

University of Greenwich

Undergraduate Final Year Project Report

Development of  
A Comprehensive Text Data Collection Platform  
for Training Language Models, Supporting Remote Workers and  
Internal QA Teams

B.Sc. (Hons) Computing (KMD, Myanmar)

COMP-1682

Banner ID - 001430678

Supervisor Name - U Hla Kyaw

# Table of Content

<b>Abstract</b>	<b>4</b>
<b>1. Introduction</b>	<b>6</b>
1.1. Company Background	6
1.2. Current Working System	6
1.3. Current Challenges in Text Data Collection	7
1.4. Aim	7
1.5. Objectives	7
1.6. Scope of the Proposed System	9
1.7. Overview of the Remaining Chapters	9
<b>2. Literature Review</b>	<b>12</b>
2.1. Text Data Collection in Local Languages	12
2.2. Existing Platforms for Text Data Collection	13
2.2.1. Clickworker	13
2.2.2. Amazon Mechanical Turk	16
2.2.3. Toloka	18
2.2.4. Remotasks	20
2.2.5. KoboCollect	22
2.2.6. Key inside of Similar systems	24
2.3. Challenges in Data Collection Systems for Local Languages	24
2.4. Legal, Ethical, Social, and Professional Considerations in Data Collection	25
2.4.1. Legal	25
2.4.2. Ethical	26
2.4.3. Social	26
2.4.4. Professional	26
<b>3. Requirement Analysis</b>	<b>28</b>
3.1. Target Users Identification	28
3.2. Functional Requirements	29
3.3. Use Case Diagrams	31
3.3.1. Use Case Diagram for Admin	31
3.3.2. Use Case Diagram for QA member	32
3.3.3. Use Case Diagram for Freelancer	33
3.4. Non-functional Requirements	34
3.4.1. Performance	34
3.4.2. Security	34
3.4.3. Usability	34
<b>4. Methodology</b>	<b>36</b>
4.1. Agile Development Methodology Overview	36
4.2. Scrum Framework for Agile Project Management	36
4.3. Sprint Planning in Scrum Framework	38

4.4. Why Agile with Scrum Framework would be an appropriate method	38
<b>5. System Design</b>	<b>41</b>
5.1. System Architecture	41
5.1.1. Class Diagram	41
5.1.2. Object Diagram	42
5.1.3. Sequence Diagram	42
5.1.3.1. Sequence diagram for task workflow	42
5.1.3.2. Sequence diagram for balance withdrawal	44
5.1.3.3. Sequence diagram for issue report	45
5.1.4. Activity Diagram	45
5.1.4.1. Activity diagram for user registration and task exploration	45
5.1.4.2. Activity diagram for job posting and task translation	47
5.1.5. Component Diagram	51
5.2. Database Design	52
5.2.1. Entity Relationship Diagram	52
5.2.2. Explanation of Database Structure	53
<b>6. Tools and Technologies</b>	<b>57</b>
6.1. Database Management System	57
6.2. Backend Technologies	57
6.3. Frontend Technologies	58
<b>7. Payment Methods</b>	<b>60</b>
7.1. Overview of Payment Integration	60
7.2. Myanmar Payment Systems	60
7.3. Global Payment Systems	60
7.4. Security Consideration in Payment Processing	61
7.5. Challenges in Payment Integration and Solution	61
<b>8. Risk Management</b>	<b>63</b>
8.1. Risk Identification	63
8.1.1. Technical Risk	63
8.1.2. Security Risk	63
8.1.3. Financial Risk	64
8.1.4. External Risk	64
8.2. Risk Matrix	65
<b>9. User Interface Design</b>	<b>68</b>
9.1 Design Principles	68
9.2 Design System Examples	69
<b>10. Sprint Planning and Execution</b>	<b>72</b>
10.1. Sprint 1	72
10.1.1. Sprint 1 Planning	72
10.1.2. Sprint 1 Execution	76
10.1.3. Sprint 1 Screen Diagrams	78

10.1.4. Sprint 1 Testing	79
10.1.5. Sprint 1 Summary	82
10.2. Sprint 2	83
10.2.1. Sprint 2 Planning	83
10.2.2. Sprint 2 Execution	85
10.2.3. Sprint 2 Screen Diagrams	87
10.2.4. Sprint 2 Testing	88
10.2.5. Sprint 2 Summary	90
10.3. Sprint 3	91
10.3.1. Sprint 3 Planning	91
10.3.2. Sprint 3 Execution	94
10.2.3. Sprint 3 Screen Diagrams	96
10.3.4. Sprint 3 Testing	97
10.3.5. Sprint 3 Summary	99
<b>11. Evaluation and Conclusion</b>	<b>101</b>
11.1. Restate the Aims and Objectives	101
11.2. Evaluation by objectives	101
11.2. Evaluation against justification made	103
11.2.1. Payment	103
11.2.2. System Language	103
11.2.3. Workflow	104
11.2.4. Design	104
11.2.5. Challenges in development process	104
11.2.6. Lesson Learnt and Personal Evaluation	105
11.2.7. Future work	106
11.3. Conclusion	107
<b>References</b>	<b>108</b>
<b>Appendices</b>	<b>111</b>
Section A - Installation and Setup	112
Section B - Screencasting Link	114
Section C - Test Results	115
Section D - Screen Diagrams Sprint 1	192
Section E - Screen Diagrams Sprint 2	199
Section F - Screen Diagrams Sprint 3	202
Section G - Source Code Samples	205

# Abstract

The purpose of this project is to design and develop a text data collection platform for a local company in Myanmar, MyanLang. The text data collection platform will have three user roles: freelancer, QA member and admin. Freelancers will be able to participate in text data collection and will get paid. Collected text will be reviewed by the QA team and Admin is responsible for posting tasks, and making payments. The platform is developed with Scrum framework with agile methodology and tech stacks used in this project are: PostgreSQL, Python, FastAPI, Typescript, and React Framework. The platform was developed with a user-centered approach prioritizing user stories and needs, ensuring quality and efficiency through role-based functionalities.

# **Chapter - 1**

## **Introduction**

# 1. Introduction

## 1.1. Company Background

MyanLang is an IT company in Myanmar founded in 2022 that is currently developing AI language models. The company slogan is "No Language Left Behind," which is the same as Meta's machine translation model name, NLLB, aiming to provide a multi-language support large language model for both Burmese and diverse minority languages in Myanmar.

Located in Yangon, MyanLang is currently running with over 20 employees, including NLP researchers, linguists, programmers, and other staff for digitizing language data and training models. MyanLang also collaborates with NGOs and other academic organizations to research various topics, such as deep learning, bias reduction in NLP models, tokenization technology, and transformer-based models that are suitable for Myanmar's underrepresented languages. MyanLang has successfully developed small language models that are capable of sentiment analysis and named entity recognition.

## 1.2. Current Working System

MyanLang's current process for developing language models includes several processes, such as researching appropriate deep learning technologies, data collection, preprocessing, and training. For data collection, the company purchases books and text data from the publishers and also uses free books from public domains. MyanLang also ethically web-scrapes data from social media, Wikipedia, and other sources without refusing to comply with website terms. Additionally, the data collection process includes extracting text data from public images with OCR technology.

The collected PDFs are digitized into text by company staff to process the developing language model. The company also hires linguists and uses translating tools to convert collected text data from one language to another for back-translation, which helps generate training data.

### 1.3. Current Challenges in Text Data Collection

However, getting enough text data from publicly available sources for developing a language model is still the company's biggest challenge, particularly in minority languages. Most of the minority languages in Myanmar are not digitized, and hiring translators for generating data will face several matters, such as hard-to-find native speakers in Yangon, and the company will also need to provide more office settings for all of the translators for several languages.

To address these challenges, the company decided to adopt a crowdsourcing approach to collect text data for minority languages. This process will involve translating the existing data into other languages and creatively writing in those languages. This process will go on until the data requirements are met, and that data will be used for training language models for both machine translation and general usage.

Considering existing crowdsourcing platforms, such as Amazon Mechanical Turk and KoboCollect, has several limitations. Most of these platforms do not support Myanmar's local languages, and most importantly, they do not integrate with Myanmar's local payment system. Additionally, these platforms may not be user-friendly for freelance translators who are not familiar with technology.

### 1.4. Aim

To develop an online platform for text data collection featuring a user-friendly interface and local language support, enabling integration with the internal QA team for data quality checks and compatibility with Myanmar online payment systems.

### 1.5. Objectives

#### **Research and Analysis**

- 1) Literature review on text data collection
- 2) Analyze similar existing system

- 3) Consider Legal, Ethical, Social, and Professional (LESP) implications
- 4) Identify target users

## **Planning**

- 1) Define functional and non-functional requirements.
- 2) Select appropriate software development methodology and implementation framework

## **System Design**

- 1) Develop system architecture, UML diagrams, and an Initial ERD
- 2) Research suitable technologies and tools for development
- 3) Assess potential risk
- 4) Design User Interface

## **Development**

- 1) Manage backlog identification for each sprint

## **Testing**

- 1) Conduct functional and non-functional testing

## **Evaluation**

- 1) Evaluate the system against the defined aims and objectives
- 2) Justification and personal satisfaction

## 1.6. Scope of the Proposed System

For now, the platform will only include text-based data collection tasks, such as translation and creative writing. Image and audio transcription will remain out of scope.

To ensure the quality of the submitted data, MyanLang will implement a quality assurance process before accepting the submission. Professional linguists, who are directly hired by the company, will review the submission to maintain the data quality.

Payments to the freelancer will be processed with the user-preferred payment system to provide accessibility to all freelance translators who are willing to contribute to this data collection process.

The platform will feature a simple and user-friendly interface to make it easy for freelancers, even with minimal technology knowledge. Data security will be one of the key factors to protect collected data. Additionally, there will also be a scoring system for the freelancer to prevent low-quality submissions from messing up the system workflow.

## 1.7. Overview of the Remaining Chapters

### **Chapter 2 - Literature Review**

This chapter involves researching literature about test data collection platforms, exploring existing data collection platforms, and analyzing challenges and Legal, ethical, social, and professional considerations in data collection platforms.

### **Chapter 3 - Requirement Analysis**

This chapter reflects identifying the target user of the project and analysis of both functional and non-functional requirements.

### **Chapter 4 - Methodology**

This chapter reflects exploring project management methodologies, selecting an appropriate methodology for this project, and explaining how and why the selected methodology (agile with a scrum framework) is suitable for this project.

## **Chapter 5 - System Design**

This chapter aims to provide the architecture of the whole system with UML diagrams and database design.

## **Chapter 6 - Tools and Technologies**

This chapter presents which technologies will be used in this project along with reasons about why these technologies are selected.

## **Chapter 7 - Payment Methods**

This chapter aims to explore why payment systems are integrated in software, how local payment systems work, and what will be the appropriate approach for integrating payment systems in this project.

## **Chapter 8 - Risk Management**

This chapter outlines how risks are identified and how mitigation strategies will be applied to overcome these risks with risk Matrix

## **Chapter 9 - User Interface Design**

This chapter presents the design system of the user interface with examples.

## **Chapter 10 - Sprint Planning and Execution**

This chapter reflects the development process of the project along with detailed sprint planning and sprint outcomes, including screen diagrams.

## **Chapter 11 - Evaluation**

This chapter will reflect evaluating the complete system with defined aims and objectives. The strengths and weaknesses of the system are analyzed, and also mention is made of personal satisfaction and future work.

## **Chapter 12 - Conclusion**

This chapter reflects the summary of the overall project, with key findings and its impacts.

## **Chapter - 2**

### **Literature Review**

## 2. Literature Review

### 2.1. Text Data Collection in Local Languages

Myanmar is rich and diverse in language and culture, and many of its cultures are underrepresented. Burmese, the language of the majority ethnic group, Bamar, is the official language of the country, and other ethnic minorities also have their own languages, cultures, and histories. There are over 130 spoken languages in Myanmar, but not all of them have a standardized written form (Manipabha and Prakash, 2024). Languages such as Jinghpaw (Kachin), Shan, Mon, S'gaw Karen, and Pwo Karen have their standardized orthographies, but most of the others do not have standardized orthographies and use the Burmese alphabet-based writing system, which poses challenges for data collection (Inglis, 2017; Kurabe and Imamura, 2016; Salem-Gervais and Ja Seng, 2022)..

Many underrepresented languages have a limited digital footprint and low-detailed annotation, making it difficult to collect text data through web scraping. To collect data for those languages, it is required to involve native speakers and professional linguists (Kholodna et al., 2024). However, this traditional method relies on professionals and employees to travel to those remote areas and do onsite oral interviews. This approach can also be inefficient and may face a lack of individuals who are willing to travel to those remote areas.

To overcome this challenge, "back-translation" technique become one of the options to break the linguistic resources limit for developing large language models. Diab, Sadat, and Semmar (2024) discussed active learning as a method to optimize data annotation for low-resource languages using LLMs. This data augmentation technique also ensures the data's accuracy by reviewing and verifying the translated text with native speakers or professional linguists. However, another problem is arising here. Back-translation in text data augmentation means translating text from one language to another and back-translating to the original language and reviewing the back-translated text with a native speaker. For the minority languages in Myanmar, existing machine translation systems do not perform well, and the resulting parallel corpus will not be that useful.

Another alternative approach to overcome this problem is the "guided back-translation" approach. "Guided" back-translation replaces the machine translation process with

crowdsourcing to native freelancers and reviews the quality of translations, and this approach will be the best practice for text data collection for minority languages in Myanmar.

## 2.2. Existing Platforms for Text Data Collection

To perform guided back-translation, a platform that enables freelancers to contribute while allowing a QA team to review translations is necessary. Here are some existing platforms with similar functionalities, many of which are primarily used by large tech companies.

### 2.2.1. Clickworker

Clickworker is a popular crowdsourcing platform, particularly in Europe and Asia. This platform offers micro-task management, such as data entry and translation, with a customized workflow to meet the business needs. Workers on this platform have to prefill the detailed skill set, and an automated task assignment system matches the worker with the appropriate tasks to ensure high accuracy and reliability. Clickworker has been an attractive option for organizations due to its quality assurance processes that include automated and manual validation, random testing, and even peer review mechanisms (Clickworker, 2023).



Figure 1: ClickWorker Website Main Page

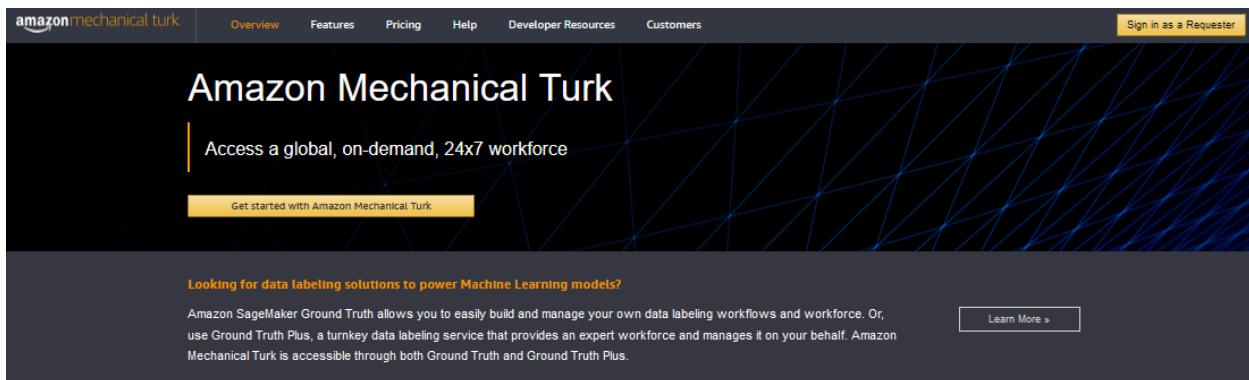
[ Image Available at: <https://www.clickworker.com/> ]

Platform	Clickworker
Primary Function	Crowdsourced microtasking
Task Types	Data entry, surveys, AI training
Payment Model	Pay-per-task (varies)
QA Review	Yes, but it varies

Local Payment System	Limited
Local Language Support	Limited
Simple UI	Yes
Automated Assignment	Partial
Mobile Friendly	Yes (app available)
Offline Functionality	No

## 2.2.2. Amazon Mechanical Turk

Amazon Mechanical Turk, also known as MTurk, is a crowdsourcing platform where requesters post tasks, and workers can browse and complete the task for a fee. The worker pool on MTurk is diverse. MTurk is also popular for its API integration with the ecosystem of Amazon's services. MTurk officially launched in 2005; the user base is growing every year, proving the platform's reliability for large-scale projects. Most of the tasks on MTurk are data annotation, content generation, translation, transcription, and surveys. MTurk also features a quality control process, which is needed for guided back-translation, including multi-level quality review to ensure the data quality.



Amazon Mechanical Turk (MTurk) is a crowdsourcing marketplace that makes it easier for individuals and businesses to outsource their processes and jobs to a distributed workforce who can perform these tasks virtually. This could include anything from conducting simple data validation and research to more subjective tasks like survey participation, content moderation, and more. MTurk enables companies to harness the collective intelligence, skills, and insights from a global workforce to streamline business processes, augment data collection and analysis, and accelerate machine learning development.

While technology continues to improve, there are still many things that human beings can do much more effectively than computers, such as moderating content, performing data deduplication, or research. Traditionally, tasks like this have been accomplished by hiring a large temporary workforce, which is time consuming, expensive and difficult to scale, or have gone undone. Crowdsourcing is a good way to break down a manual, time-consuming project into smaller, more manageable tasks to be completed by distributed workers over the Internet (also known as 'microtasks').

### Benefits

<b>Optimize efficiency</b> <small>MTurk is well-suited to take on simple and repetitive tasks in your workflows which need to be handled manually. Using MTurk to outsource microtasks ensures that work gets done quickly, while freeing up time and resources for the company – so internal staff can focus on higher value activities.</small>	<b>Increase flexibility</b> <small>Scaling up and down a workforce isn't the easiest undertaking. With access to a global, on-demand, 24x7 workforce, MTurk enables businesses and organizations to get work done easily and quickly when they need it – without the difficulty associated with dynamically scaling your in-house workforce.</small>	<b>Reduce cost</b> <small>MTurk offers a way to effectively manage labor and overhead costs associated with hiring and managing a temporary workforce. By leveraging the skills of distributed Workers on a pay-per-task model, you can significantly lower costs while achieving results that might not have been possible with just a dedicated team.</small>
--	---	--

Figure 2: Amazon Mechanical Turk Website Main Page

[ Image Available at: <https://www.mturk.com/> ]

Platform	Amazon Mechanical Turk
Primary Function	Crowdsourced task marketplace by Amazon
Task Types	Surveys, data labeling
Payment Model	Pay-per-task (set by requesters)
QA Review	Yes, but requester-controlled
Local Payment System	No
Local Language Support	Limited
Simple UI	Moderate
Automated Assignment	No
Mobile Friendly	Yes (web-based)
Offline Functionality	No

### 2.2.3. Toloka

Toloka is a crowdsourcing platform developed by Yandex that is designed to be an affordable and flexible solution for data labeling, data verification, and other micro-task-based workflows. The platform also provides API integration with highly customizable task creation. Toloka uses community-driven feedback mechanisms to maintain the task quality, and its adaptable design allows multi-language support. Toloka is popular for cost-effectiveness between academic and commercial organizations.

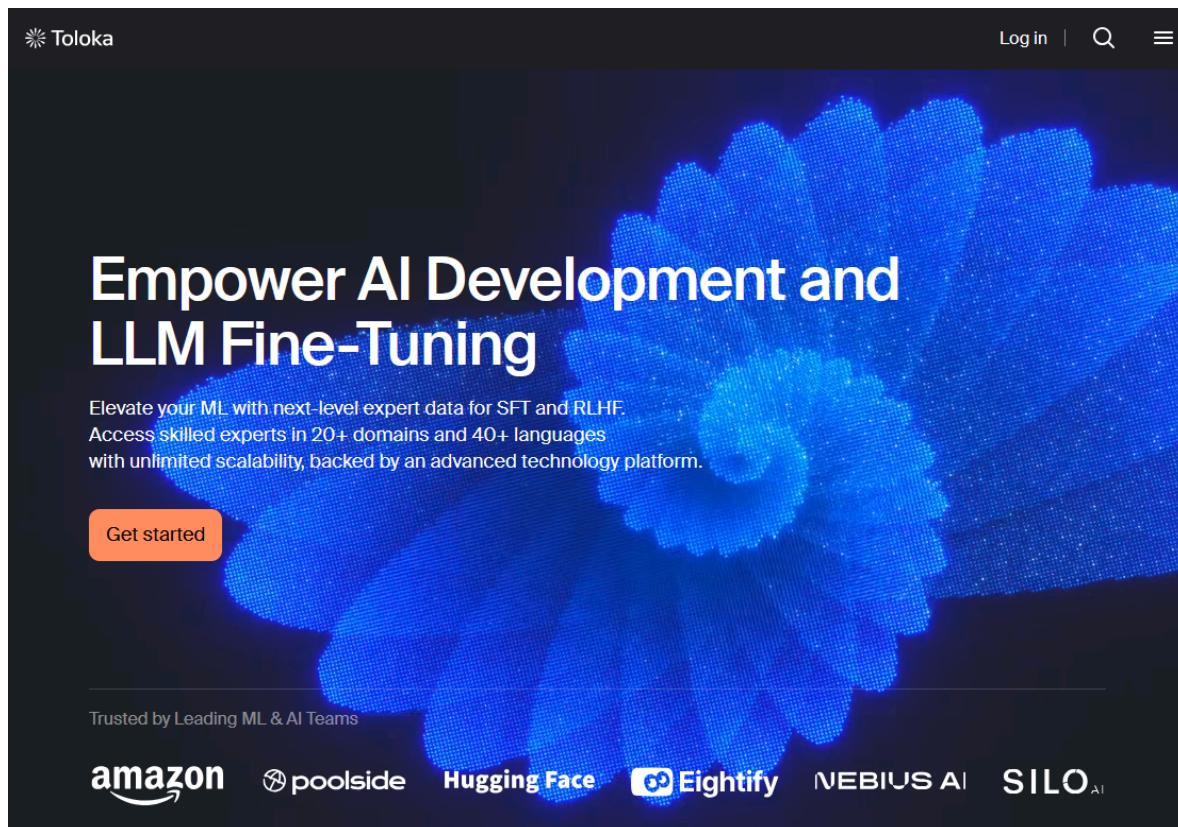


Figure 3: Toloka Website Main Page

[ Image Available at: <https://toloka.ai/> ]

Platform	Toloka
Primary Function	AI and data labeling platform by Yandex

Task Types	Image, text, and speech annotation
Payment Model	Pay-per-task (varies)
QA Review	Yes, robust QA
Local Payment System	Limited (mostly in Russia)
Local Language Support	Strong (multiple languages)
Simple UI	Yes
Automated Assignment	Yes
Mobile Friendly	Yes (app available)
Offline Functionality	No

## 2.2.4. Remotasks

Remotasks is a crowdsourcing platform established by Scale AI in 2017. This platform focuses on connecting workers from underrepresented regions with job opportunities. Most of the tasks on this platform are related to language data collection, which needs an understanding of regional and cultural contexts. The platform is designed to target local language enterprises, particularly for regional dialects, ensuring high-quality language data collection. The commitment to supporting underrepresented communities and languages is the unique selling point of this platform. One major drawback is that the platform has entirely shut down in Kenya, Nigeria, and Pakistan and has stopped accepting new sign-ups in Thailand, Vietnam, and Poland since 2024 (Brandom, 2024).

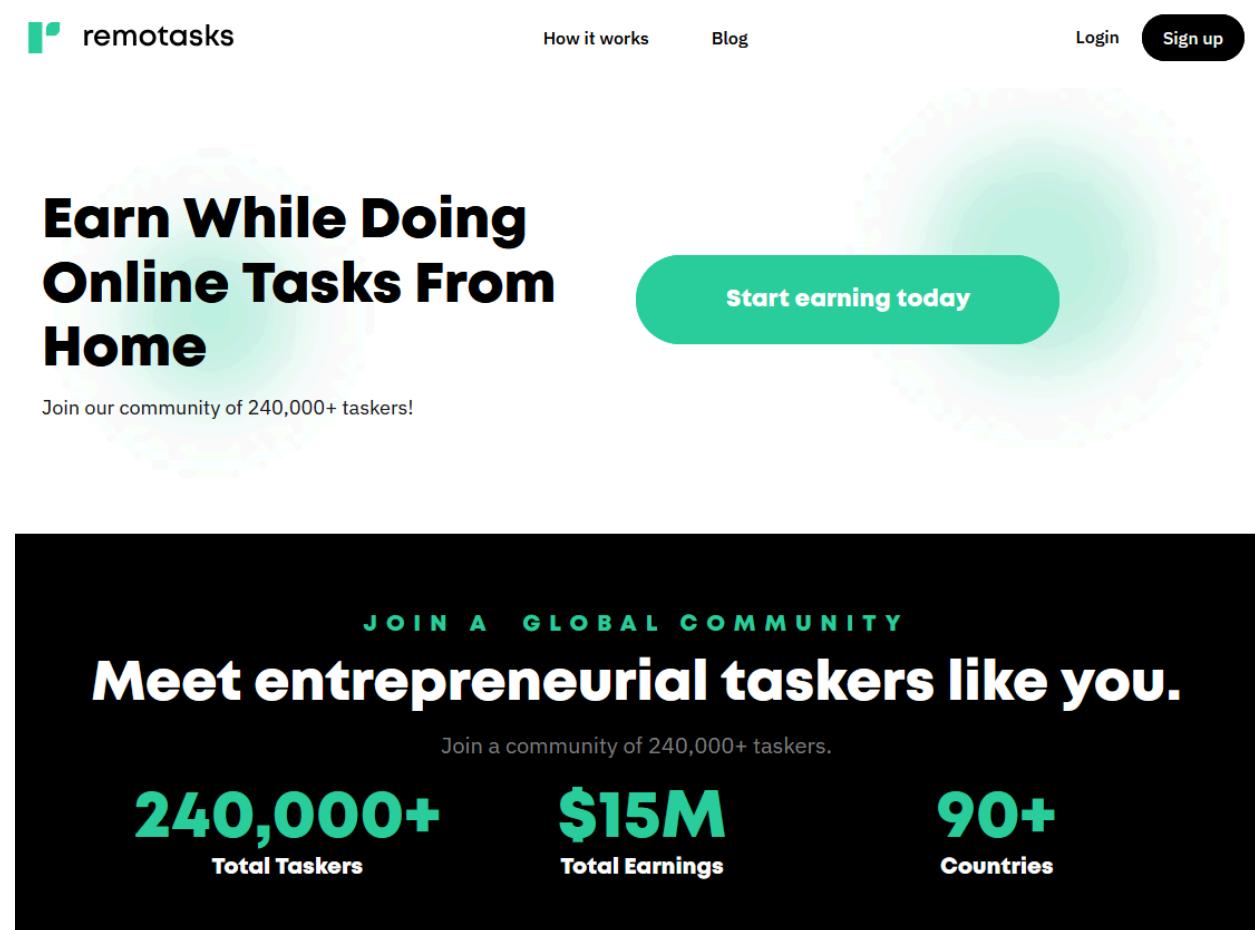


Figure 4: Remote Tasks Website Main Page

[ Image Available at: <https://www.remotasks.com/en> ]

Platform	Remotasks
Primary Function	Data labeling and annotation
Task Types	Image, video, LIDAR, NLP
Payment Model	Pay-per-task, training-based
QA Review	Yes, strict QA
Local Payment System	Limited
Local Language Support	Limited
Simple UI	Moderate
Automated Assignment	Yes (after training)
Mobile Friendly	Yes (web-based)
Offline Functionality	No

## 2.2.5. KoboCollect

KoboCollect is a data collection platform that is built for research purposes from the Harvard Humanitarian Initiative (HHI). The application was developed based on the open-source Open Data Kit (ODK) Collect app, and it was designed to make data collection in challenging environments. The application is popular for its mobile compatibility and offline functionality. Workers or volunteers can do tasks in offline mode, and data will be synced and submitted to the server when the internet connection is restored. With its mobile-friendly interface and offline capabilities, KoboCollect is a good choice for on-site data collection projects in challenging environments with limited internet access.

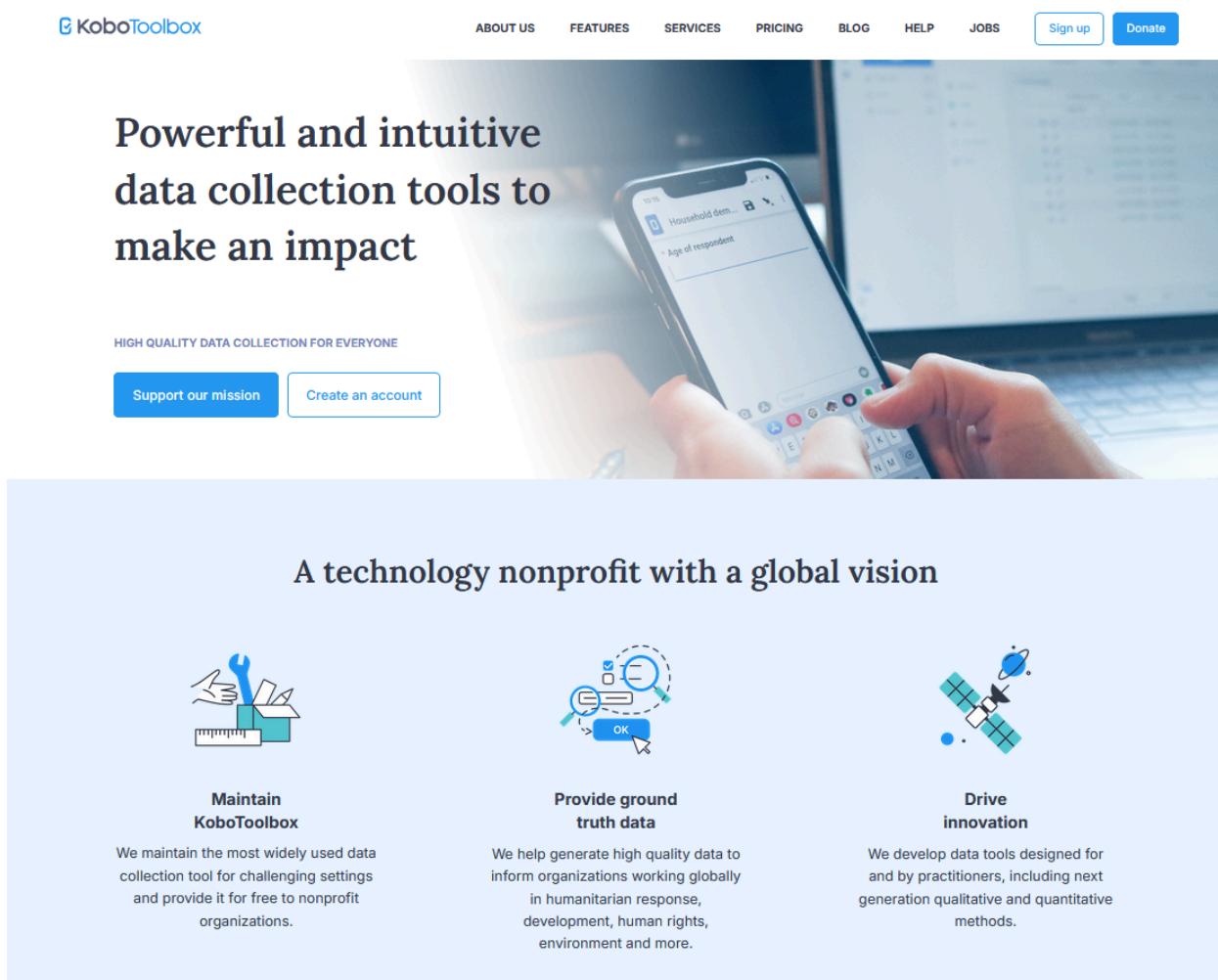


Figure 5: KoboToolbox Website Main Page

[ Image Available at: <https://www.kobotoolbox.org/> ]

Platform	KoboCollect
Primary Function	Data collection for research
Task Types	Field surveys, offline data collection
Payment Model	Open-source
QA Review	No built-in QA
Local Payment System	No
Local Language Support	Yes (customizable)
Simple UI	Yes
Automated Assignment	No
Mobile Friendly	Yes (Android app)
Offline Functionality	Yes

#### 2.2.6. Key inside of Similar systems

Clickworker, MTurk, Toloka, and Remotasks are all task-based crowdsourcing platforms, except KoboCollect, which is a tool for structured data collection instead of a pay-per-task platform. Tasks on Toloka and Remotasks are mostly related to AI training data, while others are a mix of general-purpose tasks. Remotasks is a standout for its LiDAR and image annotation, while Toloka has strong integration with Yandex's AI ecosystem as well as MTurk with the AWS ecosystem.

The QA review process on Clickworker and MTurk completely relies on requesters, and those platforms offer just basic QA features, not automated algorithms. Requesters can check and review the submitted tasks with multiple layers to ensure data quality. Toloka and Remotasks use both built-in automated algorithms and manual reviews as Clickworker and MTurk offer. On the KoboCollect application, QA can be achieved through form-based constraints rather than a QA system.

Most of those platforms do not provide a local payment system except for Toloka in some regions of Russia. Automated Assignment on Clickworker is based on user prefilled profiles, while others use performance-driven assignment. However, there is no direct automated assignment on KoboCollect. All of those platforms provide simple user interfaces for text-based tasks, while image and video annotation tasks need a steeper learning curve to use annotation tools. Among those platforms, KoboCollect, Clickworker, and Toloka are designed for mobile application compatibility, while only KoboCollect provides offline data collection.

### 2.3. Challenges in Data Collection Systems for Local Languages

Many global platforms support widely spoken languages as system languages but do not include local languages in Myanmar. This will create a gap between the freelancers and requesters since the UI elements and task instructions may not be fully translated or culturally adapted, leading to lower data quality.

While tools for image and LiDAR annotation are complex, text-only tools seem to be simple enough for freelancers in Myanmar. However, the registration system and in-app exploration are still complex on some platforms, such as MTruk and Clickworkers.

One of the main challenges of using existing platforms is that none of them provide local payment systems or currencies. For freelancers in Myanmar, this makes it difficult to receive payments and manage task prices.

For a text data collection platform targeting underrepresented languages, support for local languages with the simple user interface is essential to improve accessibility and usability for freelancers from Myanmar. A customized quality assurance mechanism can be a better approach to ensuring data quality, especially for guided back-translation.

Making payment in the native currency using familiar banks can make a more reliable data collection process and build trust in the platform. A localized data collection platform that fills these gaps can not only improve usability and data reliability but also provide a better solution for text data collection in Myanmar.

## 2.4. Legal, Ethical, Social, and Professional Considerations in Data Collection

It is important to address legal, ethical, social, and professional considerations in developing an online platform for text data collection. These help build trust between freelancers and job posters and also maintain data integrity, and show respect for the diverse language culture of Myanmar.

### 2.4.1. Legal

All the materials used for the translation are text data from purchased books and web-scraped content, and must comply with Myanmar Copyright Law. The Myanmar Copyright Act of 1914 has been repealed and replaced by the Copyright Law (Law No. 15, 2019), which came into effect on October 31, 2023 (Pyidaungsu Hluttaw, 2019). Those materials will make sure to have fully copyrighted and proper permissions are obtained. Additionally, there will also be a code of conduct between the freelancer and the platform. Freelancers who are willing to contribute to data collection have to agree that the provided materials for translation will not be used for other purposes. Regarding the payment system of the platform, it will comply with Myanmar labor laws, ensuring all the freelancers are paid transparently, fairly, and on time (Pyidaungsu Hluttaw, 2016).

#### **2.4.2. Ethical**

All the collected text data (including web scraping and purchased text) will be used ethically, respecting the original context and probity of the source materials. Freelancers will be provided information about how the collected data will be used following ethical standards in crowdsourcing. Payment will be processed on time following labor guidelines, and task prices will be set fairly and transparently. If any issue occurs, Freelancers have the right to complain and report about the issues, and the admin team will make a quick and fair resolution. The submitted task will be done without any bias for the freelancer. These ethical standards will ensure the maintenance of the high quality of the data and the reliability of the platform.

#### **2.4.3. Social**

The aim of the platform is not only data collection for underrepresented languages but also to provide economic opportunities to native speakers of minority languages. The platform will be accessible nationwide, and the user interface, including system languages and task instructions, will be localized. With a simple and clean user interface, native speakers from various technology backgrounds will be able to contribute to the data collection process easily and effectively. Another important factor is designing the quality assurance process. Submitted tasks will be reviewed not only by focusing on the meaning of the words but also by ensuring the contextual accuracy and cultural respect. Feedback from the freelancers will be prioritized for system development to guarantee a positive social impact of the platform.

#### **2.4.4. Professional**

The entire development process of the platform and data collected through the platform will be well documented following professional standards. Professional linguists, NLP researchers, and software developers will adopt best practices in software development and research ethics. This will help to support transparency of the system, and future development can be done easily. To minimize the system downtime, the system will be monitored after real-life deployment. When there are any changes in the system, the freelancers and QA team will be notified on time, and instructions for the changes will be provided if necessary. This approach will help maintain the high quality of the platform and create a productive working environment.

# **Chapter - 3**

# **Requirement Analysis**

### 3. Requirement Analysis

#### 3.1. Target Users Identification

**Freelance Translators:** Freelancers who are the main users of the platform will perform translation and creative writing tasks. They will enter written text in the provided user interface and submit the data to the system. To contribute to text data collection through the platform, they must be able to use the website to register, browse tasks, submit tasks, and request payment. They also needed to be fluent not only in the target minority language but also in the source language, especially the Burmese language, to do translation. Good typing skills in the target minority language are essential, and they need to be comfortable with the built-in text editors of the platform. While freelancers may reside in urban areas, it is important to encourage participation from those in rural regions to preserve linguistic and cultural authenticity.

**Quality assurance team:** The QA team will review the tasks submitted by the freelancers to ensure the accuracy and cultural appropriateness of the collected data. Tasks will be reviewed and accepted according to the translation quality standards and will be ensured they are free of biases. QA team members must have an in-depth understanding of the syntax of languages for both the majority language and one or more specialized minority languages in Myanmar. Training for using the platform and the reviewing process will be provided for them. The QA team will be directly hired by the company through a structured interview process.

**Platform Administrators:** Administrators manage the overall data collection process to run the process smoothly. They are in direct contact with the NLP researcher team and platform development team. Their primary jobs are to post tasks for guided back translation as the NLP team requested, manage and assign the QA team, and process payment to the freelancer. They are also responsible for resolving the issues reported by freelancers. They are the only user role of the platform that has access to the dataset and is required to maintain the data integrity with security standards.

## 3.2. Functional Requirements

The proposed platform will have the following requirements to meet the needs of all user roles and to ensure a smooth and efficient workflow.

**Freelancer Registration:** The platform will allow freelancers to register with personal details, including name, email, and age. The platform will check for an existing user with the provided email, and if there is no existing user, the platform will send a verification code to the email to complete the email verification process. Later, freelancers can log in to the platform with their credentials, email, and password.

**Managing the QA team:** The platform will enable the administrators to manage the QA team, including CRUD operations (creating new members, listing all the members, resetting passwords of members, and removing members from the team) for the QA team members.

**Job posting and task creation:** The platform will provide an interface for administrators to post jobs. Jobs are a collection of tasks that include metadata for each task, such as detailed instructions, language IDs, prices per task, maximum time given for translating tasks, and additional notes. Each task will be separately created with the linked job ID in the backend, and freelancers can do task translation through the task browsing interface of the platform.

**Assessment tasks:** Once the freelancers have registered on the platform, they will need to answer assessment tasks to review their language proficiency. The platform will provide an assessment module where freelancers can complete assessment tasks. The types of assessment tasks will be based on the language pair (two languages for translation tasks and one language for creative writing). The assessment tasks can be filtered, and freelancers can choose their preferred language pair.

**Tasks browsing:** The system will allow freelancers to browse available tasks through a filterable interface after they have completed assessment tasks for language pairs required for task translation. If the user does not complete the assessment tasks, the platform will only show the number of open and available tasks for this language pair.

**Task translation:** The platform will provide a dedicated interface for freelancers to submit translation tasks. Freelancers can type directly in the provided text editor, where instructions, task price, maximum time, and additional notes will be displayed.

**Reviewing tasks:** The platform will provide a QA reviewing mechanism with a dedicated interface. A QA member with professional language skills will review the tasks submitted by the freelancers, and the task type will include both assessment tasks and data collection tasks. Assessment task reviews determine whether freelancers are qualified to translate for a given language pair. The data collection or submitted tasks from the user will be reviewed through the defined standard, and if the translation does not meet the standard, the QA member can reject the task.

**Requesting Payment:** The freelancer can request payment for their completed tasks through the payment request form of the platform. They can select their preferred payment type and set how much money they want to withdraw from their account. The system will automatically notify the request to the admin to ensure the payment is processed on time.

**Reporting Issues:** If there are any issues in the platform, the freelancer can report the issues to the admin through a dedicated form. The issue types will include payment delay, wrong source languages, not having enough time for translation, and others. These reports will be sent to the admin with the detailed information to take action for resolution.

**Resolving Issues and Processing Payment:** The platform will also provide a dedicated interface for the admin to process the freelancer-requested payment and to resolve the reported issues.

### 3.3. Use Case Diagrams

#### 3.3.1. Use Case Diagram for Admin

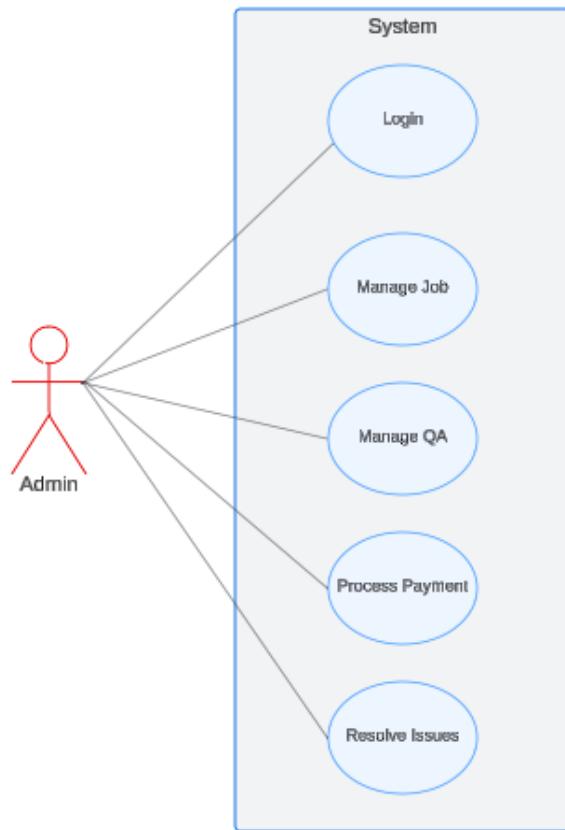
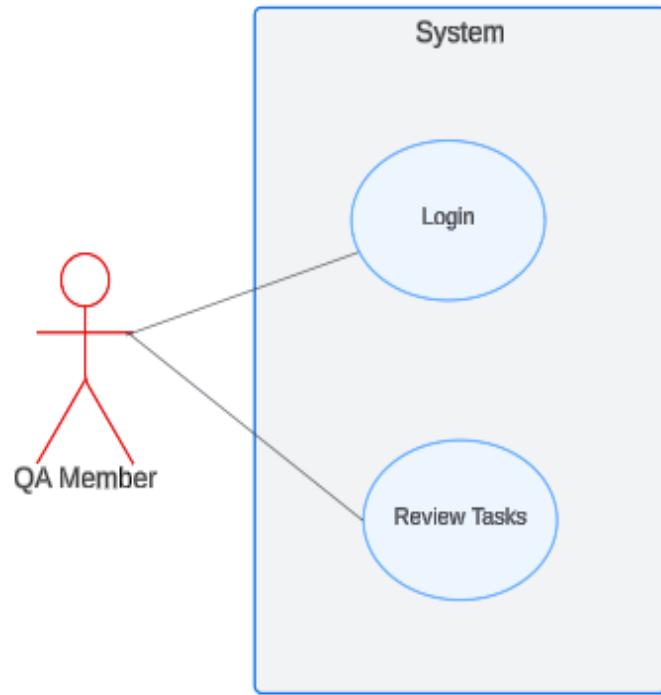


Figure 6: Admin Use Case Diagram

### 3.3.2. Use Case Diagram for QA member



*Figure 7: QA Member Use Case Diagram*

### 3.3.3. Use Case Diagram for Freelancer

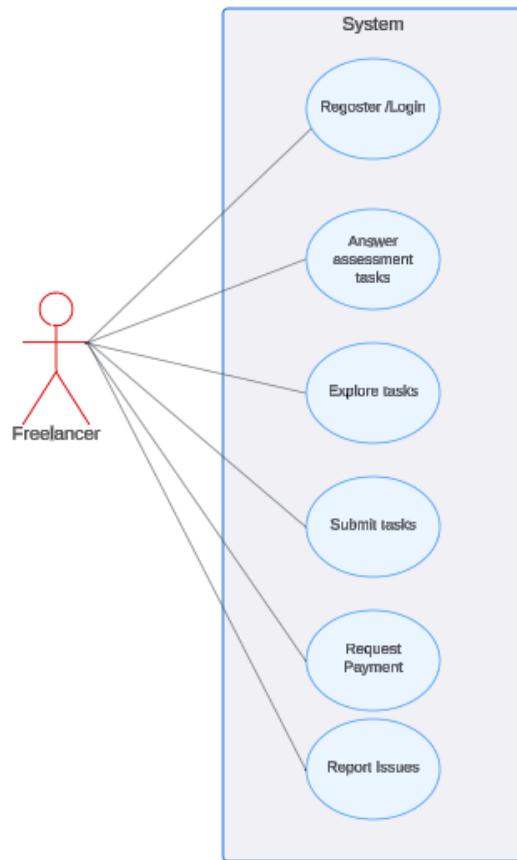


Figure 8: Freelancer Use Case Diagram

## **3.4. Non-functional Requirements**

### **3.4.1. Performance**

The system shall provide a responsive user interface and fast interactions, such as loading tasks and submitting tasks. Lazy loading should be implemented in browsing tasks to improve user experience even on low-bandwidth networks. The system must be able to handle a growing number of users and expanding task volumes. Flexible system components should be used to be able to maintain the system as needed to ensure smooth operation. The system shall be highly available; downtime must be minimized to ensure all users have access to the service continuously. To achieve this, the system should be implemented in a testing server fast and only implemented in the production server when there are no issues in the testing server. There will also be necessary backup strategies for automatic recovery if some software failure occurs.

### **3.4.2. Security**

The system will be secured using an authentication mechanism. All the sensitive data will be protected, and permission will be controlled with role-based access. Admin will have to fully access the QA team member data, including CRUD operation. Freelancer data will remain confidential, and the admin will only have reading permission. Payment will be processed through trusted channels to ensure safety and transparency. Regular reviews will be done to keep the system compliant with security standards.

### **3.4.3. Usability**

System usability is undoubtedly the prominent part of the user experience than any other features. The interface, which facilitates the median between the user and the system, will be designed to be as simple as possible to navigate the system easily for all users and complete tasks smoothly. The platform will be developed using user-friendly UI elements, in turn, pages will be linked appropriately to reduce user confusion during navigation through the platform. Users can utilize local language as system languages, and there will also be text instruction and feedback to the user to improve usability. Following that, the user interface design will comply with design principles to ensure adaptability across various devices and screen sizes.

# **Chapter - 4**

## **Methodology**

## 4. Methodology

### 4.1. Agile Development Methodology Overview

Agile is a project development approach that focuses on iterative development and customer collaboration. In Agile methodology, the project is divided into small goals, sprints, and the development process iterates through these sprints until the project meets the final goal. Each sprint cycle includes development, testing, and reviewing processes. During development cycles, discussions with customers or users are made frequently to gather feedback, which is then integrated into the next iteration. This helps ensure that the project meets customers' needs. Its iterative approach allows evolving requirements to be developed in each iteration. Agile methodology is popular in small and medium-sized projects due to its prioritized features that are most valuable to the customers. The study "Implementing agile management practices in the era of digital transformation" by Joseph Chukwunweike and Opeyemi Aro mentioned how agile methodology supports organizational needs with iterative development and user collaboration (Chukwunweike & Aro, 2024). Furthermore, in a study named "A Comparative Study of Agile and Waterfall Software Development Methodologies" by Shamsulhuda Khan and Shubhangi Sudhir Mahadik, it was presented how agile methodologies outperform traditional waterfall methods where requirements are evolving and customer feedback is important (Khan & Mahadik, 2022). By analyzing those insights, our medium-sized project, with its focus on user requirements and the potential for evolving needs, is well-suited to Agile methodology.

