

# **SAINT MICHAEL COLLEGE OF CARAGA**

---

## **ONLINE REQUEST OF CREDENTIALS FOR SAINT MICHAEL COLLEGE OF CARAGA**

A Capstone Project Presented To  
The Faculty of  
College of Computing and Information Sciences

In Partial Requirements for the Degree  
**BACHELOR OF SCIENCE IN INFORMATION TECHNOLOGY**



By

RICEL ADA S. VILLARIN  
ANGEL MAE L. BAGUIO  
JAY A. BONGADO  
ANGELO M. HINAUTAN  
ROM JONES Y. SALEM

ADVISER: JESSIE S. MAHINAY

APRIL 2025

# SAINT MICHAEL COLLEGE OF CARAGA

SAINT MICHAEL COLLEGE OF CARAGA  
College Of Computing and Information Sciences

## CERTIFICATE OF RESEARCH APPROVAL

This capstone project entitled "**ONLINE REQUEST OF CREDENTIALS FOR SAINT MICHAEL COLLEGE OF CARAGA**" presented and submitted by **RICEL ADA S. VILLARIN, ANGEL MAE L. BAGUIO, JAY A. BONGADO, ANGELO M. HINAUTAN, ROM JONES Y. SALEM** in partial fulfillment of the requirements for the degree of **BACHELOR OF SCIENCE AND INFORMATION TECHNOLOGY** is hereby accepted and recommended for Oral Examination.

**JESSIE S. MAHINAY**  
Adviser

**LEALIL I. PALACIO, MSIT (CAR)**  
Member

**MARISOL S. ROSARIO**  
Member

**MICHELLE ANN G. LUCINO**  
Member

**DAISA O. GUPIT, MIT**  
Member

**MARLON JUHN M. TIMOGAN, MIT**  
Member

**KENNETH IAN B. BARRERA, MA**  
Chairperson

ACCEPTED as partial fulfillment of the requirements for the degree of Bachelor of Science and Information Technology

APPROVED by the tribunal at the Oral Examination with the grade of \_\_\_\_\_.

**JESSIE S. MAHINAY**  
Adviser

**LEALIL I. PALACIO, MSIT (CAR)**  
Member

**MARISOL S. ROSARIO**  
Member

**MICHELLE ANN G. LUCINO**  
Member

**DAISA O. GUPIT, MIT**  
Member

**MARLON JUHN M. TIMOGAN, MIT**  
Member

**KENNETH IAN B. BARRERA, MA**  
Chairperson



---

## COLLEGE OF COMPUTING AND INFORMATION SCIENCES

### ABSTRACT

<b>TITLE:</b>	ONLINE REQUEST OF CREDENTIALS FOR SAINT MICHAEL COLLEGE OF CARAGA
<b>AUTHOR:</b>	Ricel Ada S. Villarin, Angel Mae L. Baguio, Jay A. Bongado, Angelo M. Hinautan, Rom Jones Y. Salem
<b>DEGREE:</b>	Bachelor of Science and Information Technology
<b>ADVISER:</b>	Jessie S. Mahinay
<b>PLACE OF PUBLICATION:</b>	Saint Michael College of Caraga
<b>DATE</b>	April 2025
<b>PAGES</b>	194

### DEVELOPMENTAL RESEARCH

#### I. OBJECTIVES

This study aimed to design, develop, and implement a web-based system titled “Online Request of Credentials for Saint Michael College of Caraga.” The primary objectives of the project were to streamline the credential request process by allowing students and alumni to submit and track their requests online; to integrate a tracking feature that enabled real-time monitoring of request statuses by both users and key personnel such as the Cashier, Head of the LRC, Dean/Principal, and Registrar Staff; and to establish a secure validation process for uploaded identification documents to ensure the authenticity of each request. This system minimized manual workload, reduced processing time, and enhanced the efficiency, transparency, and reliability of academic service delivery.

#### II. METHODOLOGY

The study adopted the Software Development Life Cycle (SDLC) as the primary methodology, specifically utilizing the Waterfall Model to guide each development phase in a linear and structured manner. The process began with requirements gathering, where the researchers conducted interviews and observations with the SMCC Registrar’s Office to identify the current challenges in processing credential requests. The layout, user interface, and system



---

architecture were planned in the system design phase to ensure user-friendliness and functionality. The development phase involved coding and system implementation using modern web development technologies. Once developed, the system underwent testing to identify and correct bugs and validate system features such as online request submission, tracking, and document validation. In the final deployment phase, the system was launched within a controlled environment in the Registrar's Office to evaluate real-world usability and performance. Feedback was collected for future enhancements.

### III. FINDINGS

The Online Request of Credentials System received high ratings across functional suitability, performance efficiency, and usability based on ISO 25010 standards. It earned a total weighted mean of 3.60 in functionality, 3.57 in performance efficiency, and 3.67 in usability, which were classified as "Very Functional," "Very Efficient," and "Very Usable," respectively. These results confirmed the system's effectiveness, reliability, and user-friendliness, with participants recognizing its precision, speed, and intuitive design. Continuous enhancements were recommended to ensure adaptability and maintain high standards.

### IV. RECOMMENDATIONS

To enhance the system's effectiveness, future efforts should focus on expanding user access for scalability, integrating advanced data validation and security features, improving the real-time tracking interface, offering user training for seamless navigation, and incorporating analytics for monitoring request trends. These improvements would further strengthen efficiency, reliability, and user satisfaction in processing academic credentials.

**KEYWORDS:** *Online Credential Request, Web-Based System, Real-Time Tracking, System Usability,*

*Functional Suitability*



---

## DEDICATION

This capstone research is dedicated with heartfelt appreciation to all individuals who played a meaningful role in the journey and success of this endeavor.

To the families of the researchers, sincere gratitude is extended for their unwavering love, patience, and support. Their constant encouragement and understanding served as a source of strength and inspiration, especially during the most challenging phases of this academic pursuit.

To Mr. Jessie S. Mahinay, the project adviser, the researchers express deep appreciation for his guidance, patience, and insightful contributions. His constructive feedback and steadfast support were crucial in shaping the direction and outcomes of this study. His commitment has greatly influenced the academic growth and confidence of the researchers as emerging professionals.

This project is also the result of collective effort, determination, and collaboration among the research team members. Every contribution, whether large or small, played a vital role in the successful development of a system designed to enhance the institution's academic services and support the needs of the Michaelinian community.

The completion of this capstone research would not have been possible without those who provided support throughout the journey. Their presence and encouragement are deeply appreciated.



## ACKNOWLEDGEMENTS

Foremost, the researchers extend their deepest gratitude and praise to Almighty God for His unwavering guidance, wisdom, and strength throughout this capstone project. Through His grace, the completion of this work was made possible, enabling the team to overcome challenges and persevere.

Sincere appreciation is extended to Mr. Jessie S. Mahinay for his outstanding mentorship and continuous support as the project adviser. His expertise, insightful feedback, and encouragement played a vital role in shaping the direction and outcome of this study.

The researchers also thank the instructors and committee members for their valuable input and constructive suggestions. Their guidance significantly contributed to the enhancement of the project's quality.

Gratitude is expressed to the researchers' families for their unwavering love, understanding, and emotional and financial support. Their encouragement was a constant motivation and strength throughout this academic journey.

Acknowledgement is also given to the Tertiary Education Subsidy for the financial assistance provided, which allowed the researchers to focus entirely on their academic responsibilities without added burden.

Special thanks are due to friends, fellow students, and mentors whose advice, support, and encouragement greatly contributed to the progress and completion of this work. Their involvement, whether direct or indirect, is sincerely appreciated.

Lastly, the researchers express heartfelt thanks to all individuals who offered support in any form. Every contribution, regardless of scale, made a meaningful impact on the successful realization of this capstone project.



