelancers from the database and displays the list of all freelancers. | |
| 9 | Click on a user profile from the user list. | The system extracts the user's profile and displays the profile information page. | |
| 10 | Click the "Profile" button. | The system extracts the secretary's profile from the database and displays the profile page with an "Edit" button. | |
| 11 | Click the "Edit" button on the profile page. | The system displays the profile form with editable fields and a "Submit" button. | |
| 12 | Fill up the profile form with valid details and click "Submit." | The system validates the form and displays the message: "User Profile has been updated!" | |
| 13 | Click "Submit" without filling out the form. | The system displays an error message: "Please fill out this field." and returns to the form. | |

Module: Manage Mentors

| Test No. | Test Case | Expected Results | Remarks |
|-----------------|---|--|----------------|
| 1 | Click the "Manage Applications" button. | The system extracts all application data and displays a list of applicants with their status, including a "View Applicant" button and "Filter" button. | |
| 2 | Click the "Filter" button and select a specific status (Accepted, Viewed, Not Viewed, and Withdrawn). | The system filters and sorts the list of applicants based on the selected status. | |
| 3 | Click the "View Applicant" button for a specific applicant. | The system displays the applicant's information and documents, including "Send" and "Reject" buttons. | |
| 4 | Click the "Send" button for an applicant. | The system forwards the verified application to the administrator and displays a form requiring date and time input for interview scheduling. | |
| 5 | Enter a valid date and time and click the "Set Schedule" button. | The system validates the input, updates the applicant's status to "Interview Scheduled," and notifies both the Administrator and Applicant via email and in-system notification. | |

| | | | |
|---|--|---|--|
| 6 | Enter an invalid date or time and click "Set Schedule." | The system displays an error message: "Invalid inputs." and returns to the scheduling form. | |
| 7 | Click the "Reject" button for an applicant. | The system sends a rejection notification to the applicant's email and archives the applicant's data in the database. | |
| 8 | Click the "Done" button for an applicant with "Accepted" status. | The system removes the applicant from the list and saves the data in the database. | |
| 9 | Load the application list when no records exist. | The system displays a message: "No records found." | |

Module: Manage Mentoring Sessions

| Test No. | Test Case | Expected Results | Remarks |
|----------|--|--|---------|
| 1 | Click the "Manage Registration" button. | The system displays the dashboard with "All Students," "Registration of Students," "Walk-In Registration," "Freelancer," "History" buttons, a search bar, and session lists. | |
| 2 | Click the "All Students" button. | The system displays a window containing all registered students. | |
| 3 | Click the "Registration of Students" button. | The system displays a window containing all pending student registrations. | |
| 4 | Click the "Walk-In Registration" button. | The system displays the walk-in registration process steps. | |

| | | | |
|----|--|--|--|
| 5 | Click a student from the registered student list. | The system displays the selected student's information. | |
| 6 | Click a pending student registration. | The system displays a filled-up form with proof of payment for verification. | |
| 7 | Click "Next" after verifying a student's form. | The system displays the student's scanned birth certificate for verification. | |
| 8 | Click "Next" after verifying a birth certificate. | The system displays the scanned and signed parent's consent form. | |
| 9 | Click "Next" after verifying the parent's consent form. | The system displays the proof of payment. | |
| 10 | Click "Confirm" after verifying the proof of payment. | The system saves the registration details, displays "Registration Successful," and notifies the student. | |
| 11 | Login the walk-in client's account and click "Register Now." | The system displays the registration form with payment and instrument options. | |
| 12 | Fill out the registration form and click "Next." | The system displays an interface to attach a scanned birth certificate. | |
| 13 | Attach a birth certificate and click "Next." | The system displays an interface to attach a scanned and signed parent's consent. | |
| 14 | Attach a parent's consent form and click "Next." | The system displays an interface to attach proof of payment. | |
| 15 | Attach proof of payment and click "Confirm." | The system saves the registration details, displays "Registration Successful," and notifies the student. | |