### 4.2. Scrum Framework for Agile Project Management

Scrum is one of the most popular implementation frameworks of Agile methodology. Scrum, based on empirical process control theory, is an iterative and incremental project management methodology to control risk and optimize the predictability of a project (Lei et al., 2017). The key components of the Scrum process include

**Product Backlog:** The product backlog is a list of requirements that is maintained by the product owner, and requirements can evolve based on user feedback.

**Sprints:** Sprints are time-boxed development cycles that are usually two to four weeks long.

**Sprint Backlog:** The sprint backlog is a set of selected product backlog items and a plan for delivering the product increment, which are objectives of the sprint.

**Daily Scrum Meeting:** Scrum meetings are short daily meetings (usually 15 minutes) where team members share the progress of the sprint.

**Potentially Shippable Product Increment:** The shippable product is a product increment that is delivered at the end of each sprint. During the sprint review, users test the product increment and provide feedback.

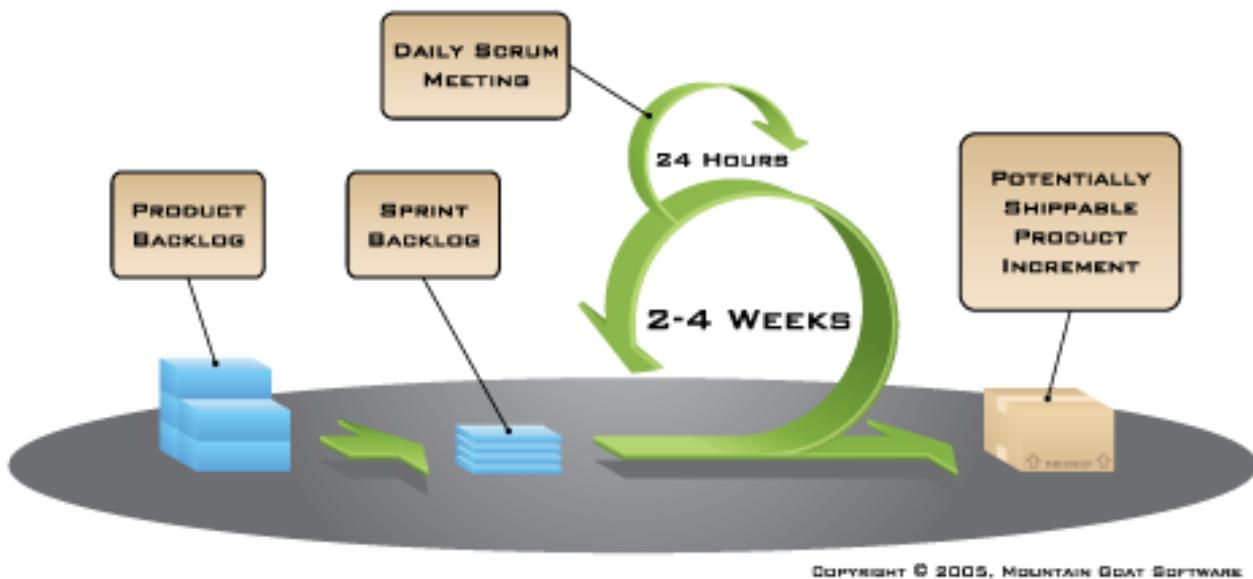


Figure 9 : Scrum Methodology Sprint Planning

[Image available at <https://www.mountaingoatsoftware.com/agile/scrum>]

A scrum team is a self-organized team and consists of a scrum master, a product owner, and developers. Team members know how to meet the goal and have controlled the project without relying on directions from people outside the team (Scrum.org, n.d.).

- Project owner manages and maintains the product backlog and is responsible for evolving the requirements of the backlog to increase the project value.
- The scrum master is responsible for planning the sprint and instructing the team on developing the product backlog items.

- Development team is a cross-functional team which is responsible for developing and delivering product increments at the end of each sprint. The developer team usually consists of 5 to 9 people who are programmers, testers, user experience designers, etc.

### 4.3. Sprint Planning in Scrum Framework

Sprint planning in the scrum framework are meetings that occur at the beginning of each sprint where all scrum team members participate. The purpose of the sprint planning is to determine what to complete and allocate an amount of time for the current sprint. Clear sprint goals and sprint backlogs are defined during sprint planning by carefully reviewing the product backlog items and prioritizing them. After setting the sprint goal, the tasks are assigned to the team members, and the sprint starts.

### 4.4. Why Agile with Scrum Framework would be an appropriate method

Using Agile with the Scrum framework is well-suited for the project development because of the following reasons:

**Minimal Project Size:** As the size of the project is small with a manageable scope, lightweight methodology is more suitable than traditional methodology. In addition, Scrum allows to identify the most important goal ,and consequently provide an increment of the product.

**Simple Complexity:** The project has a well-structured and clear framework. Accordingly, iterative cycles of scrum give assistance to split it into smaller parts, which make it easier to process and reduce the risk of overlooked details.

**Short Timeframe:** The project requires only a short duration to manage. In relation to processing time,the time-boxing of Scrum (2 to 4-week sprints) supports faster development with a focus on the most important requirement using MoSCoW prioritization. Each sprint offers incremental outcomes, and necessary adjustments can

be decided based on feedback. This helps to ensure that the must-have functionalities work properly even before the project development is completed.

**Stakeholder Involvement:** The projects involve the participation of various stakeholders, such as the admin team, QA team and users. Involving stakeholder help to be able to review the progress for each development cycle of sprint outcomes and can provide valuable feedback. This loop makes certain that the project is aligned with the user's requirements.

# **Chapter - 5**

# **System Design**

# 5. System Design

## 5.1. System Architecture

In systems engineering and IT, the architecture of an object describes the high-level structural composition of the system, including components and their interactions (Badenko et al., 2024). The following diagrams represent the system design in different views.

### 5.1.1. Class Diagram

The class diagram shows the system's classes along with their attributes and methods and the relationship among the objects created with those classes. This provides the data models for understanding the business logic of the platform.

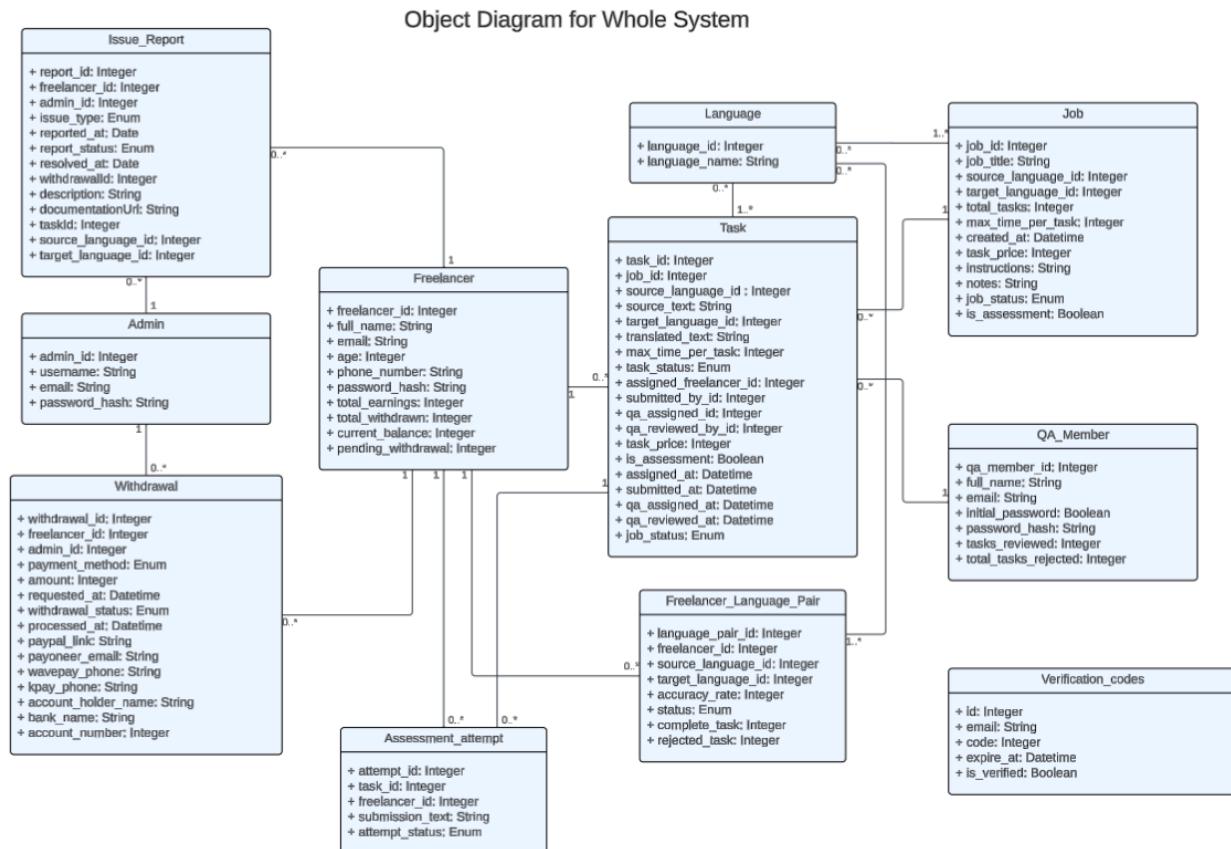


Figure 10: Object Diagram of the Whole System

## 5.1.2. Object Diagram

The object diagram illustrates examples of objects in the system with their relationships with other objects at a moment in time.

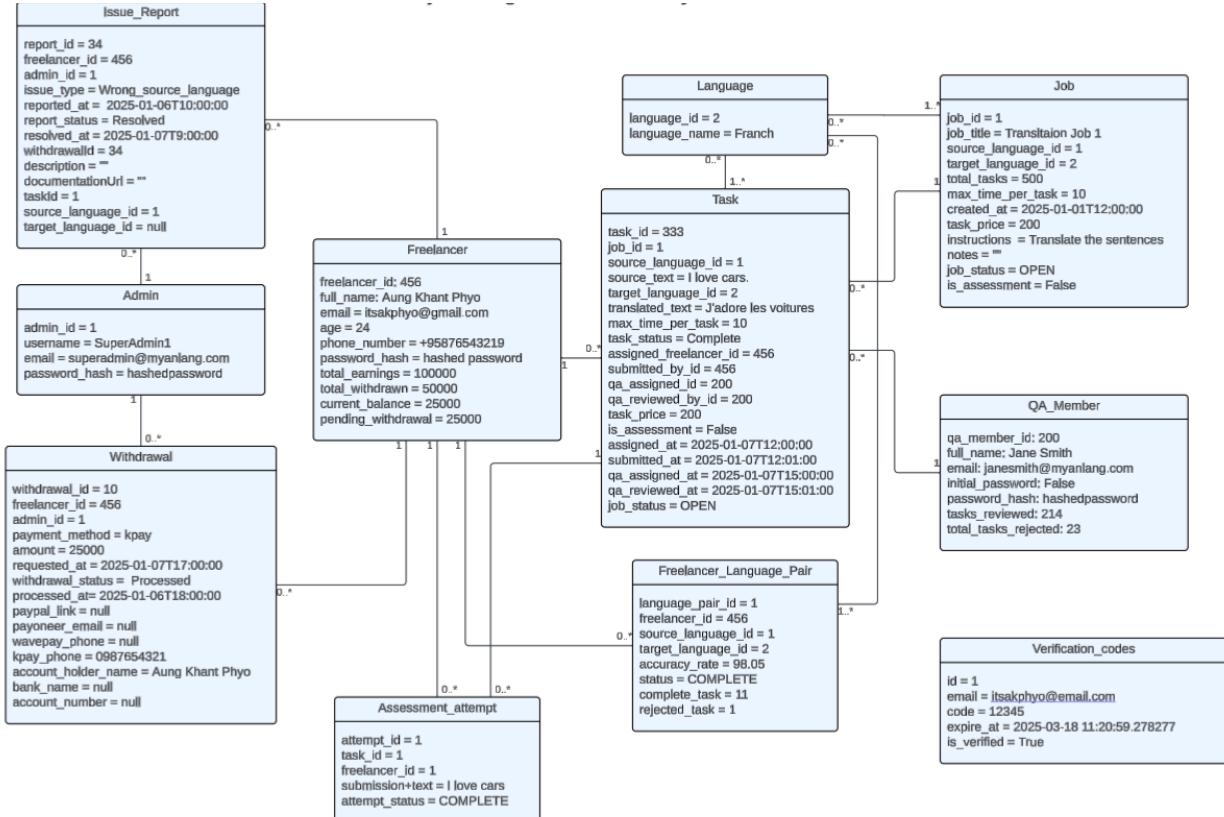


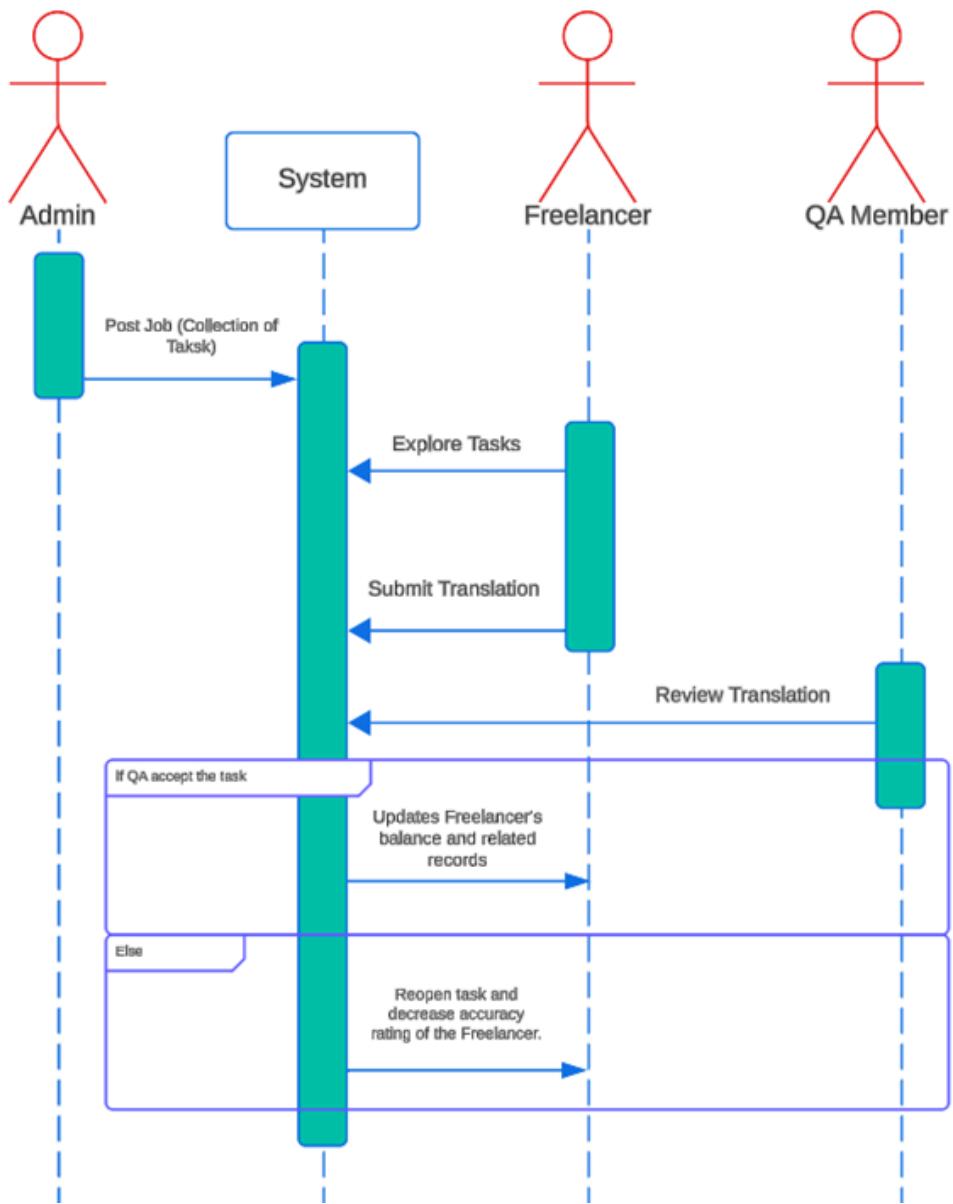
Figure 11: Class Diagram of the Whole System

## 5.1.3. Sequence Diagram

### 5.1.3.1. Sequence diagram for task workflow

This sequence diagram represents the workflow of a task within the system, and three main user roles (all user roles) are involved in this workflow. The admin starts the process by posting a job (which is a collection of tasks with metadata) to the system. A freelancer explores the task, selects a task, completes the translation, and submits the task to the system. Once a freelancer submits a task, a QA member reviews and evaluates the submitted task. Based on the quality of the translation, the QA member will

approve or reject the task, and the system will update the related record of the freelancer, including balance and accuracy.



*Figure 12: Sequence diagram for task workflow*

### 5.1.3.2. Sequence diagram for balance withdrawal

This sequence diagram represents the workflow of reporting issues within the system. Once an issue has been identified in the system, the freelancer submits a report to the system, and the system notifies the admin. After the admin takes proper action to resolve the issue, the system updates the issue status and notifies the freelancer.

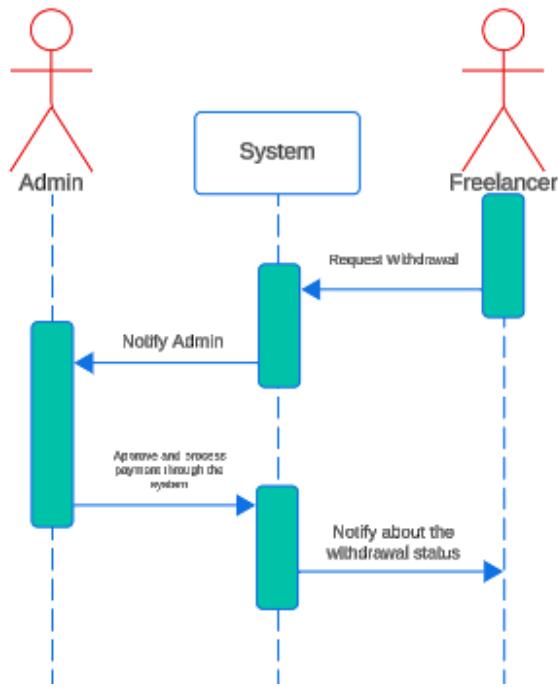


Figure 13: Sequence diagram for balance withdrawal

### 5.1.3.3. Sequence diagram for issue report

This sequence diagram represents the workflow of a freelancer withdrawing a balance as payment. Freelancers can request payment by submitting a request withdrawal form to the system, and the system notifies the admin, and the admin can process payment. Once the payment is processed, the system notifies the freelancer about the withdrawal status.

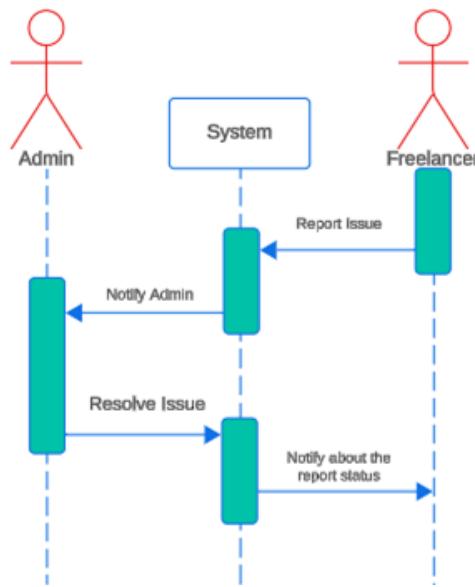


Figure 14: Sequence diagram for issue report

### 5.1.4. Activity Diagram

#### 5.1.4.1. Activity diagram for user registration and task exploration

This activity diagram illustrates how the user registration and task exploration workflow works within the system. Once a freelancer navigates to the website, there will be options for logging in and registering a new account. Old freelancers can log in to the website with their credentials, and new freelancers can register with emails. Once authentication is done, a dashboard will be displayed to the freelancer with the available tasks count for each language pair. The freelancer can see the task by selecting the language pairs that he or she is fluent in. If the freelancer does not have an answer to the assessment task, the data collection task (from the job) will not be shown to the

freelancer, and only assessment tasks for this language pair will be shown where he or she can answer and submit assessment tasks. A QA member will review the assessment task, and if passed, this freelancer will have approval for this language pair and can contribute to the data collection. If the assessment tasks do not pass QA, the system will notify the freelancer.

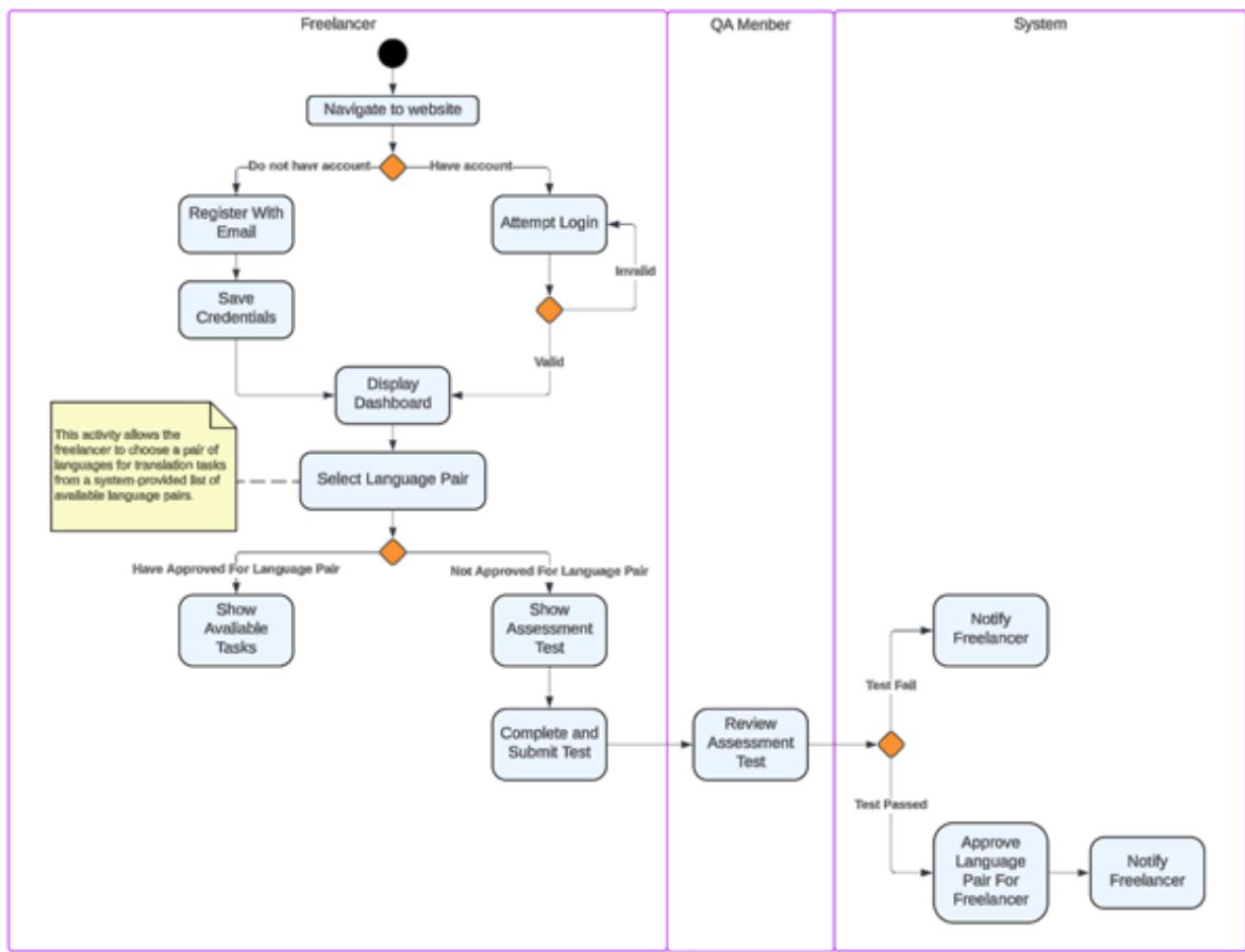


Figure 15: Activity diagram for user registration and task exploration

#### 5.1.4.2. Activity diagram for job posting and task translation

This activity diagram represents the job posting and task translation workflow with detailed interactions. Admin log in to the system and post a new job, which is a collection of tasks. The system divides the jobs into tasks and shows the tasks to the freelancers who have been approved for translation for this language pair. The freelancer logs in to the system, navigates to the task page, selects the language pair, and can see those available tasks. Once a freelancer opens an available task, the system automatically assigns the task to the freelancer and sets the task status to "assigned" to prevent two freelancers from translating the same task. If the freelancer does not submit the task within the time limit, the system will reopen the task by removing the assigned freelancer ID from the task with a background job. If the freelancer submits the task within the time limit, the system will change the task status to "under review" and show this submitted task to the QA member for an accuracy review. A QA member logs in to the system and reviews the submitted task with defined standards, and if the submission does not meet the standard, the QA member will reject the submission, and the system will reopen the task and update the accuracy rating of the freelancer. If the submission meets the standard, the QA member will accept the submission, and the system will make the task status as "complete" and will do the required update for the freelancer, such as adding a balance and updating the accuracy rating.

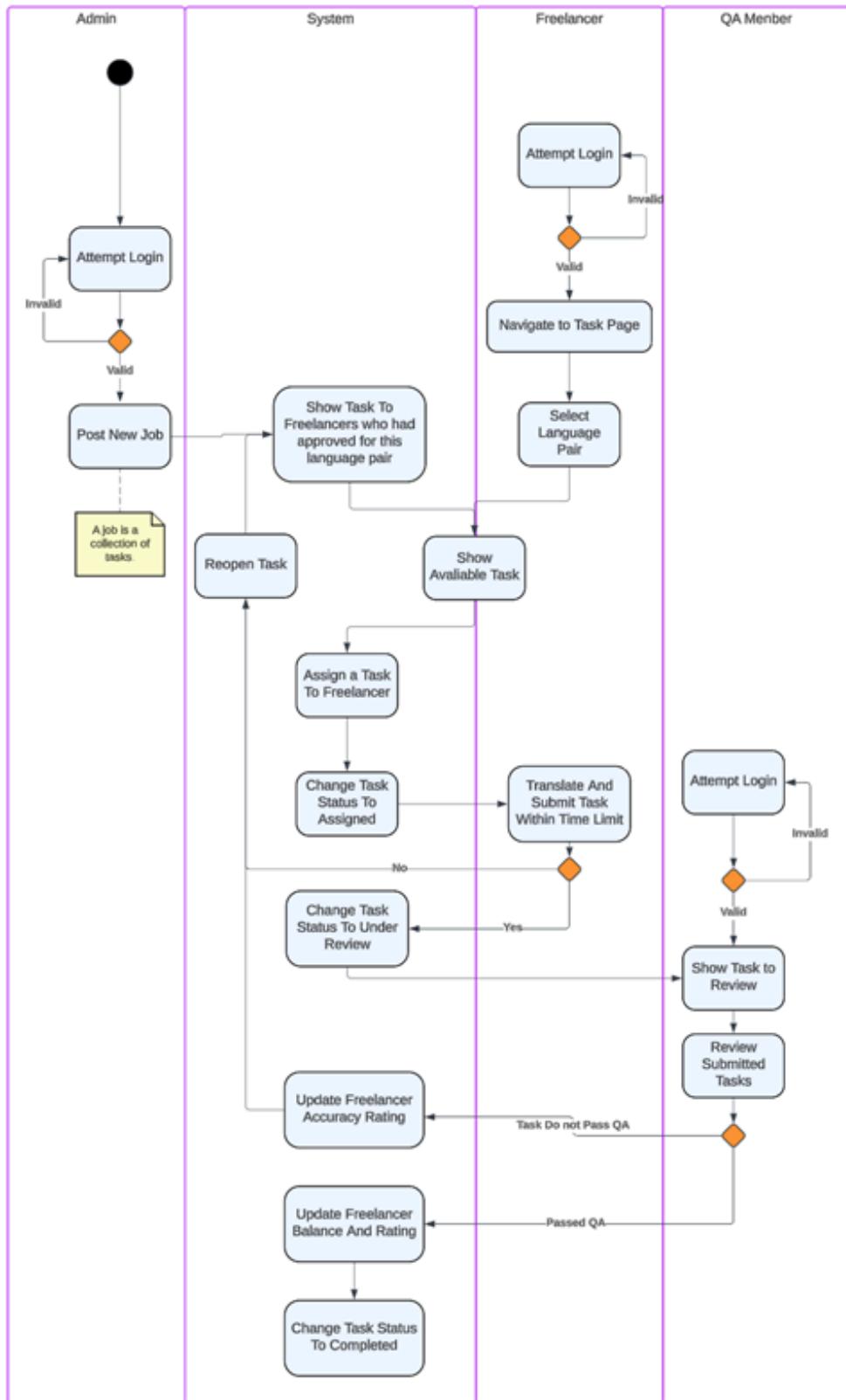


Figure 16: Activity diagram for job posting and task translation

#### 5.1.4.3. Activity diagram for issue reporting

This activity diagram represents the workflow or reporting issues, which include a freelancer and an admin as actors. There will be a dedicated issue report page, and freelancers can navigate to it after login. Freelancers can select the issue types, input information about the issues, and attach docs as evidence. Once the issue report form is submitted, the system will automatically notify the admin to take proper action for resolving.

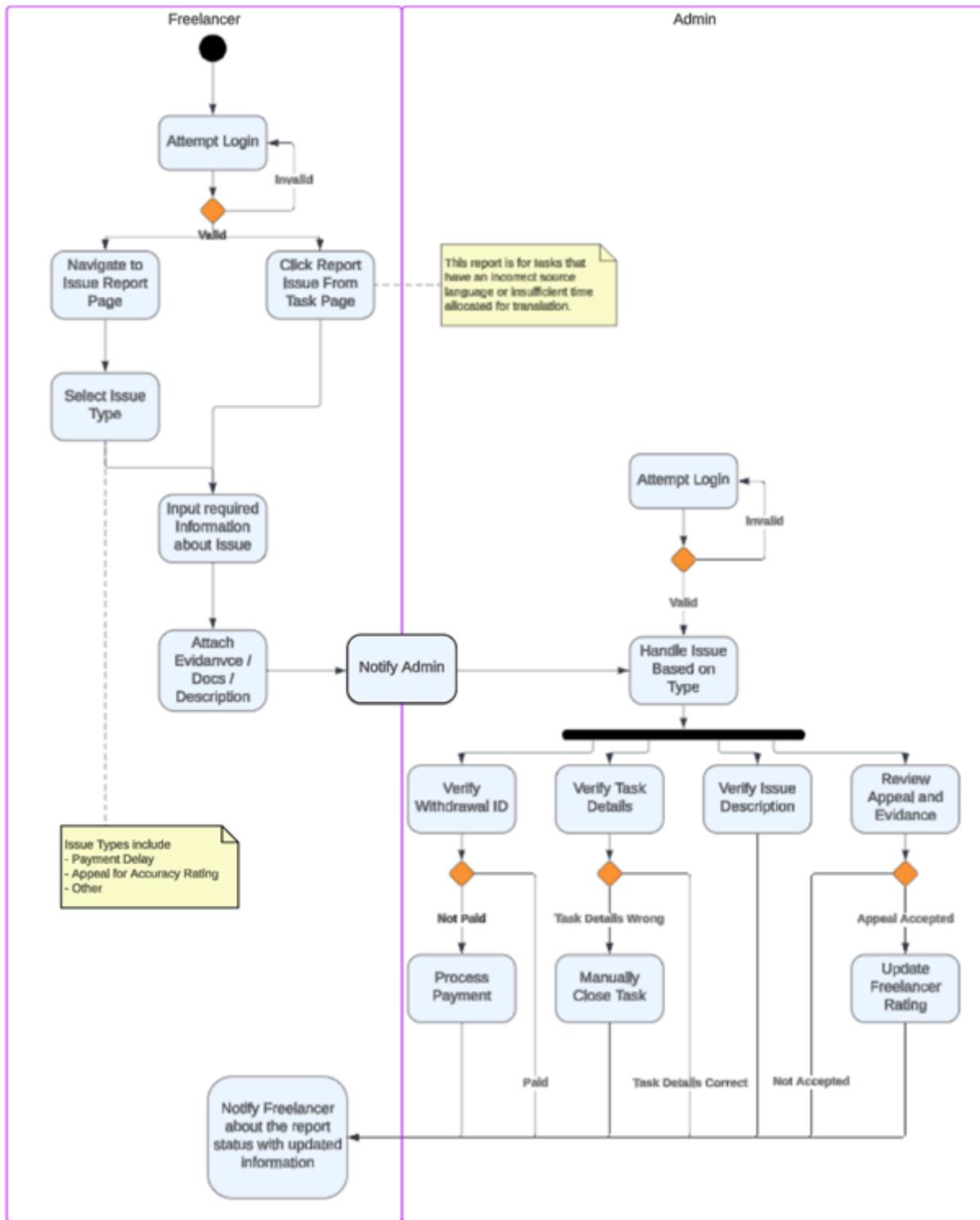


Figure 17: Activity diagram for issue reporting

### 5.1.5. Component Diagram

This component diagram represents the architecture of the entire system with interactions between each service and database. Each service operates independently, making it scalable and easier to maintain, and newer services can be added without affecting the existing system functionality.

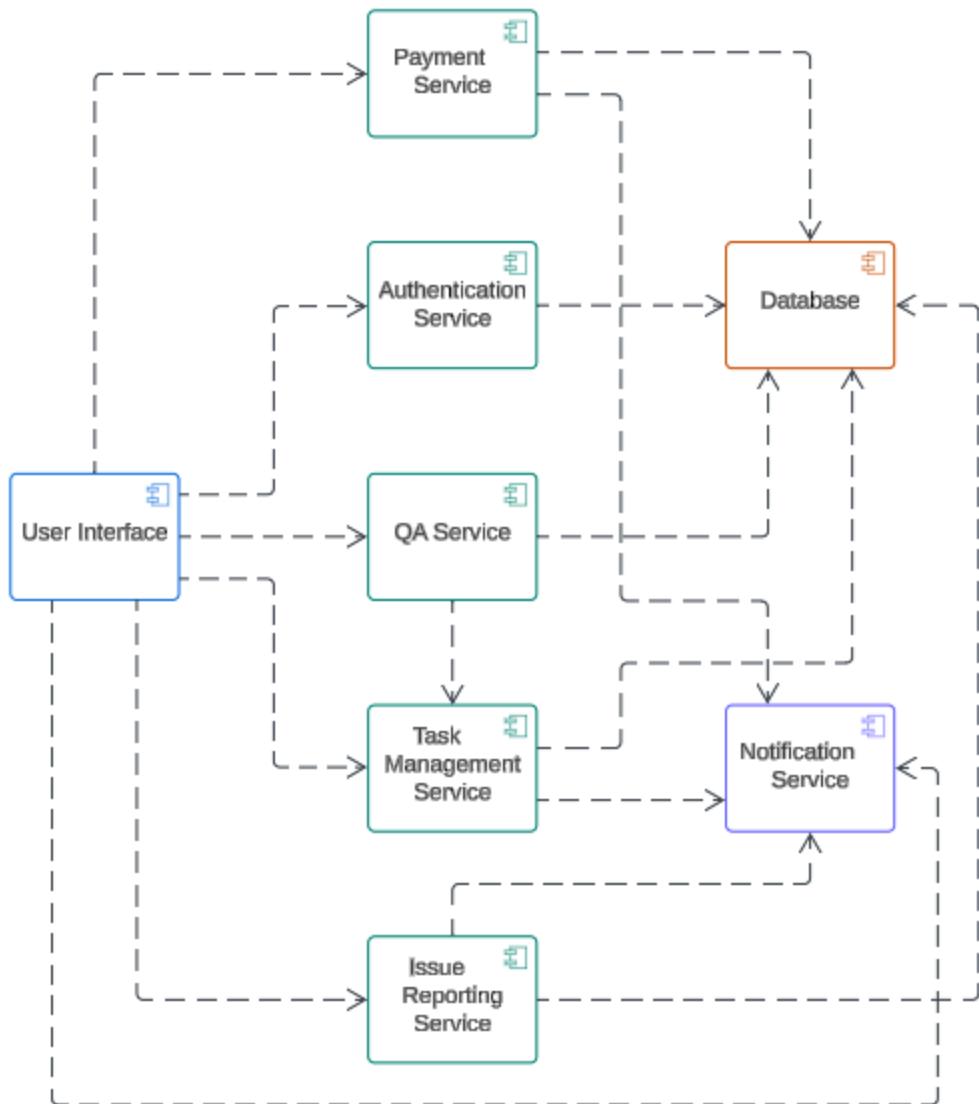


Figure 18: Component Diagram

## 5.2. Database Design

### 5.2.1. Entity Relationship Diagram

The following ERD represents the database structure of the system along with their attributes and relationships among them.

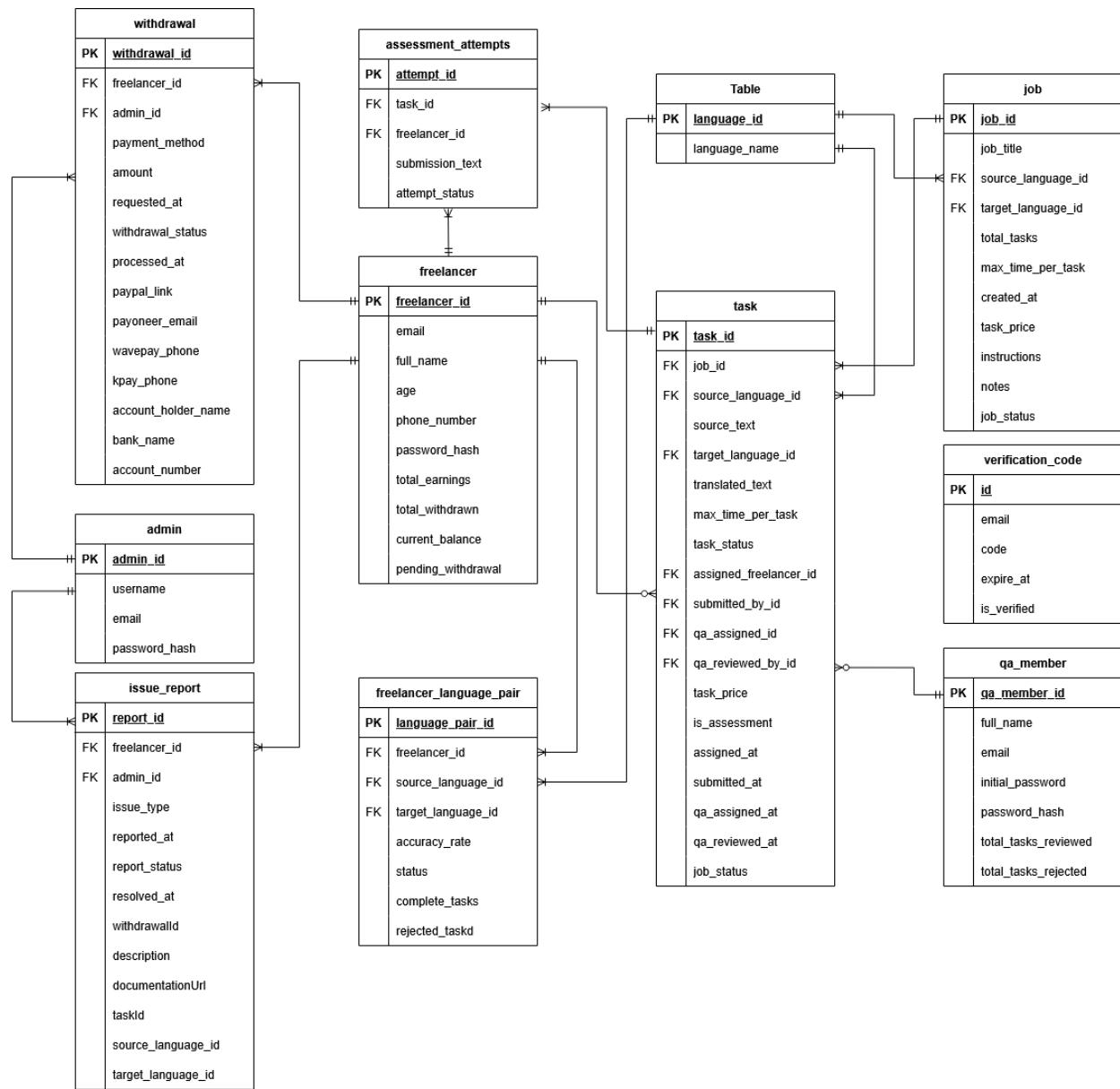


Figure 19: Entity Relationship Diagram

### 5.2.2. Explanation of Database Structure

The database is designed with multiple tables to efficiently manage the collected text data. The following are detailed explanations of each table.

- The **admin** table stores admin details such as username, email, and hashed password.

Relationships:

- One admin can resolve multiple issue reports (admin\_id as FK in issue\_report).
- One admin can make processes for multiple withdrawals (admin\_id as FK in withdrawal).

- The **freelancer** table stores freelancer details such as email, full name, age, phone number, and balance data (total earnings, current balance, pending withdrawal, etc.).

Relationships:

- One freelancer can report multiple issues (freelancer\_id as FK in issue\_report).
- One freelancer can request multiple withdrawals (freelancer\_id as FK in withdrawal).
- One freelancer can submit multiple tasks (submitted\_by\_id in task).
- One freelancer can have multiple language pairs (freelancer\_id as FK in freelancer\_language\_pair).

- The **job** table stores all metadata of a job. A job is a collection of tasks, and each job includes metadata like title, source and target languages, task count, deadlines, task price, maximum time for each task, and instructions.

Relationships:

- One job can have multiple tasks (job\_id as FK in task).
- Each job is linked to a source language and a target language (source\_language\_id and target\_language\_id as FK in language).

- The **task** table stores details of an individual task. It contains task details such as source text, translated text, task status, assigned freelancer, and QA review status.

Relationships:

- One job can have multiple tasks (job\_id as FK in task).
  - One task contains both a source language and a target language (source\_language\_id and target\_language\_id as FK in language).
  - One task can be assigned to one freelancer (assigned\_freelancer\_id as FK in task).
  - One task can be reviewed by one QA member (qa\_reviewed\_by\_id as FK in task).
- The **QA member** table stores details of a quality assurance member who reviews tasks submitted by freelancers.

Relationships:

- One QA member can review multiple tasks (qa\_reviewed\_by\_id in task).
- The **language** table stores available languages on the platform.

Relationships:

- One language can be linked to multiple jobs as a source or target language.
  - One language can be linked to multiple tasks as a source or target language.
  - One language can be linked to multiple freelancer\_language\_pair records.
- The **Freelancer Language Pair** Table stores the language proficiency data of each freelancer.

Relationships:

- One freelancer can have multiple language pairs (freelancer\_id as FK).
  - Each language pair must have a source language and a target language (source\_language\_id and target\_language\_id as FK).
- The **issue report** table stores issues reported by freelancers about tasks or payments.

Relationships:

- One freelancer can report multiple issues (freelancer\_id as FK).
  - Each report is reviewed by an admin (admin\_id as FK).

- The **withdrawal** table stores withdrawals requested by freelancers, including amount, payment method, status, and approval.

Relationships:

- One freelancer can request multiple withdrawals (freelancer\_id as FK).
- Each request is processed by an admin (admin\_id as FK).
- The **assessment attempt** table stores submission text from freelancers, which is for assessment tasks. They need to be stored separately instead of being stored in the same task table because assessment tasks are just questions for assessing user language proficiency, and one or more users can answer these tasks.

Relationships:

- One freelancer can have multiple assessment attempt submissions (freelancer\_id as FK).
- One assessment task can have multiple attempts since one or more can answer the same assessment task.
- The **verification code** table stores verification codes that are sent during email validation of user registration and resetting user passwords. This table does not have a relationship with other tables, and this just stores temporary data.

The database design is intentionally denormalized to simplify queries and improve performance. For instance, the financial data of freelancers is stored with precalculated fields such as "total\_earnings," "total\_withdrawn," and "current\_balance" instead of being calculated dynamically. These values are displayed on the user profile all the time, and precalculated data allows faster data access every time the user reloads the page. Additionally, source language ID and target language ID are stored in both the task table and the job table. This also simplifies queries for functions where freelancers can filter the available tasks by language, and tasks are displayed according to those without being grouped by job ID. Although these approaches can make the update operations more complex, they help the database design align with the application needs.

# **Chapter - 6**

## **Tools and Technologies**

## 6. Tools and Technologies

### 6.1. Database Management System

The database design of the system is well-structured, involving transactions and consistent data, as mentioned in Chapter 5 System Design. Choosing the right database management systems for the platform is based on this database design. There are 3 popular categories in database management systems: NoSQL, RDBMS, and graph databases. A graph database stores data in a graphical form with nodes and edges, which is not an appropriate option according to the initial database design of the platform. Relational database management systems are very important for solving ACID (atomicity, consistency, isolation, and durability) problems where data validity is needed (Elmasri and Navathe, 2015). Therefore, RDBMS will be used as the database management system for the platform since our system needs database transactions with ACID properties. Among RDBMS, PostgreSQL is an open-source database engine that can manage relational data, support native JSON data type, and create custom data type (PostgreSQL Documentation, 2025). PostgreSQL runs on many operations, and hence, it is selected as the database engine of the platform.

### 6.2. Backend Technologies

For backend development, Python is selected as the backend programming language due to its wide library support. Python codes are simpler than other programming languages, and they make written business logic codes readable and easier to maintain. FastAPI, a modern API development framework built on Python, is chosen for API development to connect with the frontend. It supports async operations that can handle data concurrency and can prevent users from accessing and modifying the same data. To interact with the database, an ORM is used for better query management. SQLAlchemy is an open-source Python library that provides an SQL toolkit for database interactions. Queries written with ORM are easier to maintain and manage than raw queries and also interact seamlessly with databases. For database versioning and migration, the Alembic library is selected to ensure data integrity by backing up changes and minimizing downtime during the migration process.

### 6.3. Frontend Technologies

For front-end development, the React library will be used for creating UI components with Typescript. React is created by Meta, and it is popular in front-end development for its strong community support. React allows building user interfaces with a component-based structure, and its virtual DOM helps UI updates efficiently. The selected frontend programming language, TypeScript, is an extended JavaScript with static typing. Due to type safety, TypeScript is suitable for medium and larger projects where maintainability and scalability of code are essential.

## **Chapter - 7**

# **Payment Methods**

## 7. Payment Methods

### 7.1. Overview of Payment Integration

Payment integration service plays the most important role in developing a crowdsourcing platform. The user's trust and reliability in the platform depend on how the payment system is secure and seamless. Moreover, since the platform is intended for all kinds of users, with various technology familiarity levels, the payment integration service also needs to focus on user-friendly interfaces and local payment system support.

### 7.2. Myanmar Payment Systems

Myanmar's payment system has started digital transformation in recent years by reducing cash dependency and improving interoperability between digital wallets. The Central Bank of Myanmar (CBM) is leading in the modernization of financial infrastructure and has launched the National Payment System Strategy (2020–2025) to create a strong digital payment framework (Ministry of Information Myanmar, 2022). Key components of the National Payment System Strategy are Automated Clearing House (ACH) support for retail payments and the MMQR (Myanmar's QR Code Payment Standard) system (Ministry of Information Myanmar, 2022). MMQR was launched on March 4, 2022, to enable seamless transactions across banks, mobile wallets, and merchants. MMQR allows customers to make payments between different digital wallets, such as Kpay, Wave Pay, and Mytel Pay, which are popular digital wallets in Myanmar. An updated version of MMQR provides payment for cross-border interoperability for international transactions (Ministry of Information Myanmar, 2023).

### 7.3. Global Payment Systems

International transactions can be done with global payment systems such as Stripe, Paypal, and Payoner with payment gateway APIs, which can be incorporated into software. Moreover, these gateway protects from financial frauds with minimal processing time and facilitate payment transactions in various major world currencies, securing payment systems to users worldwide.