## TABLE OF CONTENTS

	Pages
TITLE PAGE	i
CERTIFICATE OF RESEARCH APPROVAL	ii
ABSTRACT	iii
DEDICATION	v
ACKNOWLEDGEMENT	vi
TABLE OF CONTENTS	vii
LIST OF FIGURES	viii
LIST OF TABLES	xii
CHAPTER	
1                  INTRODUCTION	1
1.1 Project Context	1
1.2 Objective of the Study	2
1.3 Scope and Limitations	3
1.4 Definition of Terms	4
2                  REVIEW OF RELATED LITERATURE	5
3                  SOFTWARE REQUIREMENTS AND DESIGN SPECIFICATION	11
3.1 System Architecture	11
3.2 Conceptual Diagram	12
3.3 Use Case Diagram	13
3.4 Activity Diagram	14
3.5 Sequence Diagram	19
3.6 ERD (Database Design)	23
3.7 User Interface Design (Prototype)	24
3.8 Software Platforms, Development Environments and Tools	52
3.9 Hardware Requirements	53
3.10 Ethical Standard	54
4                  SOFTWARE DEVELOPMENT AND TESTING	59
4.1 Development Process	59
4.2 Testing Process	145
- Accuracy Test	147
- Functionality Test	148
- Performance Test	149
5                  SUMMARY, CONCLUSIONS AND RECOMMENDATIONS	151
5.1 Summary of Findings	151
5.2 Conclusion	152
5.3 Recommendations	153
REFERENCES	154
APPENDICES	158
CERTIFICATE OF TECHNOLOGY BASED ASSESSMENT	180
PUBLISHABLE JOURNAL	




---

## LIST OF FIGURES

<b>Figure No.</b>	<b>Title</b>	<b>Page</b>
1	System Architecture	11
2	Conceptual Diagram	12
3	Use Case Diagram	13
4	Activity Diagram – Students/Alumni	14
5	Activity Diagram - Admin	15
6	Activity Diagram – Head of LRC/Dean and Principal	16
7	Activity Diagram – Cashier	17
8	Activity Diagram – Registrar Staff	18
9	Sequence Diagram – Students/Alumni	19
10	Sequence Diagram – Admin	20
11	Sequence Diagram – Head of LRC/Dean and Principal	21
12	Sequence Diagram – Registrar Staff	22
13	Entity Relationship Diagram – Database Design	23
14	Admin/All Users - Login Page	24
15	Students/Alumni - Sign-Up Page	25
16	Students/Alumni - Dashboard	26
17	Students/Alumni – Request Clearance	27
18	Students/Alumni - Payment for Clearance	27
19	Students/Alumni - Payment for Requested Credentials	28
20	Students/Alumni - Clearance Status	28
21	Students/Alumni – Application Form	29
22	Students/Alumni – Feedback Form	30
23	All Users – Activity Logs	31
24	Admin - Dashboard	31
25	Admin – Requesters List	32
26	Admin – Users List	33
27	Admin – Department List	33
28	Admin – Course List	34
29	Admin – Document Type List	35




---

30	Admin – Payment Account	35
31	Admin - ISO Form Code	36
32	User (Cashier) - Dashboard	37
33	User (Cashier) – Clearance List	37
34	User (Cashier) – Clearance Disapproved List	38
35	User (Cashier) – Payment Verification	39
36	User (Cashier) – Report for Clearance	40
37	User (Dean/Principal / Head of LRC) – Dashboard	41
38	User (Dean/Principal / Head of LRC) – Clearance List	41
39	User (Dean/Principal / Head of LRC) – Clearance Disapprove List	42
40	User (Dean/Principal / Head of LRC) – Report for Clearance	43
41	User (Registrar Staff) – Dashboard	44
42	User (Registrar Staff) – Clearance List	44
43	User (Registrar Staff) – Clearance Disapprove List	45
44	User (Registrar Staff) – Request Details	46
45	User (Registrar Staff) – Processing of Documents	47
46	User (Registrar Staff) – Schedule Release Date	47
47	User (Registrar Staff) – Feedback List	48
48	User (Registrar Staff) – Report for Clearance	49
49	User (Registrar Staff) – Report for Requested Credentials	50
50	User (Registrar Staff) – Request History Report	51
51	Input-Process-Output Diagram	59
52	Admin/All Users - Login Page	61
53	Admin/All Users – Login Error Notification	61
54	Code Snippet for Admin/Users – login.php	62
55	Students/Alumni - Sign-Up Page	64
56	Code Snippet for Students/Alumni – register.php	65
57	Admin – Dashboard	67
58	Code Snippet for Admin – dashboard_admin.php	67
59	Admin – Requesters List	69
60	Code Snippet for Admin – students.php	70

---




---

61	Admin – Add User	71
62	Admin – User List	72
63	Admin – Edit User	72
64	Admin – Reset User Password	73
65	Code Snippet for Admin – users.php	74
66	Admin – Add Department	76
67	Admin – Department List	76
68	Admin – Edit Department	77
69	Code Snippet for Admin – departments.php	78
70	Admin – Add Course	80
71	Admin – Course List	80
72	Admin – Edit Course	81
73	Code Snippet for Admin – course.php	82
74	Admin – Add Document Type	84
75	Admin – Document Type List	84
76	Admin - Edit Document Type Information	85
77	Code Snippet for Admin – document_types.php	87
78	Admin – Payment Account List	88
79	Code Snippet for Admin – payment.php	89
80	Admin – ISO Form Code List	90
81	Code Snippet for Admin – iso_form_code.php	91
82	Cashier – Dashboard	93
83	Code Snippet for Cashier – dashboard_cashier.php	93
84	Cashier/Head of LRC/ Dean/Principal/Registrar – Clearance List	94
85	Code Snippet for Cashier/Head of LRC/Dean/Principal/Registrar – Clearance List	96
86	Cashier/Head of LRC/Dean/Principal/Registrar – Disapproved List	97
87	Code Snippet for Cashier/Head of LRC/Dean/Principal/Registrar – Disapproved List	98
88	Cashier – Payment List	100
89	Code Snippet for Cashier – request_verification.php	101
90	Cashier – Clearance Report	103

---




---

91	Code Snippet for Cashier – report_clearance_cashier.php	104
92	Head of LRC/ Dean/Principal – Dashboard	106
93	Code Snippet for Head of LRC / Dean/Principal dashboard_others.php	107
94	Head of LRC/ Dean/Principal – Clearance Report	108
95	Code Snippet for Head of LRC – report_clearance_librarian.php	109
96	Code Snippet for Dean/Principal – report_clearance_dean.php	111
97	Registrar – Dashboard	113
98	Code Snippet for Registrar – dashboard_registar_php	114
99	Registrar - List of Requested Documents	116
100	Registrar – Request Details	116
101	Code Snippet for Registrar – request_document.php	118
102	Registrar – For Processing	120
103	Code Snippet for Registrar – for_processing.php	121
104	Registrar – Schedule Release Date	122
105	Code Snippet for Registrar – schedule_release.php	123
106	Registrar – Feedback List	124
107	Code Snippet for Registrar – request.php	125
108	Registrar – Clearance Report	127
109	Code snippet for Registrar – report_clearance_registrar.php	128
110	Registrar – Report Requested Credentials	129
111	Code Snippet for Registrar – report_credential_registrar.php	131
112	Registrar – Request History Report	133
113	Code Snippet for Registrar – report_request_registrar.php	134
114	Students/Alumni – Dashboard	136
115	Code Snippet for Students/Alumni – dashboard_students.php	136
116	Students/Alumni – Clearance Form (Department and Course Identification)	138
117	Students/Alumni – Clearance Status	138
118	Students/Alumni – Request Form	139
119	Students/ Alumni – Payment Details	140
120	Students/Alumni – Feedback Form	140
121	Code Snippet for Students/Alumni – request.php	142

---



## **LIST OF TABLES**

Table	Page	
1	Software Requirements	52
2	Hardware Requirements	53
3	Functional Suitability	145
4	Performance Efficiency	145
5	Usability	146
6	Respondent's Distribution	146
7	Functional Suitability	147
8	Performance Efficiency	148
9	Usability	149
10	Summary Table of the Over-all Mean and Grand Distribution of the Acceptability Level	150



---

## CHAPTER 1

### INTRODUCTION

#### 1.1 Project Context

The Office of the Registrar aids students, faculty, staff, and alumni in various academic tasks, such as overseeing academic policies and procedures, managing student records, facilitating student registration, handling transcripts, coordinating course scheduling, providing commencement services, performing degree audits, offering assessment and reporting, and ensuring institutional compliance. The SMCC Registrar's Office continues to process requests for academic credentials; however, it has become increasingly burdensome for the staff, mainly because of the time-consuming manual processing and the high volume of requests they manage. As the number of credential requests grows, the risk of errors and delays also increases, further restraining the resources available.