| | | | |
|----|--|--|--|
| 16 | Click "History" to view previous sessions. | The system displays a list of previous sessions with a search bar and a date filter button. | |
| 17 | Click a session from previous session lists. | The system displays the session information and student details. | |
| 18 | Click the "Date Filter" button and enter a valid date range. | The system filters sessions by date and displays "Date filter is done." | |
| 19 | Enter an invalid date range in the date filter. | The system displays "Date range syntax is not correct." and prompts for a valid input. | |
| 20 | Search for a session ID using the search bar. | The system validates and displays the session details if the ID exists. | |
| 21 | Enter an invalid session ID in the search bar. | The system displays "Session does not exist." | |
| 22 | Click the "Freelancer" button. | The system extracts freelancer data and displays a list of freelancers with a search bar. | |
| 23 | Click a freelancer from the list. | The system displays the freelancer's profile and schedule with an "Assign Session" button. | |
| 24 | Click "Assign Session" for a freelancer. | The system displays the session schedule record form selection. | |
| 25 | Select a session and click "Assign." | The system validates the assignment and, if successful, displays "Freelancer successfully assigned." | |

Module: Generate Reports

| Test No. | Test Case | Expected Results | Remarks |
|-----------------|--|--|----------------|
| 1 | Secretary clicks on the "Enrollment" button to view registration records. | The system displays a window with "Finished" and "Unfinished" buttons. | |
| 2 | Secretary selects "Finished" in the enrollment records. | The system loads and displays the finished registration records. | |
| 3 | Secretary clicks the "Print" button to generate reports. | The system displays report sheets and charts with a secondary "Print" button. | |
| 4 | Secretary confirms the print request by clicking the secondary "Print" button. | The system displays a confirmation message: "Are you sure to Print the Generated Report?" and validates the request. | |
| 5 | Secretary clicks "OK" after confirming the print request. | The system validates and sends the file to the printer, displaying "Printing Report!" | |
| 6 | Secretary clicks the "Date Filter" button and selects a valid date range. | The system validates the date range and displays "Done!" along with the filtered records. | |
| 7 | Secretary selects an invalid date range. | The system displays an error message: "Date range syntax is not correct." and prompts for a correction. | |
| 8 | Secretary clicks the "Chart" button. | The system displays custom chart options for visualization. | |

| | | | |
|----|---|---|--|
| 9 | Secretary selects "Course" and chooses "Violin" from the course records. | The system displays all violin records and a message: "Violin Records." | |
| 10 | Secretary clicks the "Submit" button to store the generated report in the database. | The system validates and successfully stores the report. | |

Signature over Printed Name

Usability Testing Tool

After navigating the application, please answer the following (Check (✓) the rating that fits with your response):

5 - Excellent 4 - Very Good 3 - Average 2 - Fair 1 - Poor

Name of Evaluator: _____

Role being evaluated: HMS Students

| Criteria | Excellent (5) | Very Good (4) | Average (3) | Fair (2) | Poor (1) |
|--|----------------------|----------------------|--------------------|-----------------|-----------------|
| Layout of the webpage | | | | | |
| Clarity of texts (labels, font size, and font styles of texts) | | | | | |
| Multimedia of the webpage (such as images) | | | | | |
| Color Combination | | | | | |
| The interface of the system is pleasant. | | | | | |
| Tabs, buttons, and other elements used on the page are working properly. | | | | | |
| Shifting to another page | | | | | |

Signature over Printed Name of Evaluator

User Functionality Test

Name of Evaluator: _____

Role being evaluated: HMS Students

Module: Login/Logout

| Test No. | Test Case | Expected Results | Remarks |
|----------|---|---|---------|
| 1 | Student(s) inputs a valid email and password to the login page. | Redirects to the dashboard with the respected type of user (HMS Student). | |
| 2 | Student(s) inputs an invalid email and password. | Display an error message "Invalid email or password". | |
| 3 | Student(s) clicks the logout button. | Redirects to the login page. | |