## 7.4. Security Consideration in Payment Processing

Security is one of the most important approaches for payment processing to protect user information and transaction integrity. The payment system of the platform must align with standards like PCI DSS to ensure data storage, data processing, and payment information security (PCI Security Standards Council, n.d.). Encryption protocols with secure authentication mechanisms must be accomplished to keep sensitive data safe and prevent illegal transfer of personal information. By applying these security measures, a secure and reliable payment system can be combined into the platform, and trust can be built between freelancers and the platform.

## 7.5. Challenges in Payment Integration and Solution

One major challenge in this project is integrating the payment system with the gateway due to the high cost associated with API fees. Even payment system integration is technologically ready (the MMQR payment system is still not ready for full integration during this project development); the current budget for platform development has limitations. This budget limitation for payment APIs fees has significant risk for overall system functionality, and improving user experience will be more challenging.

To overcome this problem, the project will use an alternative approach that uses a manual payment with a payment workflow system. Instead of fully implementing the payment API, the payment process will go to manual transaction processing, and the system will support the user to share necessary bank or transfer details. Once the payment is made by the admin or company staff, users will be provided proof (such as a screenshot or receipt) via email. This method allows the payment transaction to be verified with the required details without needing to use paid APIs.

Since the project plan includes payment integration, the payment system can be integrated when the MMQR payment system (which is intended as the National Payment System Strategy) is ready or when more cost-effective options are available.

## **Chapter - 8**

# **Risk Management**

# 8. Risk Management

## 8.1. Risk Identification

### 8.1.1. Technical Risk

Technical risk in software development is defined as "a combination of uncertainty and magnitude of difference between actual and optimal design of product artifacts and processes" (Antinyan et al., n.d.). Technical risk mainly impacts the functionality of the product, and managing those risks makes sure the final product is as close to perfect as possible. Some technical risks are:

#### **Software Bugs or Integration Issues:**

*Example:* The QA assessment mechanism does not correctly update the translation accuracy of freelancers, resulting in a mismatched accuracy rating.

#### **Delay in System Performance:**

*Example:* When many users try to do a translation task, the system may experience delays in accessing task info due to inefficient database queries. This could lead to slower performance.

#### **Using outdated technologies:**

*Example:* One of the libraries used in the frontend development does not have an updated version that is compatible with the system technology and can cause the system to crash in the future.

### 8.1.2. Security Risk

Security risks are associated with the whole system's security and reliability and can impact the user's trust in the platform. Some security risks include:

#### **Weak authentication:**

*Example 1:* Making payments by users does not have strong authentication mechanisms.

*Example 2:* Users are having unauthorized access to the system database. Since the platform is for data collection and does not prevent unauthorized access to the database, it can cause data loss or data theft.

### 8.1.3. Financial Risk

Financial risks are related to not having efficient financial management and lead to running out of budget due to overspending money as planned. The cost of developing the platform needs to be managed carefully as decided by the stakeholders and project manager. With inefficient financial management, financial risks will arise and impact the continuation of the project. Examples of financial risks are:

#### **Unexpected expenses:**

*Example 1:* Spending more money on prototyping with multiple iterations can cause budget overruns.

*Example 2:* Spending money on third-party tools for non-core functionalities can lead to unexpected costs.

### 8.1.4. External Risk

External risks are more related to outside of the development team, such as changes in policies and rules in IT products, political problems, and changes in market needs. Examples of external risks are :

#### **Regulatory Changes:**

*Example:* changes in the digital data collection system that require new compliance measures. For instance, new data protection laws were established that need additional encryption.

## 8.2. Risk Matrix

Risk Description	Category	Likelihood	Impact	Mitigation Strategies	Risk Owner
Software Bugs or Integration Issues	Technical	Medium	High	Doing thorough testing every iterative development cycle to catch errors early.	Development Team
Delay in System Performance	Technical	Medium	Medium	Optimize the query performance and adjust server resources based on load.	Development Team
Using outdated technologies	Technical	Low	Medium	Regular technologies and tools review, and upgrade and migrate outdated technologies as needed.	Technical Leader
Weak authentication in payment systems	Security	Medium	High	Implement multi-factor authentication and encryptions.	Development Team
Unauthorized Data access	Security	Medium	High	Implement strict access controls and logging	Development Team

Over-expenses for prototyping	Financial	Low	High	Track budget and prioritize budget usage on essential features	Project Manager
Over-expenses for third-party tools	Financial	Low	High	Use open-source tools with strong community support or use an alternative approach with lower pricing	Technical Leader
Regulatory Changes in Data Protection Laws	External	Low	High	Regularly check with legal advisors and make changes as needed	Business Owner

## **Chapter - 9**

# **User Interface Design**

# 9. User Interface Design

## 9.1 Design Principles

The platform development uses the following design principles to ensure the consistency, accessibility, and scalability of the platform. These statements outline the fundamental qualities of the user interface and are integrated using Material UI's theming system.

### **Visual Consistency**

To ensure visual consistency of the platform, the color palette is defined based on the color of the company, MyanLang, logo. Primary color and secondary color used primary: (#1976d2) and secondary: (#00ACC1) to align with the logo and a neutral background (#f4f6f8) used to reduce visual attention and to make the content eye-catching.

For the font family, Roboto is used since it is a clean and modern style. Font sizes are scaled for better readability (e.g., h5 adjusts from 1.2rem to 1.5rem on larger screens).

### **Component customization**

Components of user interfaces are customized for usability. Corners are rounded (12px), and smooth transitions are used to improve the interactive feedback of the screen. Input fields and buttons also use uniform rounded corners with shadow (0 4px 6px -1px rgba(0, 0, 0, 0.1)).

### **Responsive Design**

Mobile-first breakpoints (xs: 0, sm: 600px, md: 600px, lg: 1200px) are used in the UI development to ensure seamless adaptation to the user screen size, and font size is also adjusted based on viewpoint widths.

### **Readability and accessibility**

Primary and secondary colors are defined to meet WCAG contrast ratios against the background for better readability, and fonts for headings are bold (700 weight) to improve scannability and hierarchy.

## 9.2 Design System Examples

### Palette Colors

Primary Color	primary	Background Color	background (default)
 #1976d2		 #f4f6f8	
Secondary Color	secondary	Toggle Button Group Background	MuiToggleButtonGroup backgroundColor
 #00ACC1		 #ffffff	

### Typography

#### Font Family

"Roboto", "Helvetica", "Arial", sans-serif

#### Header 5 (h5) Settings

Font Weight:	<b>700</b>
Letter Spacing:	-0.025em
Responsive Font Sizes:	
➤	For screen widths ≥ 600px: 1.2rem
➤	For screen widths ≥ 1200px: 1.5rem

#### Preview:

**This is how Header 5 (h5) looks like**

## Component Styles

### MuiButton

Text Transform:	none
Border Radius:	12px
Padding:	12px 24px
Transition:	all 0.2s cubic-bezier(0.4, 0, 0.2, 1)

Button Example

### MuiTextField

Outlined Input Border Radius: 12px

Text Field Example

### MuiToggleButtonGroup

Background Color:	#ffffff
Border Radius:	14px
Box Shadow:	0 4px 6px -1px rgba(0, 0, 0, 0.1)

Option 1   Option 2   Option 3

## **Chapter - 10**

### **Sprint Planning and Execution**

# 10. Sprint Planning and Execution

## 10.1. Sprint 1

### 10.1.1. Sprint 1 Planning

**Sprint Title :** Platform Setup and Admin Site Management Process

**Duration :** 3 week

**Start Date:** December 23, 2024

**End Date:** January 10, 2025

**Sprint Goal :** Sprint 1 goal is to implement the core features of the platform. This includes a user authentication system for freelancers, an admin dashboard for job management, assessment task management, QA member management, and assessment task submission and reviewing process.

**Scope :**

- **User Stories**

- As a freelancer I want to register.
- As a freelancer I want to login.
- As a freelancer I want to answer assessment tasks.
- As a freelancer I want to get notification from assessment reviewed results.
- As an admin I want to post assessment tasks.
- As an admin I want to post a job.
- As an admin I want to register qa members.
- As a member I want to review and assess tasks.

- **Key Features**

- User authentication (All users)

- Assessment task management (Admin)
- QA management (Admin)
- Job management (Admin)
- Assessment task submission (Freelancer)
- Assessment review (QA member)

### **Task Breakdown**

<b>Task ID</b>	<b>Sprint Backlog</b>	<b>Story Point</b>	<b>Assignee</b>
T-1	Setup initial database	3	Aung Khant Phyo
T-2	Design Login/ Register Interface	3	Aung Khant Phyo
T-3	Design Admin Dashboard Interface	3	Aung Khant Phyo
T-4	Design Assessment Task Management components	3	Aung Khant Phyo
T-5	Design Job Management components	3	Aung Khant Phyo
T-6	Design QA Member Management components	3	Aung Khant Phyo

T-7	Design Assessment Task Submission Interface	3	Aung Khant Phyoe
T-8	Design User Profile	3	Aung Khant Phyoe
T-9	Design Assessment Task Review Interface and components	3	Aung Khant Phyoe
T-10	Develop User Authentication System	5	Aung Khant Phyoe
T-11	Develop Job Management Session of Admin Dashboard	5	Aung Khant Phyoe
T-12	Develop QA Management Session of Admin Dashboard	3	Aung Khant Phyoe
T-13	Develop Assessment Task Management Session of Admin Dashboard	5	Aung Khant Phyoe
T-14	Develop User Profile for freelancer	3	Aung Khant Phyoe
T-15	Develop Assessment Task Submission for freelancer	5	Aung Khant Phyoe
T-16	Develop Assessment Task Review Interface for QA member	5	Aung Khant Phyoe

T-17	Develop Email Notification system for registration and assessment task results.	5	Aung Khant Phyo
------	---	---	-----------------

### Timeline

- **Days 1-5**

- Authentication Setup and Assessment Task Management for Admin.
  - Task include: (T-1, T-2, T-3, T-4, T-10, T-13)

- **Days 6-10**

- Job Management and QA Management for Admin, Assessment Task Submission for Freelancer.
  - Task include: (T-5, T-6, T-6, T-11, T-12, T-15)

- **Days 11-15**

- Assessment Review process for QA, Email notification system, Freelancer Profile.
  - Task include: (T-8, T-9, T-14, T-16, T-17)

### Acceptance Criteria

- Freelancer can register and login without error.
- Freelancer can recreate passwords when forgotten.
- All user roles can login without error.
- Admin can create Assessment Tasks without error.
- Admin can create QA members with an initial password.
- QA members can set their own password.

- Admin can post jobs without errors.
- Freelancer can browse available language pairs.
- Freelancer can submit assessment tasks without error.
- QA members can review assessment tasks without error.
- Freelancer will get assessment task results with email notification after the QA member has reviewed.

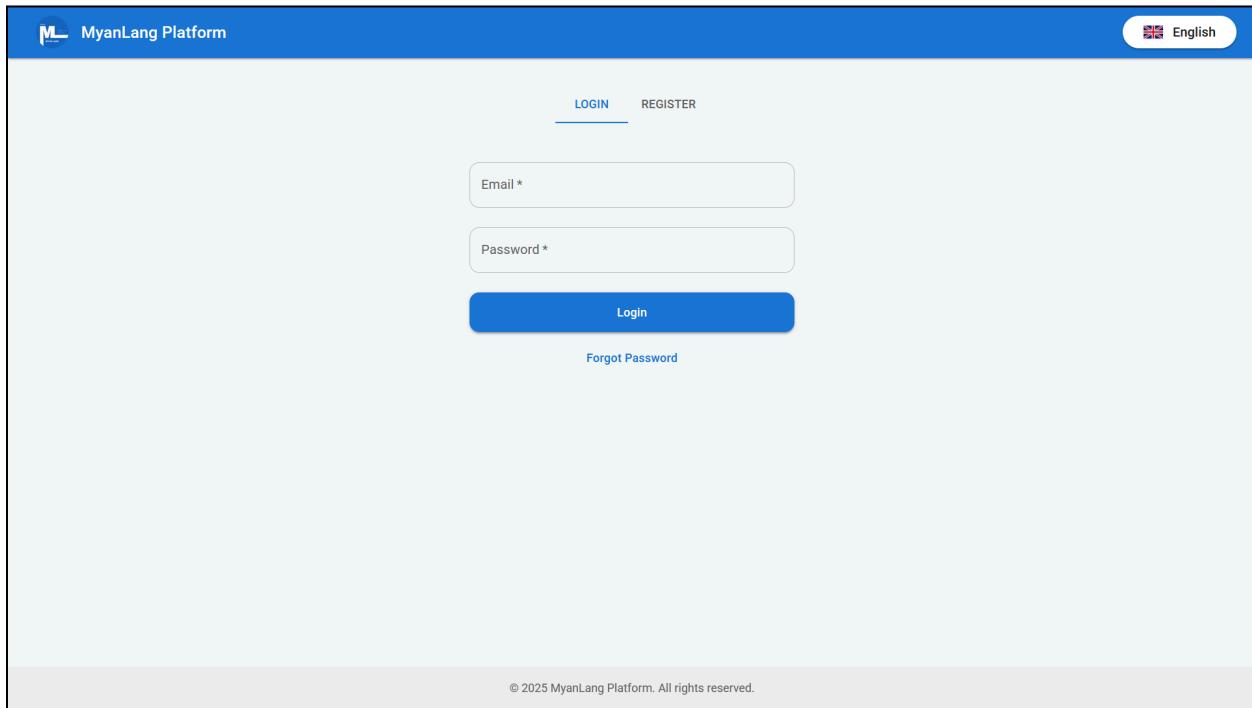
### 10.1.2. Sprint 1 Execution

#### Sprint Outcomes:

- **Completed Tasks**
  - T-1: Setup initial database
  - T-2: Design Login/ Register Interface
  - T-3: Design Admin Dashboard Interface
  - T-4: Design Assessment Task Management components
  - T-5: Design Job Management components
  - T-6: Design QA Member Management components
  - T-7: Design Assessment Task Submission Interface
  - T-8: Design User Profile
  - T-9: Design Assessment Task Review Interface and components
  - T-10: Develop User Authentication System
  - T-11: Develop Job Management Session of Admin Dashboard
  - T-12: Develop QA Management Session of Admin Dashboard
  - T-13: Develop Assessment Task Management Session of Admin Dashboard

- T-14: Develop User Profile for freelancer
  - T-15: Develop Assessment Task Submission for freelancer
  - T-16: Develop Assessment Task Review Interface for QA member
  - T-17: Develop Email Notification system for registration and assessment task results
- 
- **Unplanned Work**
    - Added verification code system for both registration and resetting password to validate user email.
    - Added logic that QA members cannot reset initial password without creating their own password first.
    - Added registering new language for task translation from admin site.
  
  - **Deliverable**
    - Functional User Authentication System
    - Admin Dashboard with Job Management, Assessment Task Management and QA Management
    - Freelancer Assessment Task Submission System
    - QA Member Assessment Task Review System
    - Email Notification System for delivering validation code and assessment task result

### 10.1.3. Sprint 1 Screen Diagrams



*Figure 20: Login Page*

See [Appendix C](#) for more screen diagrams of the sprint 1.

#### 10.1.4. Sprint 1 Testing

**Test Plans:** A set of Test Plan will be made to cover all the followings areas:

Test Area	Description	Approach
User Registration and Login	Test freelancer registration, login and resetting password. Admin login and QA member login, create password and resetting password.	Manual
Admin Functionalities	Test registering new language, assessment task management, job management, QA member management on the admin dashboard	Manual
Freelancer Assessment Task Submission	Test browsing assessment task and submitting assessment tasks for review on freelancer interface.	Manual
QA Task Review Process	Test reviewing assessment task and submitting all the reviewed results.	Manual

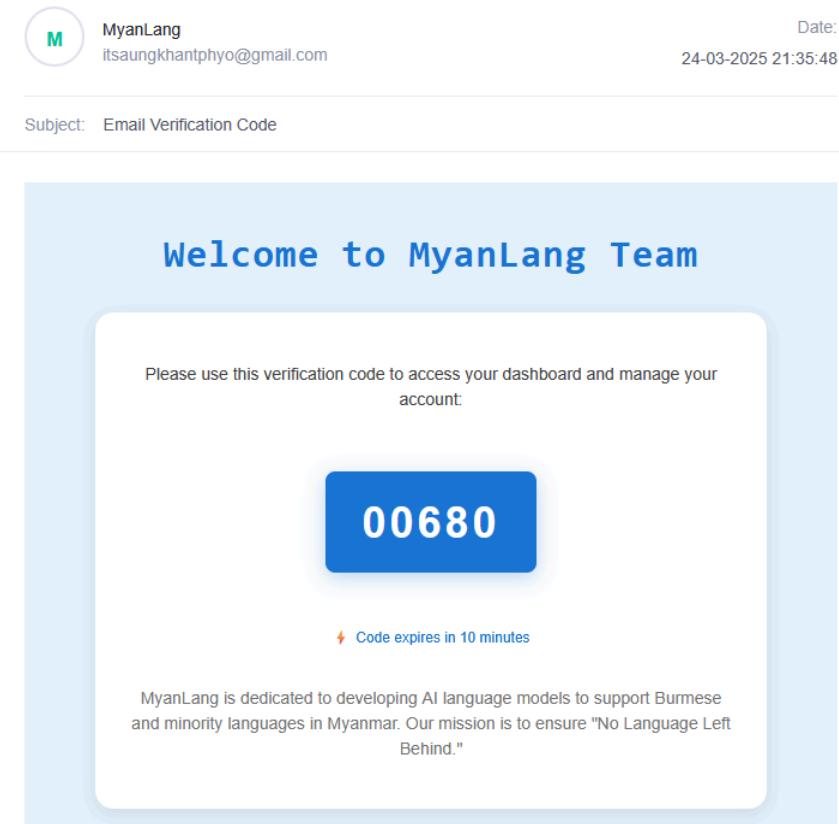
#### Testing Results

Test Case ID	REG-01
Test Area	User Registration and Login
Date	January 7, 2025
Sprint	Sprint 1
Test Scenario	Freelancer requests verification code for registration.

<b>Expected Result</b>	Email sent successfully with correct email template and validation code.
<b>Actual Result</b>	Some emails are not sent properly and some are delayed.
<b>Status</b>	Fail
<b>Bug ID</b>	BUG-101
<b>Tester</b>	Aung Khant Phyo
<b>Notes</b>	Fixed by changing sender email address with “App password” provided by Gmail.

The screenshot shows the MyanLang Platform registration process. At the top, there's a blue header bar with the logo 'ML MyanLang Platform' on the left and a language selection button 'English' on the right. Below the header, there are two buttons: 'LOGIN' and 'REGISTER', with 'REGISTER' being underlined to indicate it's the active step. A horizontal navigation bar below these buttons lists three steps: '1 Email Verification', '2 Enter Verification Code', and '3 Complete Registration'. The main content area contains a form field labeled 'Email \*' with the value 'votif16582@gamebcs.com' entered. Below this is a large grey button labeled 'Send Verification Code'.

Figure 23: Freelancer Requesting Verification Code



*Figure 24: Freelancer Receiving Verification Code with Correct Email Template*

See [Appendix B](#) for more test results from sprint 1.

### 10.1.5. Sprint 1 Summary

Being the very first sprint of the development process, the sprint started with creating a database and setting up a codebase. The database was created locally with PostgreSQL, and the codebase is set up with Vite. For the backend, Python is used, and for the frontend, JavaScript is used. For database migration, Alembic, a lightweight database migration tool, is used for better schema management. Database connection and API calling are tested with raw UI. High-level user interfaces are designed using Figma with the system theme. After that, the authentication system and admin dashboard are developed.

While developing assessment tasks for the admin dashboard, "new language registration to the system" feature was added without planning in the sprint to make sure the whole translation language in the system (not system language) is synchronized. Similarly, when developing a QA management system, a validation logic—that prevents a QA member from resetting the initial password without creating their own password—is added without planned in this sprint since somehow, when an admin resets a QA password with the initial password, after the QA member quits the job, he or she will not be able to reset the password without knowing the password set by the admin. After that, the assessment task submission system, task reviewing system, and the email notification system are implemented.

When testing, the interface has some layering errors, such as some dialog does not pop up and stays behind other components. This error is fixed during this sprint and tested again and passed. One main challenge of this sprint is the email delivery system. Initially, personal email is used for delivering email, and some email is not being sent. Therefore, the email sender address has changed to another address that supports delivering service with the "App password" of Gmail. And then, the deliverable product of this sprint was tested with manual testing to cover all the test areas that are mentioned above.

## 10.2. Sprint 2

### 10.2.1. Sprint 2 Planning

**Sprint Title :** Task Translation Process and Payment System Integration

**Sprint Goal :** The goal of this sprint is to implement a working payment system for freelancer payment and a complete task translation, submission and reviewing system.

**Duration :** 2 week

**Start Date:** January 13, 2025

**End Date:** January 24, 2025

**Scope :**

- **User Stories**

- As a freelancer I want to do task translation.
- As a freelancer I want to request payment.
- As a freelancer I want to receive notification when I get paid.
- As a freelancer I want to receive notification when a user requests payment.
- As an admin I want to pay requested payments.
- As an admin I want to see payment requests as a table.
- As a QA member I want to review user submissions.

- **Key Features**

- Translation Task Submission (Freelancer)
- Payment System Integration (Both Freelancer and Admin)
- Translation Tasks Review (QA member)

## Task Breakdown

Task ID	Sprint Backlog	Story Point	Assignee
T-18	Design Task translation interface for Freelancer	3	Aung Khant Phyo
T-19	Design Task review interface for QA member	3	Aung Khant Phyo
T-20	Design Payment Page for Freelancer	3	Aung Khant Phyo
T-21	Design Payment Page for Admin	3	Aung Khant Phyo
T-22	Develop Payment System for both Admin and Freelancer	8	Aung Khant Phyo
T-23	Develop Task translation and submission Page for Freelancer	5	Aung Khant Phyo
T-24	Develop translated task reviewing Page for QA member	5	Aung Khant Phyo

T-25	Develop email notification system for payment request and processed payment	3	Aung Khant Phy
------	---	---	----------------

### Timeline

- **Day 1-5**
  - Task Translation, submission and review system
    - Task include (T-18, T-19, T-23, T-24)
- **Day 6-10**
  - Payment System
    - Task include (T-20, T-21, T-22, T-25)

### Acceptance Criteria

- Freelancer can translate and submit tasks without error.
- Freelancer can request payment and the request will be sent as an email notification to the admin.
- Admin can process freelancer requested payment without error and payment confirmation email will be sent to the freelancer.
- QA members can review submitted translations without error.

### 10.2.2. Sprint 2 Execution

#### Sprint Outcomes :

- **Completed Tasks**

- T-18 : Design Task translation interface for Freelancer
- T-19 : Design Task review interface for QA member
- T-20 : Design Payment Page for Freelancer
- T-21 : Design Payment Page for Admin
- T-22 : Develop Payment System for both Admin and Freelancer
- T-23 : Develop Task translation and submission Page for Freelancer
- T-24 : Develop translated task reviewing Page for QA member
- T-25 : Develop email notification system for payment request and processed payment

- **Unplanned Work**

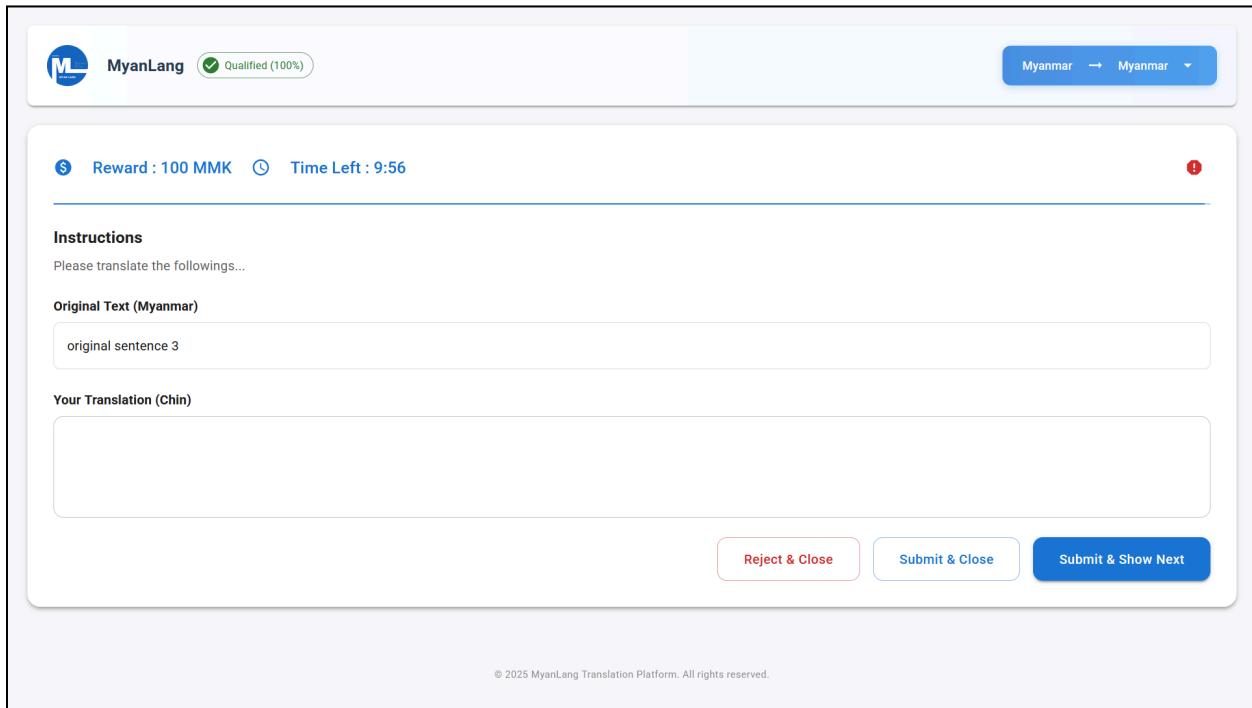
- Added “Appeal submission” feature for freelancer when their accuracy is low and cannot participate to the translation
- Changed the "Source language and target language selection" feature in "Job posting" so that the language pair can only be selected when assessment tasks are available.

*Justification : For selecting language pairs that do not have assessment tasks, freelancers will not be able to participate in tasks of this job since there is no assessment task to answer.*

- **Deliverable**

- Functional Task translation process
- Functional Payment System

### 10.2.3. Sprint 2 Screen Diagrams



*Figure 21: Task Translation Interface of Freelancer*

See [Appendix D](#) for more screen diagrams of the sprint 2.

#### 10.2.4. Sprint 2 Testing

**Test Plans:** A set of Test Plan will be made to cover all the followings areas:

Test Area	Description	Approach
Payment System Functionalities	Test requesting payment, processing payment and receiving email notification.	Manual
Task translation and submission, submitting appeal message	Test Task translation, submission, and submission appeal message of Freelancer portal. Test receiving email notification as an admin for appeal messages.	Manual
Task review process of QA member portal	Test task reviewing process and calculating freelancer language accuracy logic, and updating QA member's decision records (for acceptance rate of QA member)	Manual

#### Test Results

Test Case ID	TASK-SUBMIT-01
Test Area	Task translation and submission, and submitting appeal message
Date	January 24, 2025
Sprint	Sprint 2
Test Scenario	Freelancer submit translated task
Expected Result	Task was submitted successfully with a success toast message.
Actual Result	Task was submitted successfully with a success toast message.

Status	Pass
Bug ID	N/A
Tester	Aung Khant Phyo
Notes	N/A

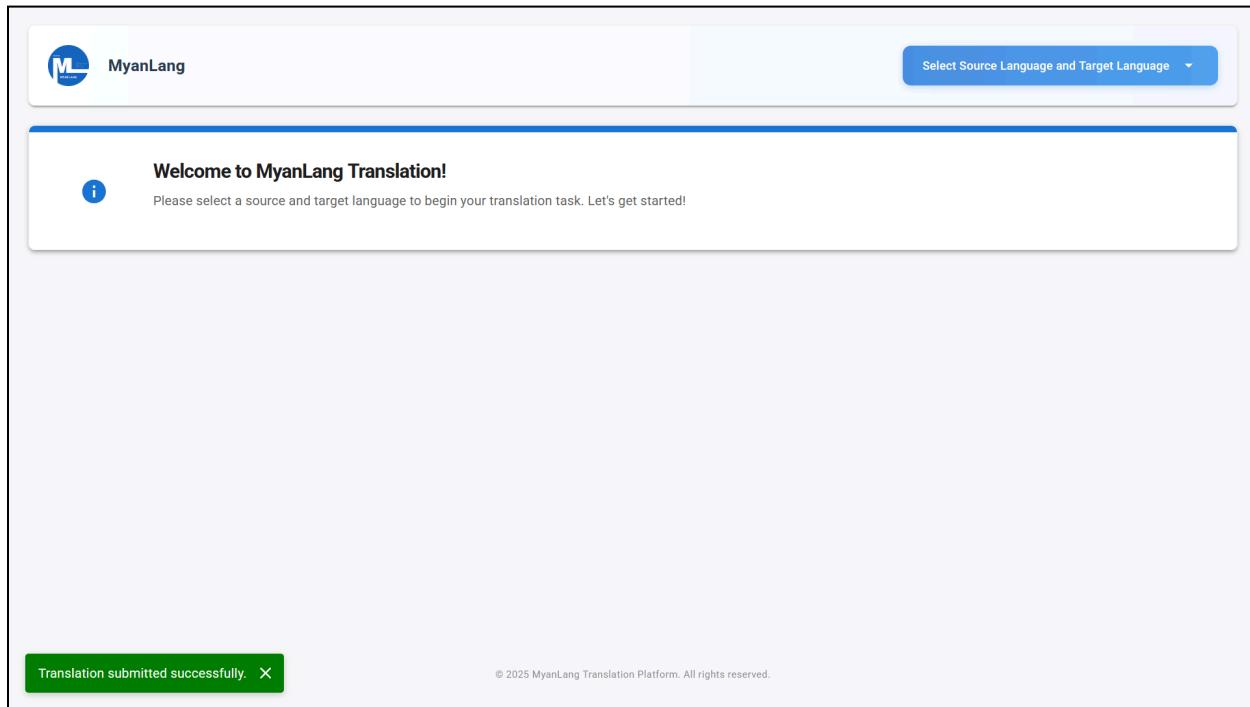


Figure 25: Success toast message displayed after a freelancer submits a task

See [Appendix B](#) for more test results from sprint 2.

### 10.2.5. Sprint 2 Summary

This sprint started with designing user interfaces for payment pages and task translation interfaces. For the task translation interface, old components from assessment task translation are used with slight adjustments.

Assigning tasks to users was made based on the following logics (see source code on [appendix E](#)):

- Once the user loaded the task translation interface, one available task was assigned to this freelancer.
- Getting available tasks was made by two rules:
  - If the task status was open, the task is an available task.
  - If the (current time - assigned\_at time) is greater than the given time for this task, this task also serves as an available task even if the task status is assigned (assuming that the user does not submit the task within the given time).
- The freelancer must submit the task within the given time, and if not, the submission will fail and the task will open again.
- If a freelancer does not submit a task, there will be no conflict in the database, and this task will be assigned to a new freelancer when the given time passes (see the second logic).

When developing task review, the "Appeal Message Submission" feature was added which was unplanned in the sprint. Freelancers can submit appeal messages when the language accuracy rate for a language pair is lower than the company-accepted percentage, and admins can reset the accuracy rate. Additionally, the "Source language and target language selection" feature in "Job posting" was changed so that the language pair can only be selected when assessment tasks are available. The payment system was developed with a manual payment approach, as mentioned in chapter 7. 5 Challenges in Payment Integration and Solution. Cloudinary, an image API, was used for uploading payment receipts, and the freelancer will receive payment with their desired payment method, and a link to the payment receipt will be displayed in the "withdrawal history" section of the freelancer portal.

## 10.3. Sprint 3

### 10.3.1. Sprint 3 Planning

**Sprint Title :** Issue Report and Multi-language Support and Final Testing

**Sprint Goal :** The goal of this sprint is to develop Issue Report features for freelancers, Issue resolve features form admin and implementing Multi-Language Support for the system. Final overall testing will be done this sprint and the deliverable product of this sprint is a fully functional platform with all the requirements.

**Duration :** 2 week

**Start Date:** January 27, 2025

**End Date:** February 7, 2025

**Scope :**

- **User Stories**

- As a freelancer I want to report when the given tasks have wrong source language.
- As a freelancer I want to report when the requested payment is delayed or not getting paid.
- As a freelancer I want to report when the time given for the task translation is not enough.
- As a freelancer I want to change the system language to a local language.
- As an admin I want to resolve user reported issues.
- As an admin I want to get notification when freelancers report an issue.
- As a freelancer I want to get email notification when the reported issue is resolved.

- **Key Features**

- Issue Report System
- Multi-language Support

### Task Breakdown

Task ID	Sprint Backlog	Story Point	Assignee
T-25	Design Issue report dialogs for each issue types	3	Aung Khant Phyo
T-26	Design Issue Page for admin dashboard	5	Aung Khant Phyo
T-27	Design System language selector dialog	3	Aung Khant Phyo
T-28	Translate system text to local languages	5	Aung Khant Phyo
T-29	Develop Issue report system for each issue types for freelancer	3	Aung Khant Phyo
T-30	Develop Issue resolution system for admin	5	Aung Khant Phyo
T-31	Develop email notification system for issues reporting	3	Aung Khant Phyo
T-32	Integrate translated system text	3	Aung Khant Phyo

T-33	Test the whole system with to cover all test area from previous sprints	5	Aung Khant Phy
------	---	---	----------------

## **Timeline**

- **Day 1-5**
  - Issue report system and system text translation
    - Task Include (T-25, T-26, T-28, T-29, T-30)
- **Day 6-10**
  - Multi-language support, email notification system and retesting previous sprints' test areas
    - Task Include (T-27, T-31, T-32, T-33)

## **Acceptance Criteria**

- Freelancers can report payment delays without error.
- Freelancers can report wrong source language tasks without error.
- Freelancers can report about not having enough time for translation without error.
- Admin will get email notification for every reported issue.
- Freelancers can change system language to their desired local language.

### **10.3.2. Sprint 3 Execution**

#### **Sprint Outcomes**

- **Completed Tasks**
  - T-25 : Design Issue report dialogs for each issue types
  - T-26 : Design Issue Page for admin dashboard
  - T-27 : Design System language selector dialog
  - T-28 : Translate system text to local languages

- T-29 : Develop Issue report system for each issue types for freelancer
- T-30 : Develop Issue resolution system for admin
- T-31 : Develop email notification system for issues reporting
- T-32 : Integrate translated system text
- T-33 : Test the whole system with to cover all test area from previous sprints

- **Unplanned Work**

- Added the main page of the platform. Justification : Previously, the login page was set as the main page of the platform which is not user-friendly. Therefore, a new main page was designed and implemented.
- Added 404 page to return when users try to navigate protected pages or invalid pages.

- **Deliverable**

- Issue report and resolve system
- Fully functional platform with all the defined requirements.
- Main Page of the Platform.

### 10.2.3. Sprint 3 Screen Diagrams



*Figure 22: Main Page of the Platform*

See [Appendix E](#) for more screen diagrams of the sprint 3.

#### 10.3.4. Sprint 3 Testing

**Test Plans:** A set of Test Plan will be made to cover all the followings areas:

Test Area	Description	Approach
Issue report and resolve system	Test reporting all issues types as freelancer and resolving as admin. Test email notification for issues reports sent accurately.	Manual
Multi-language support	Test changing system language of the platform.	Manual
Protected Routes	Test to navigate protected page for each user role.	Manual

#### Test Results

Test Case ID	ISSUE-01
Test Area	Issue report and resolve system
Date	February 7, 2025
Sprint	Sprint 3
Test Scenario	Freelancer report about payment delay.
Expected Result	Report was successfully sent to the admin with email notification.
Actual Result	Report was successfully sent to the admin with email notification.
Status	Pass
Bug ID	N/A
Tester	Aung Khant Phyo
Notes	N/A

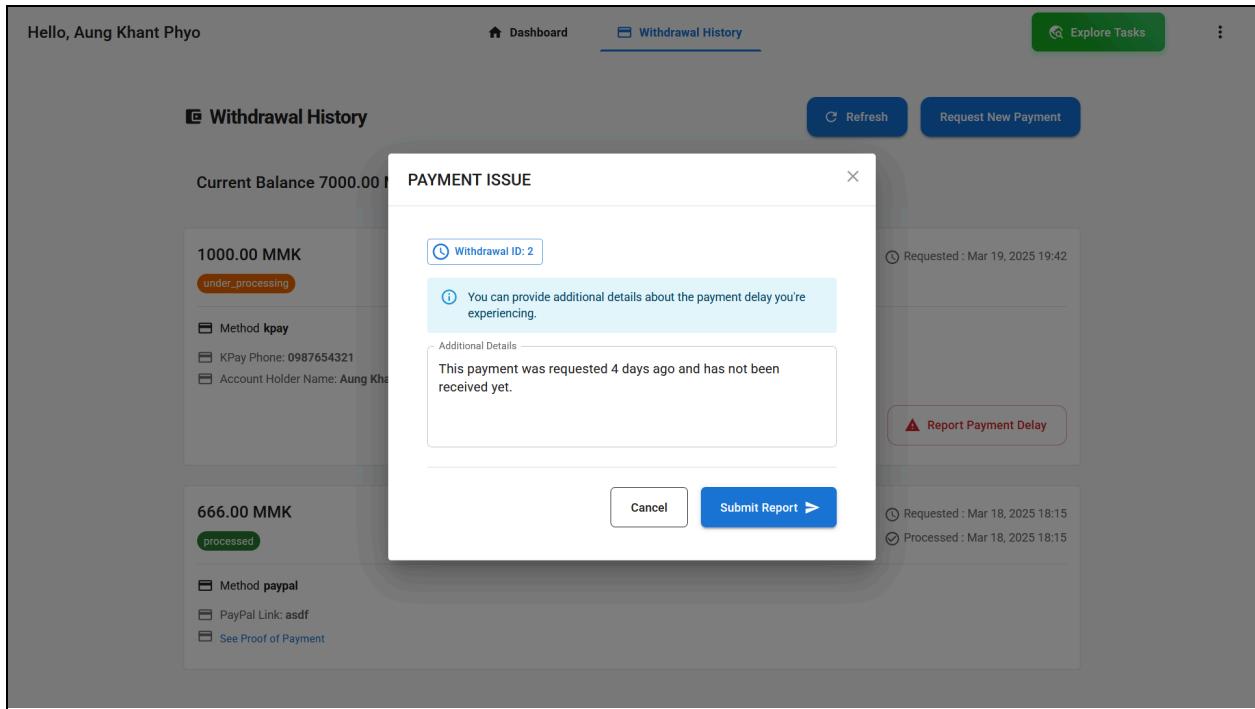


Figure 26: Freelancer reporting about payment delay

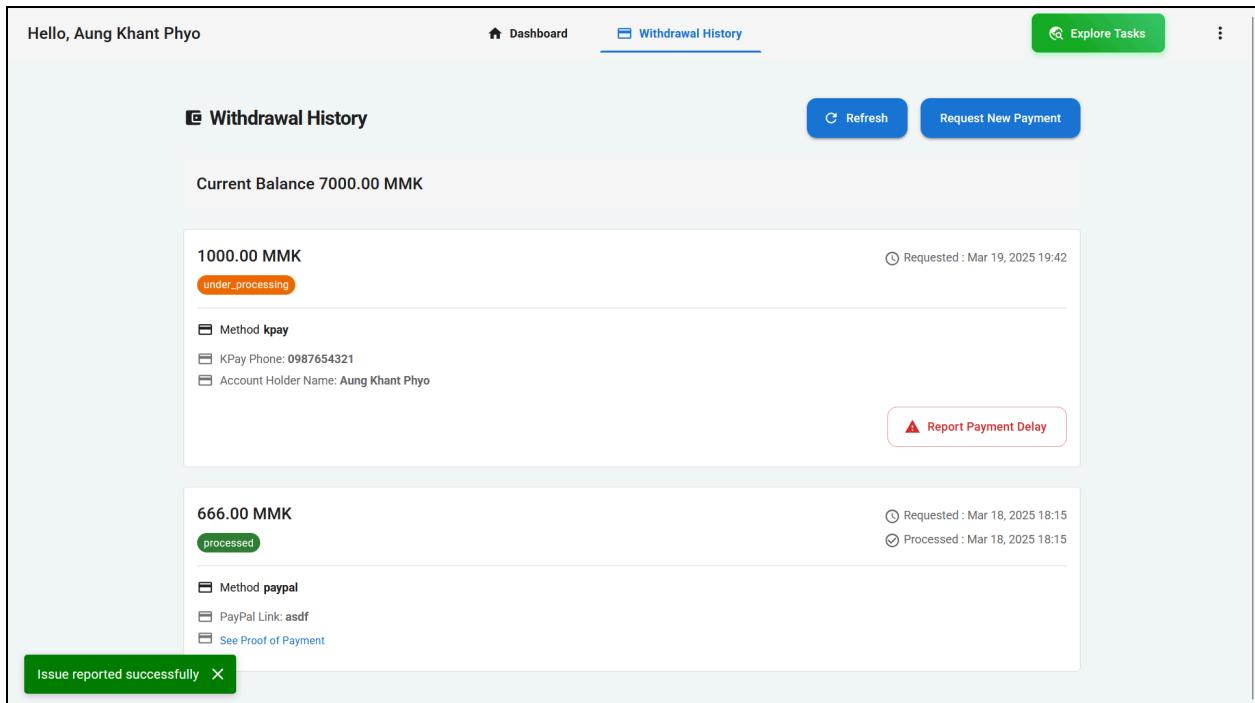


Figure 27: Success toast message displayed after a freelancer reports payment delay issue

See [Appendix B](#) for more test results from sprint 3.

### 10.3.5. Sprint 3 Summary

This sprint started with designing a user interface for issue report dialogs and the issue page for the admin dashboard. For multi-language support for system language, all the system texts are gathered in one file along with translations and displayed to the user according to the user-selected system language. This approach is more accurate, faster, and lighter than using other website localization applications. For local languages, Kachin and Chin languages are implemented since professionals for other language translations are unreachable.

During this sprint, the main page of the platform is implemented instead of using the login page as the main page for better user-friendliness. The "404 Page Not Found" page was also designed and implemented to display when users try to access protected routes and invalid links. Finally, all routes are protected by user roles, and the deliverable product of this spring, a fully functional platform, is tested to cover all the test areas.

## **Chapter - 11**

### **Evaluation and Conclusion**

# 11. Evaluation and Conclusion

## 11.1. Restate the Aims and Objectives

This project aimed to develop a text data collection platform for the MyanLang company. The text data collection platform was designed to provide easy registration for freelancers, a simple UI to improve usability, support for local payment systems, having QA members in the loop for collected text data review, and multi-language support as a system language.

To achieve this aim, several key objectives are set as follows:

- Research and Analysis
- Planning
- System Design
- User Interface Design
- Development
- Testing
- Evaluation and Justification

## 11.2. Evaluation by objectives

**Research and Analysis :** The project began with reviewing published papers about text data collection and analyzing existing similar systems. Most papers mentioned that the main challenge for minority language text data collection is not having machine translation for the "back-translation" technique, and "guided back-translation" is a solution to overcome this challenge. Although there are several similar systems to perform "guided back-translation" most of them lack support for local payment systems in Myanmar and local language support. This research phase helped define core requirements of the system, such as payment system integration and multi-language support as a system language. The LESP reviews ensured the new system design aligned with local legal requirements and professional and ethical standards.

**Planning:** During the planning phase, functional and non-functional requirements were defined along with use case diagrams for each user role. Agile methodology with scrum frame was

selected since the criteria of the project match with agile methodology (e.g., being a median project size, having user involvement with clearly defined user groups, etc.).

**System Design:** The system design phase involved developing UML diagrams (including class diagrams, objective diagrams, sequence diagrams, activity diagrams, and initial database design) and researching and selecting appropriate technologies to develop the system based on the UML diagrams. Detailed sequence diagrams for workflows were developed along with activity diagrams to clarify the functionalities of the system. For DBMS, PostgreSQL was selected for its structured approach for managing large volumes of data. Python was selected as the backend programming language, and Typescript with React was used to develop frontend interfaces. For payment system integration, a manual payment system with support for sharing receipts was implemented since payment gateways in Myanmar do not fully support covering all local and international payment systems. Risk assessment was made during this phase along with risk metrics to identify potential risks and overcome these risks.

**User Interface Design :** The initial design principles were defined during this phase, and example design examples are developed. User interfaces were fully designed during the development process to make necessary adjustments based on user feedback.

**Development:** The system development process was managed with sprint-based planning with the scrum framework. Each sprint starts with sprint planning following sprint execution and testing. The most important requirements were developed in the early sprints to ensure smooth workflow and to make necessary adjustments in late sprints. The sprint backlog developments were completed on schedule, and sprint 1 and 2 had some unplanned works that were missed during sprint planning. Both planned work and unplanned work were completed before the sprint ends, and testing for each sprint was done with detailed test cases.

**Testing:** Functional and non-functional testing were conducted during each sprint, and this objective was successfully achieved alongside the development process.

## 11.2. Evaluation against justification made

### 11.2.1. Payment

The payment system was initially intended to use a payment gateway that supported both local and international payment systems, but there is no such gateway. Additionally, the government developed payment gateway MMQR is still in an early state and is not fully documented to be integrated into the system. So payments need to be processed manually to cover all of Myanmar's most used payment methods, and the admin needs to upload the receipt as proof of payment.

Transaction security relied on the user-selected payment method provider, and there were still some concerns about misfilling information to make a transaction. A potential case of misfiling information has both a site for a freelancer and for an admin. This concern is the main drawback of not using a payment gateway, and a payment system will be the most important feature to make improvements in the future.

The other weak point is that it cannot make options for changing currency. Currently the platform supports only Myanmar Kyat, and rewards for task translation are also Myanmar Kyat. The system does not support changing the user's desired currency. This does not seem like a big deal for local usage, but this does not match social standards and should make improvements in future improvements.

### 11.2.2. System Language

Currently, the system only supports Kachin (Jingpo), Chin (Tedim), Myanmar, and English. Other local languages cannot be integrated into the system since professionals for this language are not reachable as a student. The implemented system languages were translated with local translators; for example, there are many Chin languages, and the implemented Chin language was translated by a local from the Tedim region.

Changing system language was done through a JSON file, which stores display text for each language. This approach helps the translation be accurate and faster than using other website localization tools. Supporting more system language can make the platform more reliable, making it a key area for future development.

### 11.2.3. Workflow

The translation workflow was well developed for this project to ensure the quality of the translation. Clear definition of requirements and UML diagrams of system design supported effective translation workflow. Every submitted translation needs to be reviewed by a QA member, and freelancers get paid only for valid translations. The translation accuracy is fully dependent on the QA member, who is directly hired by the company. Admin can see key metrics of the QA member (such as how many tasks rejected, how many tasks reviewed, and how many tasks accepted) for managing QA members.

### 11.2.4. Design

User interfaces for freelancers were designed emphasizing simplicity for better usability. Freelancer has only three main pages: a profile displaying personal information, a payment page where withdrawal history can be seen and can request payment, and the translation interface. The translation interface is separated from the other pages (do not render under the same app bar) to help users focus only on the task translation. Navigating to other pages from the task translation interface checked for unsaved work to prevent accidental data loss. The translation interface was used for both assessment tasks and paid task translation. Messages that display at the translation interface (such as "Assessment tasks are required" and "Assessment task under review") were properly prepared with a title and body message for clear user interaction. Both QA member interfaces and admin interfaces were also designed to match with the defined design system.

### 11.2.5. Challenges in development process

There have been several challenges faced during the development process. One main challenge was integrating the file uploader in the system. The file uploader is used to store payment receipts, and the integration has some challenges. Cloudinary is used for uploading payment receipts, and Cloudinary's built-in UI overlaps with a component: the UI buttons did not work after uploading the file and the system did not receive the uploaded file link. Customizing the Cloudinary's uploader button was required to integrate with the system, and retrieving the uploaded file link was overwritten in the system for file handling.

Another challenge was migrating databases. Managing database versions had some issues when reverting the database schema. For instance, adding new ENUM options conflicted with the existing enums and required deleting the old ENUM and recreating the new ENUM type. Rollback stratifications and manual adjustment for migration codes were needed to ensure data integrity.