According to Devikah et al., educational institutions frequently must provide academic transcripts and certificates upon student request, as these documents are essential for applying to new programs and job placements. Doing this manually can be time-consuming and require much work [1]. Also, fulfilling requests on the agency side can be just as frustrating as asking for them on the constituent side [2].

In today's digital era, technological advancements are essential in shaping credential management, as mentioned by I. E. Khairuddin et al. [3]. Additionally, K. I. R. Abang et al. stated that in any bustling institution, it is essential that each accomplished student's request be prompt and efficient [4]. To cope with these challenges, they need to grab the opportunities that technology can offer and test innovations that are not only in response to the global threat challenges but also to the future [5]. Thus, making a system that provides an online requisition of



---

credentials is wiser. In this case, embracing more effective methods in today's advancing technological landscape can aid in maintaining organization and improving productivity [4].

Throughout the years, manual credential requests have been carried out across most educational institutions, specifically at the Office of the Registrar in Saint Michael College of Caraga. The current system for requesting academic credentials at SMCC heavily relies on manual methods. Students and alumni must submit requests in person or through physical forms, leading to inefficiencies and delays in processing.

To address the issue of inefficient and cumbersome credential requests at the SMCC Registrar, an essential solution is to create a web-based system that simplifies the process of requesting academic credentials. This system can facilitate the quick and secure issuance of credentials in the Registrar's office. Students and alumni could submit requests electronically, track their status in real time, and experience reduced wait times and increased transparency. The online platform would automate much of the manual processing for administrative staff, reducing paperwork and streamlining workflow. Thus, it makes processes more accessible and convenient for everyone.

The SMCC registrar needs to implement and optimize the online request system to address existing problems in the institution. Implementing an online request system can significantly enhance the efficiency and effectiveness of administrative processes within educational institutions. The system automates the request management process, making tracking and fulfilling requests more streamlined. This transition from a paper-based system to a digital platform not only improves accessibility for students but also fosters a more organized and transparent workflow for administrative staff.



---

## 1.2 Objective of the Study

The researchers' study aimed to design, develop, and implement an Online Request for Credentials for Saint Michael College of Caraga.

Specifically, it aimed to:

1. Generate comprehensive reports on request volumes and processing times for transparency and ongoing process improvements.
2. Develop a tracking feature that enables users to monitor the real-time status of their credential requests while allowing key personnel, such as the Cashier, Head of the LRC, Dean/Principal, and Registrar Staff, to review and confirm requests with incomplete or pending requirements for clearance purposes.
3. Establish a validation process for verifying the validity of the uploaded identification document.

## 1.3 Scope and Limitations

The Online Request of Credentials System was specifically designed for Saint Michael College of Caraga. The system accommodated a single administrator account (Head of the Registrar's Office), along with designated key personnel, including the Cashier, Head of the LRC, Dean/Principal, and Registrar Staff, all authorized staff members. The admin had full access to all system features, including viewing and verifying the entire process. At the same time, the key personnel verified the requirements for credential requests.

The capstone project did not support credential requests from institutions outside the project. The daily request processing limit was set at 10, deferring additional requests to the following day. Although it incorporated necessary security features, the system did not include a detailed review or update of institutional data protection practices. Limitations also included specific software and hardware requirements, such as adequate RAM, processor capability,



---

storage, and compatibility with modern web browsers and development platforms. Furthermore, the system's functionality was restricted to processing credential requests, excluding other administrative tasks.

#### **1.4 Definition of Terms**

The terms defined below explain the basic concepts and factors discussed during the conceptualization and development of the Online Request of Credentials System and provide contextual information on how the researchers applied these concepts in their study.

**Access Control** – Mechanisms within the Online Request of Credentials System that regulate who can view, modify, or manage different aspects of the system, ensuring that only authorized personnel have access to sensitive functions and data.

**Admin** – A user role within the Online Request of Credentials System with comprehensive access to all system features. The admin can view, verify, and manage all aspects of the credential request process, including user management.

**Request Tracking** — This feature within the Online Request of Credentials System allows users to monitor the status of their credential requests in real time, including progress updates and estimated completion times.

**User Validation** – This ensures that user inputs are accurate, secure, and complete before processing any request. In ORCSMCC, it includes validating the essential information provided by end-users during the sign-up and log-in processes.

**Web-Based System** – A platform accessible through internet browsers, enabling users to interact with the system remotely without installing specific software.

**Workflow Automation** – The use of automated processes within the Online Request of Credentials System to handle repetitive tasks, such as routing requests and updating status, thereby improving efficiency.

---



---

## CHAPTER 2

### REVIEW OF RELATED LITERATURE

This chapter shows the literature related to the researchers' study to address the existing problems encountered. It focuses on the need to implement the researchers' proposed software application, its benefits, and the issues it addresses. The literature comes from electronic materials, e-books, websites, articles, and journals.

#### ***Efficiency and Transparency in Credential Request Systems***

The shift towards digitizing administrative processes in educational institutions has led to the development of online credential request systems, enhancing the issuance, verification, and management of academic credentials. These systems address the inefficiencies of manual processes, offering improved accuracy and speed. However, they also bring challenges related to data security, system compatibility, and user privacy.

The advancements in online credential request systems have transformed how credentials are created, managed, and verified, enhancing trust and efficiency in various sectors. These systems facilitate the issuance of digital credentials, allowing for interoperability and recognition across borders, which is crucial in a globalized job market [6].

According to K. I. R. Abang et al., the transition from manual to online credential request systems in higher education has notably reduced administrative bottlenecks. For instance, the City College of Tagaytay (CCT) implemented an online credential request system during the pandemic, providing a safer, faster, and more efficient solution for students and administrators [4]. However, M. Chase et al. emphasized that challenges such as ensuring data trustworthiness, managing user privacy, and maintaining usability persist [7], and institutions must address issues related to data security, particularly when handling sensitive information, to prevent potential breaches [8]. Furthermore, G.W. Matkin also concluded that institutions must adapt to these



---

changes, balancing the needs of students and employers while navigating the complexities of digital credentialing [9].

A similar system was developed by Alegado et al. for NUEST, where they introduced a web-based chat box request system to facilitate online document requests. Built using web technologies like HTML, CSS, PHP, and MySQL, this system simplified the request process, reducing the time and effort required for both the user and staff. It garnered positive feedback, with experts and users rating its effectiveness highly [10]. Despite these advantages, the issue of data privacy and security remains crucial, particularly given the increasing dependence on digital platforms in educational settings.

Heather Flanagan's insight on digital identity systems underscores the importance of security in such platforms. Flanagan notes that while digital credentials open opportunities for innovation and efficiency, they also come with inherent risks. Addressing the risks involves implementing comprehensive security and privacy that balance the needs of various stakeholders [11]. Also, ensuring user data privacy during verification processes is crucial, as systems must balance transparency with confidentiality [12]. This highlights the need for similar security protocols in the online credential request system to protect sensitive student information and comply with evolving digital identity standards [11].

I. Keck et al.'s study further discusses the technical challenges, such as system incompatibilities and the need for specialized personnel to manage the transitions. Their study emphasizes the importance of ensuring that such systems are user-friendly. They propose a framework for evaluating credential systems' effectiveness and ensuring their sustainability in the long term [13].

Rassameeroj et al. explored the impact of online systems on reducing errors and enhancing accuracy in credential processing. They found that automated systems significantly

---



---

reduced manual errors in document handling and data entry, leading to faster and more reliable services for students. However, they also raised concerns about system downtimes and the need for continuous software updates to maintain reliability [14].

The literature reflects a consensus on the potential benefits of online credential request systems, particularly regarding efficiency, accuracy, and accessibility. However, as Mishra R. et al. noted, these systems also require ongoing investment in infrastructure, security, and training to meet the evolving needs of both institutions and users. Moreover, the study emphasized the importance of ensuring data protection compliance [15].

While online credential request systems offer significant benefits to higher education institutions, including faster processing times and enhanced accuracy, their implementation requires careful consideration of technical, security, and operational challenges. The transition from manual to online systems represents a paradigm shift that can improve academic institutions' overall efficiency. Still, regular updates, proper staff training, and stringent data security measures must accompany it.