Module: Manage User Accounts

| Test No. | Test Case | Expected Results | Remarks |
|----------|--|---|---------|
| 1 | Student clicks the "Sign Up Student" button. | The system displays a "Create New Account" form. | |
| 2 | Student fills out the sign-up form and clicks "Sign" button. | The system validates the form and displays the verification page with a "Submit" button. A verification code is sent via email. | |
| 3 | Student enters the verification code and clicks "Submit." | The system validates the code and, if correct, displays "New student" | |

| | | | |
|----|---|--|--|
| | | account created successfully." | |
| 4 | Student enters an incorrect verification code. | The system displays an error message: "Verification Code Invalid!" and prompts the user to re-enter the code. | |
| 5 | Student leaves any required field in the sign-up form blank. | The system displays an error message: "Please fill out this field." and does not proceed. | |
| 6 | Student clicks the "Login" button and selects "Login Student." | The system displays a login form. | |
| 7 | Student enters valid credentials and clicks "Login." | The system validates credentials, extracts student data from the database, and displays the student's profile account. | |
| 8 | Student enters incorrect login credentials. | The system displays an error message indicating invalid credentials. | |
| 9 | Student clicks the "Edit" button in their profile. | The system retrieves and displays the student's profile information in an editable form. | |
| 10 | Student updates profile details and clicks "Submit." | The system validates the form and, if complete, updates the student profile, displaying "User Profile has been updated!" | |
| 11 | Student leaves a required field blank while updating the profile. | The system displays an error message: "Please fill out this field." and does not save changes. | |

| Test No. | Test Case | Expected Results | Remarks |
|----------|--|--|---------|
| 1 | Student fills out the registration form completely and clicks the "Next" button. | The system proceeds to the next step and displays the file attachment window. | |
| 2 | Student submits an incomplete registration form and clicks "Next." | The system displays an error message: "Fill up all the required sections." | |
| 3 | Student attaches the birth certificate and clicks "Next." | The system proceeds to the next step, displaying the attachment window for the parent's consent. | |
| 4 | Student clicks "Next" without attaching a birth certificate. | The system displays an error message: "Please attach a file." | |
| 5 | Student attaches the parent's consent and clicks "Next." | The system proceeds to the next step, displaying the attachment window for the payment slip. | |
| 6 | Student clicks "Next" without attaching the parent's consent. | The system displays an error message: "Please attach a file." | |
| 7 | Student attaches the payment slip and clicks "Confirm." | The system validates the submission and displays a confirmation message: "Registration Submitted." | |
| 8 | Student clicks "Confirm" without attaching a payment slip. | The system displays an error message: "Please attach a file." | |
| 9 | Student clicks the "Book Session" button after selecting a day. | The system validates the request, sends it to the mentor, and displays "Book Request Sent." | |

| | | | |
|----|---|---|--|
| 10 | Student clicks "View" on a session material. | The system displays the session material page. | |
| 11 | Student clicks "Add File," selects a file, and clicks "Submit." | The system validates and submits the file, displaying "Submitted!" | |
| 12 | Student clicks "Done Session" after submitting materials. | The system validates the material, opens the rating and feedback window, and allows submission. | |
| 13 | Student submits a valid feedback and rating. | The system displays "Thank you for your feedback." | |
| 14 | Student writes and posts an announcement. | The system updates the stream and displays "Your post is posted!" | |
| 15 | Student writes and submits a comment. | The system updates and displays the comment with "The comment is already posted!" | |

Module: Manage Achievements

| Test No. | Test Case | Expected Results | Remarks |
|----------|---|---|---------|
| 1 | Click the "Achievements" button | The system extracts all achievements and displays a list of "Achievements/Badge" buttons on the dashboard. If no records are found, the system displays "No records found." | |
| 2 | Click an achievement/badge (Complete Achievement) | The system validates and displays the achievement progress, certificate, and badge with "Share" and "Download" buttons. | |

| | | | |
|---|--|--|--|
| 3 | Click an achievement/badge (In-progress Achievement) | The system displays the achievement/badge progress without a certificate or badge. | |
| 4 | Click the "Download" button | The system downloads the certificate to the device and displays the message: "Download." | |
| 5 | Click the "Share" button | The system displays a sharing window with "Email," "Facebook," and "Instagram" buttons. | |
| 6 | Click the "Email" button | The system shares and sends the certificate via email. | |