Translating system language to local languages also had non-technical challenges. One main challenge was to access qualified translators for local languages. Only Kachin (Jingpo), Chin (Tedi), Myanmar, and English could be integrated. Many other local languages cannot be integrated due to not having professional translators.

#### 11.2.6. Lesson Learnt and Personal Evaluation

Being my first React project, this project was very challenging and rewarding for me. After completing this project, I understand why React is popular in web development. React uses component rendering method, and its efficiency helps the web app load faster, reducing the initial load time. Moreover, writing reusable components improves the maintainability of the code, and the written code becomes more readable and scalable.

Another lesson learned is how to use the Axios client for making API requests in a simple way. Axios supports JSON parsing, and it auto-converts server responses to JSON without additional methods. It is useful in adding auth tokens and error handling. The error message from the backend response can be easily used in the UI for a better user experience.

Moreover, I learned how to use components from the React library. React has a component library for reusable components with strong community support. These UI components are highly flexible and easy to integrate into the system.

During this project development process, I have also learned how to use local storage of user browsers effectively. Although storing user data in the local storage has some security concerns, non-confidential data stored in the local storage helps the system run faster, reducing loading time and improving user experience.

Additionally, I also learned how to define table relationships properly for backend databases and why they are important. Table relationships are the most important for maintaining data integrity and efficient queries. Having well-structured relationships and data validation enforce the business rules (such as assigning tasks to freelancers). This is the backbone of the whole program, and without properly defined relationships, data can become inconsistent, and data retrieving can be difficult.

Beyond the lessons learned, I am personally not satisfied with the limitation that users cannot change their personal information. Since the platform is intended for freelancers, users should be able to edit their personal data. The current system does not support this functionality, which I find inconvenient and something I would like to improve in the future.

#### 11.2.7. Future work

In future development of the platform, several improvements should be prioritized for a better user experience. Here are some future improvements.

**User profile edit feature:** The current system does not support editing user personal data which is inconvenient and users should be able to change their personal information such as email and year of birth.

**Enhancing the payment system:** The current system uses manual payment, which has a risk of errors for payment processing. A trustworthy payment gateway that supports both local payment methods and international payment methods should be fully integrated into the system. This change will provide a more secure and trustworthy payment system. Furthermore, the system should support multiple currencies to align with global social standards.

**Expanding system language support:** Currently the system supports a limited number of local languages. In future development, additional local languages should be integrated into the system by collaborating with qualified translators. This expansion will improve the system usability and will make the platform more accessible.

**Adding new data types for transcription:** To expand the platform capability, the system should support additional data types. This involves developing audio transcription and image transcription functionalities along with assessment tasks for those kinds of tasks. This could expand the platform capability and help meet the market needs.

### 11.3. Conclusion

I have completely designed and developed a text data collection platform for MyanLange that aligns with the aim and objectives. This project brings a solution to overcome challenges in local text data collection with the help of technology. The lessons learned during the development process helped me improve not only my technological skills but also my problem-solving skills. This project helped me transform my previously studied knowledge from the University of Greenwich and KMD Institute into a solution for a real-world problem. Developing the project by a methodology gave me hands-on practice as to how professional software developers work in companies. Using popular technologies like PostgreSQL, Python, and React, help me prepare how to provide a technological solution for a market need. In conclusion, this project taught me how technology can provide valuable solutions for the real world.

# References

Amazon Mechanical Turk (n.d.) Amazon Mechanical Turk. Homepage Screenshot Available at: <https://www.mturk.com/> (Accessed: 11 March 2025).

Antinyan, V., Staron, M., Meding, W., Henriksson, A., Hansson, J. and Sandberg, A. (n.d.) 'Defining Technical Risks in Software Development'. Available at: <https://gup.ub.gu.se/file/171785> (Accessed: 9 March 2025).

Badenko, V., Yadykin, V., Kamsky, V., Mohireva, A., Bezborodov, A., Melekhin, E., & Sokolov, N. (2024) 'Method for Developing the System Architecture of Existing Industrial Objects for Digital Representation Tasks', *Systems*, 12(9), p. 355. doi: 10.3390/systems12090355. Available at: <https://doi.org/10.3390/systems12090355> (Accessed: 23 March 2025).

Brandom, R. (2024) 'Scale AI's Remotasks platform is dropping whole countries without explanation', Rest of World, 28 March. Available at: <https://restofworld.org/2024/scale-ai-remotasks-banned-workers/> (Accessed: 11 March 2025).

Chukwunweike, J., & Aro, O. E. (2024). 'Implementing agile management practices in the era of digital transformation'. *ResearchGate*. Available at: [https://www.researchgate.net/publication/385291725\\_Implementing\\_agile\\_management\\_practices\\_in\\_the\\_era\\_of\\_digital\\_transformation](https://www.researchgate.net/publication/385291725_Implementing_agile_management_practices_in_the_era_of_digital_transformation) (Accessed: 9 March 2025).

Clickworker (2023) Clickworker Company Presentation 2023. Available at: <https://www.clickworker.com/wp-content/uploads/2023/11/clickworker-company-presentation-2023.pdf> (Accessed: 11 March 2025).

Diab, N., Sadat, F., and Semmar, N. (2024) 'Towards Guided Back-translation for Low-resource languages: A Case Study on Kabyle-French', Proceedings of the 16th International Conference on Human System Interaction (HSI), Paris, France, 8–11 July. Available at: <https://ieeexplore.ieee.org/document/10613577> (Accessed: 9 March 2025).

Elmasri, R. and Navathe, S.B. (2015) Fundamentals of Database Systems. 7th edn. Boston: Pearson. Available at: <https://www.auhd.edu.ye/upfiles/elibrary/Azal2020-01-22-12-28-11-76901.pdf> (Accessed: 19 March 2025).

Inglis, D. (2017) 'Myanmar-based Khamti Shan Orthography', Journal of the Southeast Asian Linguistics Society, 10(1), pp. xlvii–lxi. Available at:

<https://evols.library.manoa.hawaii.edu/items/385c4104-cedd-4d2e-99f2-90fc0dc99dec>

(Accessed: 9 March 2025).

Khan, S., & Mahadik, S. (2022). 'A Comparative Study of Agile and Waterfall Software Development Methodologies'. *International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)*, 2(1), pp. 399. doi: 10.48175/IJARSCT-5696.

Available at:

[https://www.researchgate.net/publication/361872079\\_A\\_Comparative\\_Study\\_of\\_Agile\\_and\\_Waterfall\\_Software\\_Development\\_Methodologies](https://www.researchgate.net/publication/361872079_A_Comparative_Study_of_Agile_and_Waterfall_Software_Development_Methodologies) (Accessed: 19 March 2025).

Kholodna, N., Julka, S., Khodadadi, M., Gumus, M.N., and Granitzer, M. (2024) 'LLMs in the Loop: Leveraging Large Language Model Annotations for Active Learning in Low-Resource Languages', *arXiv preprint arXiv:2404.02261*. Available at: <https://arxiv.org/abs/2404.02261> (Accessed: 9 March 2025).

KoboToolbox (n.d.) KoboToolbox. Available at: <https://www.kobotoolbox.org/> (Accessed: 11 March 2025).

Kurabe, K. and Imamura, M. (2016) 'Orthography and vernacular media: the case of Jinghpaw-Kachin', IIAS The Newsletter, 75, pp. 36–37. Available at:

<https://www.iias.asia/the-newsletter/article/orthography-vernacular-media-case-jinghpaw-kachin> (Accessed: 9 March 2025).

Lei, H., Ganjeizadeh, F., Jayachandran, P. K., & Ozcan, P. (2017). A statistical analysis of the effects of Scrum and Kanban on software development projects. *Robotics and Computer-Integrated Manufacturing*, 43, 59-67. doi: 10.1016/j.rcim.2015.12.001. Available at: <https://www.sciencedirect.com/science/article/pii/S0736584515301599> (Accessed: 9 March 2025).

Manipabha and Prakash, O. (2024) 'Linguistic Equity and Practices in the School Education System of Multilingual Myanmar', *International Journal of Creative Research Thoughts*, 12(11), pp. 446–447. Available at: <https://www.ijcrt.org/papers/IJCRT2411609.pdf> (Accessed: 9 March 2025).

Ministry of Information (2022) 'CBM holds coordination meeting on MMQR standardization for payment switch on 13 July', Ministry of Information. Available at:

<https://www.moi.gov.mm/moi%3Aeng/news/7634> (Accessed: 9 March 2025).

Ministry of Information (2023) 'CBM to launch Digital Payment Switch', *Central Bank of Myanmar*. Available at: <https://www.cbm.gov.mm/content/8009> (Accessed: 9 March 2025).

Pyidaungsu Hluttaw. (2016) 'Payment of Wages Law, 2016 (The Pyidaungsu Hluttaw Law No. 17, 2016)', *Myanmar Law Library*. (25 January 2016). Available at:

<https://myanmar-law-library.org/topics/myanmar-labour-health-and-safety-law/payment-of-wages-law-2016.html> (Accessed: 9 March 2025).

PCI Security Standards Council (n.d.) *PCI Security Standards Council*. Available at:

<https://www.pcisecuritystandards.org/> (Accessed: 9 March 2025).

PostgreSQL Global Development Group (n.d.). PostgreSQL documentation. Available at:

<https://www.postgresql.org/docs/> (Accessed: 23 March 2025).

Pyidaungsu Hluttaw. (2019) 'The Copyright Law (The Pyidaungsu Hluttaw Law No.15, 2019)', *Myanmar Law*. (24 May 2019). Available at:

<https://www.wipo.int/wipolex/en/legislation/details/20209> (Accessed: 9 March 2025).

Salem-Gervais, N. and Ja Seng (2022) 'From Fluidity to Discretization... and Fractal Recursivity? Opportunities and Challenges Underpinning the Introduction of Minority Languages in Kachin State's Government Schools (2011-2020)', ' Moussons, 39, pp. 5–40. Available at: <https://journals.openedition.org/moussons/9044?lang=en> (Accessed: 9 March 2025).

Scrum.org. (n.d.). 'What is Scrum?'. *Scrum.org Resources*. Available at:

<https://www.scrum.org/resources/what-scrum-module> (Accessed: 9 March 2025).

Toloka (n.d.) Toloka. Homepage Screenshot Available at: <https://toloka.ai/> (Accessed: 11 March 2025).

# Appendices

## Section A - Installation and Setup

### Prerequisites

- PostgreSQL (Version 17.4 Recommended)
- Python (Version 3.11 Recommended)
- Node.js (Version 22 Recommended)

### Installation Steps

**Project source code can be downloaded from one of here**

(zip version) -

[https://drive.google.com/file/d/1oYffh62S3PZHDQp-4XKRZX1LxBIVKIMI/view?usp=drive\\_link](https://drive.google.com/file/d/1oYffh62S3PZHDQp-4XKRZX1LxBIVKIMI/view?usp=drive_link)

(no zip version) -

[https://drive.google.com/drive/folders/1cuBhm-CxRK3DCfvILtnln6R0foW2yDT?usp=drive\\_link](https://drive.google.com/drive/folders/1cuBhm-CxRK3DCfvILtnln6R0foW2yDT?usp=drive_link)

#### Clone the project

- cd project\_folder
- npm install

#### Setup Backend

- cd project\_folder/backend
- python -m venv venv
- **(Make sure the venv is correctly named and it must be under backend folder)**

#### Activate Virtual Environment

(Window)

- project\_folder\backend\venv\Scripts\activate

(macOS/Linux)

- source project\_folder/backend/venv/bin/activate

## Install Dependencies

- pip install -r requirements.txt
- pip install alembic pydantic pydantic[email] pydantic\_settings python-jose passlib fastapi\_mail python-multipart pandas bcrypt fastapi psycopg2-binary uvicorn

*Note: Make sure PostgreSQL is installed before installing psycopg2-binary.*

## Database Setup

In local PostgreSQL create the database with the name “texta”.

- CREATE DATABASE texta;

After creating the database, change the Database URL in the .env file as needed.

## Run Alembic migrations in the activated venv of backend:

- alembic upgrade head

## Start the Application

- cd project\_folder
- npm run dev (this will start frontend and backend automatically)

## User Accounts

### Freelancer Registration

Freelancer accounts can be registered with email.

### Admin Credentials

Email: superadmin@email.com

Password: 123456

### QA Member Creation

QA members can be created from the QA member manage section of Admin Dashboard.

### **Adding new language to the database**

Before creating any tasks, the admin needs to add language to the database. Choosing language for task posting can be selected only from the added language from the database. This language name will be displayed to the freelancer and it is recommended to name them properly.

### **Job CSV format**

When creating job and assessment tasks, the CSV file must have a sentences column with no header.

### **Emails**

Currently,

- itsaungkhanphyo@gmail.com is used as company email (sending system notification to freelancers) and
- itsakphyo@gmail.com (receiving notification as admin) is used as an admin email.

Admin email can be changed easily in the “.env” file and changing company email in “.env” file will require credentials such as app password.

### **Notes**

- Internet connection is required to use file uploading service (e.g., uploading receipt to cloudinary).

## **Section B - Screencasting Link**

<https://youtu.be/7jabVNqLNj8>

## Section C - Test Results

<b>Test Case ID</b>	REG-02
<b>Test Area</b>	User Registration and Login
<b>Date</b>	January 7, 2025
<b>Sprint</b>	Sprint 1
<b>Test Scenario</b>	Freelancer verifies email with incorrect validation code.
<b>Expected Result</b>	Error message shows invalid verification code.
<b>Actual Result</b>	Error message shows invalid verification code.
<b>Status</b>	Pass
<b>Bug ID</b>	N/A
<b>Tester</b>	Aung Khant Phyo
<b>Notes</b>	N/A

The screenshot shows the MyanLang Platform registration page. At the top, there is a blue header bar with the platform's logo and name "MyanLang Platform" on the left, and a language selection button "English" with a UK flag icon on the right. Below the header, there are two buttons: "LOGIN" and "REGISTER", with "REGISTER" being underlined to indicate it is the active tab. A horizontal progress bar shows three steps: 1. Email Verification (with a checked blue circle), 2. Enter Verification Code (with a blue circle), and 3. Complete Registration (with a grey circle). Below the progress bar, an error message "Invalid verification code" is displayed in a red box with a small red circular icon containing a question mark. Below the message, there is a placeholder text "Enter the verification code sent to your email" and a row of five empty input fields for entering the code. At the bottom of the page, a copyright notice reads "© 2025 MyanLang Platform. All rights reserved."

Figure 28: Error message displayed when freelancer verified with incorrect code

<b>Test Case ID</b>	REG-03
<b>Test Area</b>	User Registration and Login
<b>Date</b>	January 7, 2025
<b>Sprint</b>	Sprint 1
<b>Test Scenario</b>	Freelancer verifies with correct validation code.
<b>Expected Result</b>	Reach to the next state (filling personal info page) of registration.
<b>Actual Result</b>	Reach to the next state (filling personal info page) of registration.
<b>Status</b>	Pass
<b>Bug ID</b>	N/A
<b>Tester</b>	Aung Khant Phyo
<b>Notes</b>	N/A

The screenshot shows the MyanLang Platform registration interface. At the top, there is a blue header bar with the logo 'MyanLang Platform' and a language selection button 'English'. Below the header, there are two tabs: 'LOGIN' and 'REGISTER', with 'REGISTER' being underlined, indicating the current step. There are three checked checkboxes: 'Email Verification', 'Enter Verification Code', and 'Complete Registration'. The 'Complete Registration' checkbox has a small number '3' next to it. Below these checkboxes are three input fields: 'Full Name \*', 'Year of Birth', and 'Password \*'. Under the 'Password \*' field, there is a note: 'Password must meet the following requirements:' followed by a list of requirements: 'At least 8 characters', 'At least one uppercase letter', 'At least one lowercase letter', 'At least one number', and 'At least one special character'. At the bottom of the form, there is a checkbox labeled 'I agree to the terms and conditions', a link to 'Terms and Conditions', and a link to 'Privacy Policy'.

Figure 29: Navigated to step 3 of registration when verification code is correct

<b>Test Case ID</b>	REG-04
<b>Test Area</b>	User Registration and Login
<b>Date</b>	January 7, 2025
<b>Sprint</b>	Sprint 1
<b>Test Scenario</b>	Freelancer registers without filling all the required personal information.
<b>Expected Result</b>	Message displays to fill all the required fields.
<b>Actual Result</b>	Message displays to fill all the required fields.
<b>Status</b>	Pass
<b>Bug ID</b>	N/A
<b>Tester</b>	Aung Khant Phyo
<b>Notes</b>	N/A

The screenshot shows the registration page of the MyanLang Platform. At the top, there is a blue header bar with the logo 'ML MyanLang Platform' on the left and a language selection dropdown 'English' on the right. Below the header, there are two buttons: 'LOGIN' and 'REGISTER', with 'REGISTER' being underlined to indicate it is active. Underneath these buttons, there are three checkboxes: 'Email Verification' (checked), 'Enter Verification Code' (checked), and 'Complete Registration'. The main form area contains two input fields: 'Full Name \*' and 'Password \*'. The 'Full Name' field has a red border and a dark gray placeholder box containing the text 'Please fill out this field.' The 'Password' field also has a red border. Below the password field, a note says 'Password must meet the following requirements:' followed by five bullet points: 'At least 8 characters', 'At least one uppercase letter', 'At least one lowercase letter', 'At least one number', and 'At least one special character'. At the bottom of the form, there is a checkbox labeled 'I agree to the terms and conditions', a link to 'Terms and Conditions', another link to 'Privacy Policy', and a large blue 'Register' button.

Figure 30: Message displayed when a freelancer registers without filling in personal information  
(message varies by browser type)

<b>Test Case ID</b>	REG-05
<b>Test Area</b>	User Registration and Login
<b>Date</b>	January 7, 2025
<b>Sprint</b>	Sprint 1
<b>Test Scenario</b>	Freelancer registers without agreeing terms and conditions.
<b>Expected Result</b>	Error message shows to agree to the terms and conditions.
<b>Actual Result</b>	Error message shows to agree to the terms and conditions.
<b>Status</b>	Pass
<b>Bug ID</b>	N/A
<b>Tester</b>	Aung Khant Phyo
<b>Notes</b>	N/A

The screenshot shows a registration page with the following details:

- Header:** LOGIN **REGISTER**
- Checkboxes:** Email Verification, Enter Verification Code, Complete Registration
- Error Message:** You must agree to the terms and conditions (highlighted in red)
- Form Fields:**
  - Full Name \*: Aung Khant Phyo
  - Year of Birth: 1999
  - Password \*: [REDACTED]
- Requirements:** Password must meet the following requirements:
  - At least 8 characters
  - At least one uppercase letter
  - At least one lowercase letter
  - At least one number
  - At least one special character
- Agreement:** I agree to the terms and conditions (checkbox)
- Links:** Terms and Conditions, Privacy Policy
- Buttons:** Register

Figure 31: Error message displayed when freelancer register without agreeing to terms.

<b>Test Case ID</b>	REG-06
<b>Test Area</b>	User Registration and Login
<b>Date</b>	January 7, 2025
<b>Sprint</b>	Sprint 1
<b>Test Scenario</b>	Freelancer logins to the platform.
<b>Expected Result</b>	Reach to the profile page of freelancer portal.
<b>Actual Result</b>	Reach to the profile page of freelancer portal.
<b>Status</b>	Pass
<b>Bug ID</b>	N/A
<b>Tester</b>	Aung Khant Phyo
<b>Notes</b>	N/A

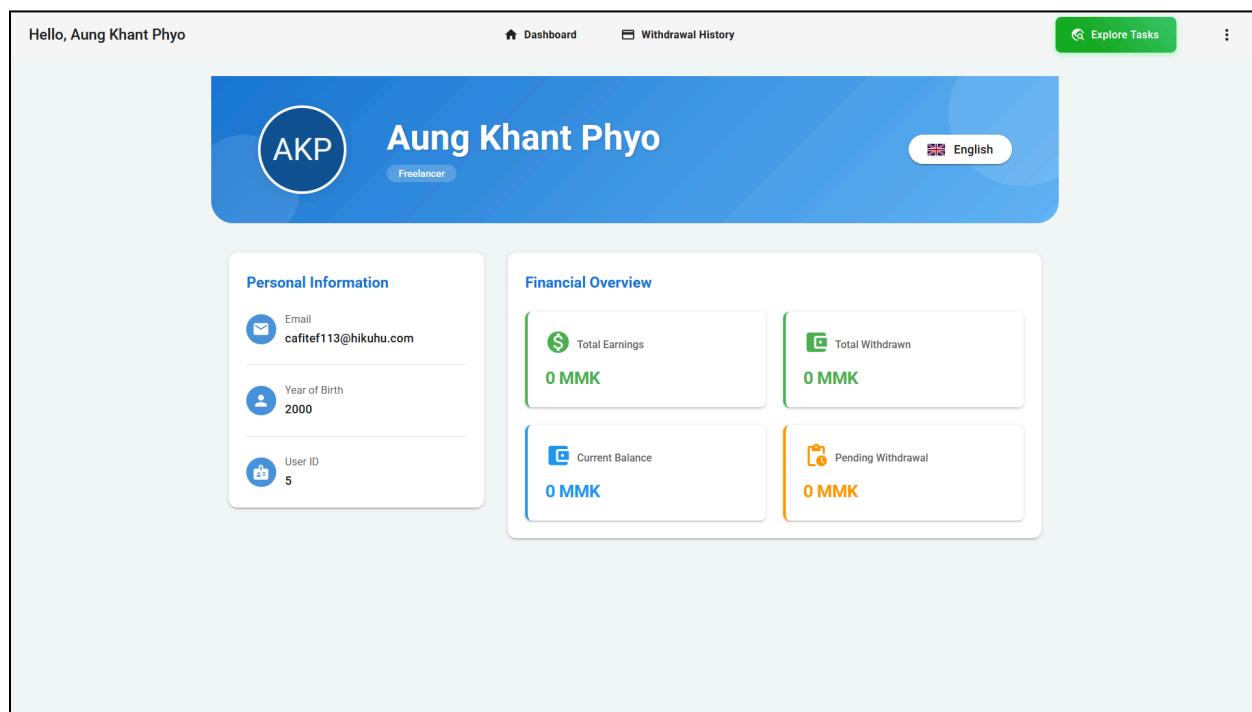


Figure 32: Freelancer successfully login to the platform and reaches the profile page.

<b>Test Case ID</b>	REG-07
<b>Test Area</b>	User Registration and Login
<b>Date</b>	January 7, 2025
<b>Sprint</b>	Sprint 1
<b>Test Scenario</b>	Freelancer logs in with invalid credentials.
<b>Expected Result</b>	Error message shows invalid credentials.
<b>Actual Result</b>	Error message shows invalid credentials.
<b>Status</b>	Pass
<b>Bug ID</b>	N/A
<b>Tester</b>	Aung Khant Phyo
<b>Notes</b>	N/A

The screenshot shows the MyanLang Platform login interface. At the top, there is a blue header bar with the logo 'ML MyanLang Platform' on the left and a language selection button 'English' on the right. Below the header, there are two buttons: 'LOGIN' (underlined) and 'REGISTER'. A red error message box contains the text '① Invalid email or password'. Below the message box are two input fields: 'Email \*' containing 'itsakphyo@gmail.com' and 'Password \*' containing a series of dots. At the bottom of the form is a large blue 'Login' button. Below the button, there is a link 'Forgot Password'. At the very bottom of the page, there is a small copyright notice: '© 2025 MyanLang Platform. All rights reserved.'

Figure 33: Error message displayed invalid credentials after freelancer login with wrong credentials

<b>Test Case ID</b>	REG-08
<b>Test Area</b>	User Registration and Login
<b>Date</b>	January 7, 2025
<b>Sprint</b>	Sprint 1
<b>Test Scenario</b>	Freelancer registers with the email of existing users.
<b>Expected Result</b>	Error message shows user already exists.
<b>Actual Result</b>	Error message shows user already exists.
<b>Status</b>	Pass
<b>Bug ID</b>	N/A
<b>Tester</b>	Aung Khant Phyo
<b>Notes</b>	N/A

The screenshot shows the registration page of the MyanLang Platform. At the top, there is a blue header bar with the logo 'ML MyanLang Platform' on the left and a language selection button 'English' on the right. Below the header, there are two buttons: 'LOGIN' and 'REGISTER', with 'REGISTER' being underlined to indicate it is the active tab. A vertical registration flow is displayed: 1 Email Verification, 2 Enter Verification Code, and 3 Complete Registration. In the 'Email \*' input field, the value 'itsakphyo@gmail.com' is entered. Below the input field, a red rectangular box contains the error message 'User already exists'. At the bottom of the form is a large blue button labeled 'Send Verification Code'.

*Figure 34: Error message displayed “User already exists” when freelancers register with the email of existing users.*

<b>Test Case ID</b>	REG-09
<b>Test Area</b>	User Registration and Login
<b>Date</b>	January 7, 2025
<b>Sprint</b>	Sprint 1
<b>Test Scenario</b>	Admin login with correct credentials.
<b>Expected Result</b>	Reach to the main dashboard of the admin portal.
<b>Actual Result</b>	Reach to the main dashboard of the admin portal.
<b>Status</b>	Pass
<b>Bug ID</b>	N/A
<b>Tester</b>	Aung Khant Phyo
<b>Notes</b>	N/A

The screenshot shows the Admin Dashboard interface. On the left, there is a sidebar with navigation links: Jobs Dashboard, Payments, QA Management, Assessment Tasks, and User Reports. The main area displays two task cards. The first card, titled 'Job Title : Customer Service Replies' (ID: 5), has a progress bar at 0/20, a time of 10 min, and a price of 100.00 MMK. Its instructions are 'Please translate the followings...'. The second card, titled 'Job Title : Myanmar Creative writing' (ID: 7), has a progress bar at 4/20, a time of 30 min, and a price of 500.00 MMK. Its instructions are 'Please write creatively'. At the bottom left, there is a 'Logout' button, and at the bottom right, a blue circular button with a '+' sign.

Figure 35: Admin successfully logs in with correct credentials and accesses the main dashboard

<b>Test Case ID</b>	REG-10
<b>Test Area</b>	User Registration and Login
<b>Date</b>	January 7, 2025
<b>Sprint</b>	Sprint 1
<b>Test Scenario</b>	QA member login with incorrect credentials.
<b>Expected Result</b>	Error message shows invalid credentials.
<b>Actual Result</b>	Error message shows invalid credentials.
<b>Status</b>	Pass
<b>Bug ID</b>	N/A
<b>Tester</b>	Aung Khant Phyo
<b>Notes</b>	N/A

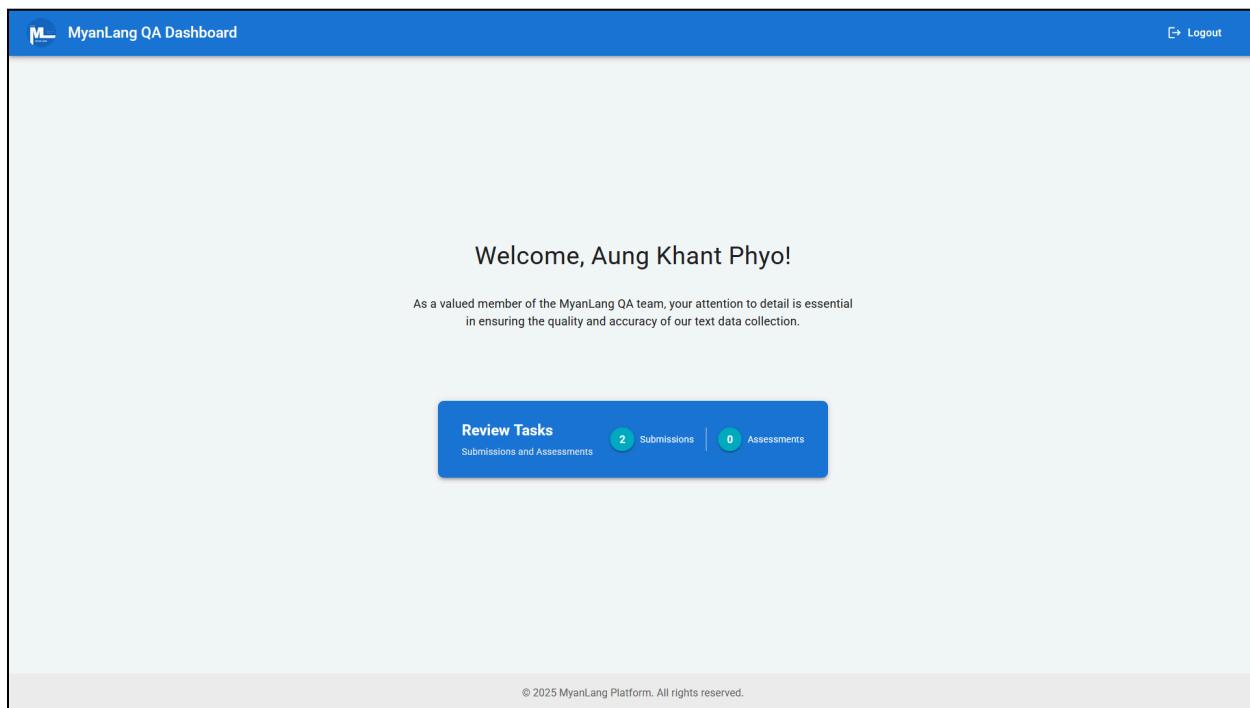


Figure 35: QA member successfully logs in with correct credentials and accesses the main page

<b>Test Case ID</b>	REG-11
<b>Test Area</b>	User Registration and Login
<b>Date</b>	January 7, 2025
<b>Sprint</b>	Sprint 1
<b>Test Scenario</b>	QA members reset the initial password (set by admin) without creating their own password.
<b>Expected Result</b>	Error message shows unable to reset.
<b>Actual Result</b>	Error message shows unable to reset.
<b>Status</b>	Pass
<b>Bug ID</b>	N/A
<b>Tester</b>	Aung Khant Phyo
<b>Notes</b>	N/A

The screenshot shows a password reset interface for the MyanLang Platform. At the top, there is a blue header bar with the platform's logo and name. Below the header, a navigation bar indicates three steps: 1. Email Verification, 2. Enter Code, and 3. Reset Password. A red error message box is displayed, stating "Failed to send reset code". Below the error message is an input field for the email address, containing "deniel@qa.com". A blue button labeled "Send Verification Code" is located below the email input field. The rest of the page is white and contains no other visible content.

Figure 37: Error message displayed when QA members reset the initial password (set by admin) without creating their own password.

<b>Test Case ID</b>	ADMIN-01
<b>Test Area</b>	Admin Functionalities
<b>Date</b>	January 7, 2025
<b>Sprint</b>	Sprint 1
<b>Test Scenario</b>	Admin adds a new language to the database.
<b>Expected Result</b>	New language was successfully added to the database with a successful toast message.
<b>Actual Result</b>	The language add dialog is behind the layer of the main dashboard.
<b>Status</b>	Fail
<b>Bug ID</b>	BUG-102
<b>Tester</b>	Aung Khant Phyo
<b>Notes</b>	Fixed by changing the zIndex of the frontend components.

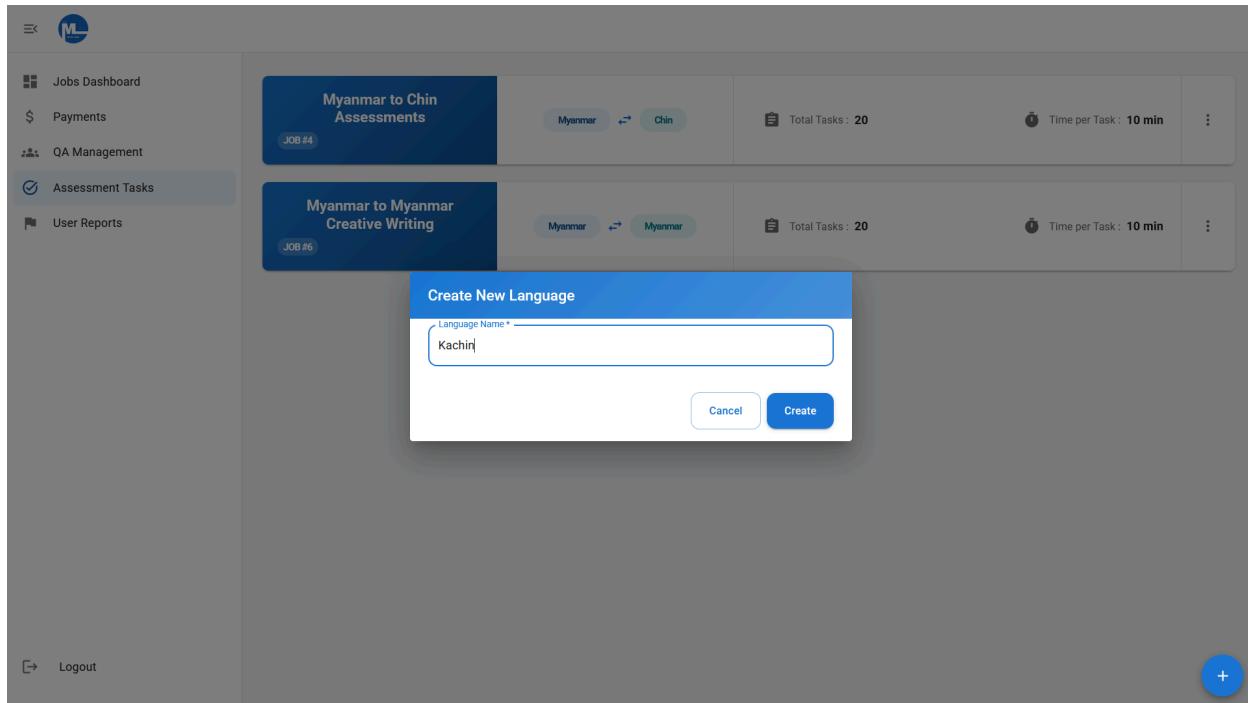


Figure 38: Admin adding new language to the database with language create dialog

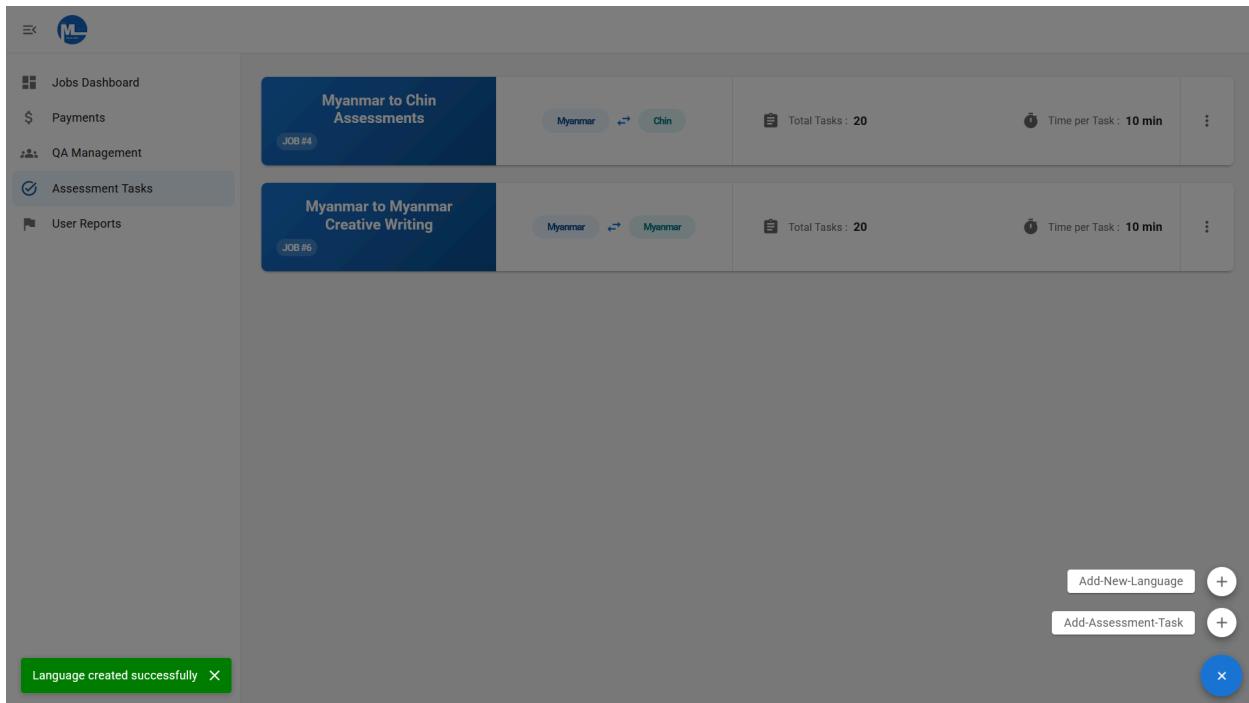


Figure 39: A success toast message displayed after admin added new language to the database

<b>Test Case ID</b>	ADMIN-02
<b>Test Area</b>	Admin Functionalities
<b>Date</b>	January 7, 2025
<b>Sprint</b>	Sprint 1
<b>Test Scenario</b>	Admin posts a collection of assessment tasks.
<b>Expected Result</b>	Assessment tasks were successfully added to the database with a successful toast message.
<b>Actual Result</b>	The assessment tasks add dialog is behind the layer of the main dashboard.
<b>Status</b>	Fail
<b>Bug ID</b>	BUG-103
<b>Tester</b>	Aung Khant Phyo
<b>Notes</b>	Fixed by changing the zIndex of the frontend components.

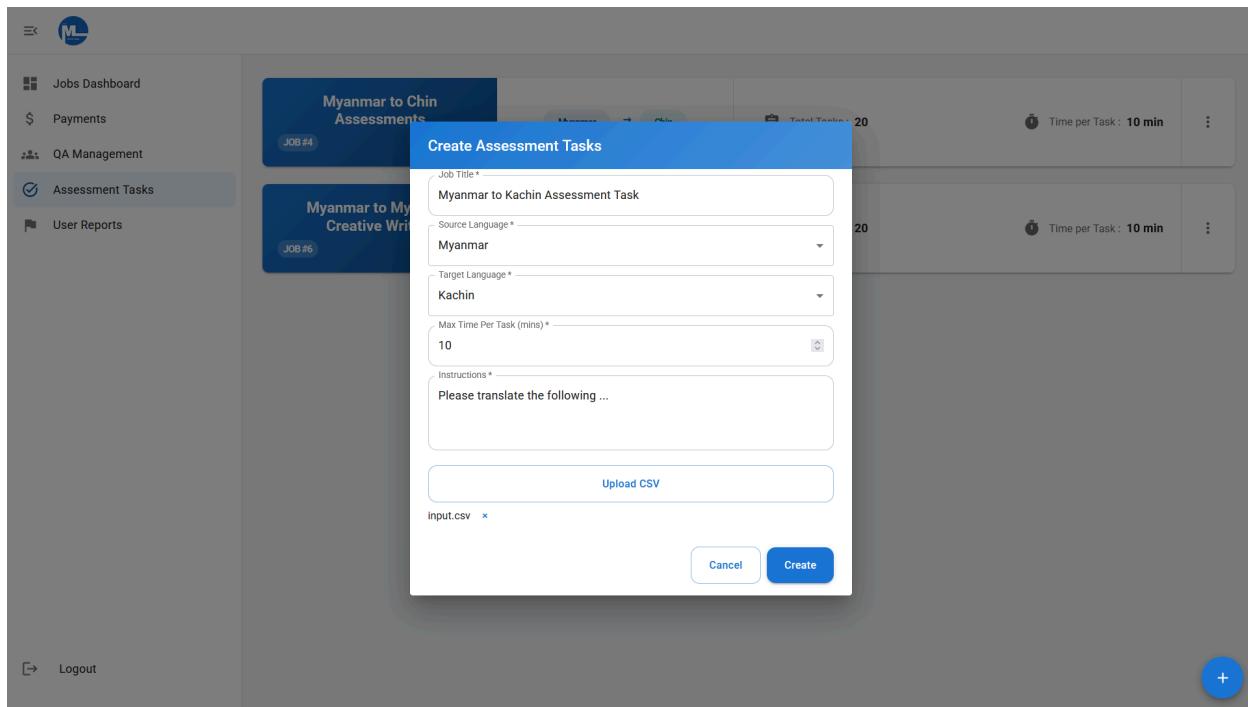


Figure 40: Admin creating assessment task with “assessment task create” dialog

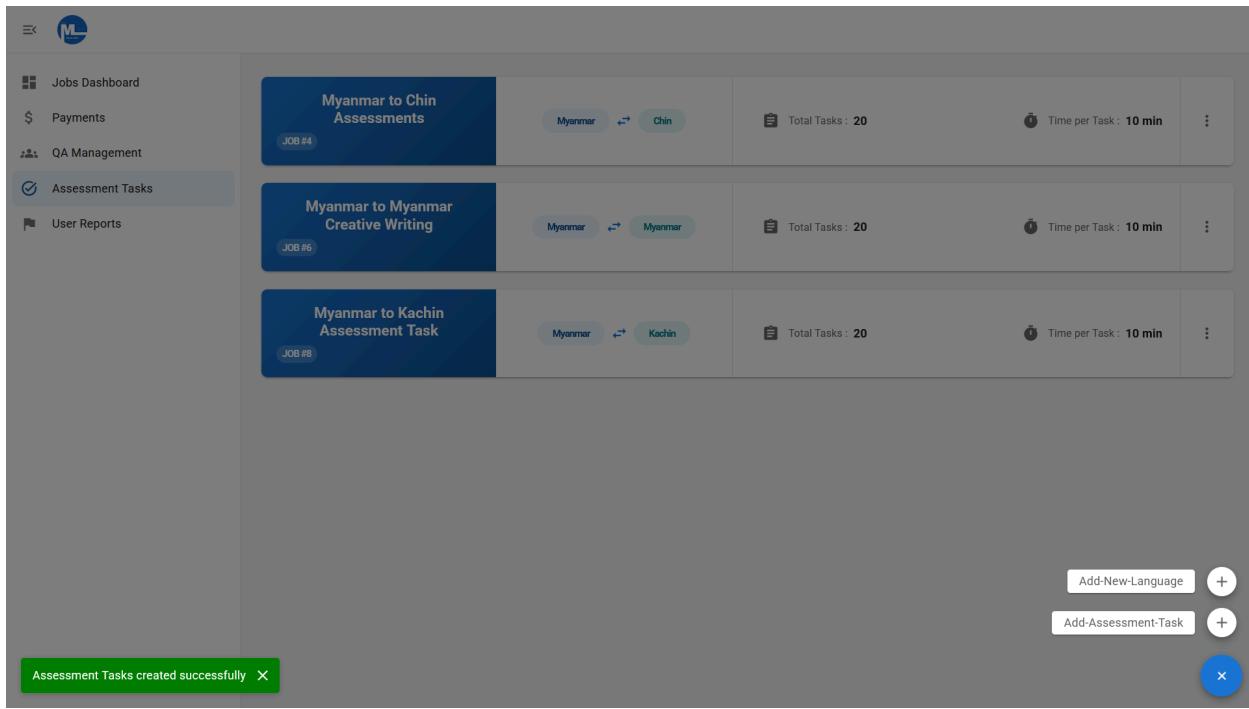


Figure 41: Success toast message displayed when admin successfully created a new assessment task

<b>Test Case ID</b>	ADMIN-03
<b>Test Area</b>	Admin Functionalities
<b>Date</b>	January 7, 2025
<b>Sprint</b>	Sprint 1
<b>Test Scenario</b>	Admin posts a collection of assessment tasks without filling all the required fields.
<b>Expected Result</b>	Error message shows to fill all the required fields.
<b>Actual Result</b>	Error message shows to fill all the required fields.
<b>Status</b>	Pass
<b>Bug ID</b>	N/A
<b>Tester</b>	Aung Khant Phyo
<b>Notes</b>	N/A

The screenshot shows a user interface for creating assessment tasks. The main area displays two job entries: 'Myanmar to Chin Assessments' (Job #4) and 'Myanmar to My Creative Writ' (Job #6). A modal window titled 'Create Assessment Tasks' is open, prompting the user to enter details for a new task. The 'Job Title' field is populated with 'Myanmar to Kachin Assessment Task'. The 'Source Language' dropdown is set to 'Myanmar', and the 'Target Language' dropdown is set to 'Kachin'. The 'Max Time Per Task (mins)' input field is empty and highlighted in red, with an error message 'Please enter a number.' displayed below it. At the bottom of the modal are 'Cancel' and 'Create' buttons.

Figure 42: Message displayed when admin creates assessment tasks without filling all required fields (message varies by browser type)

<b>Test Case ID</b>	ADMIN-04
<b>Test Area</b>	Admin Functionalities
<b>Date</b>	January 7, 2025
<b>Sprint</b>	Sprint 1
<b>Test Scenario</b>	Admin post a job (a collection of translation tasks) without filling all required fields.
<b>Expected Result</b>	Error message shows to fill all the required fields.
<b>Actual Result</b>	Job posting dialog is behind the layer of the main dashboard.
<b>Status</b>	Fail
<b>Bug ID</b>	BUG-104
<b>Tester</b>	Aung Khant Phyo
<b>Notes</b>	Fixed by changing the zIndex of the frontend components.

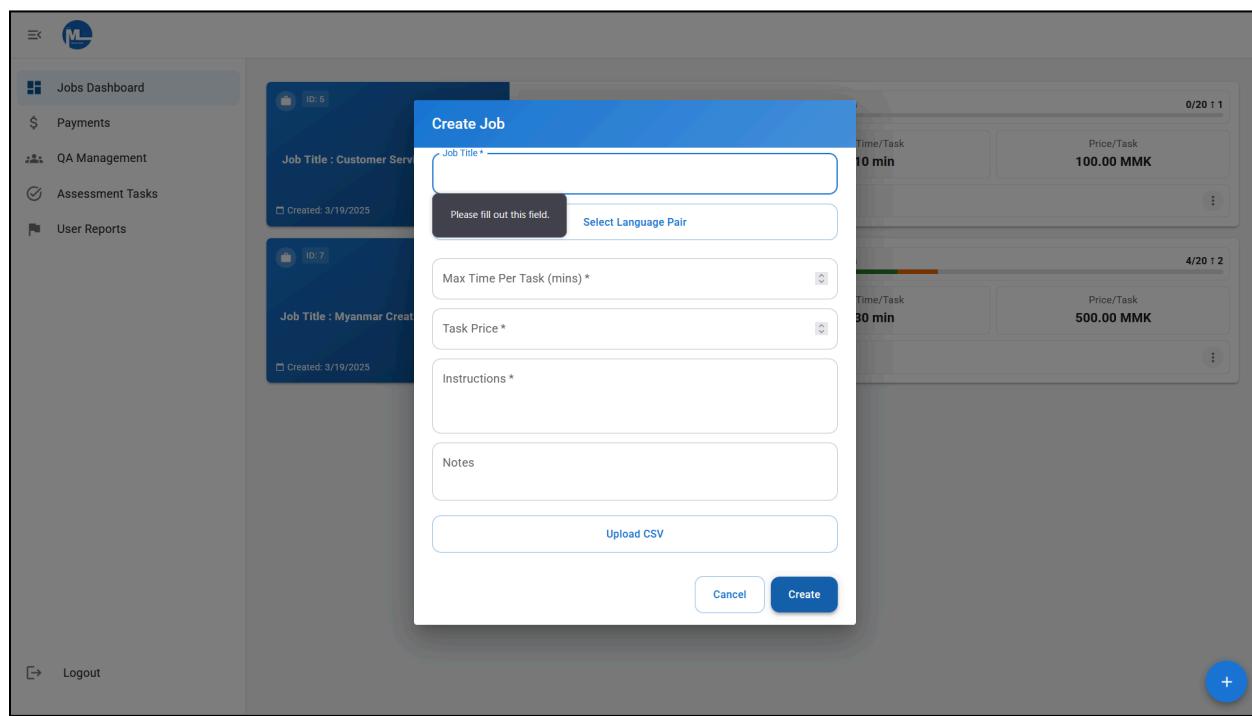


Figure 42: Message displayed when admin creates a new job (a collection of translation tasks) without filling all required fields (message varies by browser type)

<b>Test Case ID</b>	ADMIN-05
<b>Test Area</b>	Admin Functionalities
<b>Date</b>	January 8, 2025
<b>Sprint</b>	Sprint 1
<b>Test Scenario</b>	Admin posts a job with all required fields.
<b>Expected Result</b>	Job was successfully posted with a success toast message.
<b>Actual Result</b>	Job was successfully posted with a success toast message.
<b>Status</b>	Pass
<b>Bug ID</b>	N/A
<b>Tester</b>	Aung Khant Phyo
<b>Notes</b>	N/A

The screenshot shows the Admin Dashboard with a modal dialog titled "Create Job". The dialog form includes the following fields:

- Job Title \***: Daily conversations
- Max Time Per Task (mins) \***: 10
- Task Price \***: 100
- Instructions \***: Please translate the following...
- Notes**: (empty)
- Upload CSV**: (button)

Below the dialog, there is a file input field labeled "input.csv" with a delete icon. At the bottom right of the dialog are "Cancel" and "Create" buttons. The background of the dashboard shows two other job entries:

- ID: 5, Job Title: Customer Serv, Created: 3/19/2025
- ID: 7, Job Title: Myanmar Creat, Created: 3/19/2025

Figure 44: Admin posting a new job with “Create Job” dialog

The screenshot shows a user interface for managing jobs. On the left, a sidebar menu includes 'Jobs Dashboard' (selected), 'Payments', 'QA Management', 'Assessment Tasks' (with a checkmark), and 'User Reports'. The main area displays three job entries:

- Job Title : Customer Service Replies** (ID: 5)  
Created: 3/19/2025  
Myanmar → Chin  
Total Tasks: 20 | Progress: 0/20 11 | Time/Task: 10 min | Price/Task: 100.00 MMK  
Instructions: Please translate the following...
- Job Title : Myanmar Creative writing** (ID: 7)  
Created: 3/19/2025  
Myanmar → Myanmar  
Total Tasks: 20 | Progress: 4/20 12 | Time/Task: 30 min | Price/Task: 500.00 MMK  
Instructions: Please write creatively
- Job Title : Daily conversations** (ID: 10)  
Created: 3/24/2025  
Burmese → Kachin  
Total Tasks: 20 | Progress: 0/20 10 | Time/Task: 10 min | Price/Task: 100.00 MMK  
Instructions: Please translate the following...