#### ***Credential Request Processes Through Automation and Real-Time Tracking***

The transition to digitized credential request and approval processes has revolutionized administrative workflows in educational and professional institutions. These systems offer increased efficiency, accessibility, and security, enabling users to request, process, and verify credentials seamlessly. It highlights the critical verification steps, eligibility assessment, and the authorities' role in ensuring compliance with institutional standards. The process typically involves multiple steps, including documentation review, verification of qualifications, and recommendations from authorized personnel.

According to Gaikwad et al., institutions are moving towards digitizing the credential request systems to streamline administrative processes, thereby increasing convenience for

---



---

students and alumni. Online platforms allow users to submit requests for academic documents, such as transcripts and certificates, at any time and location, significantly reducing barriers [16]. Gune et al. explained that the approval process for credential requests involves several key steps and methodologies that ensure efficiency and security. Initially, a request is submitted and categorized based on type to determine the appropriate approver. This is facilitated by systems that utilize visual representations of approval elements, allowing users to define their processes effectively [17]. Chiahsin et al. also stated that credential verification may involve multiple service providers, ensuring that the requested credentials are validated securely [18].

Integrating digital certificates and facial recognition technology can streamline verification, reduce administrative burdens, and enhance security against counterfeiting [19]. Furthermore, non-fungible tokens (NFTs) and blockchain technology provide a decentralized, secure method for issuing and validating academic credentials, ensuring compliance with regulations like GDPR [20]. Blockchain solutions can accelerate verification processes by approximately 8% while maintaining data integrity [21]. Countries like China and the U.S. offer valuable insights into competency validation, which can be adapted to improve systems in other regions [22]. Lastly, a design thinking approach to distributed ledger technology can revolutionize credential verification, ensuring transparency and fraud prevention [23].

A web-based system can streamline the approval request process, making it more efficient. While these systems enhance the approval process, challenges such as manual errors and inefficiencies in traditional methods persist, highlighting the need for continuous improvement in approval workflows.

#### ***Validation of Uploaded Documents through Admin Review in Online Credential Systems***

Document validation is a crucial element in online credential systems. Effective document verification processes ensure that uploaded documents, such as identification cards and receipts,

---



---

adhere to institutional requirements, safeguarding the integrity and security of transactions. As the rapid development of information sharing and exchange continues, an increasing number of businesses and individual users are adopting digital documents due to their convenience and efficiency [24]. However, while digital documents simplify accessibility, verifying their authenticity remains a significant challenge [25].

The implication arising from the problem of fake documentation is that it is causing serious and alarming impacts and needs to be urgently considered. This technological revolution, while enabling greater flexibility in document handling, has also introduced vulnerabilities. Validating documents remains laborious and time-consuming, further exacerbating concerns about fraudulent documents [26]. The implications of counterfeit documents are severe, posing risks to security, institutional trust, and transactional integrity. Addressing this issue requires immediate attention and adopting modern, robust methods for issuing and verifying significant documents to safeguard against potential misuse [27].

According to Hamidi, H., inaccurate verification methods can lead to significant security vulnerabilities, allowing unauthorized access to sensitive information and systems. High error rates in scanned identity verification may increase susceptibility to identity theft and other fraudulent activities, underscoring the necessity of precise assessments [28]. However, C. Avram et al. stated that using uploaded IDs and documents for validation enhances security. This process helps reduce identity theft and unauthorized access to sensitive information, creating a safer online environment [29]. Using copies of official documents, such as birth certificates or government IDs, allows organizations to validate users' identities effectively [30].

According to Alsulaiman, A., utilizing user-centric design in credential verification also indirectly supports security. When users find the design intuitive, they are less likely to make errors that could compromise their credentials during the verification process. User-friendly

---



---

interfaces can guide individuals through complex security steps, ensuring that security protocols are followed without frustrating users [31]. Additionally, the studies of C. Avram et al. and M. Schmitt et al. also stated that uploaded ID validation simplifies the user experience by allowing for quick and easy verification. Users can upload their identification documents directly from their mobile devices, which minimizes the time spent manually entering data. This convenience increases user satisfaction rates as the process becomes hassle-free and efficient [29], [32].

The validation of identification documents, through optimized admin review and advanced technological solutions, is vital for securing online credential systems. Institutions can effectively address challenges like fraudulent documentation and enhance overall system reliability by leveraging robust verification methods, user-centric designs, and emerging technologies. These advancements protect sensitive information and foster user trust and satisfaction, paving the way for more efficient and secure digital transactions in the future.



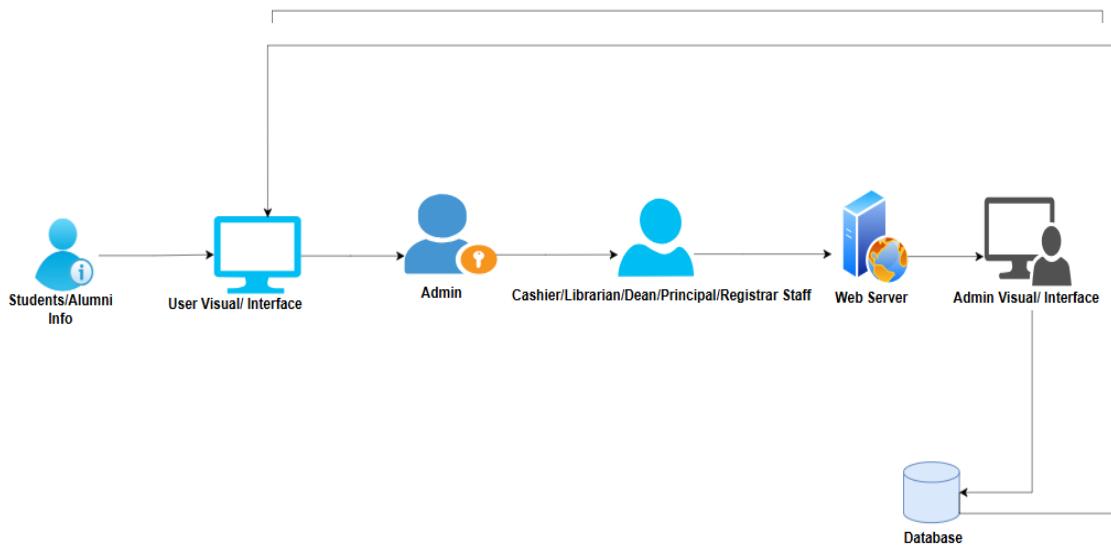
---

## CHAPTER 3

### SOFTWARE REQUIREMENTS AND DESIGN SPECIFICATION

#### 3.1 System Architecture

This chapter presents a detailed analysis of the technical components in designing and developing the proposed system, including software and hardware requirements specifications. Additionally, it provides a thorough overview of the functionality of the Online Request for Credentials System (ORCS).



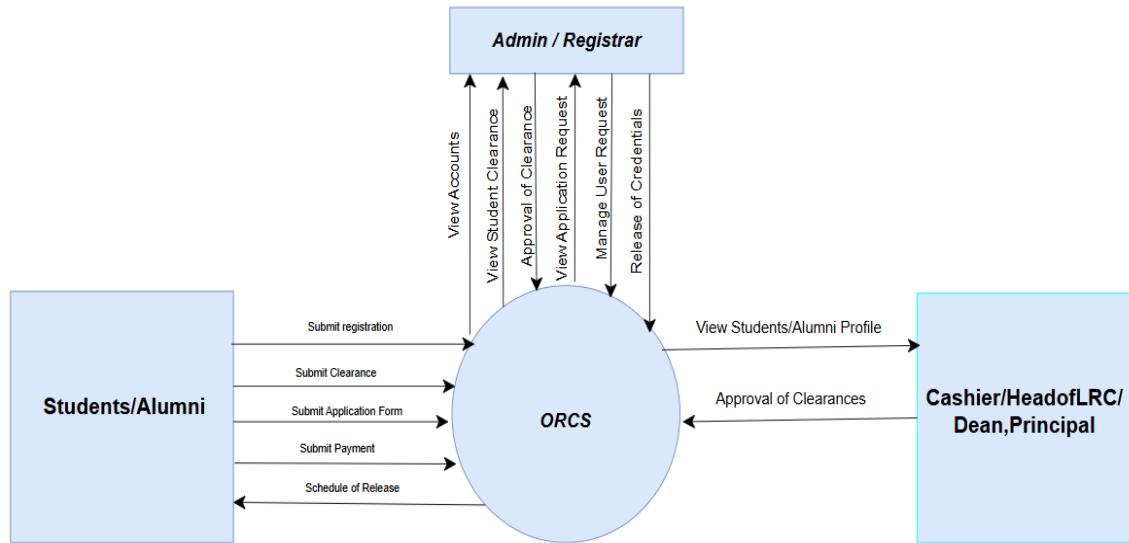
*Figure 1. System Architecture*

Figure 1 illustrates the fundamental process of the Online Request of Credentials System (ORCS). This system facilitates user interaction among various roles, including the administrator (Head of the Registrar's Office), users (Registrar Staff, Cashier, Head of the LRC, and Dean/Principal), as well as students and alumni, all of whom can access the platform via the Internet. End-users must create an account to use the leading platform. The Cashier, Head of the LRC, Dean/Principal, and Registrar Staff review student requests seeking credentials by checking their backgrounds and approving clearance. Meanwhile, the administrator manages credential