Signature over Printed Name

Usability Testing Tool

After navigating the application, please answer the following (Check (✓) the rating that fits with your response):

5 - Excellent 4 - Very Good 3 - Average 2 - Fair 1 - Poor

Name of Evaluator: _____

Role being evaluated: HMS Mentors/Freelancers

| Criteria | Excellent (5) | Very Good (4) | Average (3) | Fair (2) | Poor (1) |
|--|----------------------|----------------------|--------------------|-----------------|-----------------|
| Layout of the webpage | | | | | |
| Clarity of texts (labels, font size, and font styles of texts) | | | | | |
| Multimedia of the webpage (such as images) | | | | | |
| Color Combination | | | | | |
| The interface of the system is pleasant. | | | | | |
| Tabs, buttons, and other elements used on the page are working properly. | | | | | |
| Shifting to another page | | | | | |

Signature over Printed Name of Evaluator

User Functionality Test

Name of Evaluator: _____

Role being evaluated: HMS Mentors/Freelancers

Module: Login/Logout

| Test No. | Test Case | Expected Results | Remarks |
|----------|--|---|---------|
| 1 | Mentor(s)/Freelancer(s) inputs a valid email and password to the login page. | Redirects to the dashboard with the respected type of user (HMS <u>Mentor/Freelancer</u>). | |
| 2 | Mentor(s)/Freelancer(s) inputs an invalid email and password. | Display an error message "Invalid email or password". | |
| 3 | Mentor(s)/Freelancer(s) clicks the logout button. | Redirects to the login page. | |

Module: Manage User Accounts

| Test No. | Test Case | Expected Results | Remarks |
|----------|--|---|---------|
| 1 | Click the button to display account management options | The system displays the dashboard with "Sign Up New Account" and "Login" dropdowns. | |
| 2 | Click the "Login" dropdown | The system displays the list of "Login Student" and "Login Mentor" buttons. | |
| 3 | Click the "Login Mentor" button | The system displays the login form. | |
| 4 | Input credentials and click the "Login" button | The system validates the credentials, extracts mentor account details | |

| | | | |
|----|---|--|--|
| | | from the database, and displays the mentor profile. | |
| 5 | Click the "Edit" button on the mentor profile | The system extracts the mentor profile information and displays the form with a submit button. | |
| 6 | Fill up the form and click the submit button | The system validates the input. If valid and complete, it displays a message: "User Profile has been updated!" | |
| 7 | Click the "Sign Up New Account" dropdown | The system displays "Sign Up Student" and "Sign Up Mentor" buttons. | |
| 8 | Click the "Sign Up Mentor" button | The system displays the mentor sign-up form. | |
| 9 | Fill up the sign-up form and click the "Sign Up" button | The system validates the form. If valid and complete, it displays a verification page and sends a verification code via email. | |
| 10 | Input verification code and click the "Submit" button | The system validates the code. If valid, it displays a message: "New student account created successfully." | |
| 11 | Leave any required field blank in the login form | The system displays an error message: "Please fill out this field." | |
| 12 | Leave any required field blank in the edit profile form | The system displays an error message: "Please fill out this field." | |
| 13 | Leave any required field blank in the sign-up form | The system displays an error message: "Please fill out this field." | |
| 14 | Enter an invalid verification code | The system displays an error message: "Verification Code Invalid!" | |

Module: Manage Mentors

| Test No. | Test Case | Expected Results | Remarks |
|-----------------|--|--|----------------|
| 1 | Click the "Apply Mentorship" button | The system displays the application interface with options to choose between "Full-Time Mentor" or "Freelancer". | |
| 2 | Select "Full-Time Mentor" and proceed through document submission steps (Resume, PRC License, Medical Certificate, NBI Clearance, Portfolio) | The system displays each document submission window sequentially, and after submitting all, it displays a confirmation window summarizing the documents. | |
| 3 | Confirm document submission | The system updates the application status to "Under Review," records documents in the database, and sends an email notification confirming the submission. | |
| 4 | Check the application status after submission | The system displays the current status: 'Under Review', 'Accepted', 'Rejected', or 'Withdrawn'. | |
| 5 | If the status is "Accepted," complete the onboarding module by signing the contract, reading guidelines, and watching tutorials | The system records the signed contract, confirms completion of guidelines and tutorials, and grants the mentor full access to their account. | |
| 6 | If the contract is not signed, attempt to proceed | If the contract is not signed, attempt to proceed | |