A green toast message at the bottom left says 'Job created successfully' with a close button 'X'. On the right, there are 'Add-Job' and a '+' button.

Figure 45: A success toast message displayed after the admin posted a new job.

<b>Test Case ID</b>	ADMIN-05
<b>Test Area</b>	Admin Functionalities
<b>Date</b>	January 8, 2025
<b>Sprint</b>	Sprint 1
<b>Test Scenario</b>	Admin creates a new QA member.
<b>Expected Result</b>	A new QA member was successfully created with a success toast message.
<b>Actual Result</b>	A new QA member was successfully created with a success toast message.
<b>Status</b>	Pass
<b>Bug ID</b>	N/A
<b>Tester</b>	Aung Khant Phyo
<b>Notes</b>	N/A

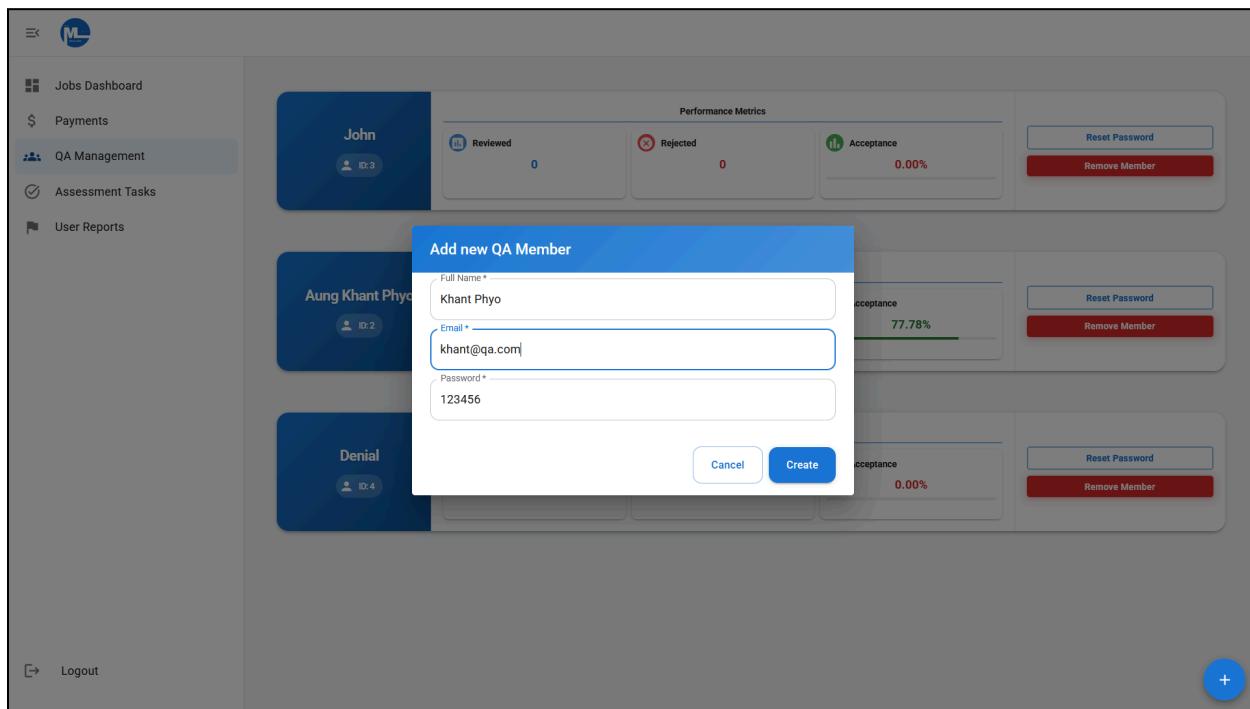


Figure 46: Admin creating new QA member with “Add new QA member” dialog



Figure 47: A success toast message displayed after the admin successfully created a new QA member

<b>Test Case ID</b>	ADMIN-05
<b>Test Area</b>	Admin Functionalities
<b>Date</b>	January 8, 2025
<b>Sprint</b>	Sprint 1
<b>Test Scenario</b>	Admin reset QA member password.
<b>Expected Result</b>	QA member password was reset successfully.
<b>Actual Result</b>	QA member password was reset successfully.
<b>Status</b>	Pass
<b>Bug ID</b>	N/A
<b>Tester</b>	Aung Khant Phyo
<b>Notes</b>	N/A

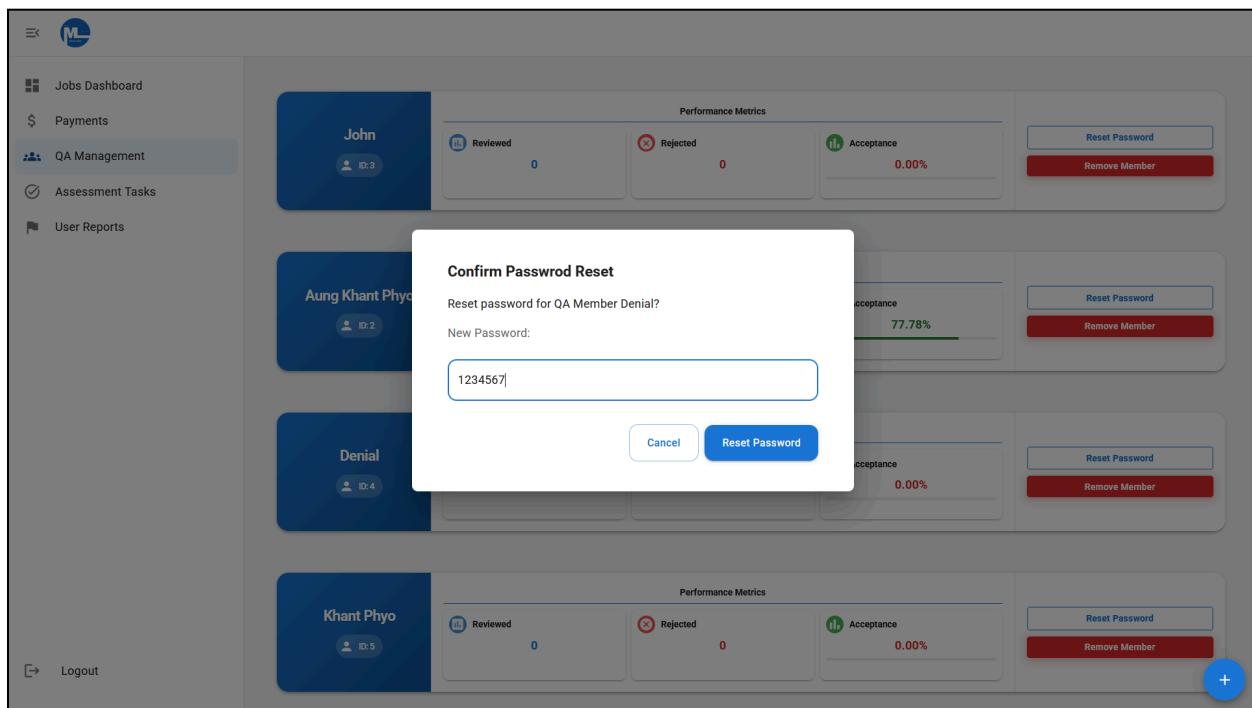


Figure 48: Admin resetting QA member password with a dialog

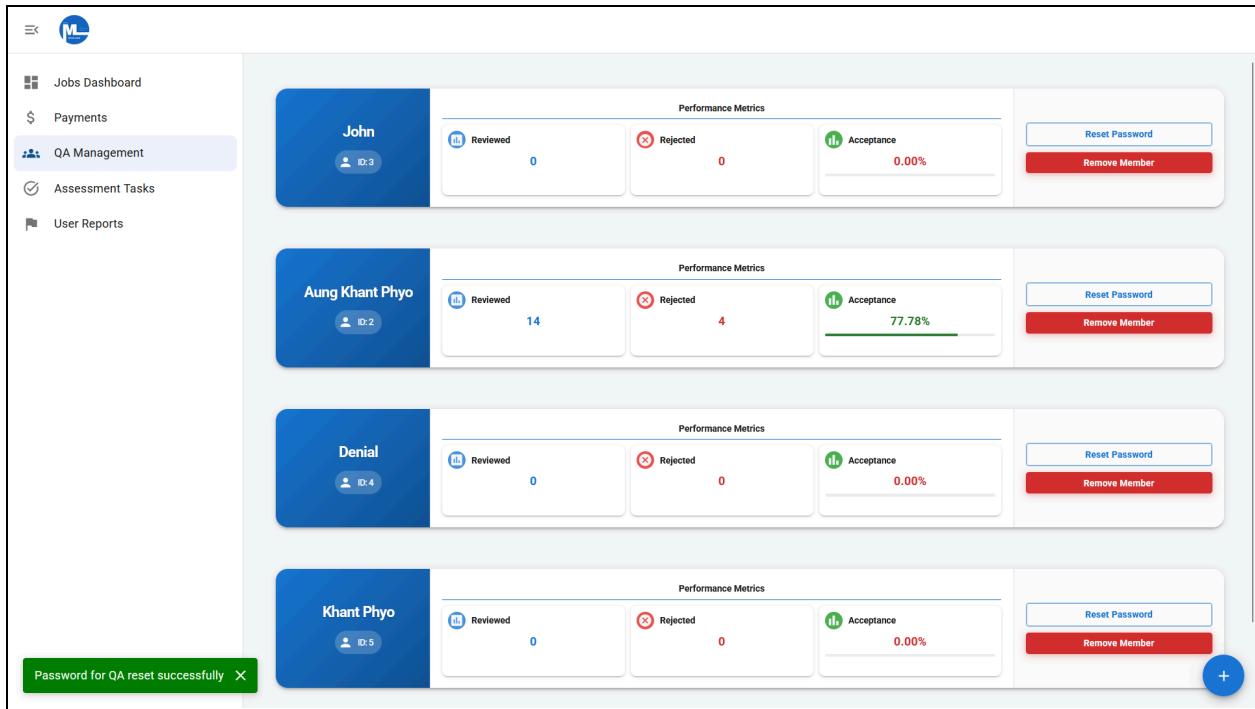


Figure 49: A success toast message displayed after admin successfully reset QA member password

<b>Test Case ID</b>	ASSESSMENT-SUBMIT-01
<b>Test Area</b>	Freelancer Assessment Task Submission
<b>Date</b>	January 8, 2025
<b>Sprint</b>	Sprint 1
<b>Test Scenario</b>	Freelancer browses tasks for language pairs for which he or she has not yet completed the assessment tasks.
<b>Expected Result</b>	Show “Take assessment tasks” button.
<b>Actual Result</b>	Show “Take assessment tasks” button.
<b>Status</b>	Pass
<b>Bug ID</b>	N/A
<b>Tester</b>	Aung Khant Phyo
<b>Notes</b>	N/A

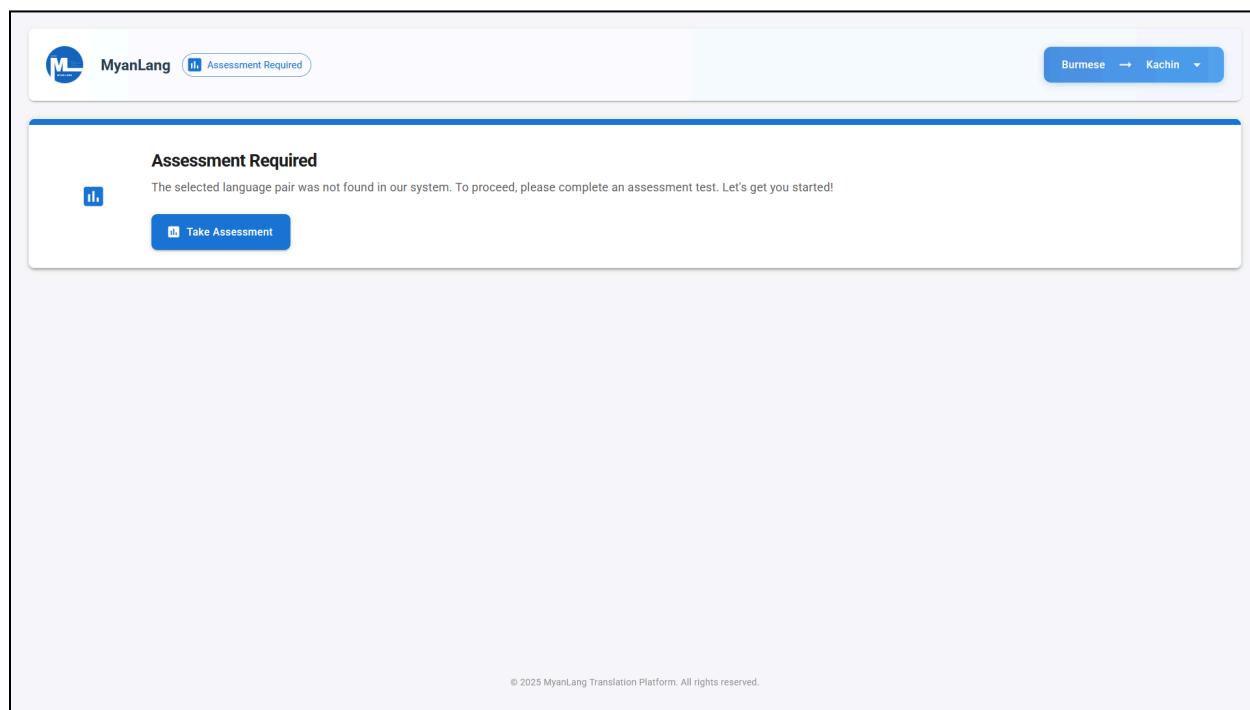
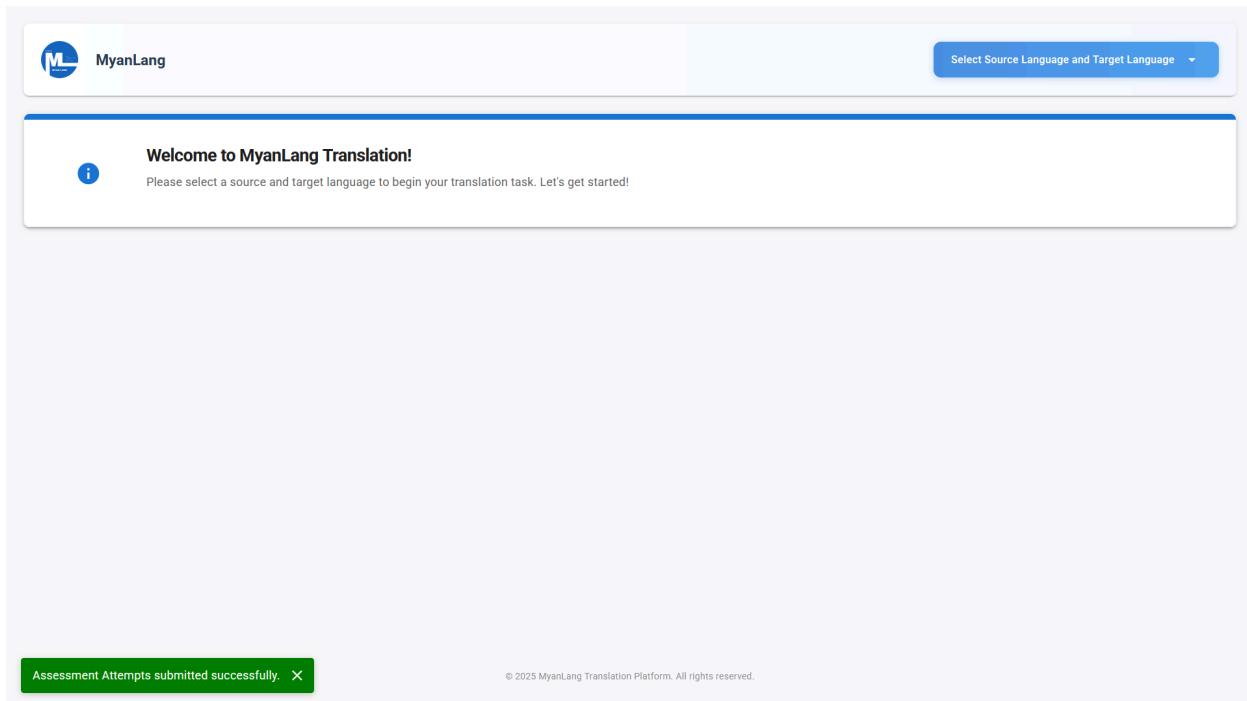


Figure 50: “Take assessment” button displayed when a freelancer browses tasks for language pairs for which he or she has not yet completed the assessment tasks.

<b>Test Case ID</b>	ASSESSMENT-SUBMIT-02
<b>Test Area</b>	Freelancer Assessment Task Submission
<b>Date</b>	January 8, 2025
<b>Sprint</b>	Sprint 1
<b>Test Scenario</b>	Freelancer submits assessment tasks.
<b>Expected Result</b>	Assessment tasks are submitted successfully with a success toast message.
<b>Actual Result</b>	Assessment tasks are submitted successfully with a success toast message.
<b>Status</b>	Pass
<b>Bug ID</b>	N/A
<b>Tester</b>	Aung Khant Phyo
<b>Notes</b>	N/A

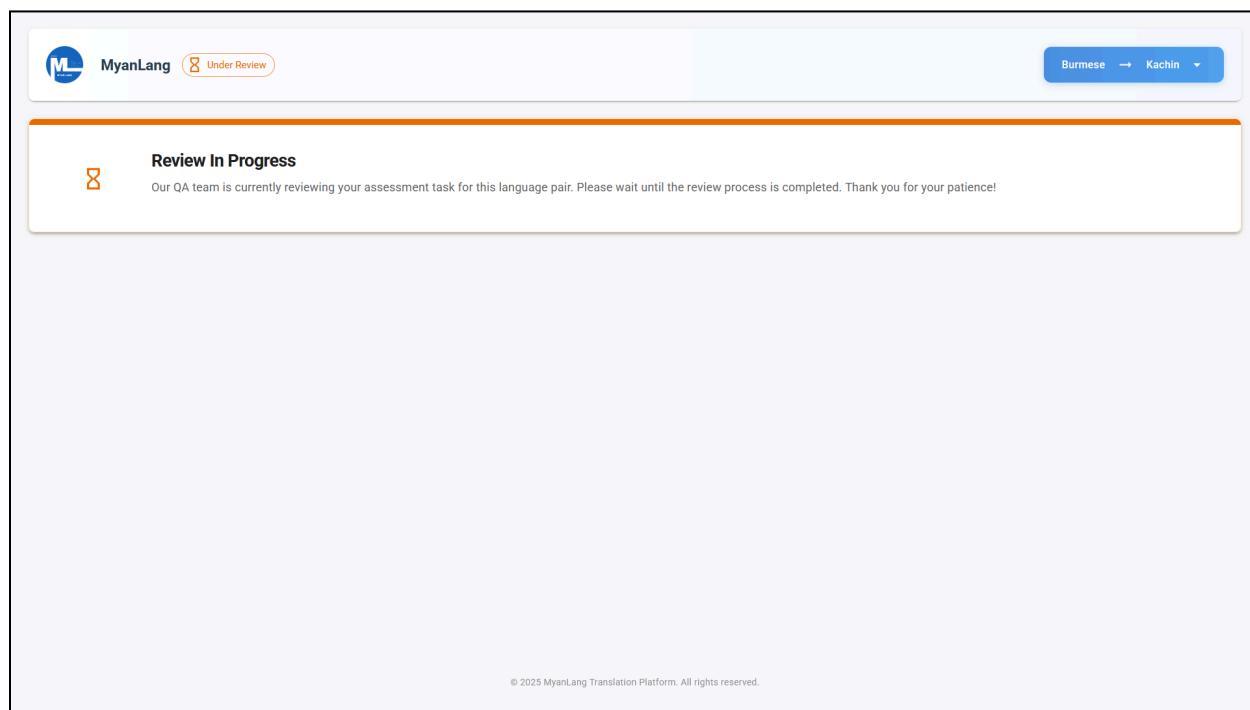
The screenshot shows a translation interface for the MyanLang platform. At the top, there's a logo for 'MyanLang' and a status message 'Assessment Required'. On the right, it shows the language pair 'Burmese → Kachin'. Below this, a timer indicates 'Time allowed for each task: 10 minutes'. The main area contains instructions: 'Please translate the following...'. Under 'Original Text (Burmese)', there is a text input field containing 'original sentence 5'. Under 'Your Translation (Kachin)', there is a larger text input field containing 'translated sentence 5'. At the bottom left is a 'Previous' button, in the center is 'Task 5 of 5', and on the right is a green 'Submit Tasks' button. A small copyright notice at the bottom states '© 2025 MyanLang Translation Platform. All rights reserved.'

Figure 51: Freelancer taking assessment tasks



*Figure 52: A success toast message displayed after freelancer successfully submitted assessment tasks*

<b>Test Case ID</b>	ASSESSMENT-SUBMIT-03
<b>Test Area</b>	Freelancer Assessment Task Submission
<b>Date</b>	January 8, 2025
<b>Sprint</b>	Sprint 1
<b>Test Scenario</b>	Freelancer browses tasks for language pairs for which the assessment tasks review process is not completed yet.
<b>Expected Result</b>	Show a message to wait until the review process completes.
<b>Actual Result</b>	Show a message to wait until the review process completes.
<b>Status</b>	Pass
<b>Bug ID</b>	N/A
<b>Tester</b>	Aung Khant Phyo
<b>Notes</b>	N/A



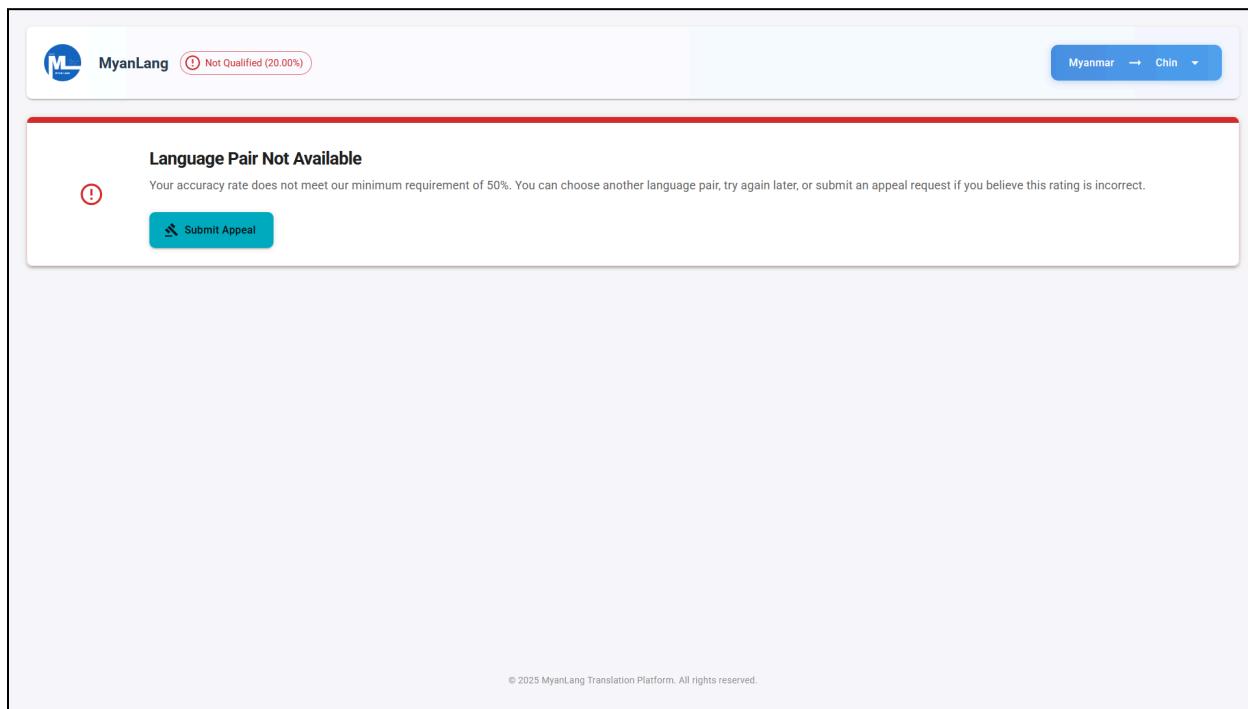
*Figure 53: “Review in progress” message displayed when a freelancer browses tasks for language pairs for which the assessment tasks review process is not completed yet.*

<b>Test Case ID</b>	ASSESSMENT-SUBMIT-04
<b>Test Area</b>	Freelancer Assessment Task Submission
<b>Date</b>	January 8, 2025
<b>Sprint</b>	Sprint 1
<b>Test Scenario</b>	Freelancer browses tasks when he or she completed assessment tasks.
<b>Expected Result</b>	Show available tasks.
<b>Actual Result</b>	Show available tasks.
<b>Status</b>	Pass
<b>Bug ID</b>	N/A
<b>Tester</b>	Aung Khant Phyo
<b>Notes</b>	N/A

The screenshot shows a web-based translation application interface. At the top, there's a header with the MyanLang logo, a green badge indicating 'Qualified (100%)', and a language selection dropdown set to 'Myanmar → Myanmar'. Below the header, the main content area displays a task card. The card includes a reward of '100 MMK' and a time left of '9:54'. It has sections for 'Instructions' (asking to translate the following), 'Original Text (Myanmar)' containing 'original sentence 4', and 'Your Translation (Chin)' with an empty input field. At the bottom of the card are three buttons: 'Reject & Close' (red), 'Submit & Close' (light blue), and 'Submit & Show Next' (blue). A small note at the bottom of the page says '© 2025 MyanLang Translation Platform. All rights reserved.'

*Figure 54: One available task displayed when a freelancer browses tasks when he or she completed assessment tasks.*

<b>Test Case ID</b>	ASSESSMENT-SUBMIT-05
<b>Test Area</b>	Freelancer Assessment Task Submission
<b>Date</b>	January 8, 2025
<b>Sprint</b>	Sprint 1
<b>Test Scenario</b>	Freelancer browses tasks when his or her assessment task is rejected.
<b>Expected Result</b>	Show a message about language proficiency is not acceptable form the company.
<b>Actual Result</b>	Show a message about language proficiency is not acceptable form the company.
<b>Status</b>	Pass
<b>Bug ID</b>	N/A
<b>Tester</b>	Aung Khant Phyo
<b>Notes</b>	N/A



*Figure 55: “Language Pair Not Available” message was displayed when a freelancer browses tasks when his or her assessment task is rejected.*

<b>Test Case ID</b>	ASSESSMENT-REVIEW-01
<b>Test Area</b>	Freelancer Assessment Task Submission
<b>Date</b>	January 8, 2025
<b>Sprint</b>	Sprint 1
<b>Test Scenario</b>	QA members review and accept the assessment tasks.
<b>Expected Result</b>	Result was stored correctly in the database and sent an email notification to the freelancer about the result. QA members' reviewed task counts are calculated correctly and displayed at the admin dashboard.
<b>Actual Result</b>	Result was stored correctly in the database and sent an email notification to the freelancer about the result. QA members' reviewed task counts are calculated correctly and displayed at the admin dashboard.
<b>Status</b>	Pass
<b>Bug ID</b>	N/A
<b>Tester</b>	Aung Khant Phyo
<b>Notes</b>	N/A

The screenshot shows a user interface for reviewing multiple translation pairs. At the top, there are navigation links for 'REVIEW ASSESSMENTS' and 'REVIEW SUBMISSIONS'. A dropdown menu indicates the current task is 'burmese → kachin'. Below this, the main area is titled 'Batch Translation Review (Burmese → Kachin)'. It displays five separate review cards, each containing an original sentence in Burmese and its corresponding translated sentence in Kachin. Each card includes 'Reject' and 'Approve' buttons. At the bottom of the page are three buttons: 'Cancel', 'Submit and Show Next', and 'Submit All Reviews'.

Original Text (Burmese)  
original sentence 1

Submitted Text (Kachin)  
translated sentence 1

Original Text (Burmese)  
original sentence 2

Submitted Text (Kachin)  
translated sentence 2

Original Text (Burmese)  
original sentence 3

Submitted Text (Kachin)  
translated sentence 3

Original Text (Burmese)  
original sentence 4

Submitted Text (Kachin)  
translated sentence 4

Original Text (Burmese)  
original sentence 5

Submitted Text (Kachin)  
translated sentence 5

Reject Approve

Reject Approve

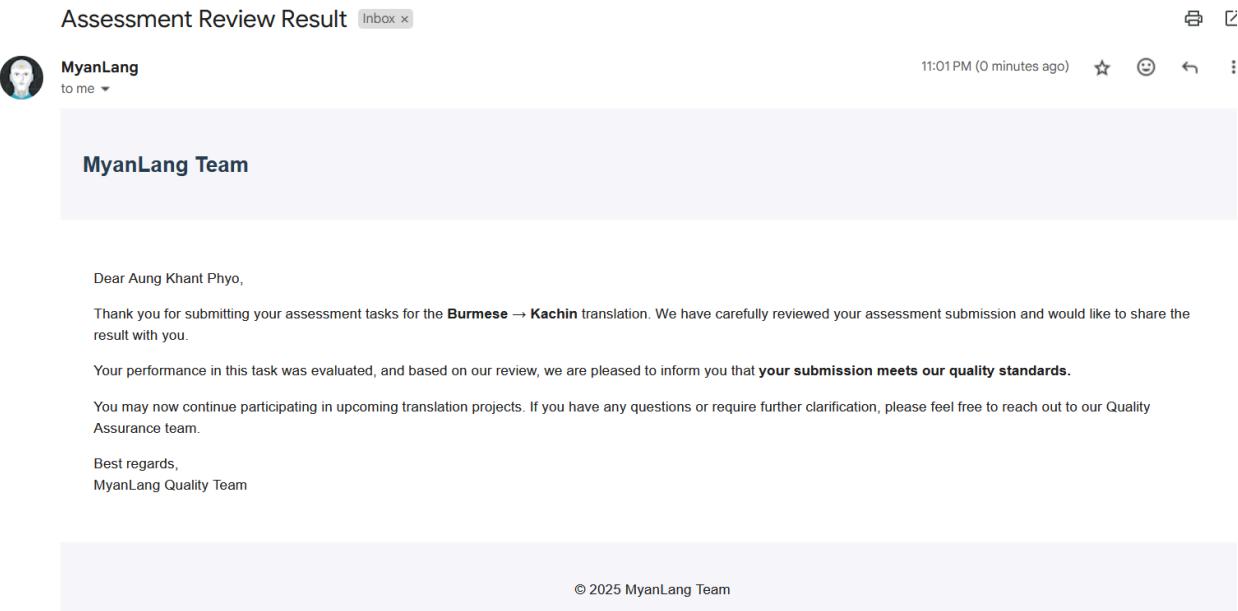
Reject Approve

Reject Approve

Reject Approve

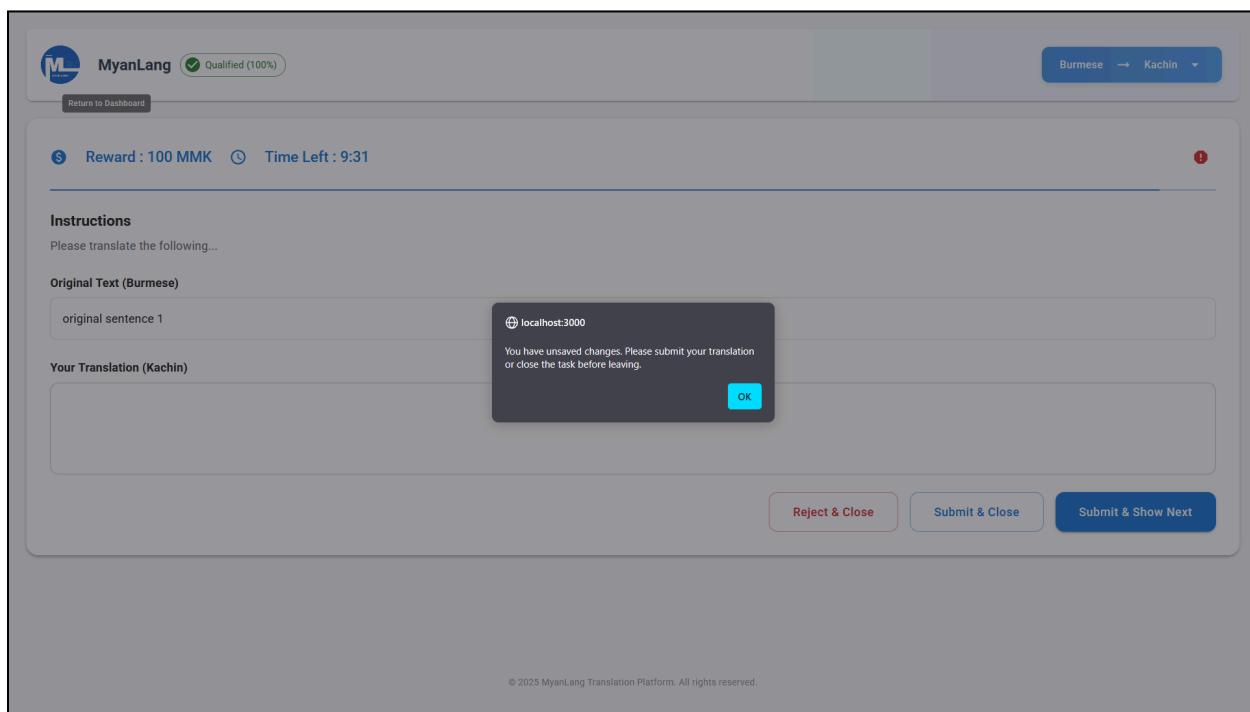
Cancel Submit and Show Next Submit All Reviews

Figure 56: QA member reviewing all the submitted assessment task of a freelancer



*Figure 57: An email about assessment task result received with the result*

<b>Test Case ID</b>	TASK-SUBMIT-02
<b>Test Area</b>	Task translation and submission, and submitting appeal message
<b>Date</b>	January 24, 2025
<b>Sprint</b>	Sprint 2
<b>Test Scenario</b>	Freelancer navigates to another page from the task translation interface without submitting.
<b>Expected Result</b>	Alert freelancers to submit the task or reject the task.
<b>Actual Result</b>	Alert freelancers to submit the task or reject the task.
<b>Status</b>	Pass
<b>Bug ID</b>	N/A
<b>Tester</b>	Aung Khant Phyo
<b>Notes</b>	N/A



*Figure 58: An alert displayed to freelancers when he or she tries to navigate to another page from the task translation interface without submitting.*

<b>Test Case ID</b>	TASK-SUBMIT-03
<b>Test Area</b>	Task translation and submission, and submitting appeal message
<b>Date</b>	January 24, 2025
<b>Sprint</b>	Sprint 2
<b>Test Scenario</b>	Freelancer submit task with submit and show next button.
<b>Expected Result</b>	Task was submitted successfully with a success toast message and the next available task was displayed to the freelancer.
<b>Actual Result</b>	Task was submitted successfully with a success toast message and the next available task was displayed to the freelancer.
<b>Status</b>	Pass
<b>Bug ID</b>	N/A
<b>Tester</b>	Aung Khant Phyo
<b>Notes</b>	N/A

The screenshot shows a translation task interface. At the top, it displays the user's profile: 'MyanLang' with a green checkmark indicating 'Qualified (100%)'. It also shows the language pair 'Burmeese → Kachin'. Below this, the task details are shown: 'Reward : 100 MMK' and 'Time Left : 9:58'. A red warning icon is visible in the top right corner. The main area contains instructions: 'Please translate the following...' and two text input fields: 'Original Text (Burmeese)' containing 'original sentence 2' and 'Your Translation (Kachin)'. At the bottom right are three buttons: 'Reject & Close' (red), 'Submit & Close' (light blue), and 'Submit & Show Next' (dark blue). A green toast message at the bottom left says 'Translation submitted successfully.' with a close button 'X'. The footer notes '© 2025 MyanLang Translation Platform. All rights reserved.'

*Figure 59: A success task submit toast message displayed and a new available task was displayed to freelancer*

<b>Test Case ID</b>	TASK-SUBMIT-04
<b>Test Area</b>	Task translation and submission, and submitting appeal message
<b>Date</b>	January 24, 2025
<b>Sprint</b>	Sprint 2
<b>Test Scenario</b>	Freelancers select language pairs that do not have any available tasks.
<b>Expected Result</b>	Show no available task message with retry button.
<b>Actual Result</b>	Show no available task message with retry button.
<b>Status</b>	Pass
<b>Bug ID</b>	N/A
<b>Tester</b>	Aung Khant Phyo
<b>Notes</b>	N/A

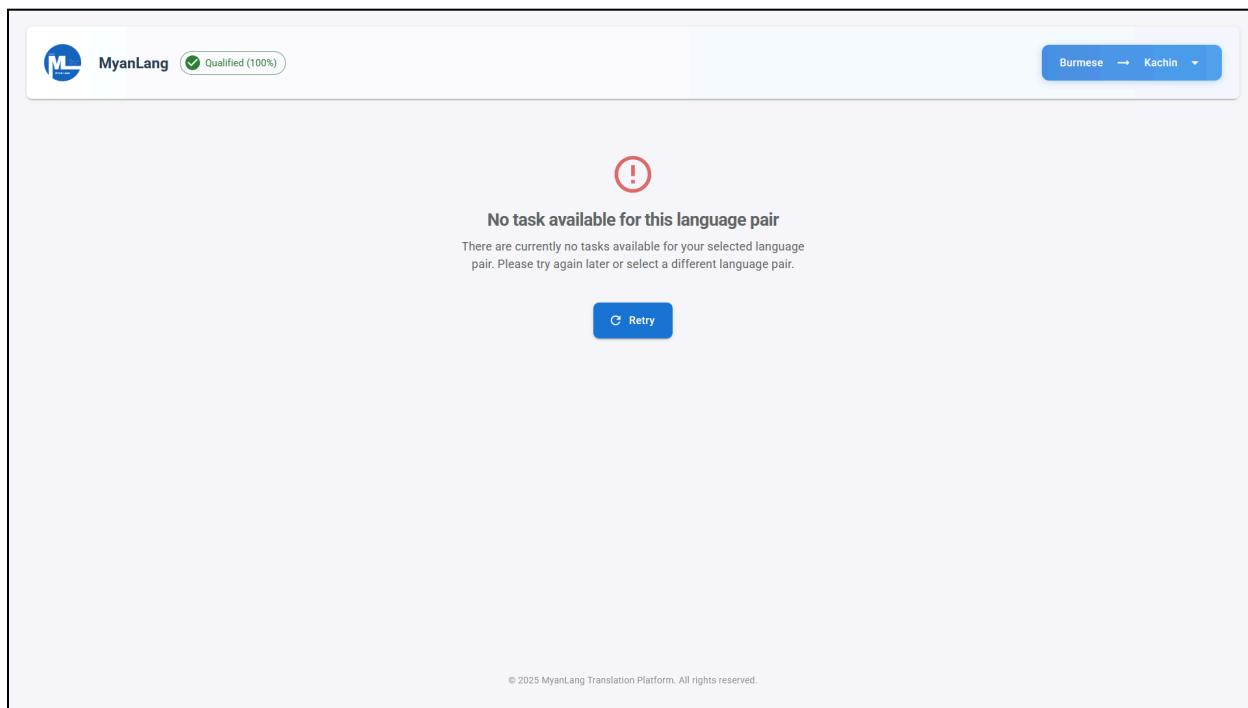
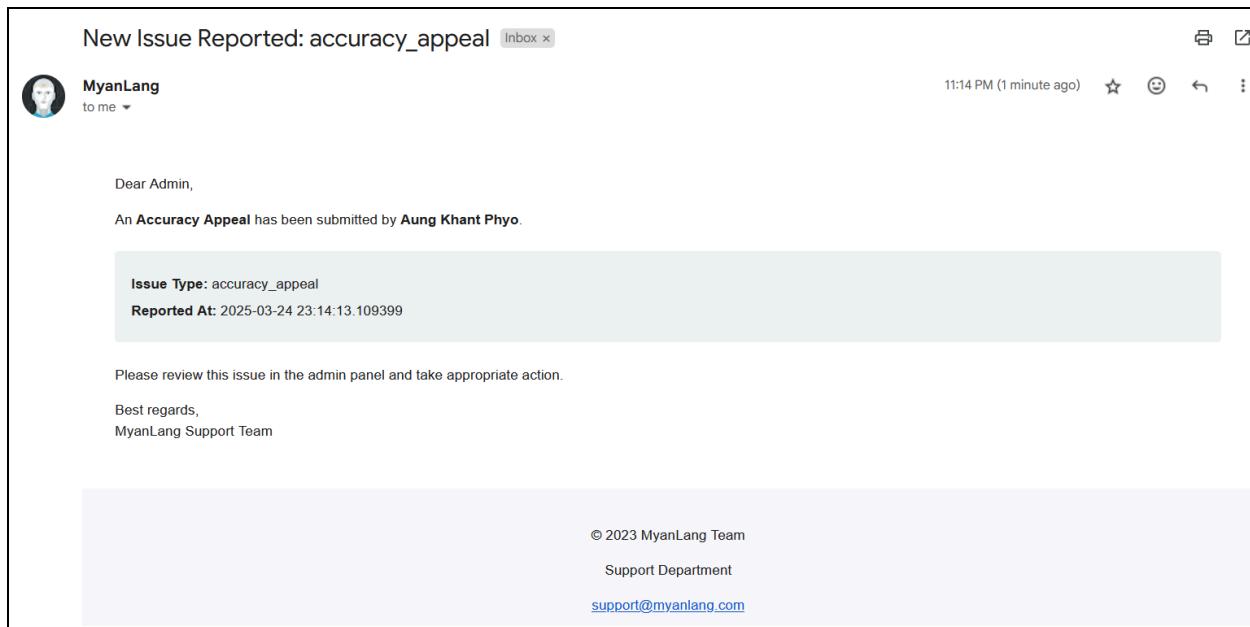


Figure 60: “No task available” message displayed to freelancers when selecting language pairs that do not have any available tasks.

<b>Test Case ID</b>	TASK-SUBMIT-04
<b>Test Area</b>	Task translation and submission, and submitting appeal message
<b>Date</b>	January 24, 2025
<b>Sprint</b>	Sprint 2
<b>Test Scenario</b>	Freelancers submit appeal messages to admin for language pairs that accuracy rate do not meet the requirement.
<b>Expected Result</b>	Appeal message was successfully sent to the admin with email notification.
<b>Actual Result</b>	Appeal message was successfully sent to the admin with email notification.
<b>Status</b>	Pass
<b>Bug ID</b>	N/A
<b>Tester</b>	Aung Khant Phyo
<b>Notes</b>	N/A

The screenshot shows a web application interface for 'MyanLang'. At the top, there's a navigation bar with the logo 'MyanLang', a status indicator 'Not Qualified (20.00%)', and a language pair selector 'Myanmar → Chin'. Below the header, a message box displays 'Language Pair Not Available' with a note about accuracy and a 'Submit Appeal' button. A larger modal window titled 'APPEAL REQUEST' is overlaid. It contains an 'Appeal Message' field with the text: 'I believe my translation accuracy was evaluated too low—could you please review it for possible reconsideration?'. There's also a section for 'Additional Documentations' with a file upload area showing a recent upload. At the bottom of the modal are 'Cancel' and 'Submit Appeal' buttons. The footer of the page includes a copyright notice: '© 2025 MyanLang Translation Platform. All rights reserved.'

*Figure 61: Freelancer submitting appeal message to admin about accuracy was evaluated too low*



*Figure 62: Admin receive notification about accuracy appeal*

<b>Test Case ID</b>	TASK-REVIEW-01
<b>Test Area</b>	Task review process of QA member portal
<b>Date</b>	January 25, 2025
<b>Sprint</b>	Sprint 2
<b>Test Scenario</b>	QA members review and accept the tasks.
<b>Expected Result</b>	Task status of the database changed to “CLOSE” and reward was added to freelancer balance.
<b>Actual Result</b>	Task status of the database changed and reward was added to freelancer balance.
<b>Status</b>	Pass
<b>Bug ID</b>	N/A
<b>Tester</b>	Aung Khant Phyo
<b>Notes</b>	N/A

The screenshot shows a user profile for 'Aung Khant Phyo' (Freelancer). The top navigation bar includes 'Hello, Aung Khant Phyo', 'Dashboard', 'Withdrawal History', 'Explore Tasks', and a settings icon. The main area displays 'Personal Information' (Email: itsakphyo@gmail.com, Year of Birth: 2005, User ID: 1) and 'Financial Overview' (Total Earnings: 8666 MMK, Total Withdrawn: 666 MMK, Current Balance: 7000 MMK, Pending Withdrawal: 1000 MMK).