requests by validating user requirements and ensuring all necessary approvals are in place.

### 3.2 Conceptual Diagram



*Figure 2. Conceptual Diagram (DFD Level 0)*

Figure 2 illustrates the operational capabilities assigned to different users within the ORCS architecture. The admin has the authority to review and manage the credential requests submitted by students and alumni. Additionally, they can perform functions such as validating requirements. Conversely, the Cashier and other offices are responsible for maintaining a list of individuals who have requested credentials and their associated duties within the ORCS.



### 3.3 Use Case Diagram

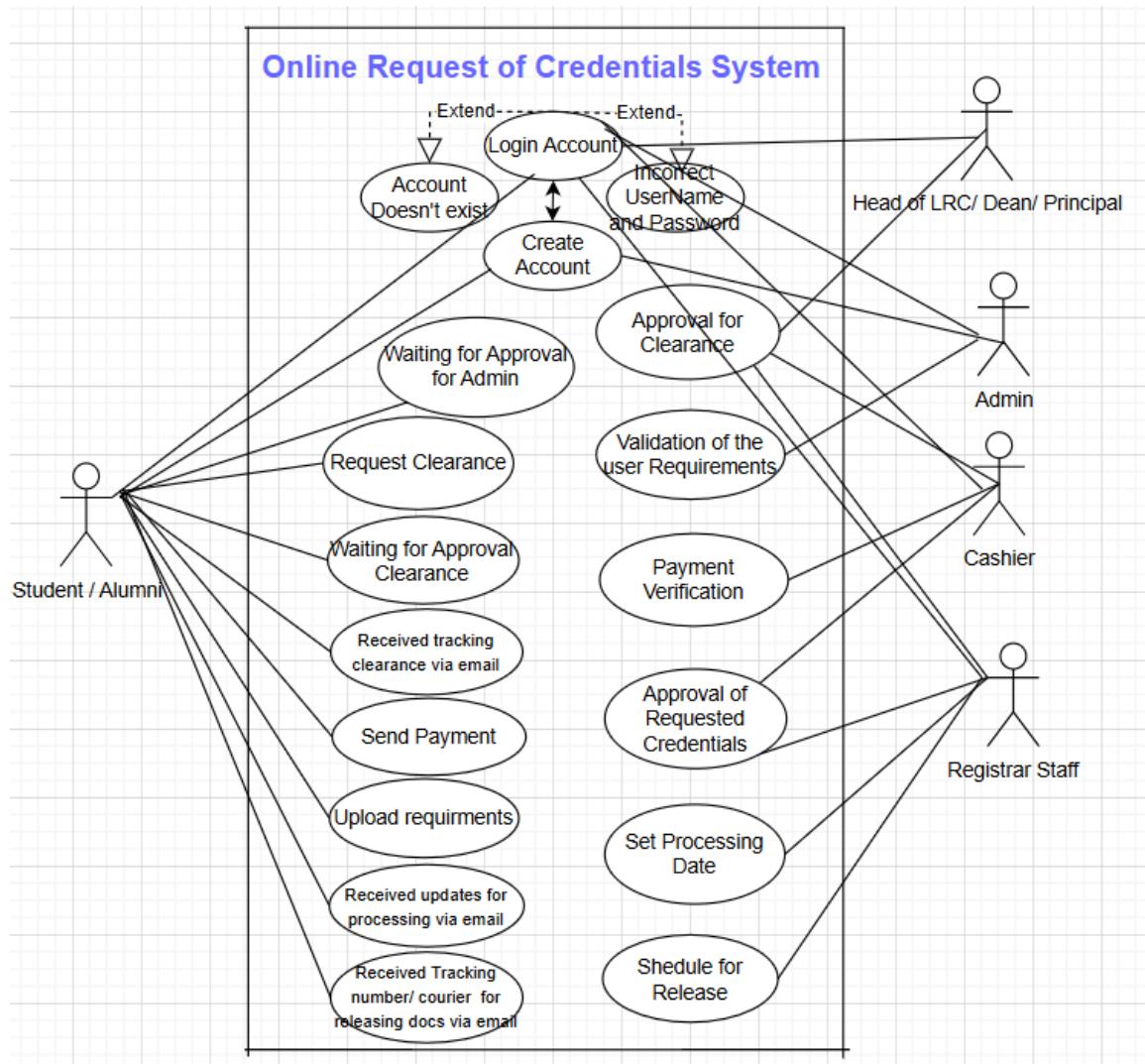


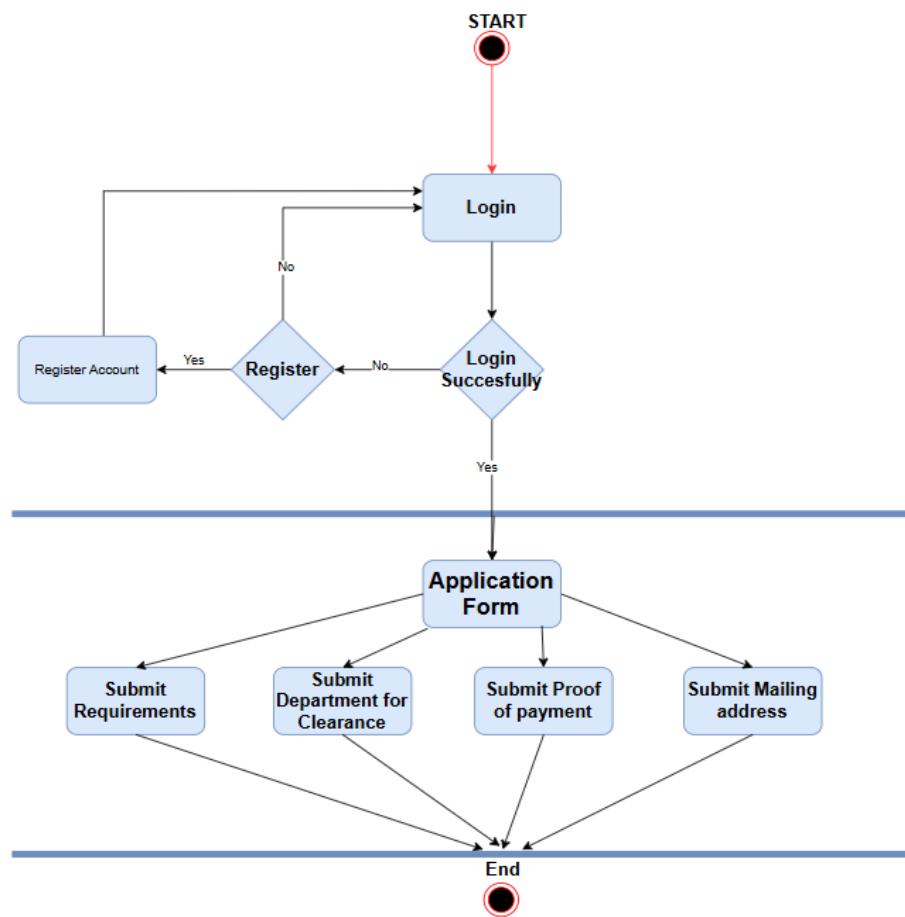
Figure 3. Use Case Diagram

Figure 3 outlines the key features available to each type of user and their roles within the system. The administrator is responsible for validating user requirements and creating accounts for system users such as the Cashier, Head of the LRC, Dean/Principal, and Registrar Staff, as they do not have self-registration privileges. Students and alumni must create an account before logging in and are responsible for completing their request forms. The Registrar Staff manages the approval and release of the requested documents. The system also generates a schedule for



issuing credentials, with appointments available from Monday to Friday. The Cashier is responsible for verifying clearance and payment for the requested credentials. Similarly, the Head of the LRC and the Dean/Principal are tasked with reviewing and confirming requests that have incomplete or pending requirements for clearance purposes.