| | | | |
|----|---|---|--|
| 7 | If the application is rejected, view the response | The system displays the message: "Sorry you have been rejected." | |
| 8 | Withdraw the application by clicking "Withdraw Application" and confirming with "Yes" | The system updates the status to "Withdrawn," sends an email notification, and displays the message "Your application has been successfully withdrawn." | |
| 9 | Click "No" when prompted for withdrawal confirmation | The system closes the confirmation window and returns to the "Manage Application" interface. | |
| 10 | Click on "Guidelines" and mark them as read | The system confirms the guidelines have been read and closes the window. | |
| 11 | Click on "Tutorials" and mark them as completed | The system confirms the tutorials have been completed and closes the window. | |

Module: Manage Mentoring Sessions

| Test No. | Test Case | Expected Results | Remarks |
|----------|---|--|---------|
| 1 | Mentor clicks "Manage Sessions" button. | The system displays all classes of the mentors and the "Create class" button. | |
| 2 | Mentor clicks a specific class. | The system displays information about the class with navigation buttons: "Home," "Calendar," "Stream," "Classwork," "Students," "Grade," | |

| | | | |
|----|--|--|--|
| | | "Booking Requests," and "Attendance." | |
| 3 | Mentor clicks the "Create class" button and fills up the form. | If valid, the system creates a new class and displays it on the homepage. If empty or invalid, an error message prompts: "Please fill up the remaining field." | |
| 4 | Mentor clicks "Stream" and posts an announcement. | The system updates and posts the announcement, displaying the message: "Your post is posted!" | |
| 5 | Mentor clicks "Stream" and adds a comment. | The system updates and posts the comment, displaying the message: "Your comment is added!" | |
| 6 | Mentor clicks "Classwork" and creates session materials. | The system validates, updates, and posts the material, displaying the message: "Posted!" | |
| 7 | Mentor clicks "Students" and invites a student. | If valid, the system sends the invitation and displays the message: "Invite Sent." If empty, an error message prompts: "Please fill up this field." | |
| 8 | Mentor clicks "Booking Requests" and accepts a request. | The system validates, updates the mentor's schedule, and sends the student a message: "Your request has been accepted." | |
| 9 | Mentor clicks "Booking Requests" and declines a request. | The system rejects the request and sends the student a message: "Your request has been declined." | |
| 10 | Mentor clicks "Booking Requests" and schedules a session. | The system updates the mentor's schedule, displaying the message: | |

| | | | |
|----|--|---|--|
| | | "Your schedule has been updated!" | |
| 11 | Mentor clicks "Attendance" and marks students. | The system updates attendance and displays the message: "Attendance has been saved!" If empty, an error message prompts: "Please fill up the remaining field." | |
| 12 | Mentor clicks "Calendar" and marks a session as done. | The system updates the session and displays the message: "Session Done!" | |
| 13 | Mentor updates availability by selecting a day and clicking "Available." | The system updates the schedule. If a day is selected, the "Available" button changes to "Unavailable." | |
| 14 | Mentor clicks "Unavailable" to cancel bookings. | The system updates the schedule, cancels all booked reservations, and notifies students: "Your booked session has been canceled, due to the mentor's schedule." | |