Figure 63: Freelancer balance is 8666 Myanmar Kyat before QA member review translated task

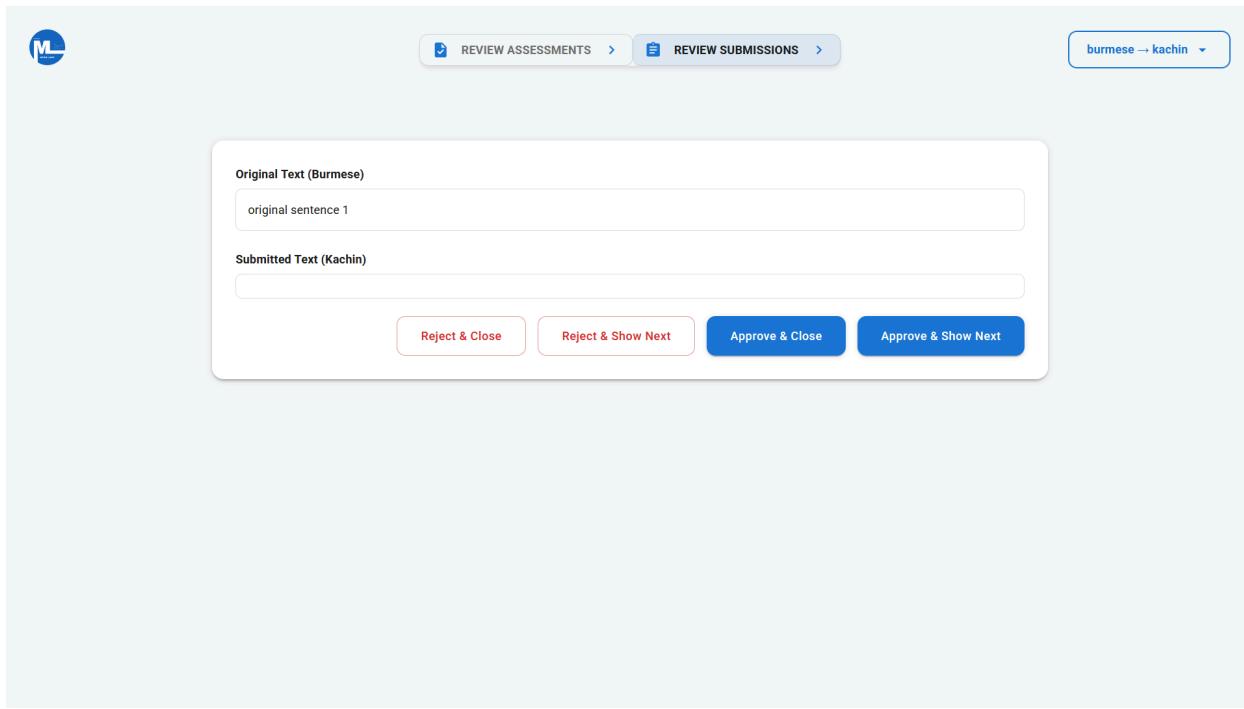


Figure 64: QA member reviewing submitted task

A screenshot of a freelancer's profile page. The top header includes 'Hello, Aung Khant Phyo', 'Dashboard', 'Withdrawal History', 'Explore Tasks', and a language switcher for 'English'. The profile card for 'Aung Khant Phyo' (Freelancer) features a circular icon with 'AKP'. The 'Personal Information' section lists an email ('itsakphyo@gmail.com'), year of birth ('2005'), and user ID ('1'). The 'Financial Overview' section displays the following data:

Total Earnings	Total Withdrawn
8866 MMK	666 MMK

Current Balance	Pending Withdrawal
7200 MMK	1000 MMK

Figure 65: Freelancer balance is 8866 Myanmar Kyat (added 200 MMK reward) after QA member accept the translated task

<b>Test Case ID</b>	TASK-REVIEW-02
<b>Test Area</b>	Task review process of QA member portal
<b>Date</b>	January 25, 2025
<b>Sprint</b>	Sprint 2
<b>Test Scenario</b>	QA members review and reject the tasks.
<b>Expected Result</b>	Task status of the database changed to “OPEN” and freelancer language pair accuracy was updated.
<b>Actual Result</b>	Task status of the database changed to “OPEN” and freelancer language pair accuracy was updated.
<b>Status</b>	Pass
<b>Bug ID</b>	N/A
<b>Tester</b>	Aung Khant Phyo
<b>Notes</b>	N/A

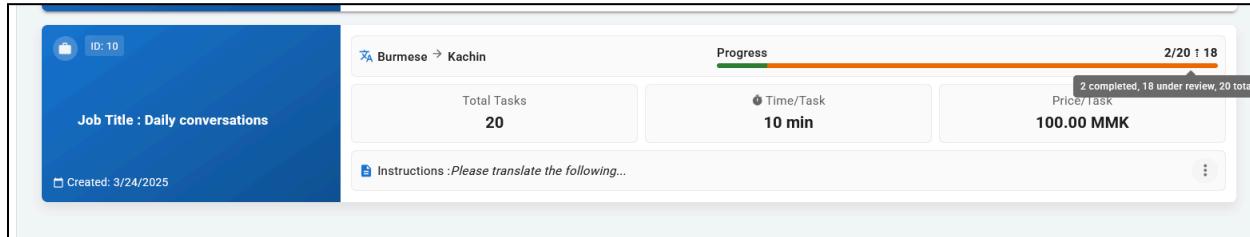


Figure 66: Before QA members reviewing the submitted task, there are 18 under review tasks at job ID 10.

The screenshot shows a user interface for reviewing language submissions. At the top, there are navigation links: 'REVIEW ASSESSMENTS' and 'REVIEW SUBMISSIONS'. A dropdown menu indicates the translation pair: 'burmese → kachin'. The main area contains two text input fields: 'Original Text (Burmese)' containing 'original sentence 12' and 'Submitted Text (Kachin)' which is currently empty. Below these are four action buttons: 'Reject & Close' (red), 'Reject & Show Next' (red), 'Approve & Close' (blue), and 'Approve & Show Next' (blue). A green success message at the bottom left states 'Reviewed Task submitted successfully'.

Figure 67: QA member rejecting all tasks from review interface

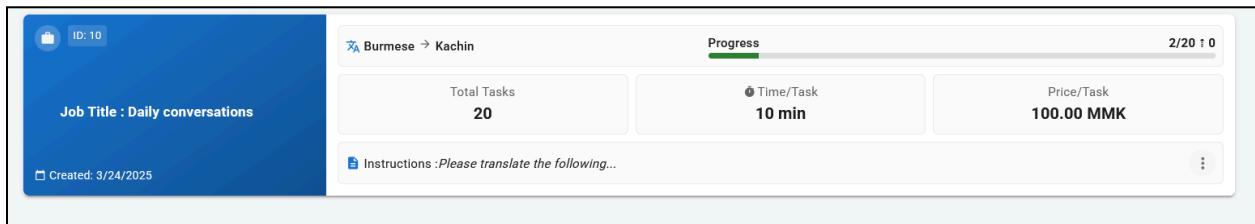


Figure 68: The “under\_review” 18 task are now open after QA member rejected all the submitted 18 tasks

<b>Test Case ID</b>	PAY-01
<b>Test Area</b>	Payment System Functionalities
<b>Date</b>	January 25, 2025
<b>Sprint</b>	Sprint 2
<b>Test Scenario</b>	Freelancer requests payment without having enough available balance
<b>Expected Result</b>	Show error message about not enough available balance
<b>Actual Result</b>	Error message does not show instantly at the dialog and the backend responses about unsuccessful.
<b>Status</b>	Fail
<b>Bug ID</b>	BUG-104
<b>Tester</b>	Aung Khant Phyo
<b>Notes</b>	Fixed after adding checking available balance logic at frontend.

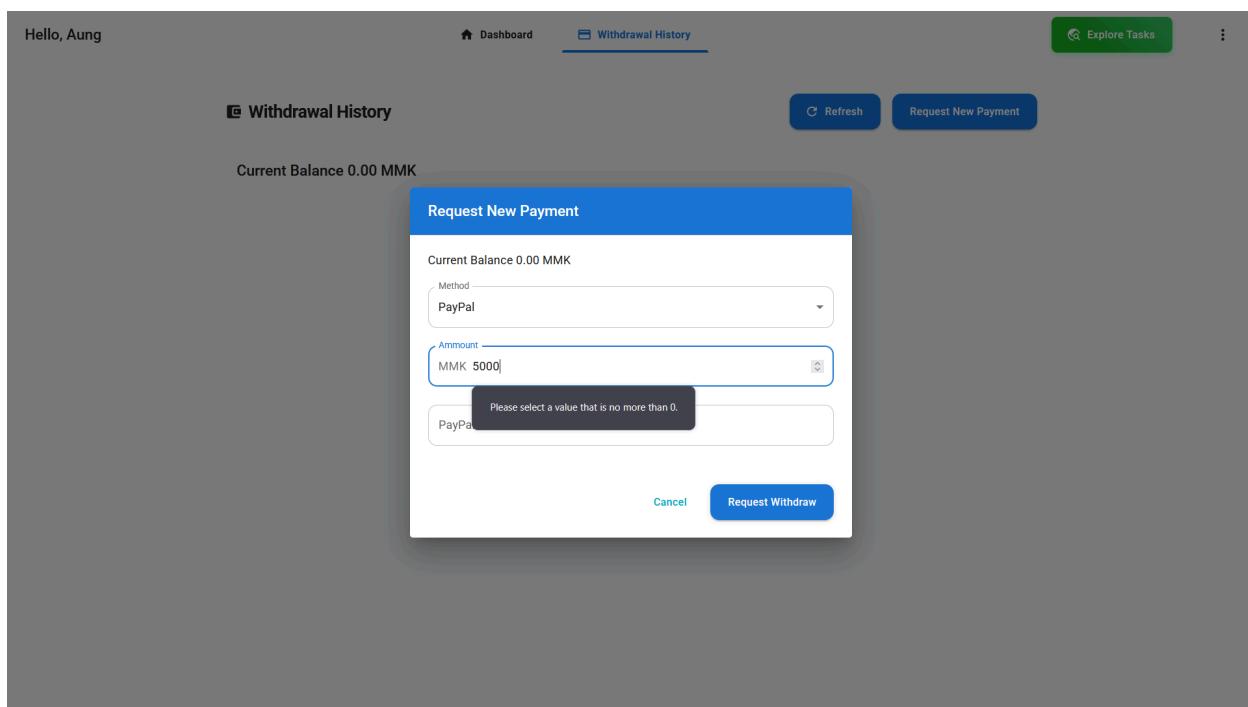


Figure 69: Error message displayed when entering an amount that exceeds the available balance.

<b>Test Case ID</b>	PAY-02
<b>Test Area</b>	Payment System Functionalities
<b>Date</b>	January 25, 2025
<b>Sprint</b>	Sprint 2
<b>Test Scenario</b>	Freelancer requests payment with paypal.
<b>Expected Result</b>	Payment request was successfully sent with a success toast message; admin received email notification about the payment request.
<b>Actual Result</b>	Payment request was successfully sent with a success toast message; admin received email notification about the payment request.
<b>Status</b>	Pass
<b>Bug ID</b>	N/A
<b>Tester</b>	Aung Khant Phyo
<b>Notes</b>	N/A

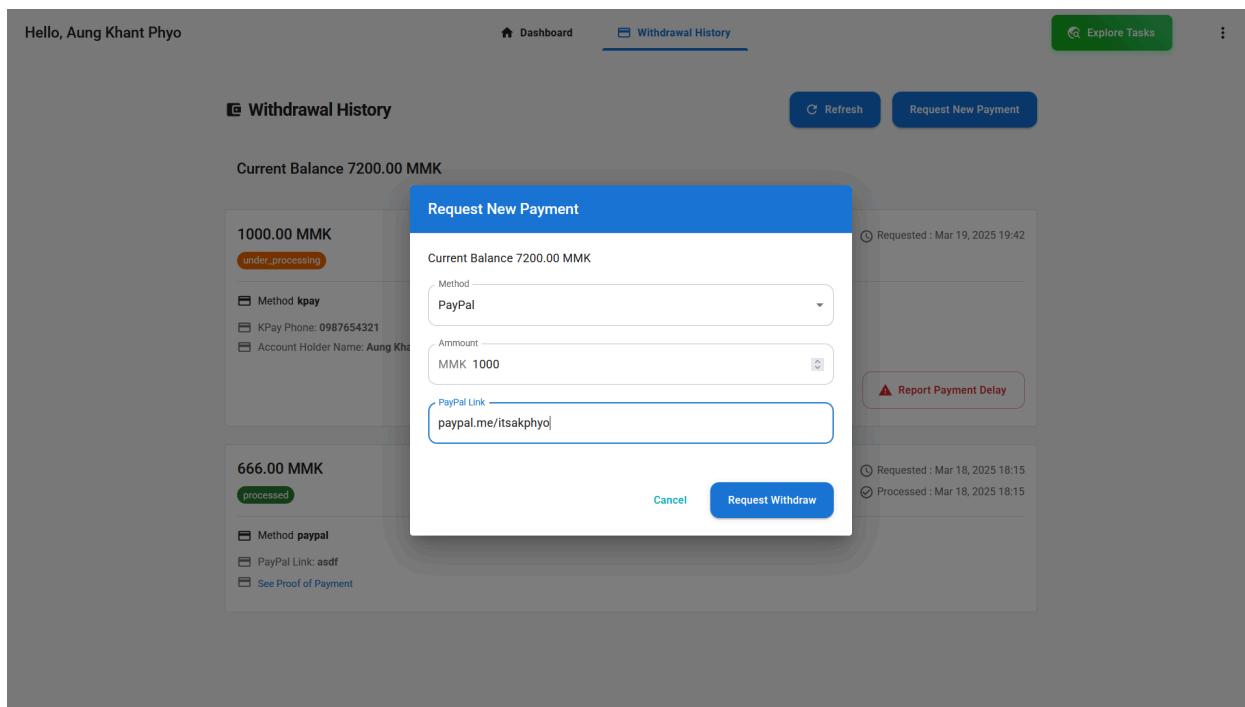


Figure 70: Freelancer requesting payment with paypal.



MyanLang  
to me ▾

11:37 PM (2 minutes ago)

Dear Admin,

A new payment request has been submitted by **Aung Khant Phy**.

**Payment Details:**

- **Freelancer Name:** Aung Khant Phy
- **Payment Method:** paypal
- **Amount:** \$1000.0

Please review the request and proceed with the payment.

Best regards,  
MyanLang Finance Team

© 2025 MyanLang Team

Finance Department | [support@myanlang.com](mailto:support@myanlang.com)

*Figure 71: Email notification was sent to admin about the payment request*

<b>Test Case ID</b>	PAY-03
<b>Test Area</b>	Payment System Functionalities
<b>Date</b>	January 25, 2025
<b>Sprint</b>	Sprint 2
<b>Test Scenario</b>	Freelancer requests payment with Kpay.
<b>Expected Result</b>	Payment request was successfully sent with a success toast message; admin received email notification about the payment request.
<b>Actual Result</b>	Payment request was successfully sent with a success toast message; admin received email notification about the payment request.
<b>Status</b>	Pass
<b>Bug ID</b>	N/A
<b>Tester</b>	Aung Khant Phyo
<b>Notes</b>	N/A

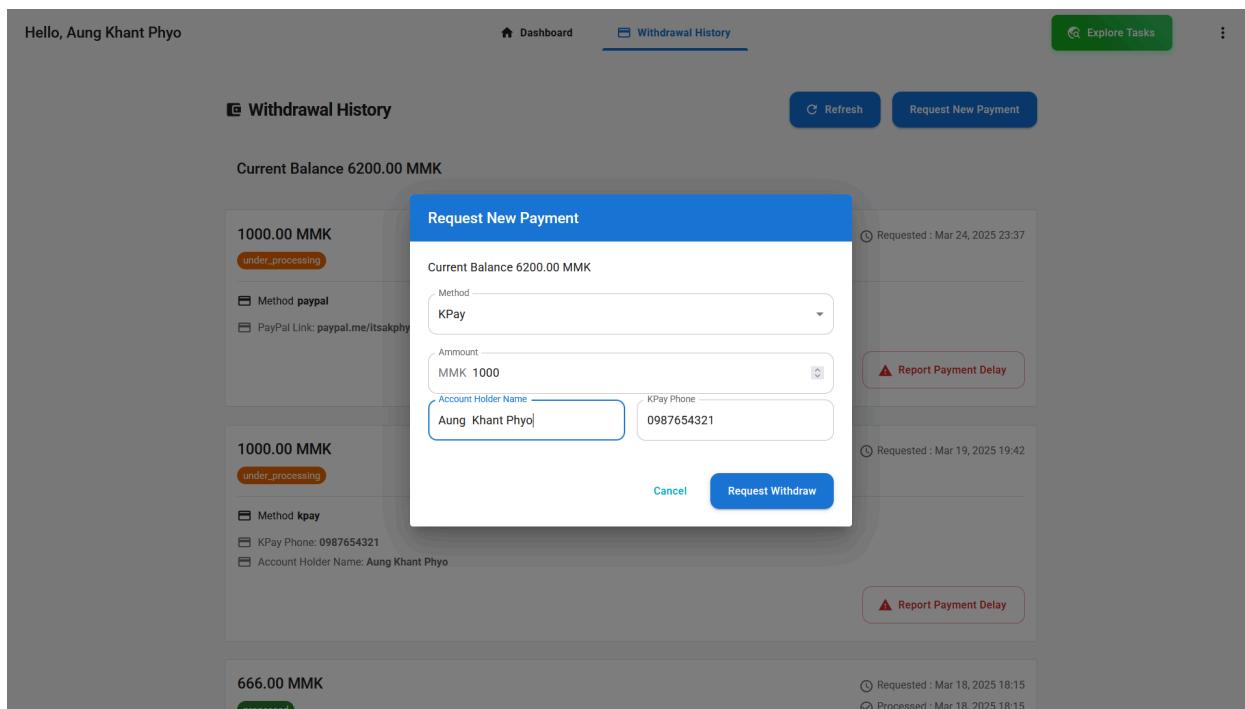


Figure 72: Freelancer requesting payment with KPay.



*Figure 73: Email notification was sent to admin about the payment request*

<b>Test Case ID</b>	PAY-04
<b>Test Area</b>	Payment System Functionalities
<b>Date</b>	January 25, 2025
<b>Sprint</b>	Sprint 2
<b>Test Scenario</b>	Freelancer requests payment with WavePay.
<b>Expected Result</b>	Payment request was successfully sent with a success toast message; admin received email notification about the payment request.
<b>Actual Result</b>	Payment request was successfully sent with a success toast message; admin received email notification about the payment request.
<b>Status</b>	Pass
<b>Bug ID</b>	N/A
<b>Tester</b>	Aung Khant Phyo
<b>Notes</b>	N/A

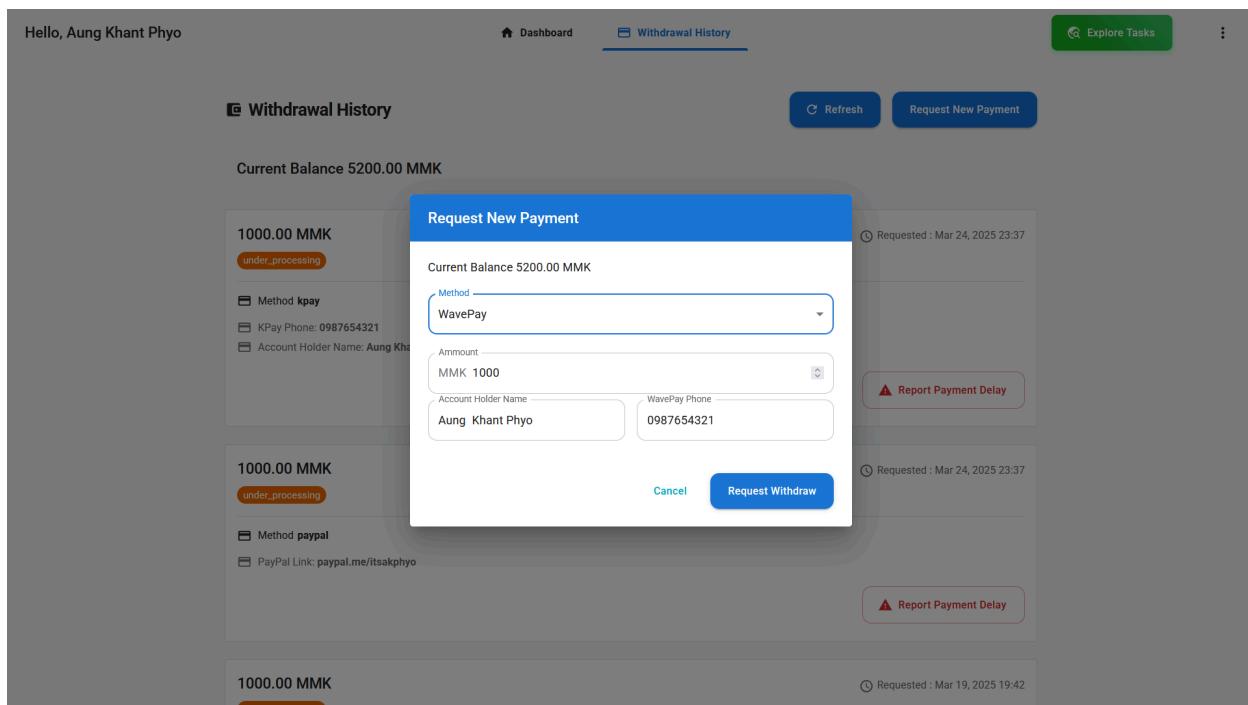


Figure 74: Freelancer requesting payment with WavePay



MyanLang  
to me ▾

11:42 PM (0 minutes ago)



Dear Admin,

A new payment request has been submitted by **Aung Khant Phyo**.

**Payment Details:**

- **Freelancer Name:** Aung Khant Phyo
- **Payment Method:** wavepay
- **Amount:** \$1000.0

Please review the request and proceed with the payment.

Best regards,  
MyanLang Finance Team

\*\*\*

*Figure 75: Email notification was sent to admin about the payment request*

<b>Test Case ID</b>	PAY-05
<b>Test Area</b>	Payment System Functionalities
<b>Date</b>	January 25, 2025
<b>Sprint</b>	Sprint 2
<b>Test Scenario</b>	Freelancer requests payment with Payoneer.
<b>Expected Result</b>	Payment request was successfully sent with a success toast message; admin received email notification about the payment request.
<b>Actual Result</b>	Payment request was successfully sent with a success toast message; admin received email notification about the payment request.
<b>Status</b>	Pass
<b>Bug ID</b>	N/A
<b>Tester</b>	Aung Khant Phyo
<b>Notes</b>	N/A

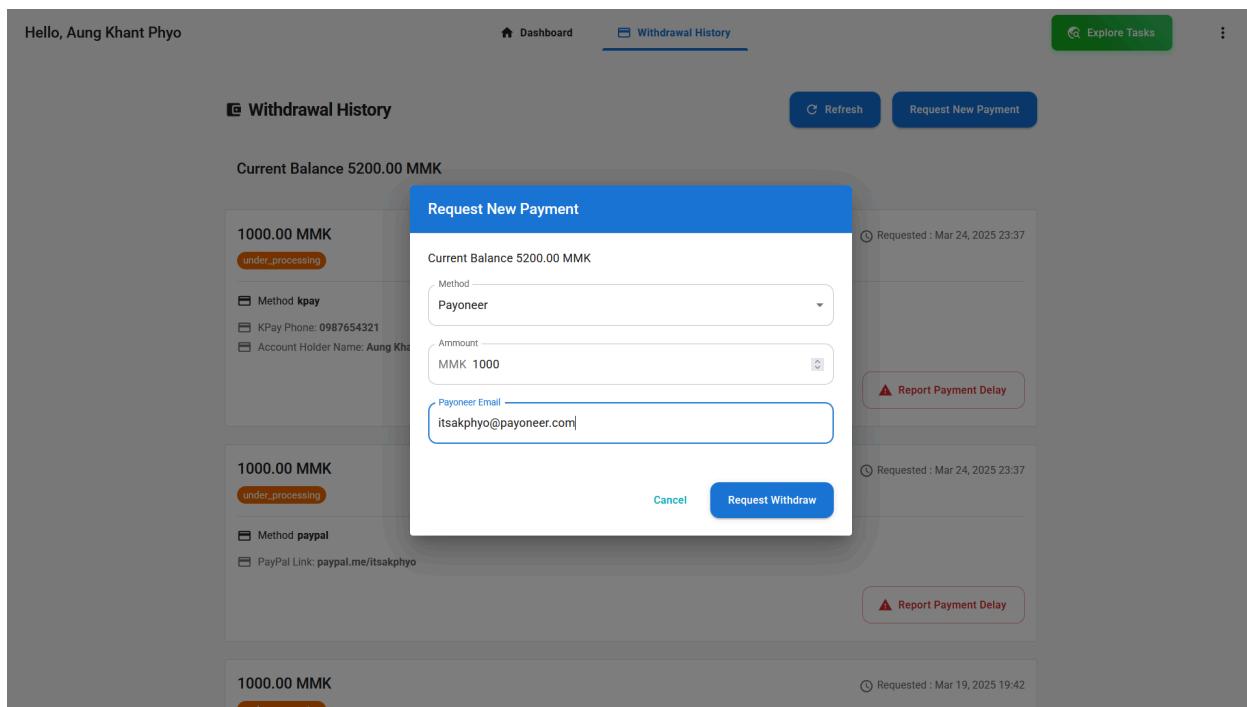


Figure 76: Freelancer requesting payment with Payoneer.



*Figure 77: Email notification was sent to admin about the payment request*

<b>Test Case ID</b>	PAY-06
<b>Test Area</b>	Payment System Functionalities
<b>Date</b>	January 25, 2025
<b>Sprint</b>	Sprint 2
<b>Test Scenario</b>	Freelancer requests payment with a Bank Account.
<b>Expected Result</b>	Payment request was successfully sent with a success toast message; admin received email notification about the payment request.
<b>Actual Result</b>	Payment request was successfully sent with a success toast message; admin received email notification about the payment request.
<b>Status</b>	Pass
<b>Bug ID</b>	N/A
<b>Tester</b>	Aung Khant Phyo
<b>Notes</b>	N/A

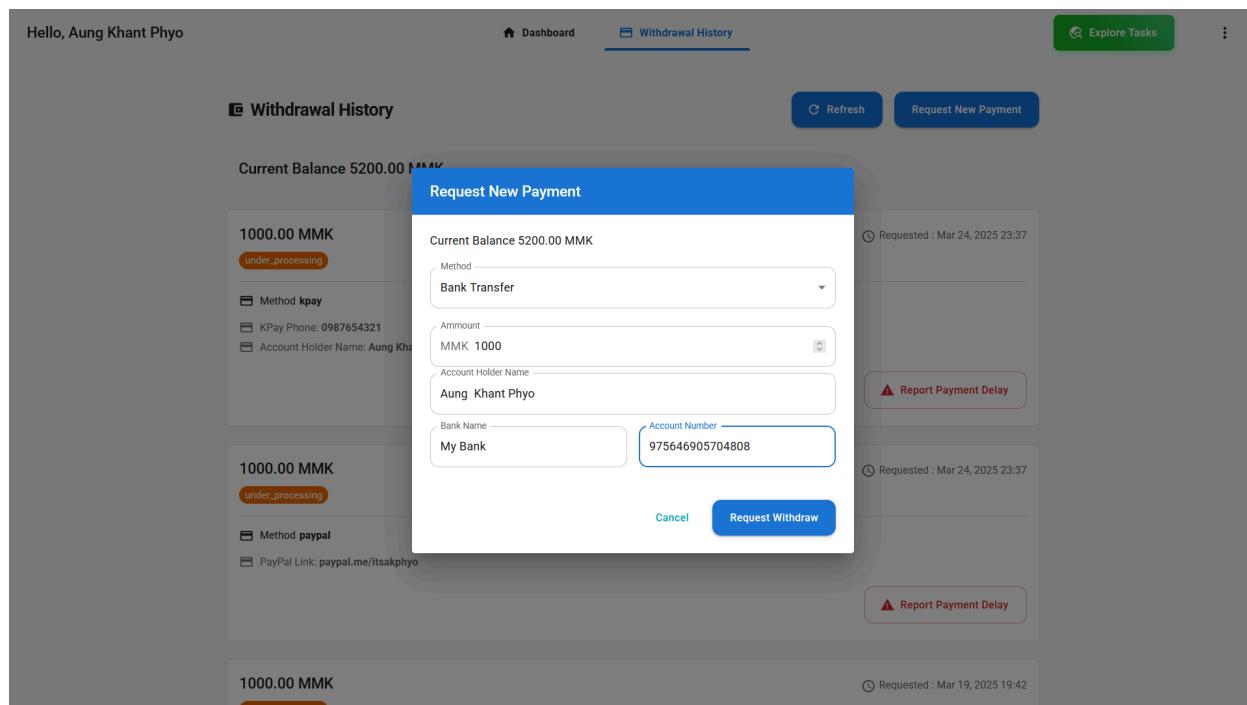
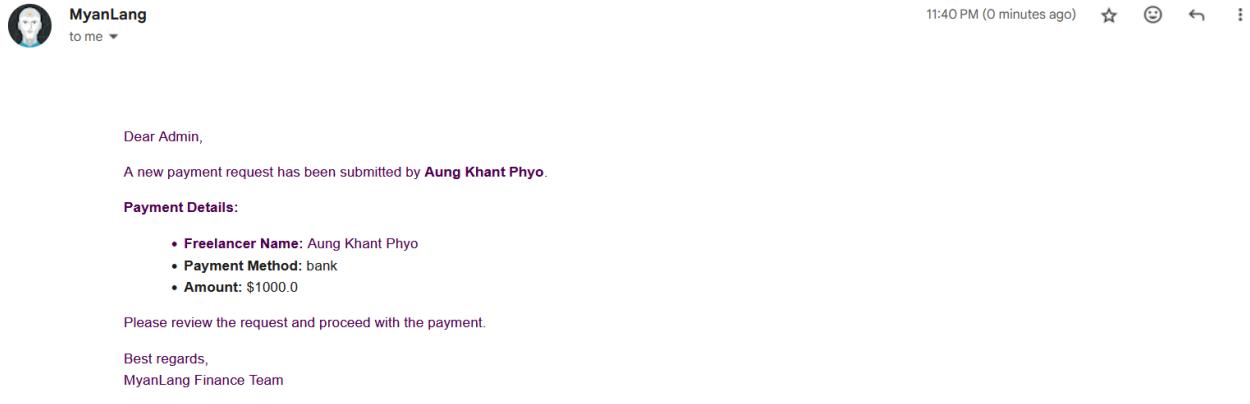


Figure 78: Freelancer requesting payment with Bank Transfer



*Figure 79: Email notification was sent to admin about the payment request*

<b>Test Case ID</b>	PAY-07
<b>Test Area</b>	Payment System Functionalities
<b>Date</b>	January 25, 2025
<b>Sprint</b>	Sprint 2
<b>Test Scenario</b>	Admin process payment manually and upload a receipt.
<b>Expected Result</b>	Payment was processed successfully and email notification about payment update was sent to freelancer
<b>Actual Result</b>	Payment was processed successfully and email notification about payment update was sent to freelancer
<b>Status</b>	Pass
<b>Bug ID</b>	N/A
<b>Tester</b>	Aung Khant Phyo
<b>Notes</b>	N/A

The screenshot shows the 'Payments' section of the admin dashboard. On the left, there's a sidebar with links: Jobs Dashboard, Payments (which is active and highlighted in blue), QA Management, Assessment Tasks, and User Reports. The main area is titled 'Payment Withdrawal Requests' and describes it as 'Manage and process freelancer payment withdrawal requests'. It features a search bar and filters for 'Payment Method' (All Methods) and 'Status' (All Statuses). Below is a table with the following data:

ID	Freelancer ID	Amount	Method	Requested	Status	Actions
7	1	1000.00 MMK	payoneer	3/24/2025, 11:42:47 PM	Processing	<a href="#">View</a>
6	1	1000.00 MMK	wavepay	3/24/2025, 11:42:38 PM	Processing	<a href="#">View</a>
5	1	1000.00 MMK	bank	3/24/2025, 11:40:06 PM	Processing	<a href="#">View</a>
4	1	1000.00 MMK	kpay	3/24/2025, 11:37:59 PM	Processing	<a href="#">View</a>
3	1	1000.00 MMK	paypal	3/24/2025, 11:37:22 PM	Processing	<a href="#">View</a>
2	1	1000.00 MMK	kpay	3/19/2025, 7:42:59 PM	Processing	<a href="#">View</a>
1	1	666.00 MMK	paypal	3/18/2025, 6:15:09 PM	Complete	<a href="#">View</a>

At the bottom, it says 'Showing 7 of 7 requests'.

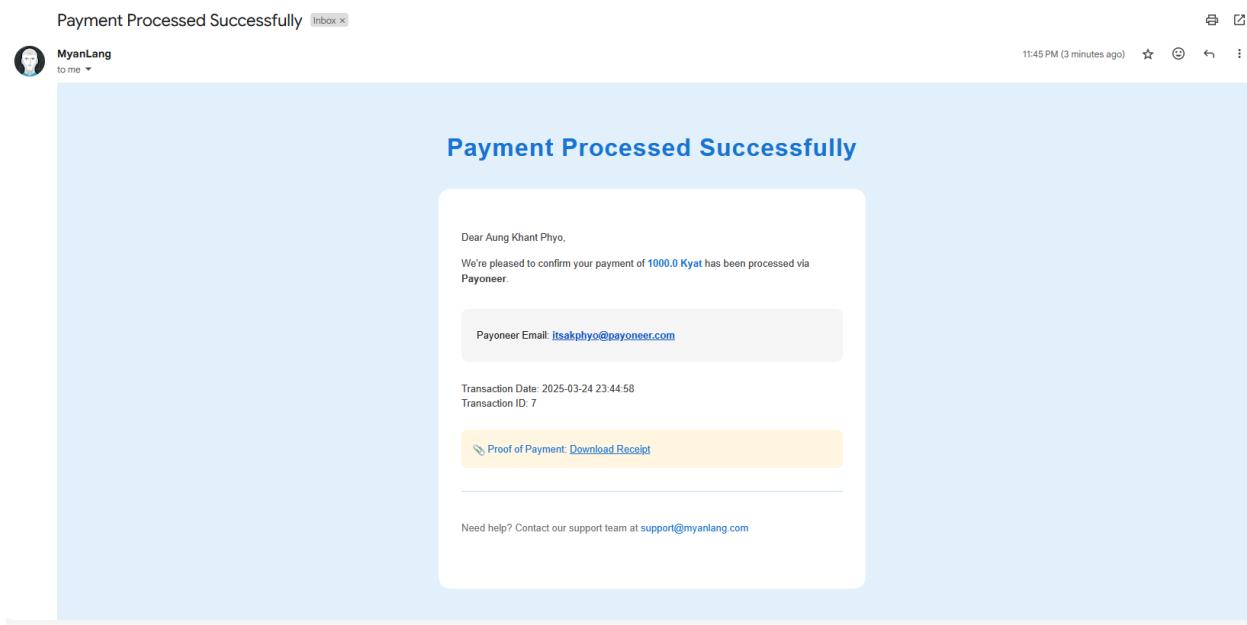
Figure 80: Admin receive payment requests in a table

The screenshot shows a web-based application interface for managing payment withdrawal requests. On the left, there is a sidebar with navigation links: Jobs Dashboard, Payments (which is selected and highlighted in blue), QA Management, Assessment Tasks, and User Reports. The main content area has a title "Payment Withdrawal Requests" with a subtitle "Manage and process freelancer payment withdrawal requests". Below this are filters for "Payment Method" (set to "All Methods") and "Status" (set to "All Statuses"), and a search bar. The main table lists 7 requests. The first request, ID 7, has a modal open titled "Withdrawal Request #7" containing details: Freelancer ID: 1, Amount: 1000.00 MMK, Payment Method: payoneer, Requested At: 3/24/2025, 11:42:47 PM, and Status: Processing. It also includes a link "View Uploaded Proof". At the bottom of the modal are "Cancel" and "Approve" buttons. The table rows show other requests with various statuses like Processing, Complete, and Pending. A footer at the bottom of the page says "Showing 7 of 7 requests" and includes a "Logout" link.

*Figure 81: Admin manually processes payment with payoneer and uploads receipt as proof of payment.*

This screenshot shows the same application interface after the admin has processed the withdrawal requests. The sidebar and top navigation are identical to Figure 81. The main table now displays all 7 requests with their status set to "Complete". The first request (ID 7) still has its original details and the "View Uploaded Proof" link. The footer indicates "Showing 7 of 7 requests" and includes a "Logout" link.

*Figure 82: Request status was set to complete after admin make payment and upload receipt*

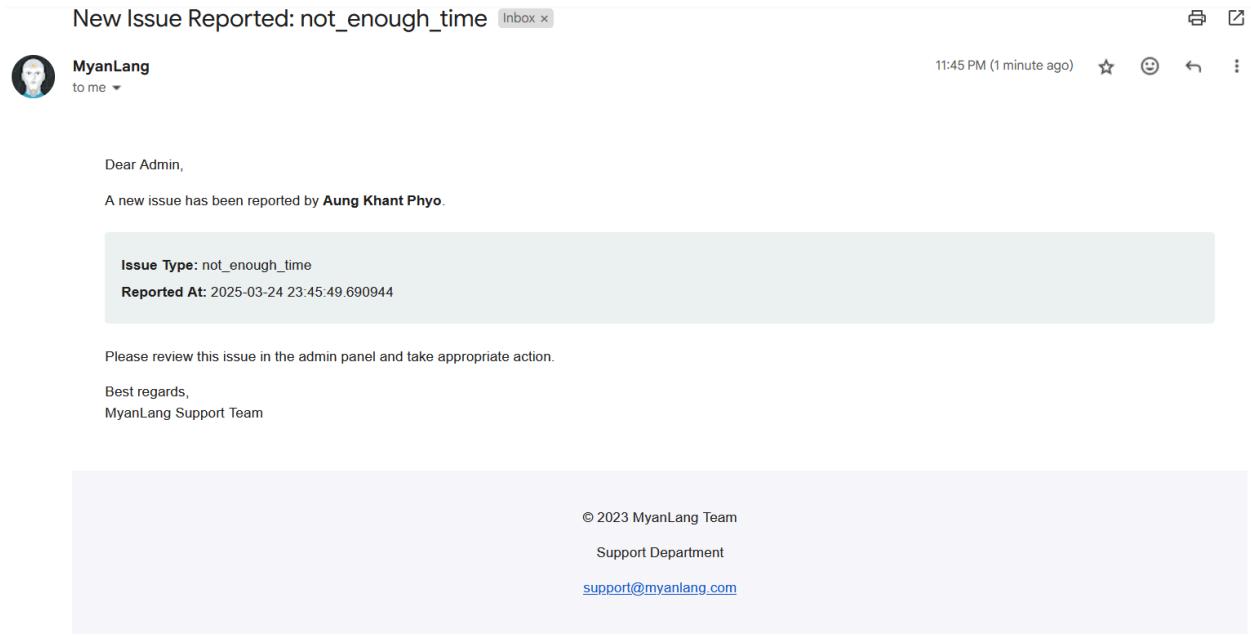


*Figure 83: Freelancer received notification about payment processed successfully.*

<b>Test Case ID</b>	ISSUE-02
<b>Test Area</b>	Issue report and resolve system
<b>Date</b>	February 7, 2025
<b>Sprint</b>	Sprint 3
<b>Test Scenario</b>	Freelancer reports about “Not having enough time” for translation.
<b>Expected Result</b>	Report was successfully sent to the admin with email notification.
<b>Actual Result</b>	Report was successfully sent to the admin with email notification.
<b>Status</b>	Pass
<b>Bug ID</b>	N/A
<b>Tester</b>	Aung Khant Phyo
<b>Notes</b>	N/A

The screenshot shows a web-based translation application interface. At the top, there's a header with the MyanLang logo, a 'Qualified (100%)' badge, and a language selection dropdown set to 'Myanmar → Myanmar'. Below the header, a progress bar indicates 'Reward : 100 MMK' and 'Time Left : 9:54'. On the left, there's a sidebar with 'Instructions' and 'Please translate the followings...'. Under 'Original Text (Myanmar)', the text 'original sentence 3' is shown. Below it, under 'Your Translation (Chin)', there's a large empty text input field. A modal window titled 'ISSUE REPORT' is open in the center. It contains fields for 'Task ID : 83' and 'Issue Type' (set to 'Not enough time for translation'). At the bottom of the modal are 'Cancel' and 'Submit Report' buttons. In the background, there are other UI elements like 'Close', 'Submit & Close', and 'Submit & Show Next' buttons. The footer of the page includes a copyright notice: '© 2025 MyanLang Translation Platform. All rights reserved.'

Figure 84: Freelancer reporting issue for not having enough time for translation



*Figure 85: Admin received email notification about issue report*

<b>Test Case ID</b>	ISSUE-03
<b>Test Area</b>	Issue report and resolve system
<b>Date</b>	February 7, 2025
<b>Sprint</b>	Sprint 3
<b>Test Scenario</b>	Freelancer reports about “wrong source language”.
<b>Expected Result</b>	Report was successfully sent to the admin with email notification.
<b>Actual Result</b>	Report was successfully sent to the admin with email notification.
<b>Status</b>	Pass
<b>Bug ID</b>	N/A
<b>Tester</b>	Aung Khant Phyo
<b>Notes</b>	N/A

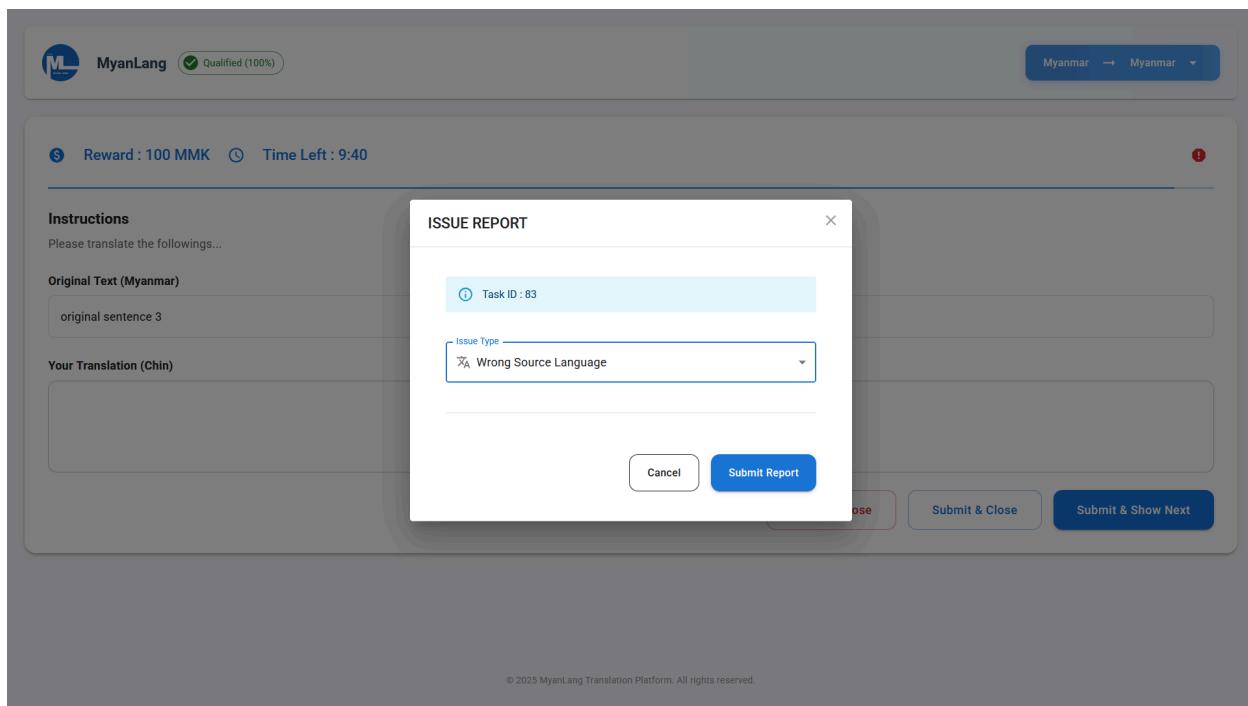


Figure 86: Freelancer reporting issue about having wrong source language

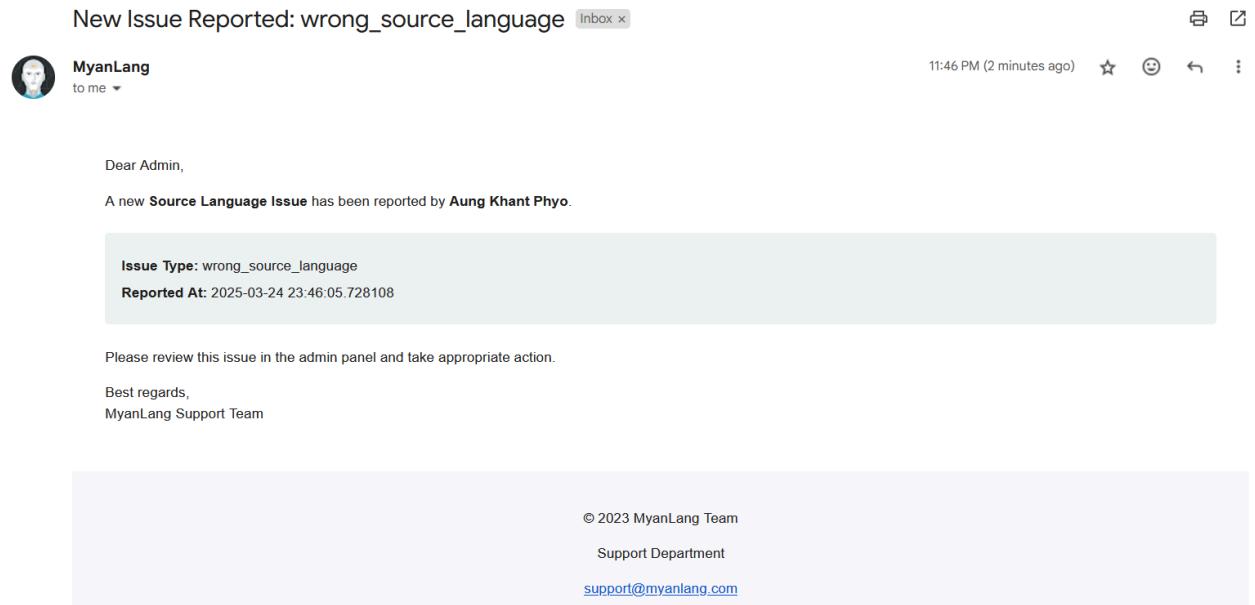


Figure 87: Admin receive email notification about issue report

<b>Test Case ID</b>	ISSUE-04
<b>Test Area</b>	Issue report and resolve system
<b>Date</b>	February 7, 2025
<b>Sprint</b>	Sprint 3
<b>Test Scenario</b>	Admin resolve for payment delay: pay manually and close the report status.
<b>Expected Result</b>	Report was closed successfully and sent an email notification to the freelancer.
<b>Actual Result</b>	Report was closed successfully and sent an email notification to the freelancer.
<b>Status</b>	Pass
<b>Bug ID</b>	N/A
<b>Tester</b>	Aung Khant Phyo
<b>Notes</b>	N/A

ID	Issue Type	Status	Reported At	Actions
#1	Payment Delay	under review	Mar 19, 2025, 07:54 PM	<button>Resolve</button>
#2	Wrong Source Language	under review	Mar 19, 2025, 07:55 PM	<button>Resolve</button>
#3	Not Enough Time	under review	Mar 19, 2025, 07:55 PM	<button>Resolve</button>
#4	Accuracy Appeal	under review	Mar 19, 2025, 07:55 PM	<button>Resolve</button>
#5	Payment Delay	under review	Mar 24, 2025, 09:45 PM	<button>Resolve</button>
#6	Accuracy Appeal	under review	Mar 24, 2025, 11:14 PM	<button>Resolve</button>
#7	Not Enough Time	under review	Mar 24, 2025, 11:45 PM	<button>Resolve</button>

Figure 88: Admin assess issue reports as a table

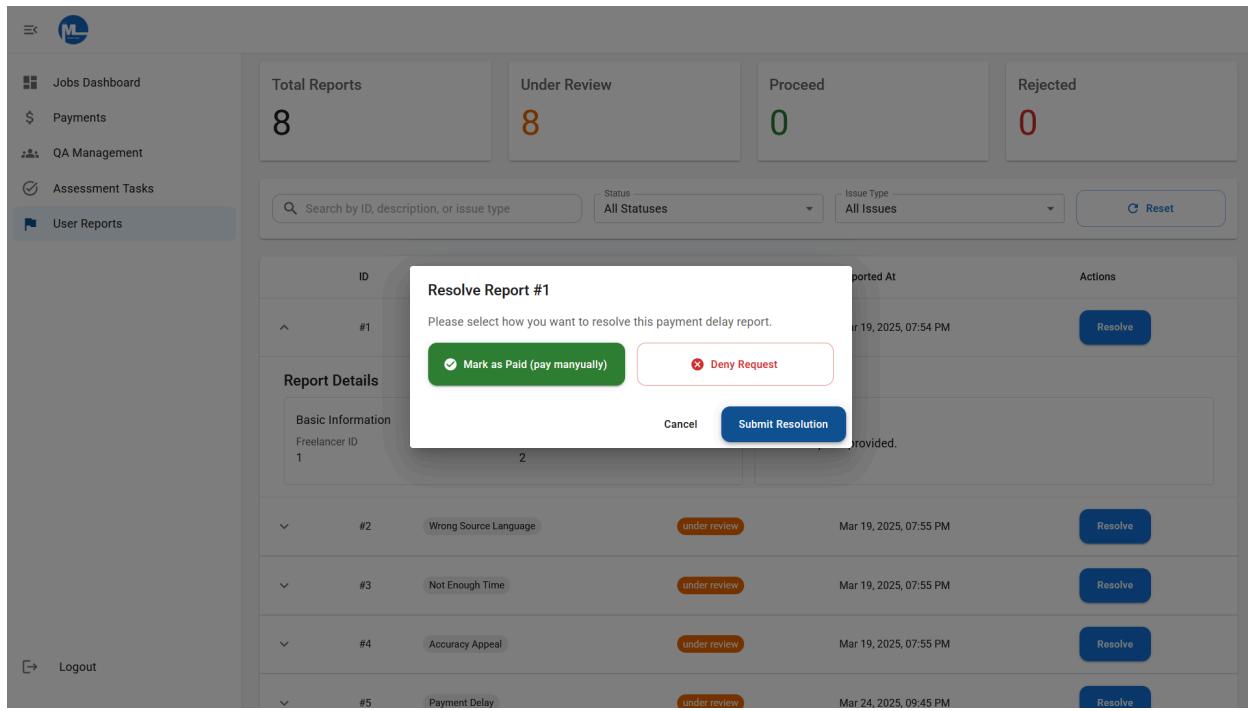


Figure 89: Admin resolved payment delay issue by paying manually and set the report status to complete

<b>Test Case ID</b>	ISSUE-05
<b>Test Area</b>	Issue report and resolve system
<b>Date</b>	February 7, 2025
<b>Sprint</b>	Sprint 3
<b>Test Scenario</b>	Admin resolve user accuracy appeal messages by resetting user accuracy status.
<b>Expected Result</b>	Freelancer accuracy was reset successfully and sent notification to freelancer email.
<b>Actual Result</b>	Freelancer accuracy was reset successfully and sent notification to freelancer email.
<b>Status</b>	Pass
<b>Bug ID</b>	N/A
<b>Tester</b>	Aung Khant Phyo
<b>Notes</b>	N/A

The screenshot shows a user interface for managing reports. On the left, there's a sidebar with navigation links: Jobs Dashboard, Payments, QA Management, Assessment Tasks, and User Reports (which is currently selected). The main area displays a table of reports with columns: ID, Issue Type, Status, Reported At, and Actions. There are five reports listed:

- #2: Wrong Source Language, under review, Mar 19, 2025, 07:55 PM, Resolve button
- #3: Not Enough Time, under review, Mar 19, 2025, 07:55 PM, Resolve button
- #4: Accuracy Appeal, under review, Mar 19, 2025, 07:55 PM, Resolve button
- #5: Payment Delay, proceed, Mar 24, 2025, 09:45 PM, Resolve button
- #6: Accuracy Appeal, proceed, Mar 24, 2025, 11:14 PM, Resolve button

A modal window is open over the third report (#4) titled "Resolve Report #4". It contains a message: "Please select how you want to resolve this accuracy appeal report." Below the message are two buttons: "Reset Freelancer Accuracy" (green background, checked) and "Deny Appeal" (red background). At the bottom of the modal are "Cancel", "Submit Resolution" (blue), and "Proceed" buttons.