### 3.4 Activity Diagram



*Figure 4. Activity Diagram – Students/Alumni*

Figure 4 illustrates the series of activities available to students within the system. It begins with the login page. If a user does not have an account, they can create one before returning to the login interface. Once logged in successfully, students and alumni are directed to the ORCS homepage, where they can view and select various forms, including the student/alumni request



form, requirements form, and release schedule form (note that only students from SMCC are eligible for the release schedule form). Additionally, users can log out of their accounts from this page.

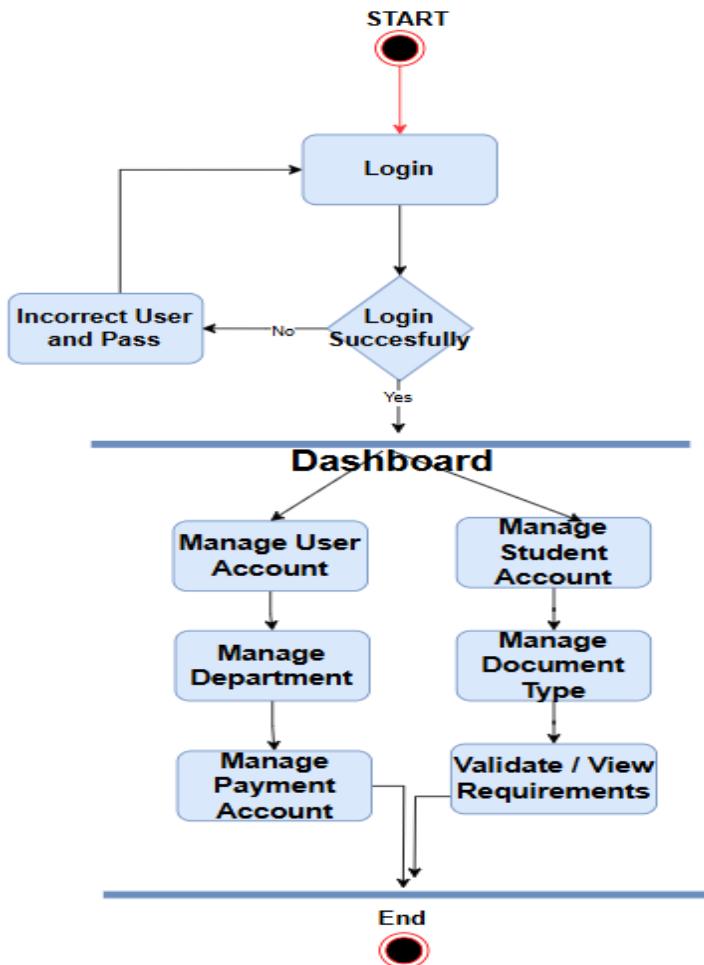


Figure 5. Activity Diagram - Admin

Figure 5 illustrates the system's administrative functions. After a successful login, the administrator gains access to the dashboard, which provides an overview of key metrics, including the number of users, students, pending requests, and completed requests. The administrator manages user and student accounts, configures the payment account, adds departments and document types, and validates or views user requirements. These functions allow the administrator to maintain and oversee the overall operation of the system efficiently.

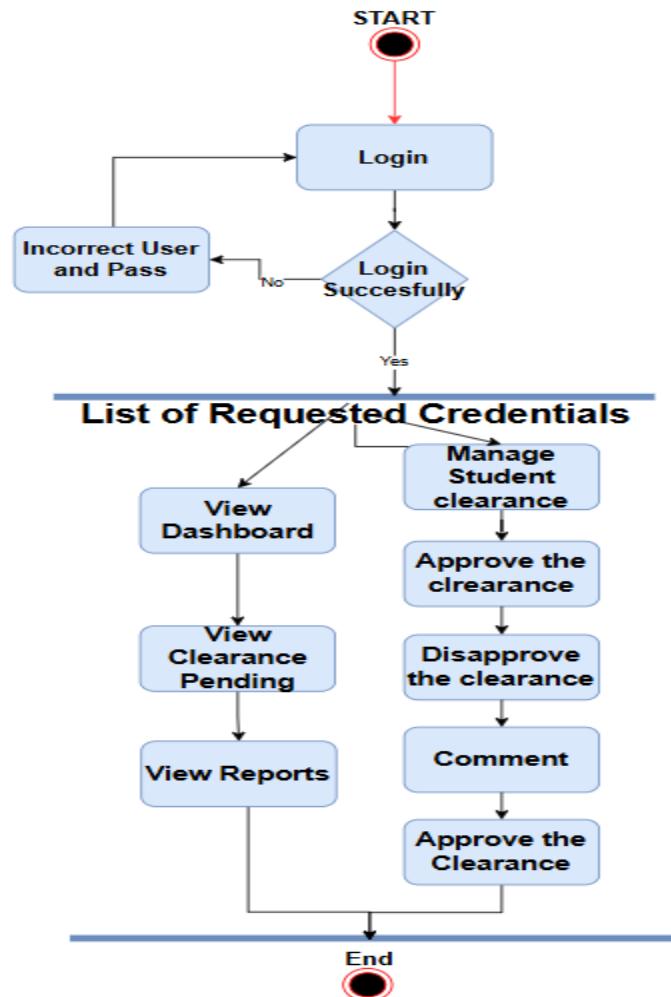


Figure 6. Activity Diagram – Head of LRC/Dean and Principal

Figure 6 illustrates the series of activities available to the user. If a user does not have an account, they cannot access the homepage on the login page. The administrator can register accounts for the Head of the LRC, the Principal, and the Deans. Once logged in successfully, the user's capabilities are limited to viewing and managing the list of individuals who have filled out the form. This includes checking backgrounds, approving clearances, and adding comments if the students have not completed the clearance requirements.