Module: Manage Achievements

| Test No. | Test Case | Expected Results | Remarks |
|----------|---|--|---------|
| 1 | Mentor clicks the "Students" button to view student data. | System extracts and displays a list of students with a button for further actions. | |

| | | | |
|---|---|---|--|
| | | | |
| 2 | Mentor clicks on a student from the list. | System displays the student's profile and achievement progress. | |
| 3 | Mentor clicks the "Create Achievements" button and fills up the form with valid data, then clicks "Post." | System validates the form and displays: "Achievement successfully created!" | |
| 4 | Mentor submits an incomplete or invalid achievement form. | System displays: "Please fill out this field." and prompts the mentor to complete the form. | |
| 5 | Mentor clicks the "Request" button to view achievement requests. | System displays the list of certificate requests. | |
| 6 | Mentor clicks a request from the list. | System displays the certificate form with an edit button. | |
| 7 | Mentor fills up the certificate form, attaches a signature, and clicks "Submit." | System validates the form and displays: "Certificate successfully submitted!" | |
| 8 | Mentor submits a certificate request form with missing or invalid fields. | System displays: "Please fill out this field." and prompts the mentor to complete the form. | |
| 9 | Mentor submits a certificate request without attaching a signature. | System displays: "Please input signature." and prompts the mentor to attach a signature. | |

Signature over Printed Name

CHAPTER IV

RECOMMENDATIONS

After conducting the study, the researchers developed a web-based application aimed at addressing various management challenges experienced by Herliz Music Studio and Instruments. The researchers identified key challenges in managing enrolment and mentorship in Herliz Music Studio and Instruments, including manual enrollment processes, delayed communications between students and mentors, lack of centralized booking and session records, and the absence of systematic achievement tracking and issuance. These issues have led to inefficient enrollment records, miscommunication between students and mentors, and inefficient tracking of student progress and achievements.

The researchers addressed these challenges and limitations by developing an integrated, real-time system that allows users to register online, book and manage sessions, exchange messages instantly, and issue and track student achievements in a centralized platform. This approach significantly reduces administrative workload, enhances transparency, and improves the overall learning experience at Herliz Music Studio and Instruments. The researchers have taken into account the feedback from the client following the system testing and system checking process.

Based on the observed usage and evaluation results, the researchers came up with the following recommendations to further enhance the usability, accessibility, and scalability of the ConnectingNotes system:

1. Develop a dedicated mobile application version to accommodate users who frequently access the system using smartphones or tablets.
2. Implement SMS and push notification alerts to ensure timely updates regarding class bookings, cancellations, and announcements.
3. Enhance mentor profile features to showcase portfolios, credentials, or sample videos within their profiles.
4. Offer training and onboarding for first-time users to help new users (especially parents or less tech-savvy users) understand how to use the system effectively.

CHAPTER V

IMPLEMENTATION PLAN

This chapter presents the implementation plan of the ConnectingNotes: A Web Application for Music Mentorship and Learning, including the conversion plan, deployment, and policies of the said system.

ConnectingNotes works on any web browser supporting HTML 5 (Google Chrome, Firefox, Microsoft Edge, etc.). The minimum hardware requirements to run the system are any operating system but the Windows 10 operating system is recommended and the latest with an AMD Ryzen 3 or Intel Core i3 or higher processor, 4 GB random access memory (RAM), and an internet connection.

In deploying the system, the researcher selects a hosting provider that best suits the needs and budget. The system is uploaded to a web hosting service such as Digital Ocean and to ensure the website is accessible to users and is secure, the research buys a budget domain name system and updates the Domain Name System (DNS) records to match the hosting server's IP address is necessary. After updating the DNS records, the website is launched and made available to users.

To transition from the existing manual processes to the digital platform, a pilot conversion method will be used. This approach begins by rolling out the system to a small, controlled group of users such as the administrator, a secretary, two mentors, and four students to test core functionalities, gather feedback, and address any initial challenges. Once the pilot implementation has been evaluated

and approved, the system will be fully deployed to all users within Herliz Music Studio.

The conversion plan will take three (3) months. It includes the following: preparation of hardware and software requirements, populating the database, user training, and pilot implementation.

| Activity | Months | | | | | | | | | | | |
|--|--------|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | | | | 2 | | | | 3 | | | |
| | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| Preparation of Hardware and Software Requirements | | | | | | | | | | | | |
| Populating Database | | | | | | | | | | | | |
| User Training | | | | | | | | | | | | |
| Pilot Implementation | | | | | | | | | | | | |

Figure 1. System Conversion Plan

Preparation of Hardware and Software Requirements

This phase focuses on establishing the necessary hardware and software for the system. Identifying and setting up a reliable web hosting provider is essential to ensure seamless operation upon deployment. This preparation phase will take two (2) weeks.