Figure 90: Admin reset freelancer accuracy status after reviewing appeal message

Appeal Accepted Inbox ×



MyanLang  
to me ▾

**MyanLang Team**

Dear Aung Khant Phyo,

We've reviewed your appeal regarding the accuracy assessment for your Myanmar → Chin translations and are pleased to inform you that your request has been approved.

Your accuracy score for this language pair has been reset, and you may now resume participation in translation projects.

Please don't hesitate to contact our quality assurance team if you need any clarification or support.

Best regards,  
MyanLang Quality Team

© 2025 MyanLang Team

Quality Assurance Department

[support@myanlang.com](mailto:support@myanlang.com)

✉ Reply ↗ Forward 😊

*Figure 91: Freelancer receive email notification about accuracy score was reset*

<b>Test Case ID</b>	ISSUE-06
<b>Test Area</b>	Issue report and resolve system
<b>Date</b>	February 7, 2025
<b>Sprint</b>	Sprint 3
<b>Test Scenario</b>	Admin close task for the task which has wrong source language.
<b>Expected Result</b>	Task was closed successfully and sent an email notification to the freelancer.
<b>Actual Result</b>	Task was closed successfully and sent an email notification to the freelancer.
<b>Status</b>	Pass
<b>Bug ID</b>	N/A
<b>Tester</b>	Aung Khant Phyo
<b>Notes</b>	N/A

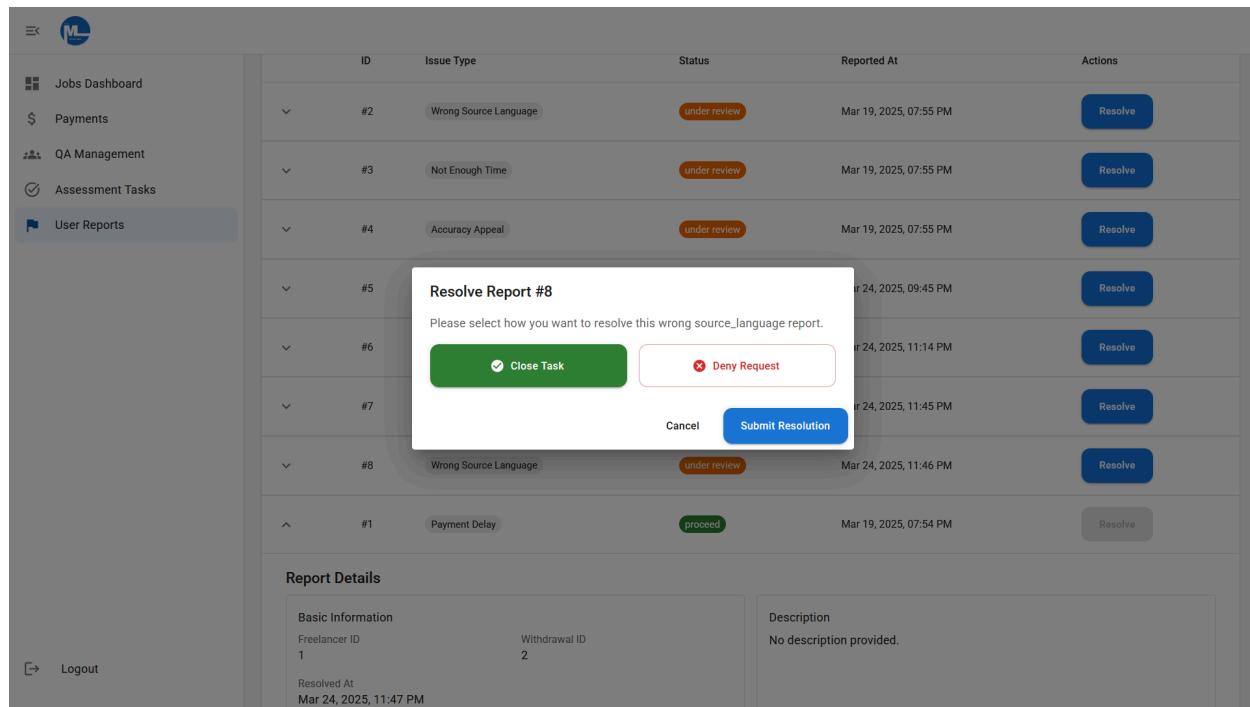


Figure 92: Admin resolve “wrong source language” issue by closing the task

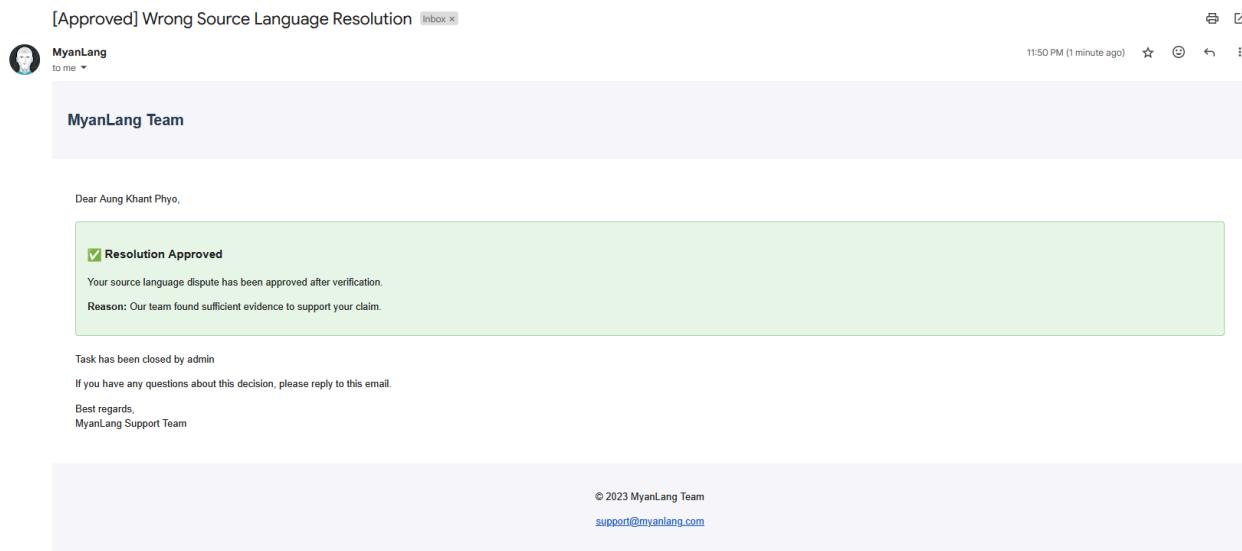


Figure 93: Freelancer receive email notification about report status update

<b>Test Case ID</b>	ISSUE-07
<b>Test Area</b>	Issue report and resolve system
<b>Date</b>	February 7, 2025
<b>Sprint</b>	Sprint 3
<b>Test Scenario</b>	Admin update given to the task that users report about not having enough time.
<b>Expected Result</b>	Given time for the task was updated successfully and sent email notification to the freelancer who reported this issue.
<b>Actual Result</b>	Given time for the task was updated successfully and sent email notification to the freelancer who reported this issue.
<b>Status</b>	Pass
<b>Bug ID</b>	N/A
<b>Tester</b>	Aung Khant Phyo
<b>Notes</b>	N/A

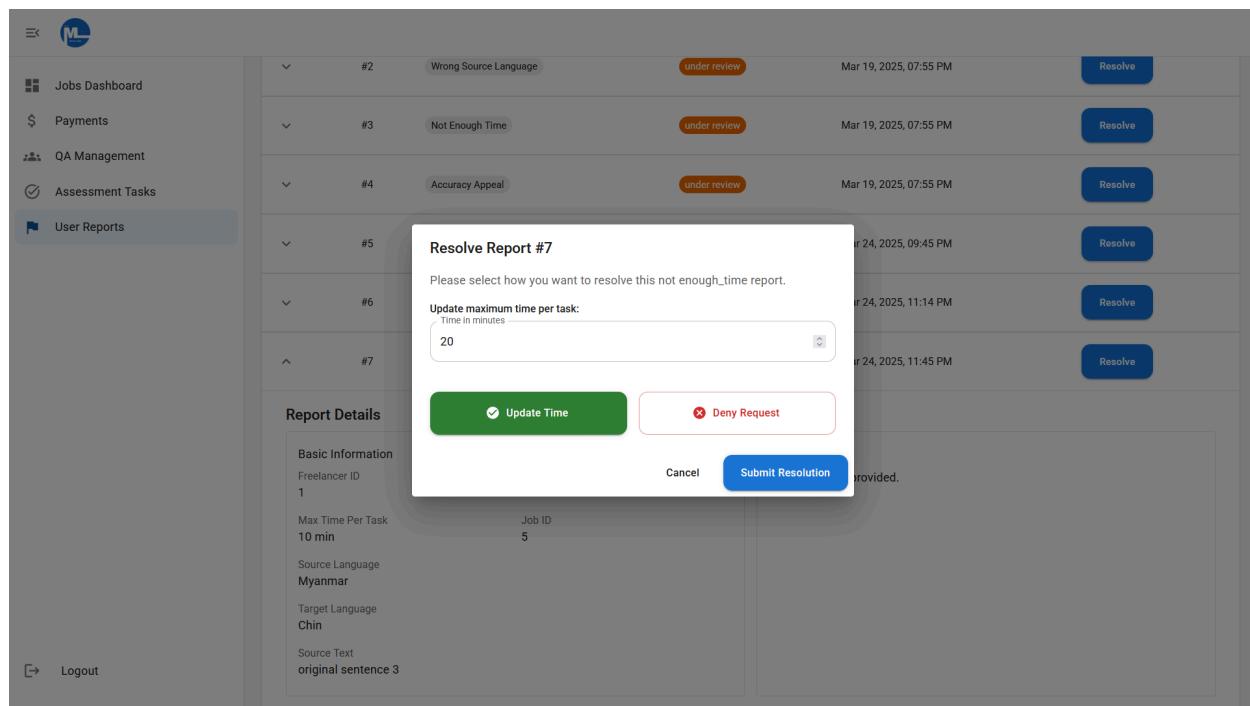


Figure 94: Admin resolve “not enough time” issue by updating the given time for the task

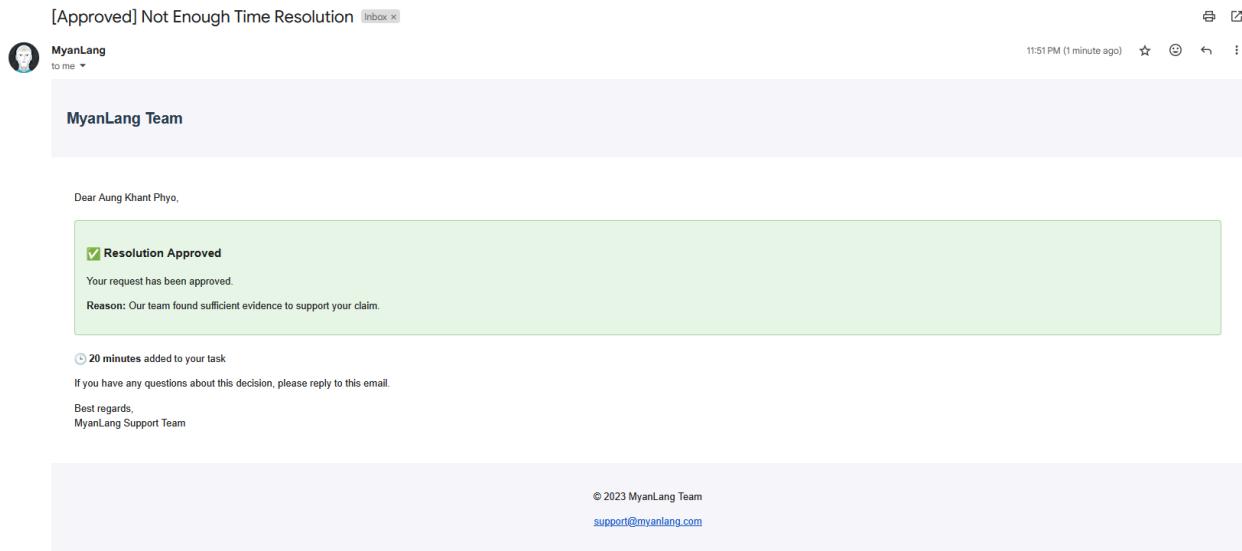


Figure 95: Freelancer receive email notification about report status update

<b>Test Case ID</b>	SYSTEM-LANGUAGE-01
<b>Test Area</b>	Multi-language support
<b>Date</b>	February 7, 2025
<b>Sprint</b>	Sprint 3
<b>Test Scenario</b>	Freelancer changes the system language to English.
<b>Expected Result</b>	System language was successfully changed to English.
<b>Actual Result</b>	System language was successfully changed to English.
<b>Status</b>	Pass
<b>Bug ID</b>	N/A
<b>Tester</b>	Aung Khant Phyo
<b>Notes</b>	N/A

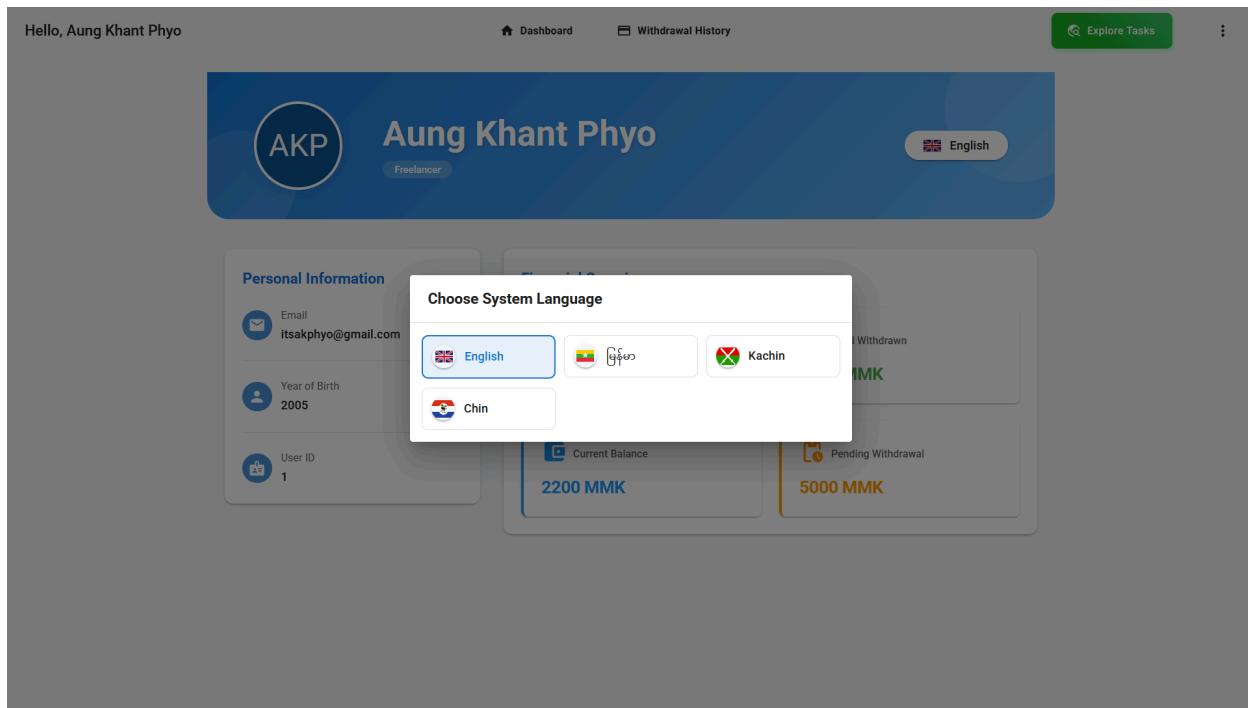


Figure 96: Freelancer changing the system language to English with a selector dialog

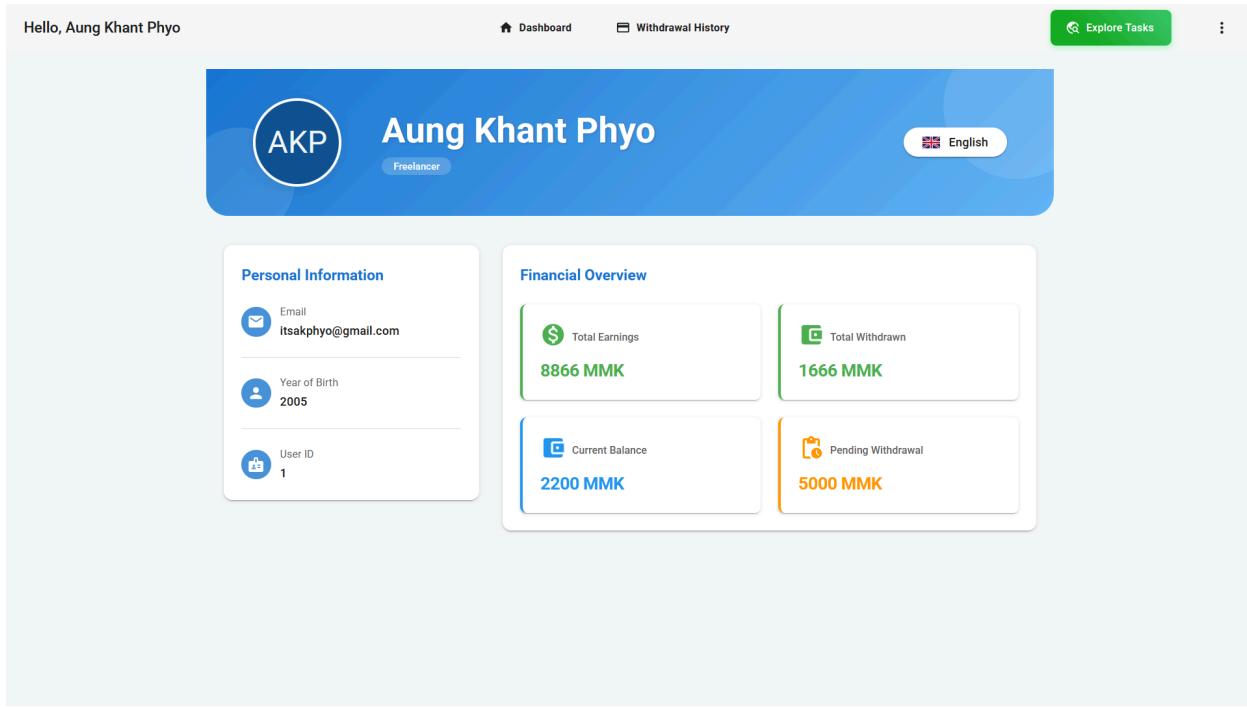


Figure 97: System language was changed to English

<b>Test Case ID</b>	SYSTEM-LANGUAGE-02
<b>Test Area</b>	Multi-language support
<b>Date</b>	February 7, 2025
<b>Sprint</b>	Sprint 3
<b>Test Scenario</b>	Freelancer changes the system language to Myanmar.
<b>Expected Result</b>	System language was successfully changed to Myanmar.
<b>Actual Result</b>	System language was successfully changed to Myanmar.
<b>Status</b>	Pass
<b>Bug ID</b>	N/A
<b>Tester</b>	Aung Khant Phyo
<b>Notes</b>	N/A

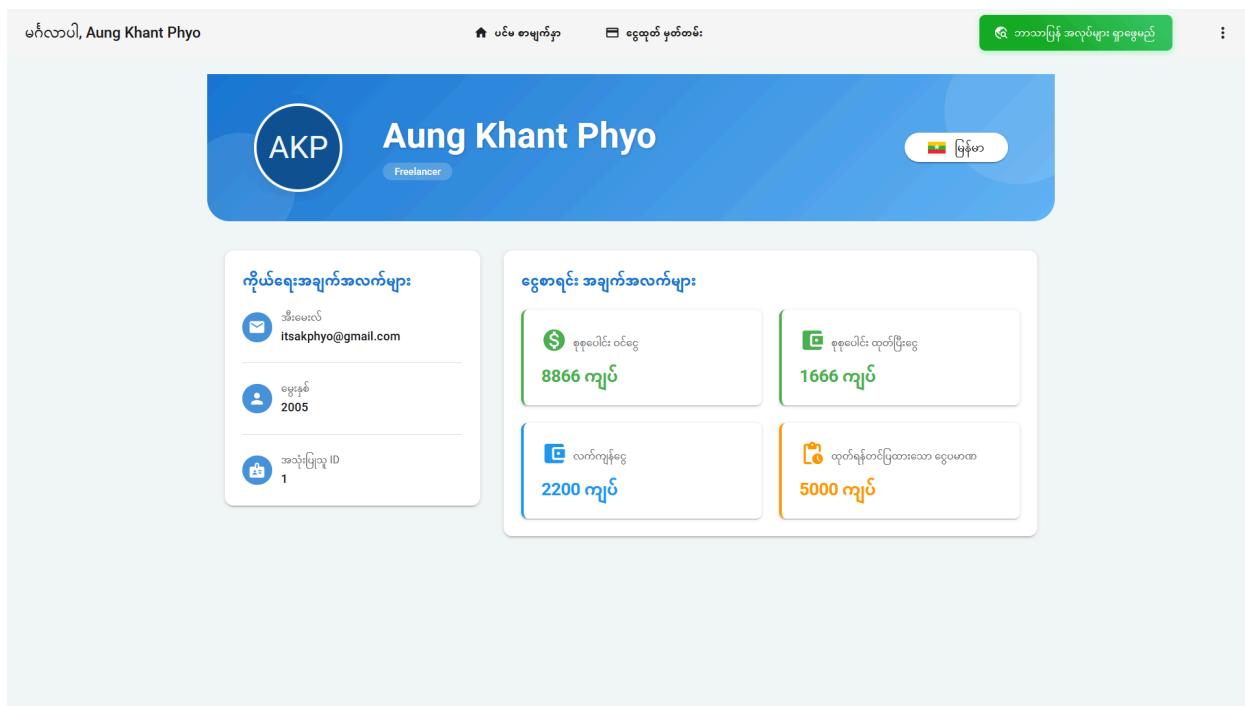


Figure 98: Freelancer changed the system language to Myanmar

<b>Test Case ID</b>	SYSTEM-LANGUAGE-03
<b>Test Area</b>	Multi-language support
<b>Date</b>	February 7, 2025
<b>Sprint</b>	Sprint 3
<b>Test Scenario</b>	Freelancer changes the system language to Chin.
<b>Expected Result</b>	System language was successfully changed to Chin.
<b>Actual Result</b>	System language was successfully changed to Chin.
<b>Status</b>	Pass
<b>Bug ID</b>	N/A
<b>Tester</b>	Aung Khant Phyo
<b>Notes</b>	N/A

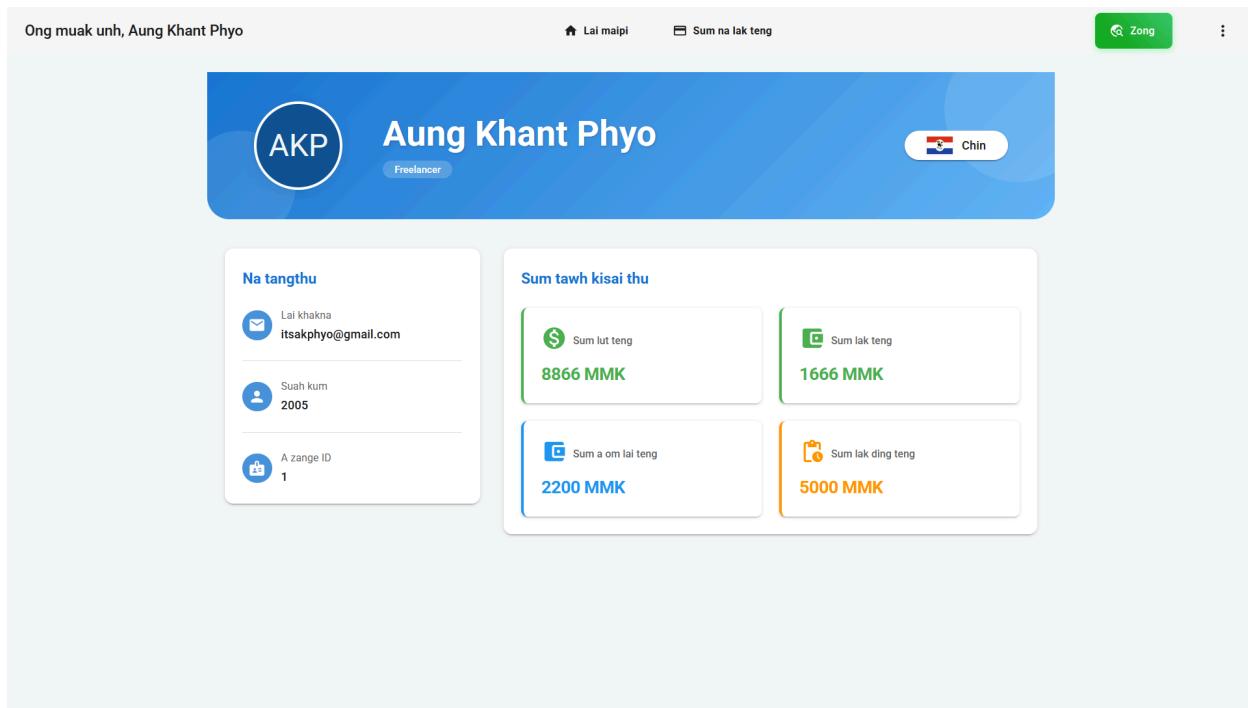


Figure 99: Freelancer changed the system language to Chin

<b>Test Case ID</b>	SYSTEM-LANGUAGE-04
<b>Test Area</b>	Multi-language support
<b>Date</b>	February 7, 2025
<b>Sprint</b>	Sprint 3
<b>Test Scenario</b>	Freelancer changes the system language to Kachin.
<b>Expected Result</b>	System language was successfully changed to Kachin.
<b>Actual Result</b>	System language was successfully changed to Kachin.
<b>Status</b>	Pass
<b>Bug ID</b>	N/A
<b>Tester</b>	Aung Khant Phyo
<b>Notes</b>	N/A

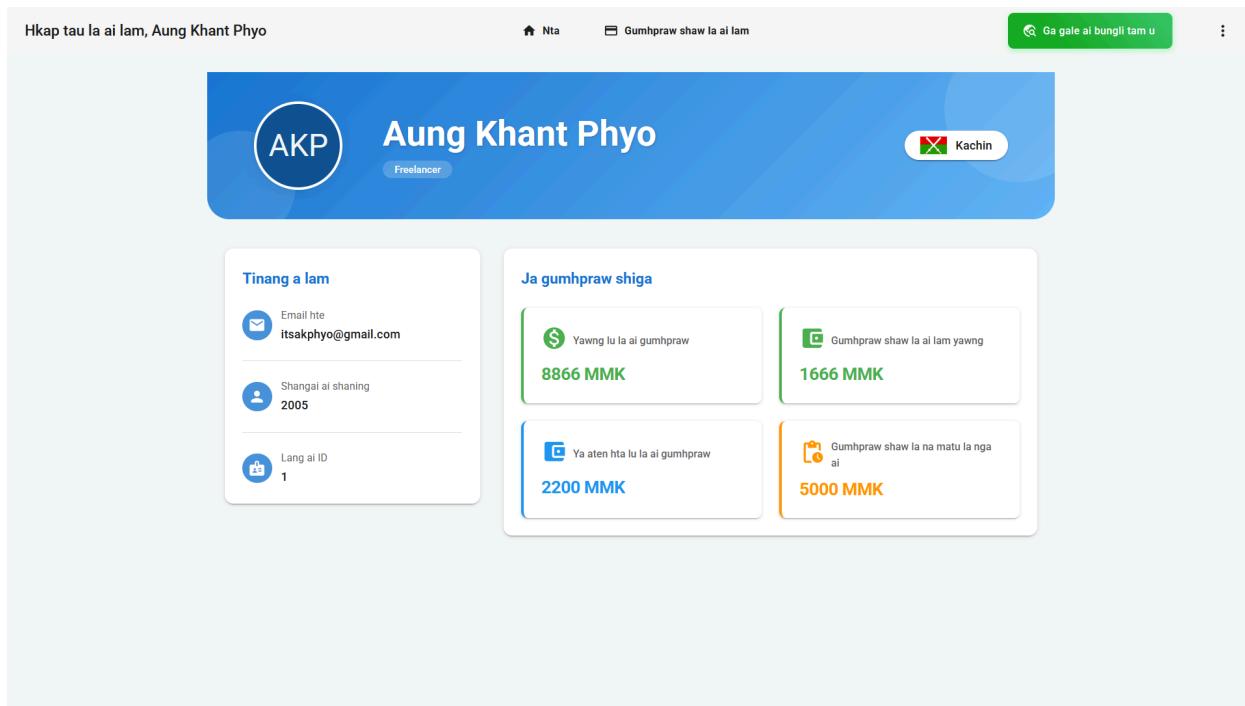


Figure 100: Freelancer changed the system language to Kachin

<b>Test Case ID</b>	ROUTE-01
<b>Test Area</b>	Protected Routes
<b>Date</b>	February 7, 2025
<b>Sprint</b>	Sprint 3
<b>Test Scenario</b>	Freelancer navigates to the admin route.
<b>Expected Result</b>	“404 Page not found” page was returned successfully.
<b>Actual Result</b>	“404 Page not found” page was returned successfully.
<b>Status</b>	Pass
<b>Bug ID</b>	N/A
<b>Tester</b>	Aung Khant Phyo
<b>Notes</b>	N/A

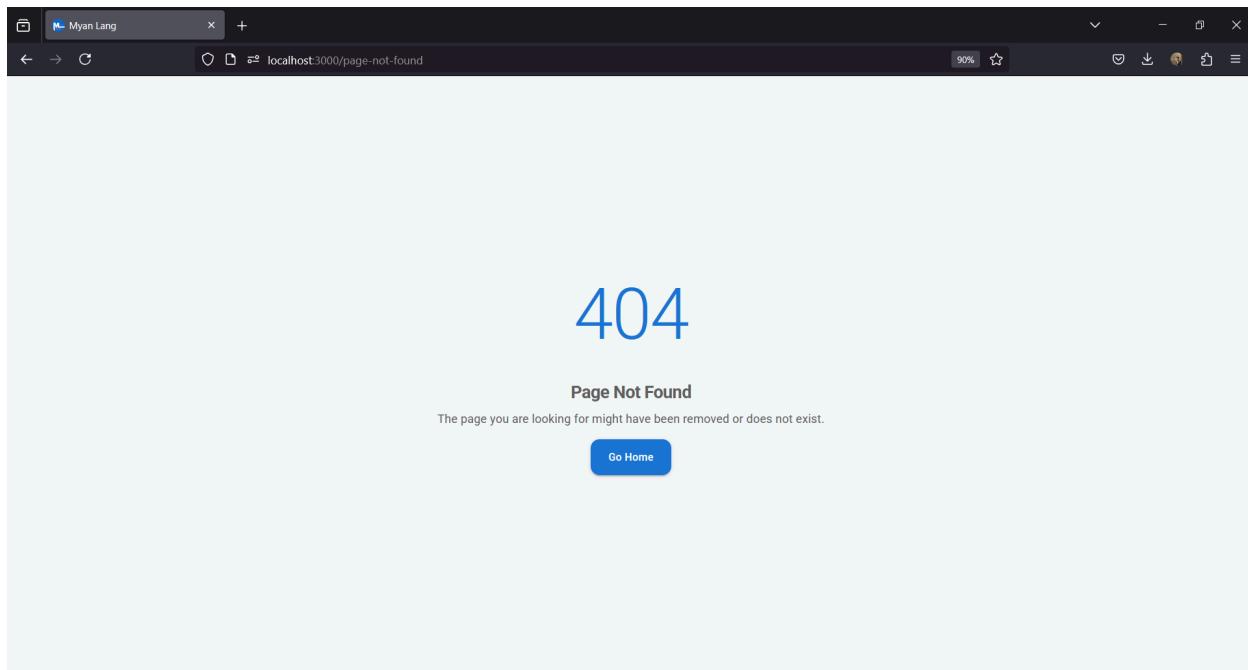
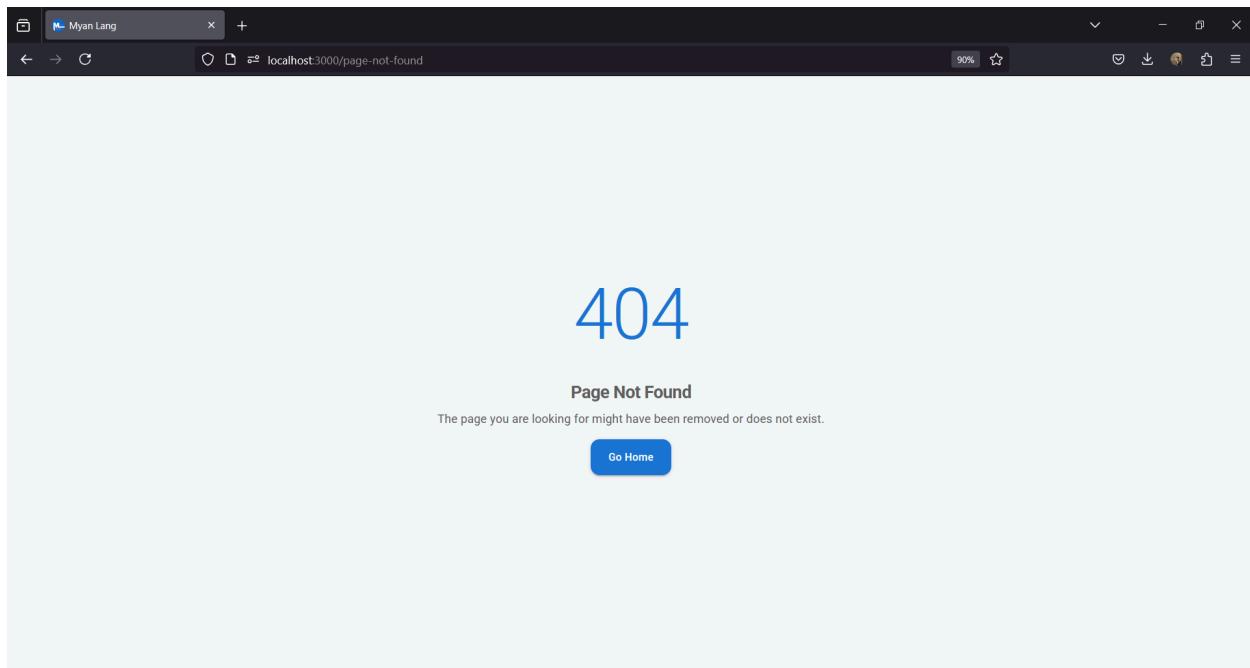


Figure 101: “404 Page not found” page was returned when freelancer navigates to the admin route.

<b>Test Case ID</b>	ROUTE-02
<b>Test Area</b>	Protected Routes
<b>Date</b>	February 7, 2025
<b>Sprint</b>	Sprint 3
<b>Test Scenario</b>	Freelancer navigates to the QA member route.
<b>Expected Result</b>	“404 Page not found” page was returned successfully.
<b>Actual Result</b>	“404 Page not found” page was returned successfully.
<b>Status</b>	Pass
<b>Bug ID</b>	N/A
<b>Tester</b>	Aung Khant Phyo
<b>Notes</b>	N/A



*Figure 102: “404 Page not found” page was returned when freelancer navigates to the QA member route.*

<b>Test Case ID</b>	ROUTE-03
<b>Test Area</b>	Protected Routes
<b>Date</b>	February 7, 2025
<b>Sprint</b>	Sprint 3
<b>Test Scenario</b>	Admin navigates to the freelancer route.
<b>Expected Result</b>	“404 Page not found” page was returned successfully.
<b>Actual Result</b>	“404 Page not found” page was returned successfully.
<b>Status</b>	Pass
<b>Bug ID</b>	N/A
<b>Tester</b>	Aung Khant Phyo
<b>Notes</b>	N/A

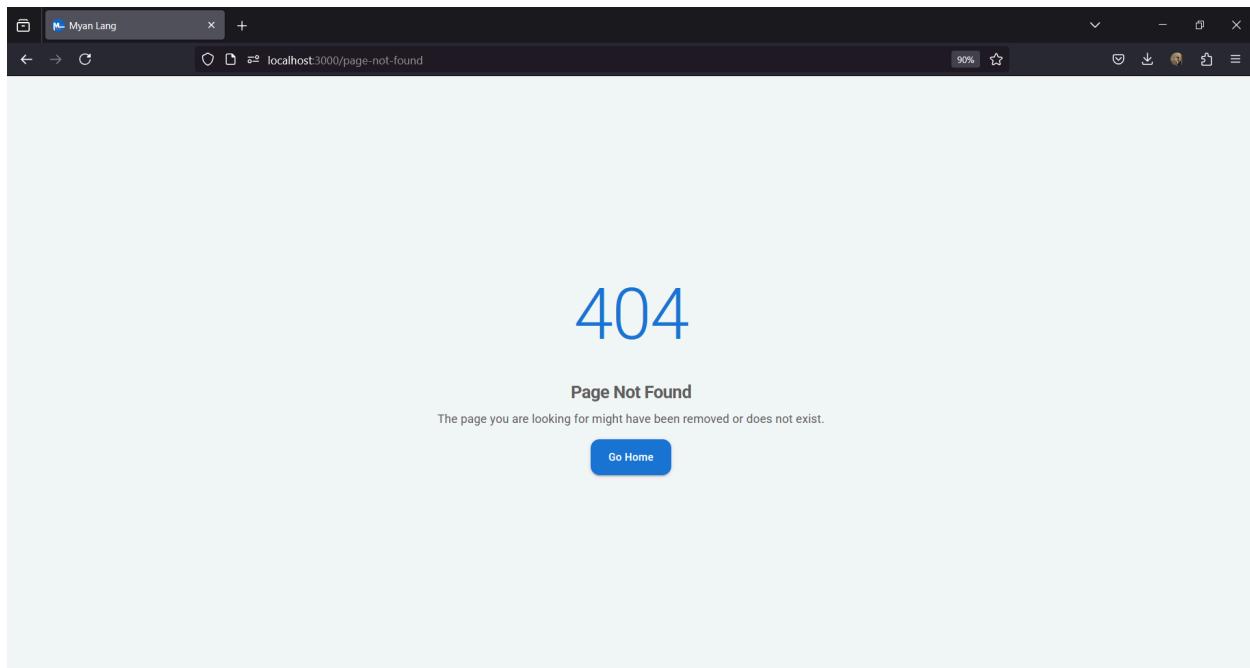


Figure 103: “404 Page not found” page was returned when admin navigates to the freelancer route.

<b>Test Case ID</b>	ROUTE-04
<b>Test Area</b>	Protected Routes
<b>Date</b>	February 7, 2025
<b>Sprint</b>	Sprint 3
<b>Test Scenario</b>	Admin navigates to the QA member route.
<b>Expected Result</b>	“404 Page not found” page was returned successfully.
<b>Actual Result</b>	“404 Page not found” page was returned successfully.
<b>Status</b>	Pass
<b>Bug ID</b>	N/A
<b>Tester</b>	Aung Khant Phyo
<b>Notes</b>	N/A

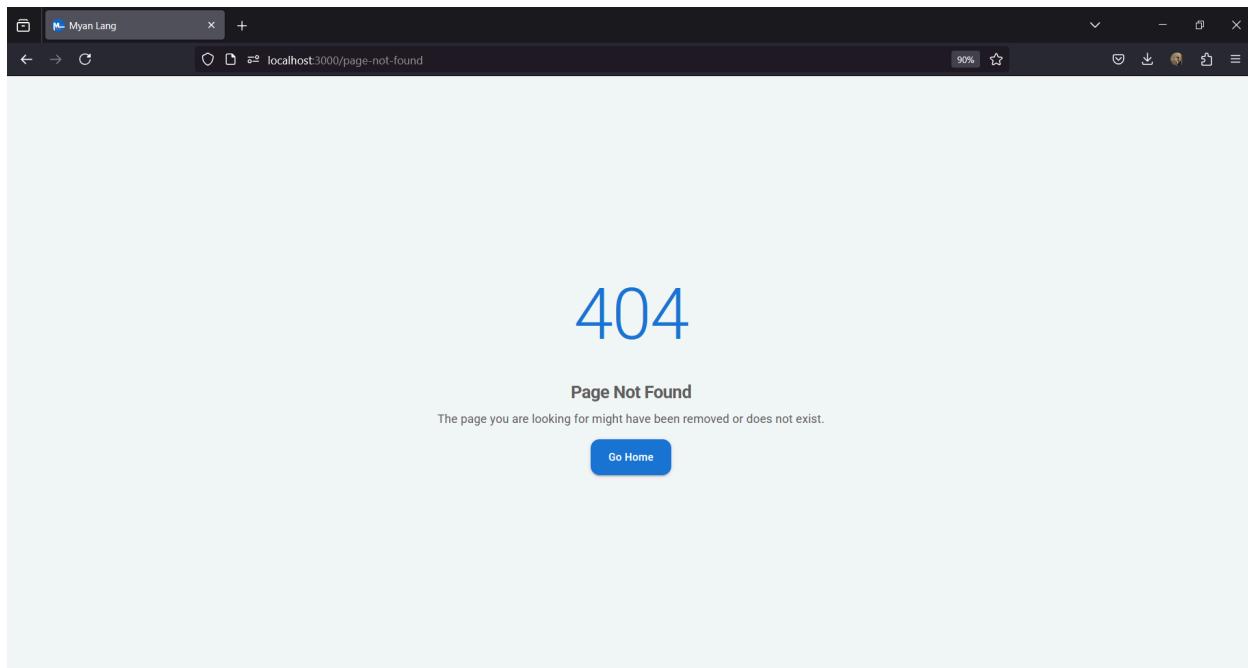


Figure 104: “404 Page not found” page was returned when admin navigates to the QA member route.