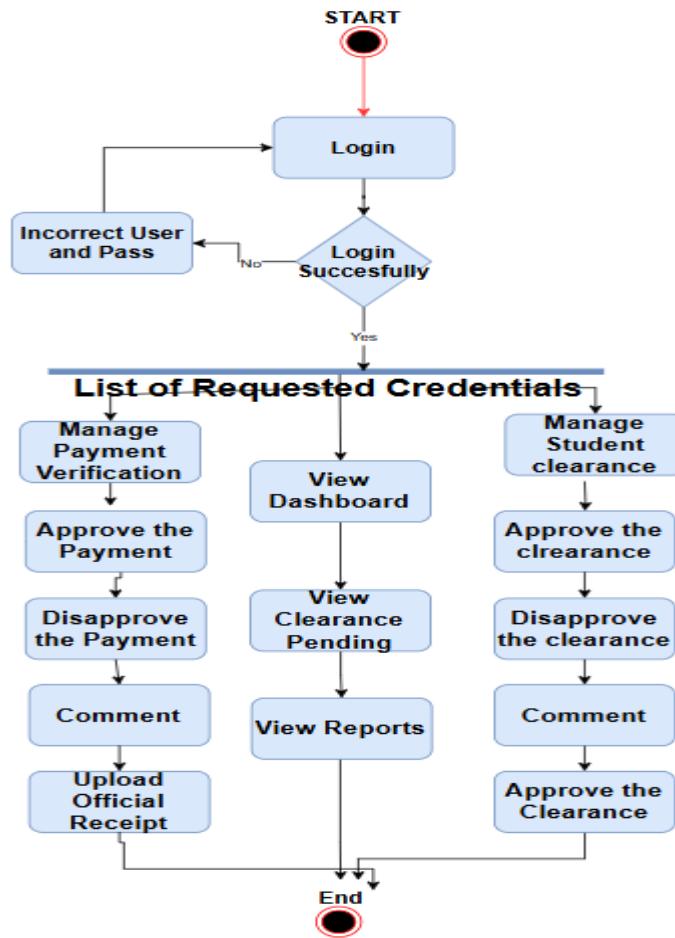
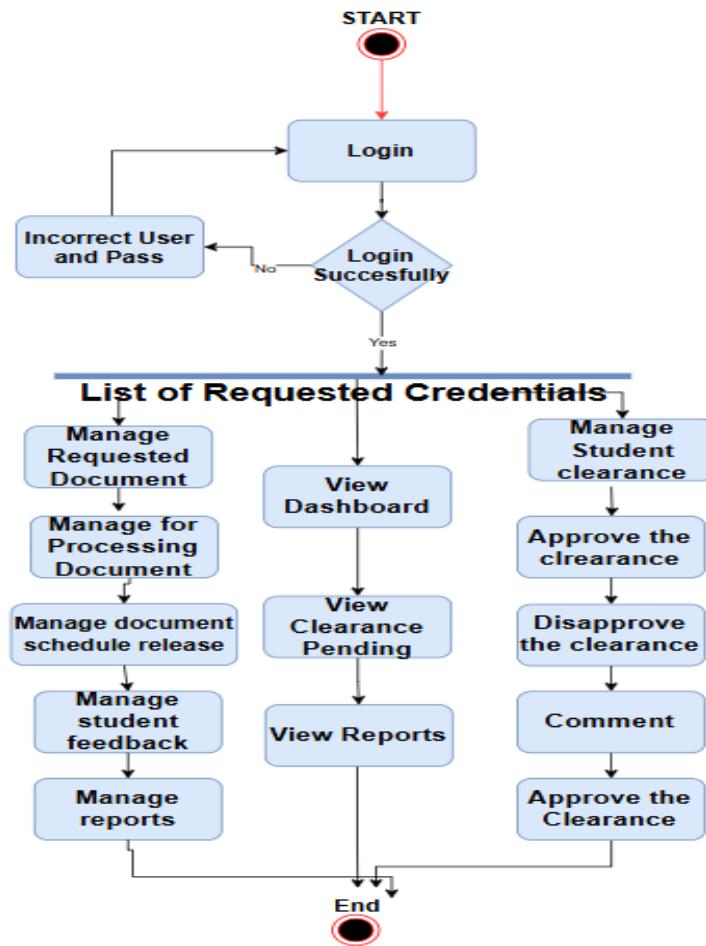


Figure 7. Activity Diagram – Cashier

Figure 7 illustrates the series of activities available to users, focusing on the cashier's role within the system. On the login page, a user without an account cannot proceed to the homepage or access the system, as only the administrator has the authority to create accounts for personnel such as the cashier. Once successfully logged in, the cashier can approve or disapprove clearance requests. If a request is disapproved, the cashier can provide comments, particularly if the requester has an outstanding balance. The cashier is also responsible for validating student payments during the payment verification. After validation, the cashier uploads the official receipt to confirm that the user has completed the payment.



*Figure 8. Activity Diagram – Registrar Staff*

Figure 8 illustrates the series of activities available to the user, explicitly focusing on the role of the registrar staff. Users without an account cannot access the homepage on the login page, as only the administrator has the authority to create accounts for registrar staff. Once successfully logged in, the registrar staff can view and manage the list of individuals who have submitted request forms. Their tasks include approving clearances, managing requested documents, and adding comments if students have not completed the necessary clearance requirements. The registrar staff also sets the processing date, and once the document is processed, the request is scheduled for release. The user is then notified that their requested document will be available for pickup as soon as possible.



### 3.5 Sequence Diagram

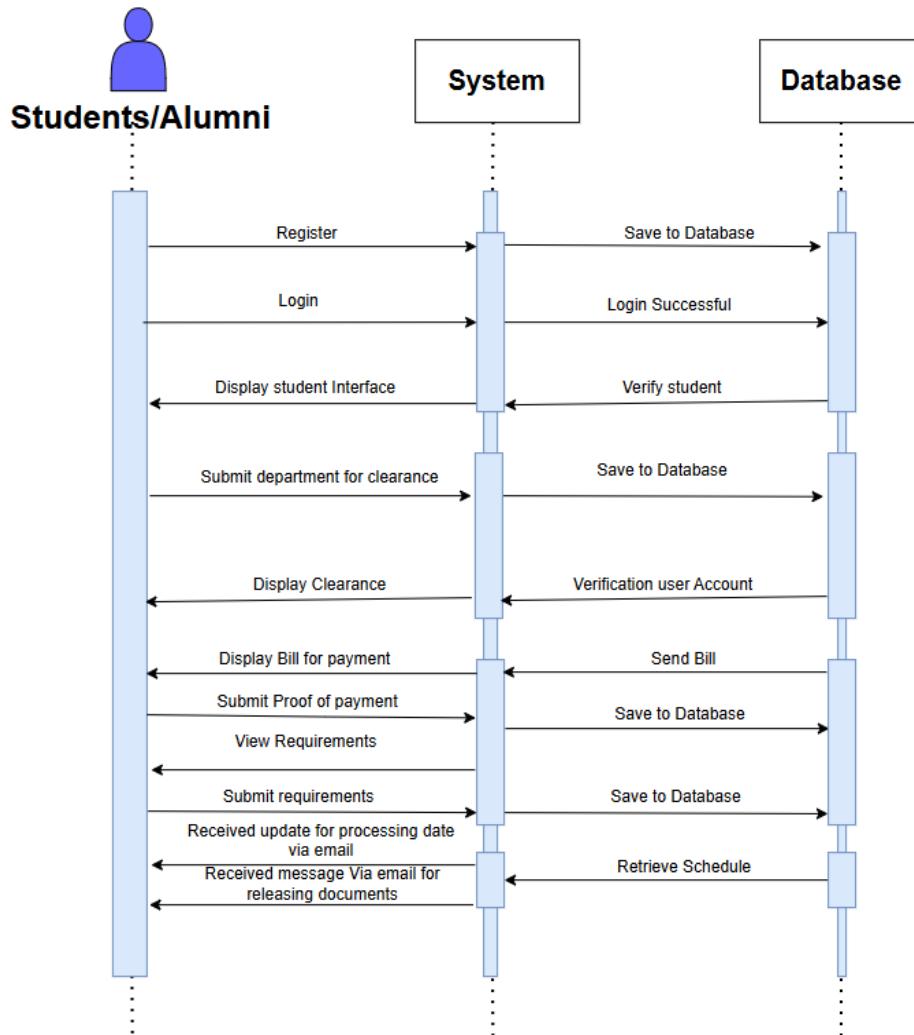
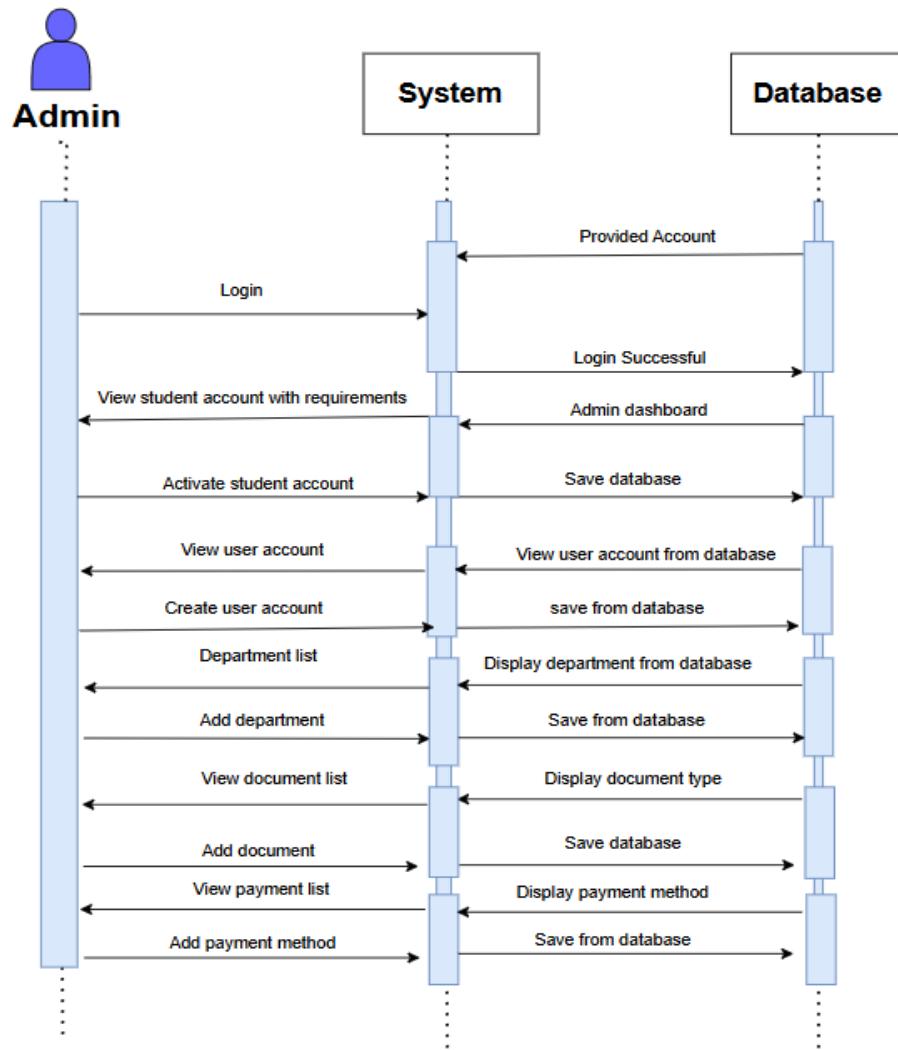