Populating the Database

In this phase, all data required for the database will be gathered, including valid records of stranded marine wildlife incidents and sighting data. Additionally, species information and guidelines will be included. This comprehensive data collection is crucial for the system's accuracy and readiness. This phase will also take two (2) weeks.

User Training

This phase involves creating a training module that covers the various functionalities of the system. The module will be implemented to orient users effectively. User feedback will be welcomed and utilized for future enhancements. This training activity is expected to be completed in three (3) weeks.

Pilot Implementation

During this phase, the system will be introduced to a small group of users for testing and evaluation. Users will provide feedback and suggestions, and once the test group approves the system, it will be fully implemented across the 30 coastal areas in Bohol. The pilot implementation is expected to be completed in five (5) weeks and conclude after three (3) months when full system implementation will commence.

Policies of the System

For a successful implementation and utilization of the system, the following policies will be enforced:

1. Only the administrator or secretary can deactivate user accounts to ensure data accuracy and control access.
2. Mentors and students must upload valid and verifiable documents. Accuracy of their provided information is strictly adhered and false submissions may lead to account suspension or deletion.
3. Role-based access will be strictly implemented. For example, students cannot access mentor settings, and only administrators can generate official reports.
4. The administrator and secretary will conduct monthly database backups to ensure data security and integrity.
5. Data deletion is prohibited in the system; however, data can be archived. Authorized users can retrieve inactive data and unarchive it if necessary.

REFERENCES

- [1] J. Balanskat, S. Blamire, and M. Kefala. "The ICT Impact Report: A Review of Studies of ICT Impact on Schools in Europe." Education and Information Technologies, 2006. DOI: <https://bit.ly/3zfCGsk>
- [2] T. Ali, A. Behaz, and D. Djoudi. "Adaptive e-learning environment based on learning styles and its impact on development students' engagement." International Journal of Educational Technology in Higher Education, 2019. DOI: <https://bit.ly/45GNFaz>

- [3] Jamshidi, F., & Marghitu, D. 2020. A Web-Based Platform to Teach Music Online. ProQuest Number: INFORMATION TO ALL USERS, 29288786. bit.ly/3xMtU4P.
- [4] Askeroth, J. 2019. Online learning through emerging innovations and platforms: Digital Badges and MOOCs. ProQuest LLC. <https://bit.ly/4bwnAfG>
- [5] Legaspi, J. B. C., De Angel, R., Lagman, A., & Ortega, J. H. J. C. 2019. Web-based course scheduling system using a greedy algorithm. International Journal of Simulation--Systems, Science & Technology, 20, 141-147. bit.ly/3LbzosU.
- [6] Tang, M. M. 2022. College vocal music teaching design based on the internet platform. Wireless Communications and Mobile Computing, 2022(1), 3590597. <https://bit.ly/3ZDQfwl>
- [7] Douglas, T., Owens. (2024). Interactive Composing in School Jazz Ensembles. 708-730. doi: 10.1093/oxfordhb/9780197574874.013.34 <http://bit.ly/4gGkIA9>
- [8] Pesek, M., Klavz, F., Savli, F., Marolt, M. (2022). Online and in-class evaluation of a music theory e-learning platform, MDPI. <https://bit.ly/421tqV3>
- [9] Chen, Y. and Safian, A.R. (2024) A review of online learning pedagogy and Vocal Music Education, International Journal of Academic Research in Business and Social Sciences. <https://bit.ly/3DWf6De>
- [10] Shreya Jadhav. (2024). Mentor Management System. International Journal for Research in Applied Science and Engineering Technology. <https://bit.ly/4abfppO>
- [11] Andrade, C., Alves, P., Fernandes, J. E., & Coutinho, F. (2020, June). A web platform to support mentoring programs in higher education. In 2020, 15th Iberian Conference on Information Systems and Technologies (CISTI) (pp. 1-6). IEEE. <https://bit.ly/3OCfOYE>