<b>Test Case ID</b>	ROUTE-05
<b>Test Area</b>	Protected Routes
<b>Date</b>	February 7, 2025
<b>Sprint</b>	Sprint 3
<b>Test Scenario</b>	QA member navigates to the admin route.
<b>Expected Result</b>	“404 Page not found” page was returned successfully.
<b>Actual Result</b>	“404 Page not found” page was returned successfully.
<b>Status</b>	Pass
<b>Bug ID</b>	N/A
<b>Tester</b>	Aung Khant Phyo
<b>Notes</b>	N/A

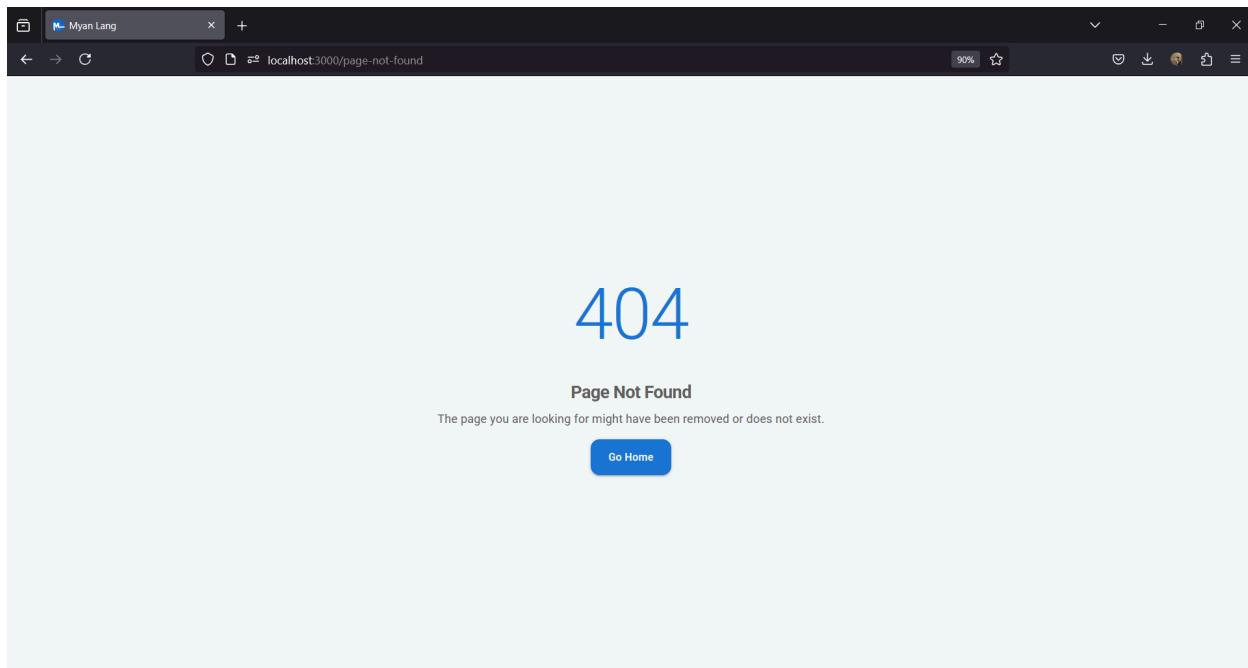
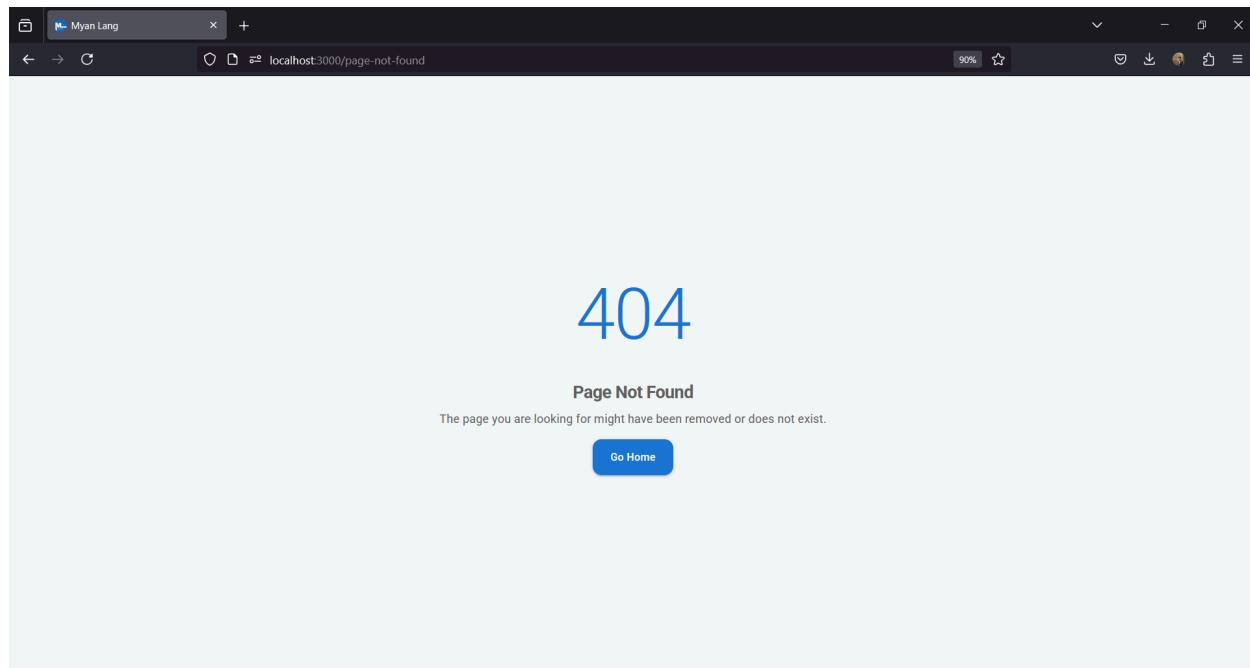


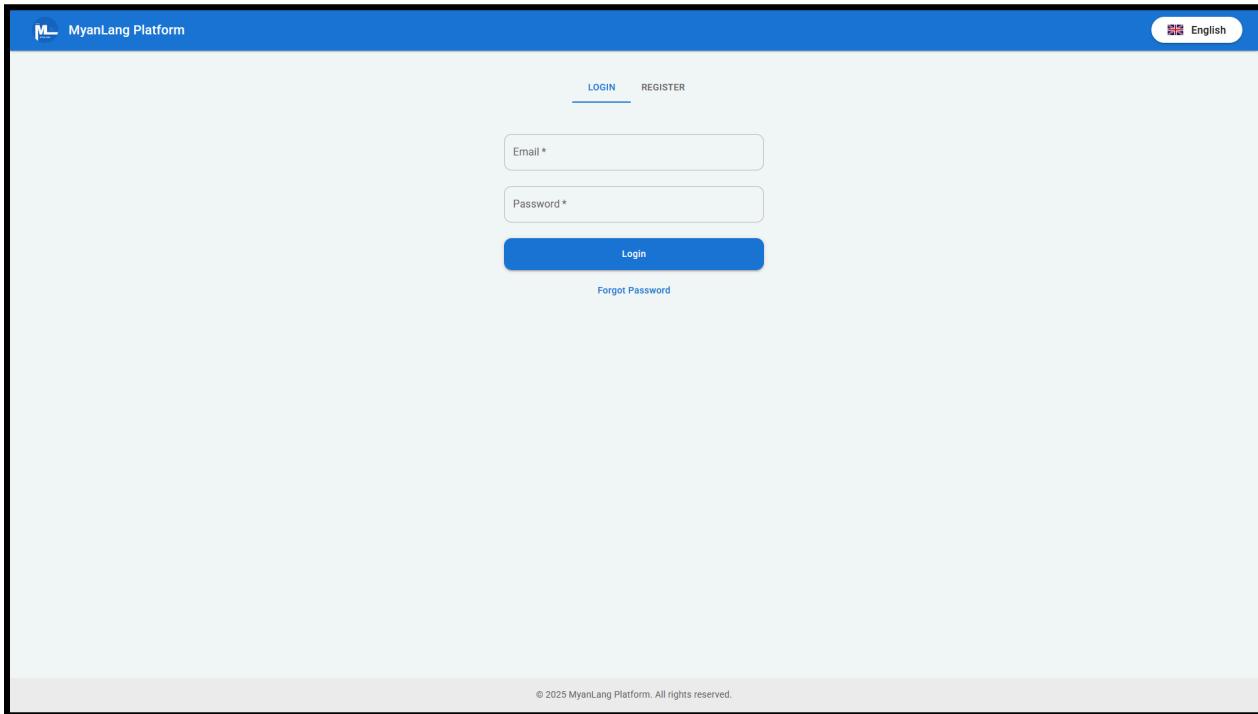
Figure 105: “404 Page not found” page was returned when QA member navigates to the admin route.

<b>Test Case ID</b>	ROUTE-06
<b>Test Area</b>	Protected Routes
<b>Date</b>	February 7, 2025
<b>Sprint</b>	Sprint 3
<b>Test Scenario</b>	QA member navigates to the freelancer route.
<b>Expected Result</b>	“404 Page not found” page was returned successfully.
<b>Actual Result</b>	“404 Page not found” page was returned successfully.
<b>Status</b>	Pass
<b>Bug ID</b>	N/A
<b>Tester</b>	Aung Khant Phyo
<b>Notes</b>	N/A



*Figure 106: “404 Page not found” page was returned when the QA member navigated to the freelancer route.*

## Section D - Screen Diagrams Sprint 1



*Figure 107: Login page of the platform*

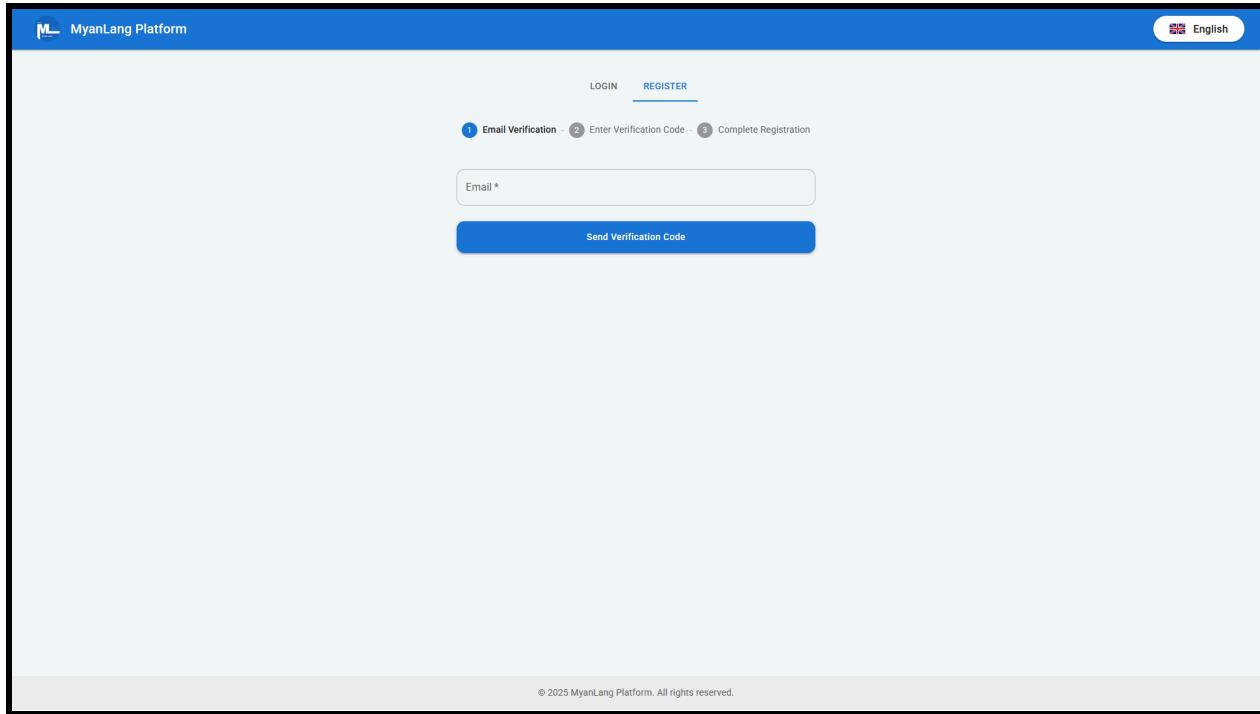


Figure 108: Register step 1 page (requesting verification code)

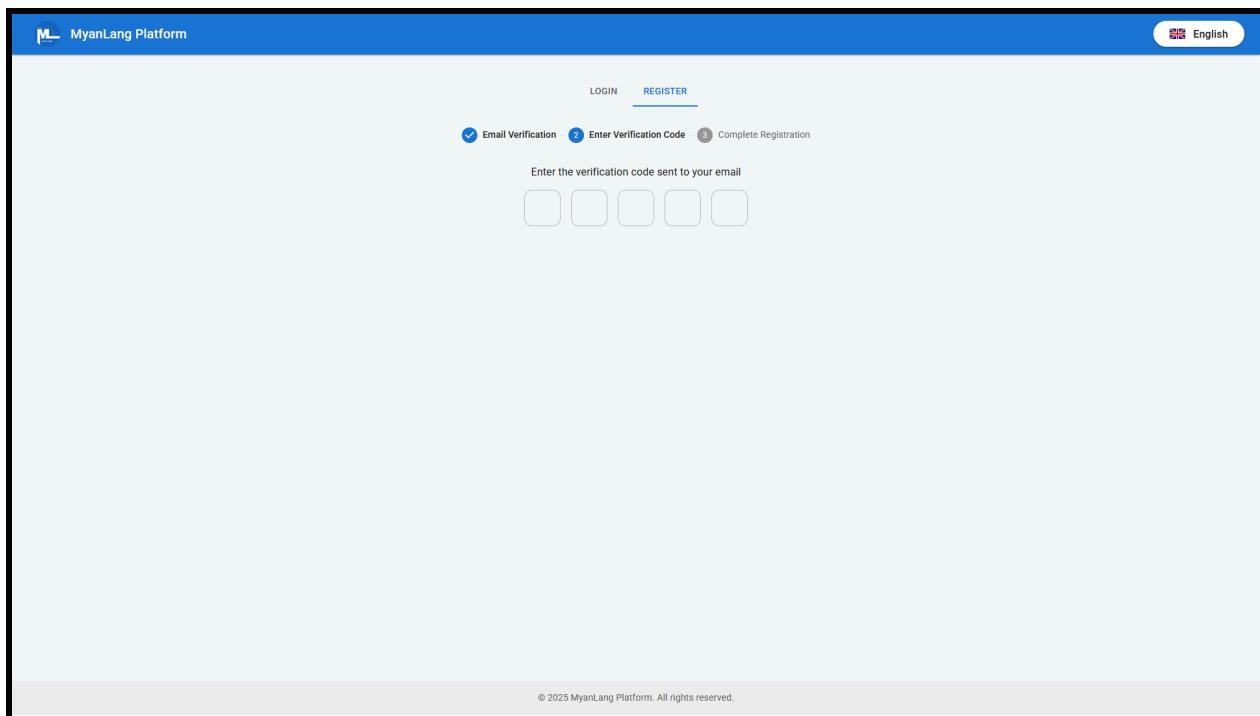


Figure 109: Register step 1 page (entering verification code)

The screenshot shows the 'REGISTER' step of the registration process. At the top, there are three tabs: 'LOGIN', 'REGISTER' (which is underlined), and 'Forgot Password?'. Below these are three checkboxes: 'Email Verification' (checked), 'Enter Verification Code' (checked), and 'Complete Registration' (unchecked). The main form fields include 'Full Name \*', 'Year of Birth', and 'Password \*'. A note below the password field states: 'Password must meet the following requirements: At least 8 characters, At least one uppercase letter, At least one lowercase letter, At least one number, At least one special character'. There is also a checkbox for 'I agree to the terms and conditions' and links to 'Terms and Conditions' and 'Privacy Policy'. A large blue 'Register' button is at the bottom.

*Figure 110: Register step 3 page (filling personal information)*

The screenshot shows the 'Jobs Dashboard' section of the admin dashboard. On the left, a sidebar lists 'Jobs Dashboard', 'Payments', 'QA Management', 'Assessment Tasks', and 'User Reports'. The main area displays two job tasks. Task 1 (ID: 5) is titled 'Job Title : Customer Service Reply' and is set for 'Myanmar → China'. It has 20 total tasks, a 10 min time/task, and a 100.00 MMK price/task. Instructions are: 'Please translate the followings...'. Task 2 (ID: 7) is titled 'Job Title : Myanmar Creative writing' and is set for 'Myanmar → Myanmar'. It has 20 total tasks, a 30 min time/task, and a 500.00 MMK price/task. Instructions are: 'Please write creatively'. At the bottom left is a 'Logout' link, and at the bottom right is a blue '+' button.

*Figure 111: Job Management section of Admin Dashboard (main page for admin)*

The screenshot shows the 'Assessment Tasks' section of the Admin Dashboard. On the left, a sidebar menu includes 'Jobs Dashboard', 'Payments', 'QA Management' (which is highlighted in blue), 'Assessment Tasks' (also highlighted in blue), and 'User Reports'. The main area displays two assessment jobs: 'Myanmar to Chin Assessments' (Job #4) and 'Myanmar to Myanmar Creative Writing' (Job #6). Each job card shows the source language ('Myanmar'), target language ('Chin' or 'Myanmar'), total tasks (20), time per task (10 min), and a three-dot menu icon. A large blue '+' button is located in the bottom right corner of the main content area.

Figure 112: Assessment Task Manage Section of Admin Dashboard

The screenshot shows the 'QA Member Manage' section of the Admin Dashboard. The sidebar menu is identical to Figure 112. The main area displays two QA members: 'John' (ID: 3) and 'Aung Khant Phyto' (ID: 2). Each member card includes a profile picture, ID number, and a 'Performance Metrics' section showing 'Reviewed' (0 vs 14), 'Rejected' (0 vs 4), and 'Acceptance' (0.00% vs 77.78%). To the right of each card are 'Reset Password' and 'Remove Member' buttons. A large blue '+' button is located in the bottom right corner of the main content area.

Figure 113: QA Member Manage Section of Admin Dashboard

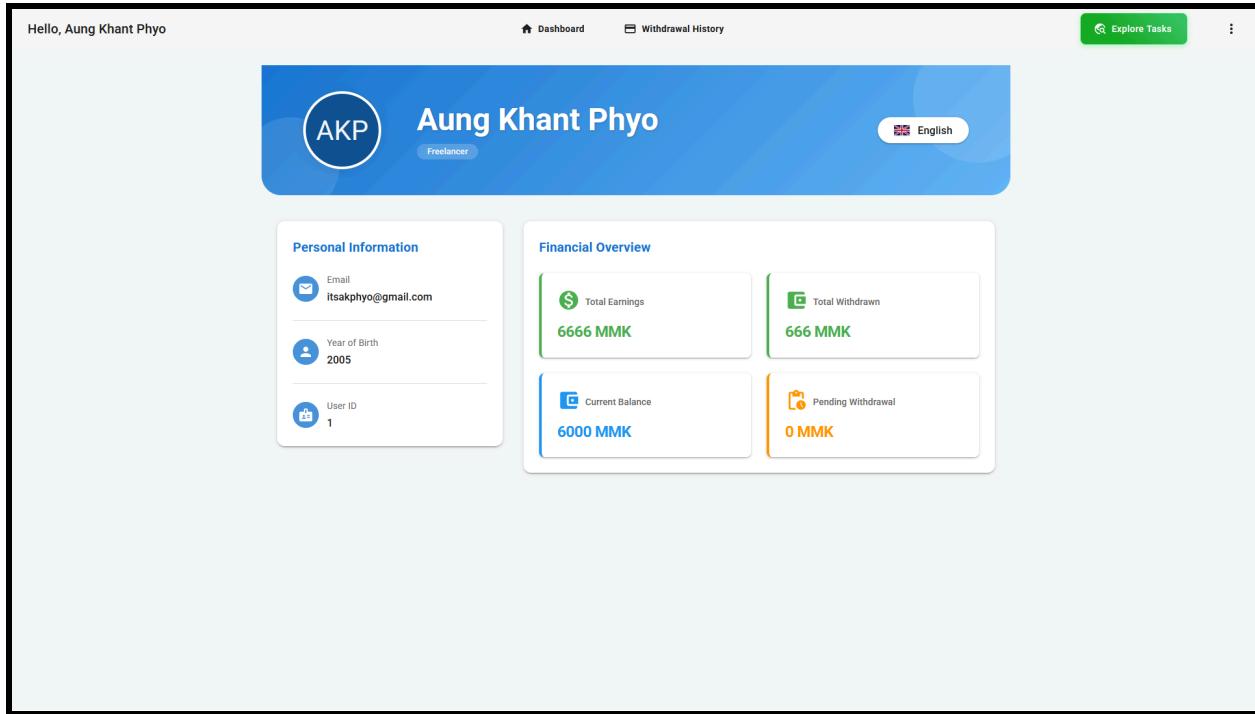


Figure 114: User Profile of Freelancer Portal

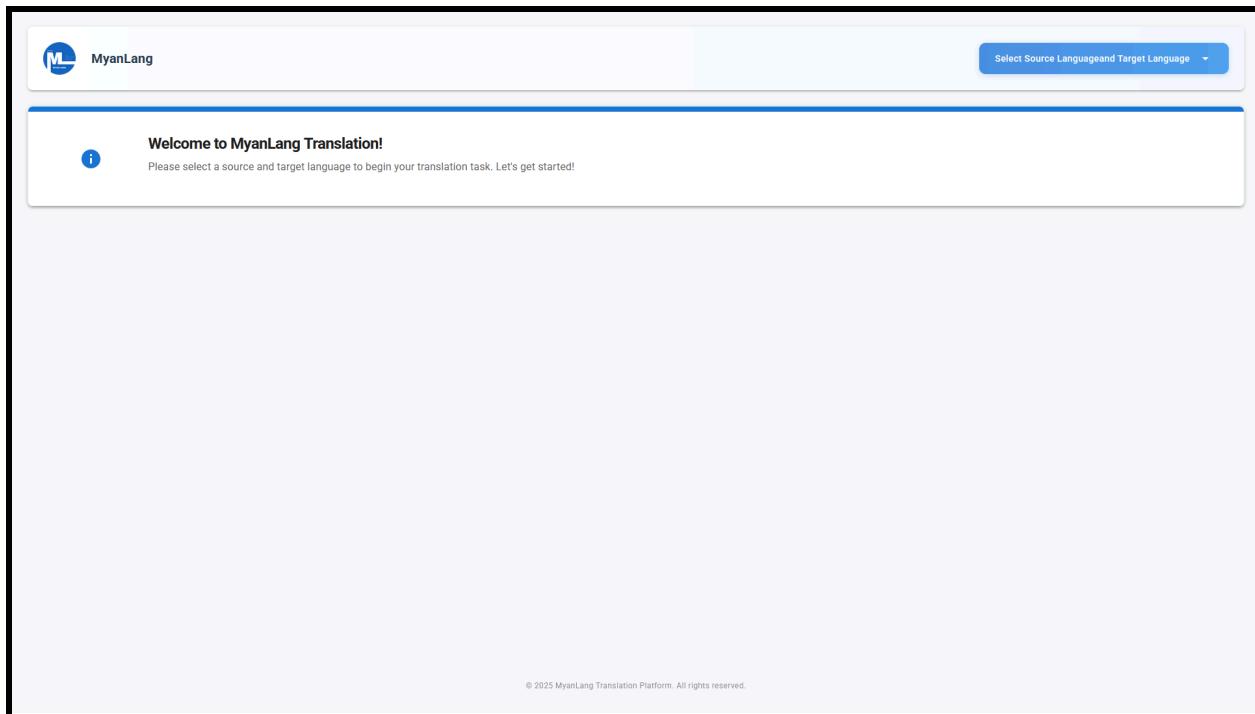


Figure 115: Language Translation Page of Freelancer Portal

The screenshot shows the assessment task submission interface for the Freelancer Portal. At the top left is the MyanLang logo and a status message 'Assessment Required'. At the top right are language selection buttons for 'Myanmar' and a dropdown menu. A blue banner at the top indicates a time limit of '10 minutes'. The main area contains instructions: 'Please write creatively'. Below this are two text input fields: 'Original Text (Myanmar)' containing 'original sentence 5' and 'Your Translation (Myanmar)' containing 'translated sentence 5'. To the right of the translation field is a green circular icon with a white letter 'G'. Navigation buttons at the bottom include '← Previous', 'Task 5 of 5', and a large green 'Submit Tasks' button.

Figure 116: Assessment Task Submission Interface of Freelancer Portal

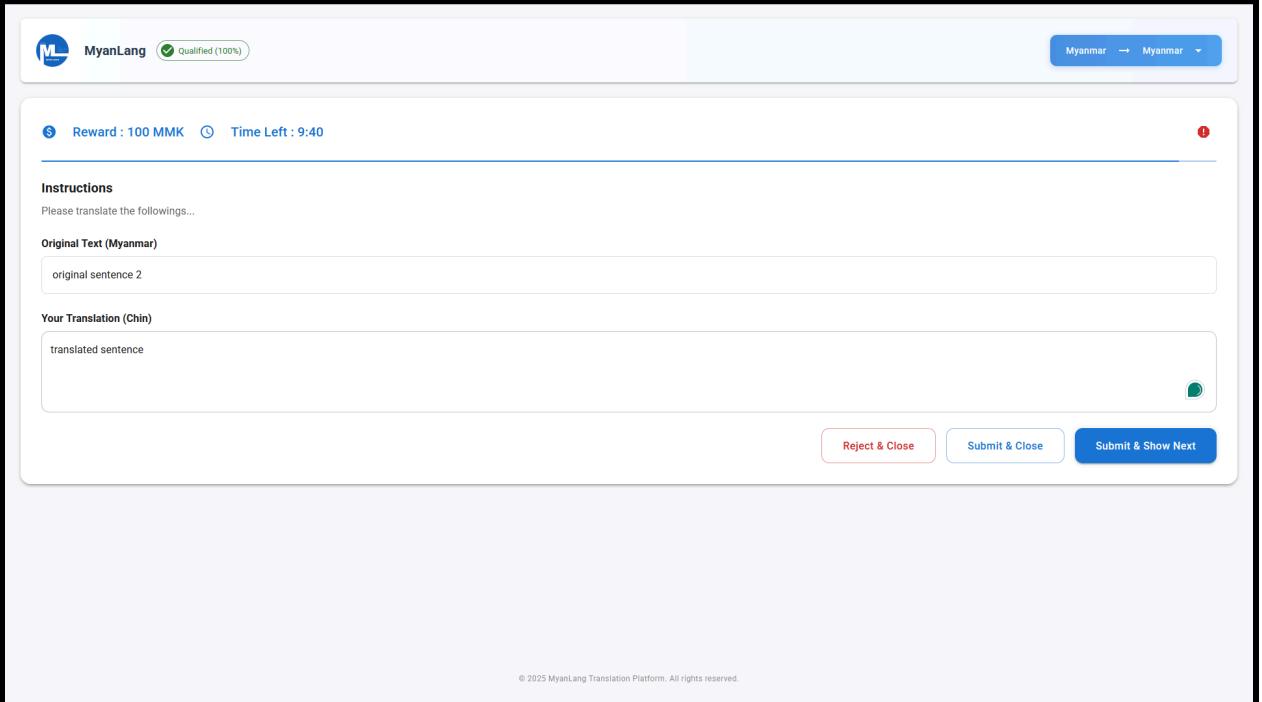
**Batch Translation Review (Myanmar → Myanmar)**

<b>Original Text (Myanmar)</b>	<input type="text" value="original sentence 1"/>	<b>Submitted Text (Myanmar)</b>	<input type="text"/>	<b>Reject</b>	<b>Approve</b>
<b>Original Text (Myanmar)</b>	<input type="text" value="original sentence 2"/>	<b>Submitted Text (Myanmar)</b>	<input type="text"/>	<b>Reject</b>	<b>Approve</b>
<b>Original Text (Myanmar)</b>	<input type="text" value="original sentence 3"/>	<b>Submitted Text (Myanmar)</b>	<input type="text"/>	<b>Reject</b>	<b>Approve</b>
<b>Original Text (Myanmar)</b>	<input type="text" value="original sentence 4"/>	<b>Submitted Text (Myanmar)</b>	<input type="text"/>	<b>Reject</b>	<b>Approve</b>
<b>Original Text (Myanmar)</b>	<input type="text" value="original sentence 5"/>	<b>Submitted Text (Myanmar)</b>	<input type="text" value="translated sentence 5"/>	<b>Reject</b>	<b>Approve</b>

**Cancel** **Submit and Show Next** **Submit All Reviews**

Figure 117: Assessment task review interface of QA member portal

## Section E - Screen Diagrams Sprint 2



The image shows a screenshot of the MyanLang translation platform's task translation interface. At the top left is the MyanLang logo and a green circular badge indicating 'Qualified (100%)'. At the top right is a language selection bar showing 'Myanmar → Myanmar'. Below the header, the reward is listed as 'Reward : 100 MMK' and the time left is 'Time Left : 9:40'. A red exclamation mark icon is visible in the top right corner of the main content area. The main content area has a light gray background and contains the following sections:

- Instructions**: A placeholder text 'Please translate the followings...'.
- Original Text (Myanmar)**: A text input field containing 'original sentence 2'.
- Your Translation (Chin)**: A text input field containing 'translated sentence'.

At the bottom right of the main content area are three buttons: 'Reject & Close' (red), 'Submit & Close' (light blue), and 'Submit & Show Next' (dark blue). A small copyright notice at the bottom center reads '© 2025 MyanLang Translation Platform. All rights reserved.'

Figure 118: Task Translation Interface of Freelancer Portal

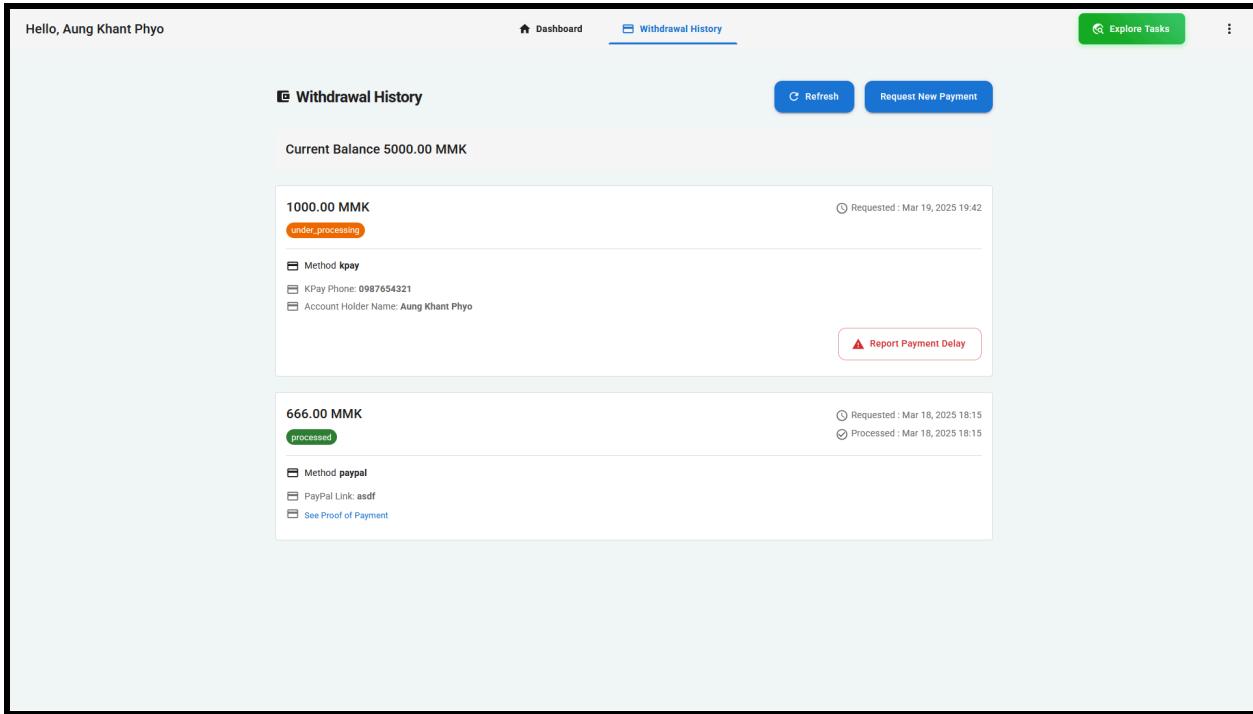


Figure 116: Withdrawal History Page of Freelancer Portal

ID	Freelancer ID	Amount	Method	Requested	Status	Actions
2	1	1000.00 MMK	kpay	3/19/2025, 7:42:59 PM	Processing	<a href="#">View</a>
1	1	666.00 MMK	paypal	3/18/2025, 6:15:09 PM	Complete	<a href="#">View</a>

Figure 120: Payment Page of Admin portal

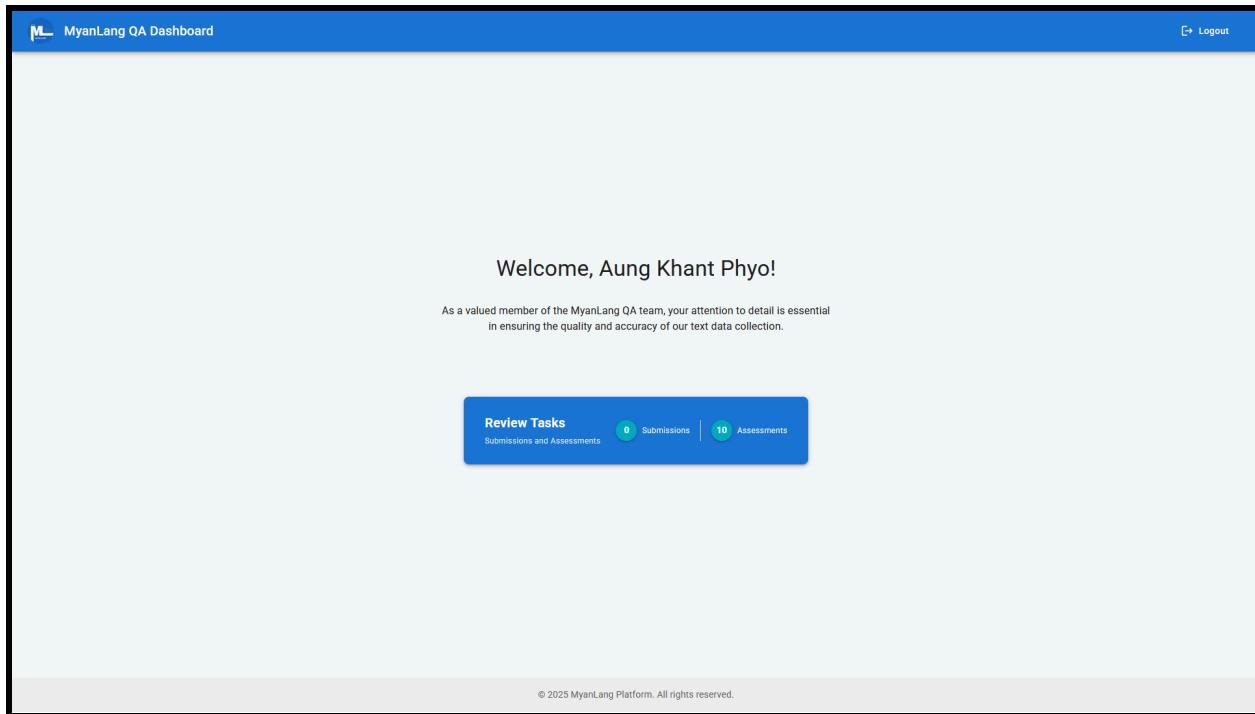


Figure 121: Main page of QA member

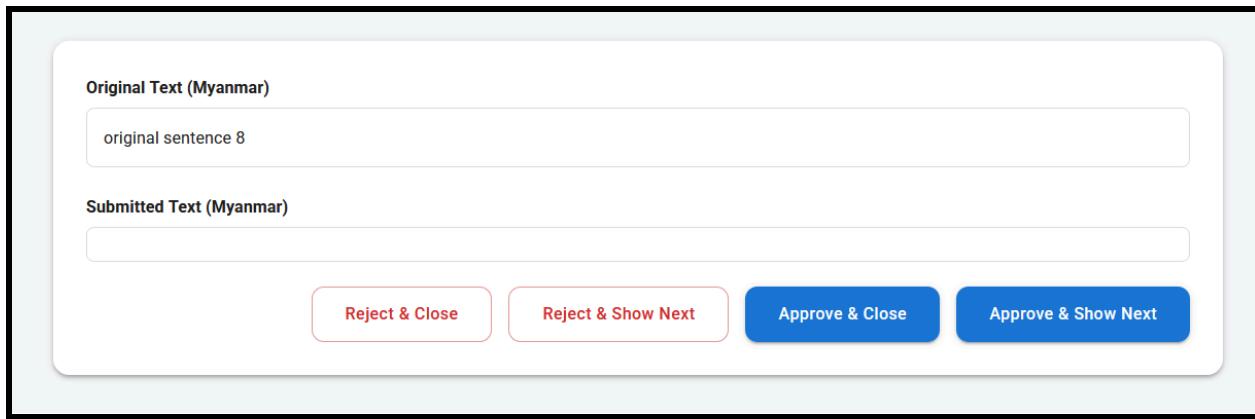


Figure 122: Task Review Interface of QA member

## Section F - Screen Diagrams Sprint 3

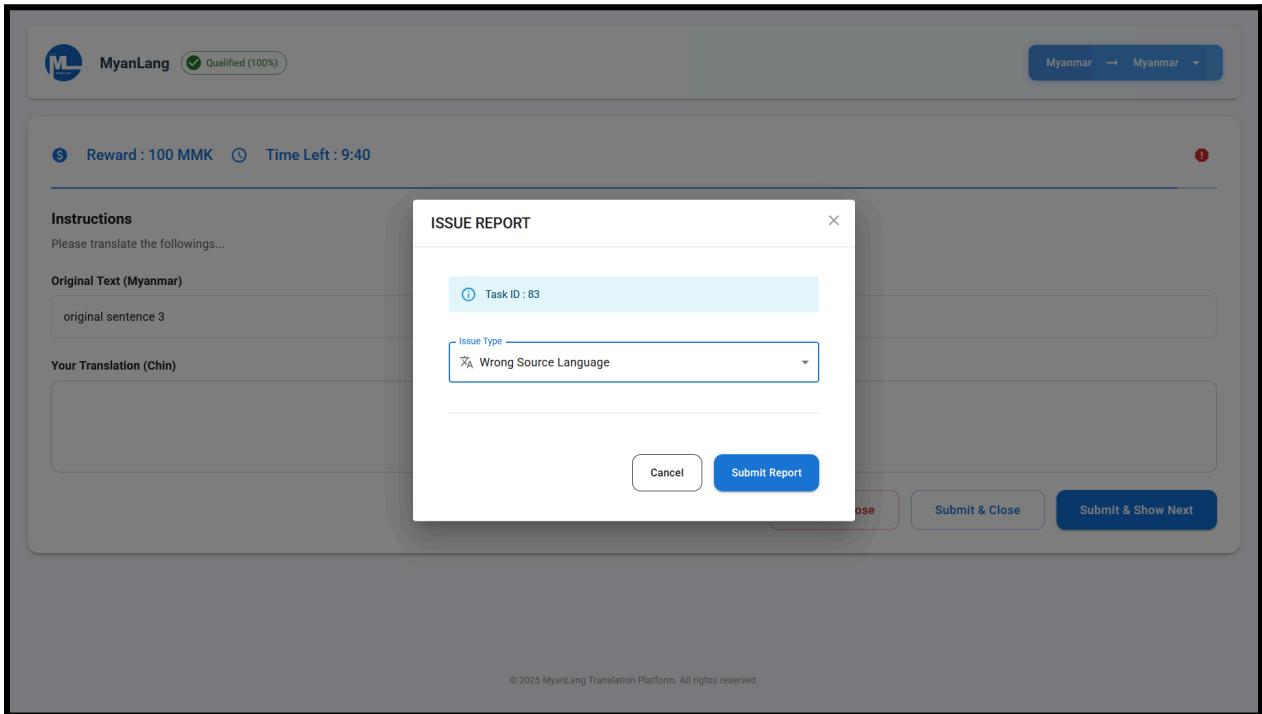
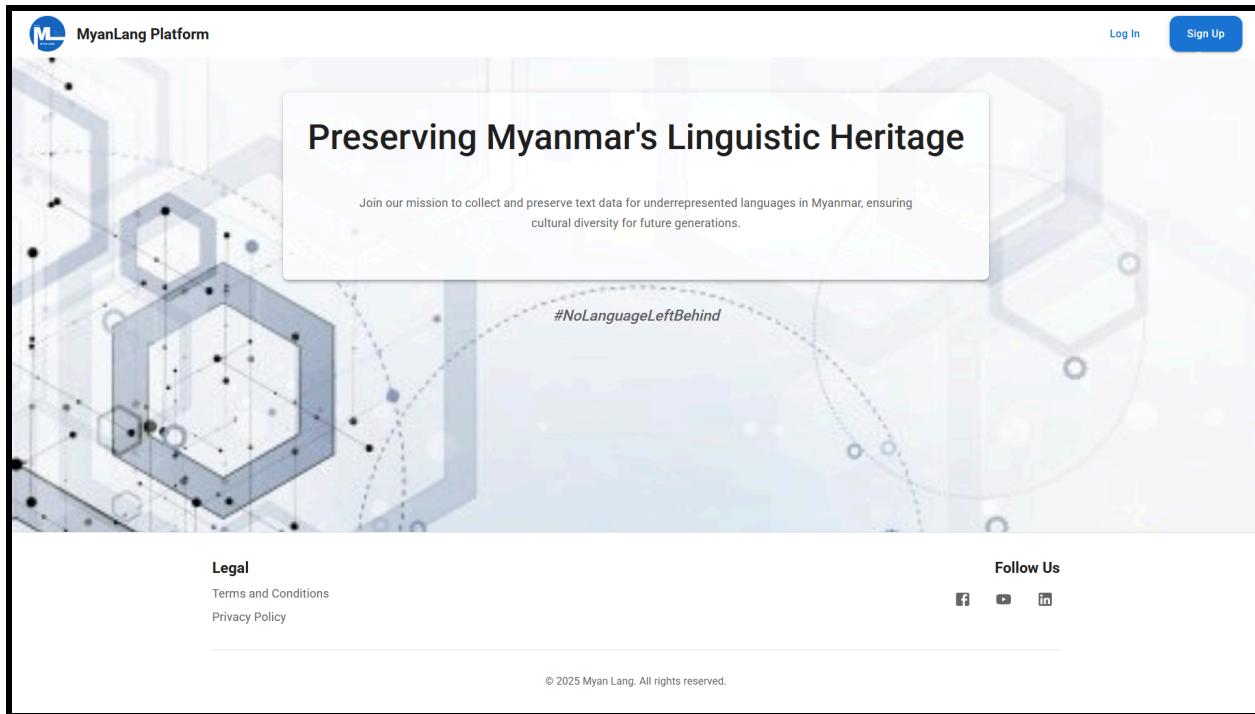


Figure 123: Issue report dialog of Freelancer portal



*Figure 124: Main page of the platform*

Transaction ID	Method	Amount	Status	Date
1	Payoneer	1000.00 ကျပ်	processed	Mar 24, 2025 23:42
2	WavePay	1000.00 ကျပ်	under_processing	Mar 24, 2025 23:42
3	WavePay	1000.00 ကျပ်	under_processing	Mar 24, 2025 23:40

*Figure 125: Withdrawal History Page in Burmese Language*

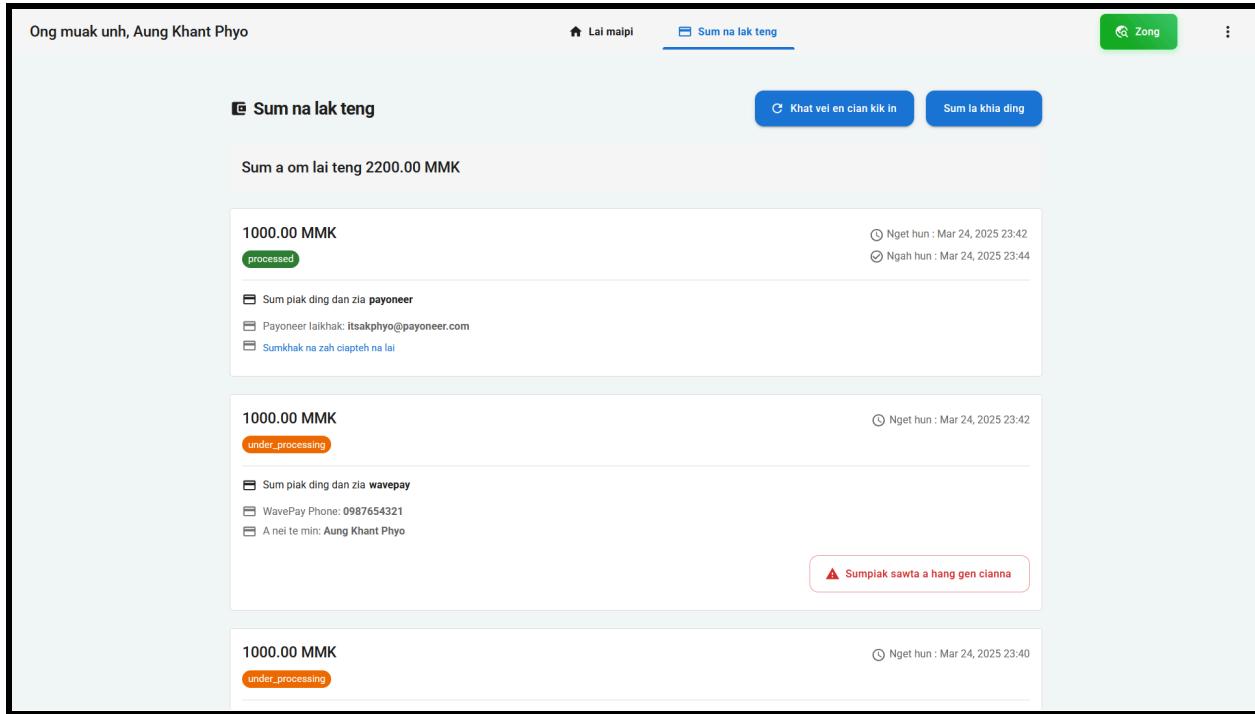


Figure 125: Withdrawal History Page in Chin Language

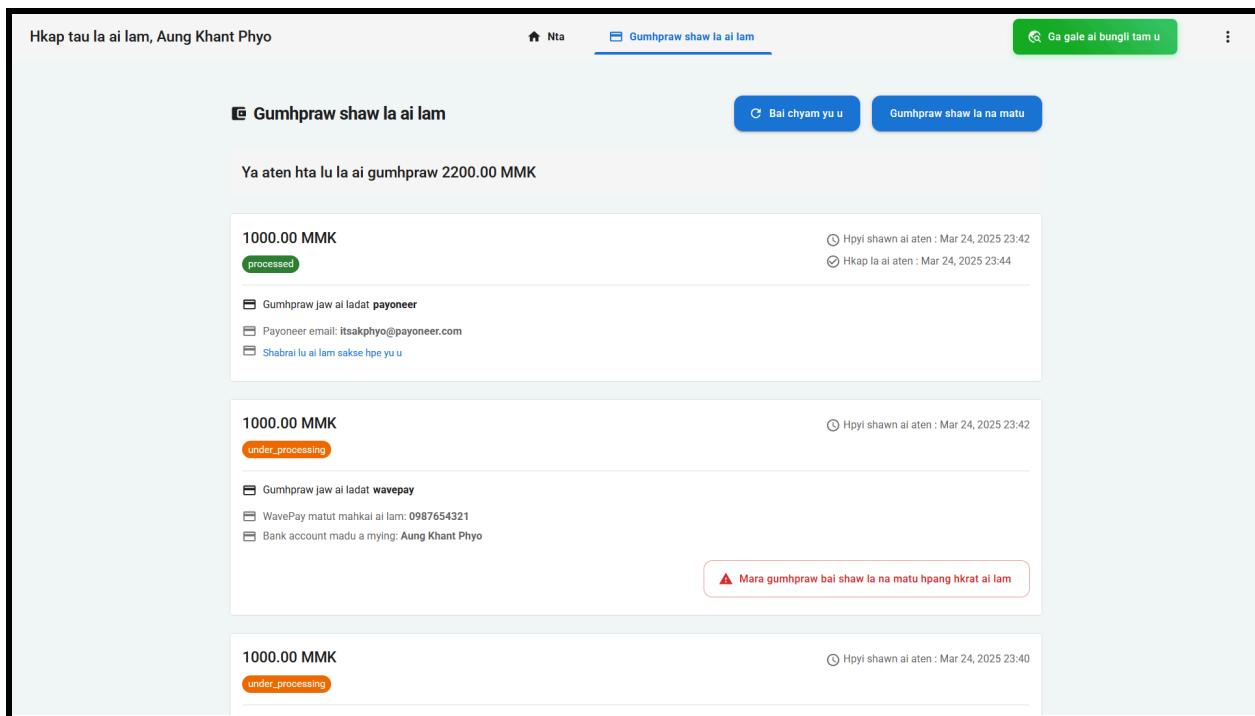


Figure 125: Withdrawal History Page in Kachin Language

## Section G - Source Code Samples

Backend API route that retrieves and assigns an open task based on language criteria and task status.

```
@router.get("/open", response_model=Optional[OpenTaskResponse])
def get_open_task(
    params: OpenTaskRequest = Depends(),
    db: Session = Depends(get_db)
):
    try:
        now = datetime.now(timezone.utc)
        SourceLanguage = aliased(Language)
        TargetLanguage = aliased(Language)

        task_data = (
            db.query(
                Task.task_id,
                Task.source_text,
                Task.translated_text,
                Task.max_time_per_task,
                Task.task_price,
                Job.instructions.label("instruction"),
                SourceLanguage.language_name.label("source_language_name"),
                TargetLanguage.language_name.label("target_language_name"),
                Task.task_status,
            )
            .join(Job, Task.job_id == Job.job_id)
            .join(SourceLanguage, Task.source_language_id ==
SourceLanguage.language_id)
            .join(TargetLanguage, Task.target_language_id == TargetLanguage.language_id)
            .filter(
                Task.source_language_id == params.source_language_id,
                Task.target_language_id == params.target_language_id,
                Task.is_assessment == "f",
                (
                    Task.task_status == "OPEN"
                ) |
                (
                    (Task.task_status == "ASSIGNED_TO_FL") &
                    (Task.assigned_at.isnot(None)) &
                    text("(task.assigned_at + (task.max_time_per_task || ' minutes')::interval) < now()")
```

```

        )
        )
        )
        .order_by(Task.task_id)
        .with_for_update(skip_locked=True)
        .first()
    )

if not task_data:
    return

task_to_update = db.query(Task).filter(Task.task_id == task_data.task_id).first()
task_to_update.assigned_freelancer_id = params.freelancer_id
task_to_update.assigned_at = now
task_to_update.task_status = "ASSIGNED_TO_FL"
db.commit()

return OpenTaskResponse(
    task_id=task_data.task_id,
    instruction=task_data.instruction,
    max_time_per_task=task_data.max_time_per_task,
    price=task_data.task_price,
    source_text=task_data.source_text,
    translated_text=task_data.translated_text,
    source_language_name=task_data.source_language_name,
    target_language_name=task_data.target_language_name,
)

except SQLAlchemyError as e:
    logger.error(f"Database error: {str(e)}")
    db.rollback()
    raise HTTPException(status_code=500, detail="Internal server error. Please try again later.")

except Exception as e:
    logger.error(f"Unexpected error: {str(e)}")
    db.rollback()
    raise HTTPException(status_code=500, detail="An unexpected error occurred.")

```