Figure 9. Sequence Diagram – Students/Alumni

Figure 9 illustrates the procedural steps available to students and alumni. After successfully registering, users can start using the login interface. Once logged in, students and alumni can access various functions, such as submitting request forms and requirements forms for database storage. Additionally, the system is designed to send mail to inform students that their documents are being processed. After completing the requirements, users are expected to receive their credentials.



*Figure 10. Sequence Diagram – Admin*

Figure 10 illustrates the procedural steps available to the administrator. They can utilize the login interface to create an account in the SQL database for enhanced system security. Upon successful login, administrators access the dashboard, which displays the number of active users, students, and alumni who have created accounts. The administrator can also manage accounts for students, alumni, and users. Additionally, they can access and manage the list of document types stored in the database, set the payment method, and designate the department responsible for approving student clearances.

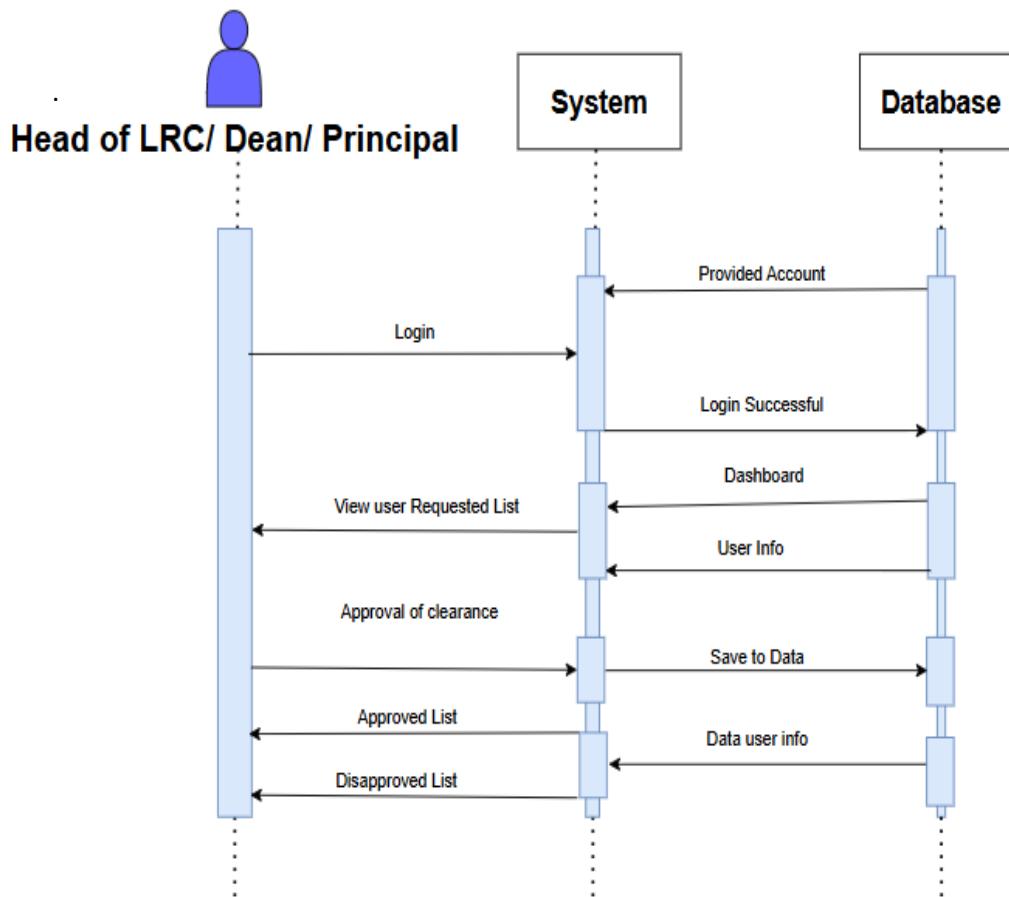


Figure 11. Sequence Diagram – Head of LRC/Dean and Principal

Figure 11 illustrates the procedural steps available to users. The user can use the login interface to complete the login process, which an admin-provided account facilitates. Once the login is successful, the user can access the requested list or user information, allowing them to approve the clearance.

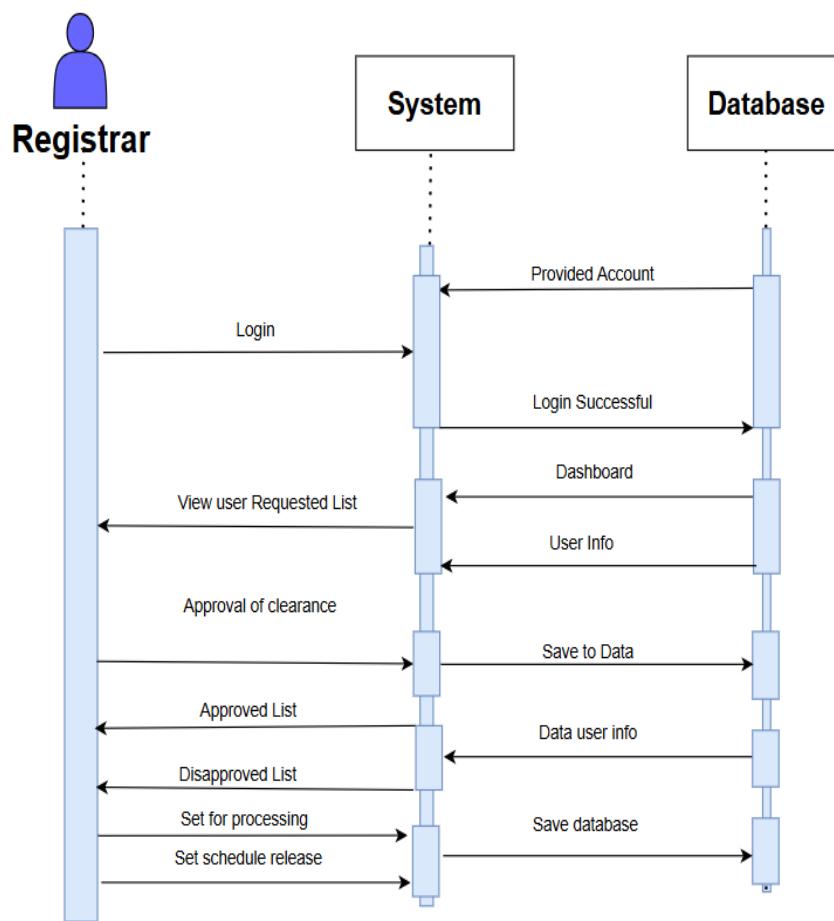
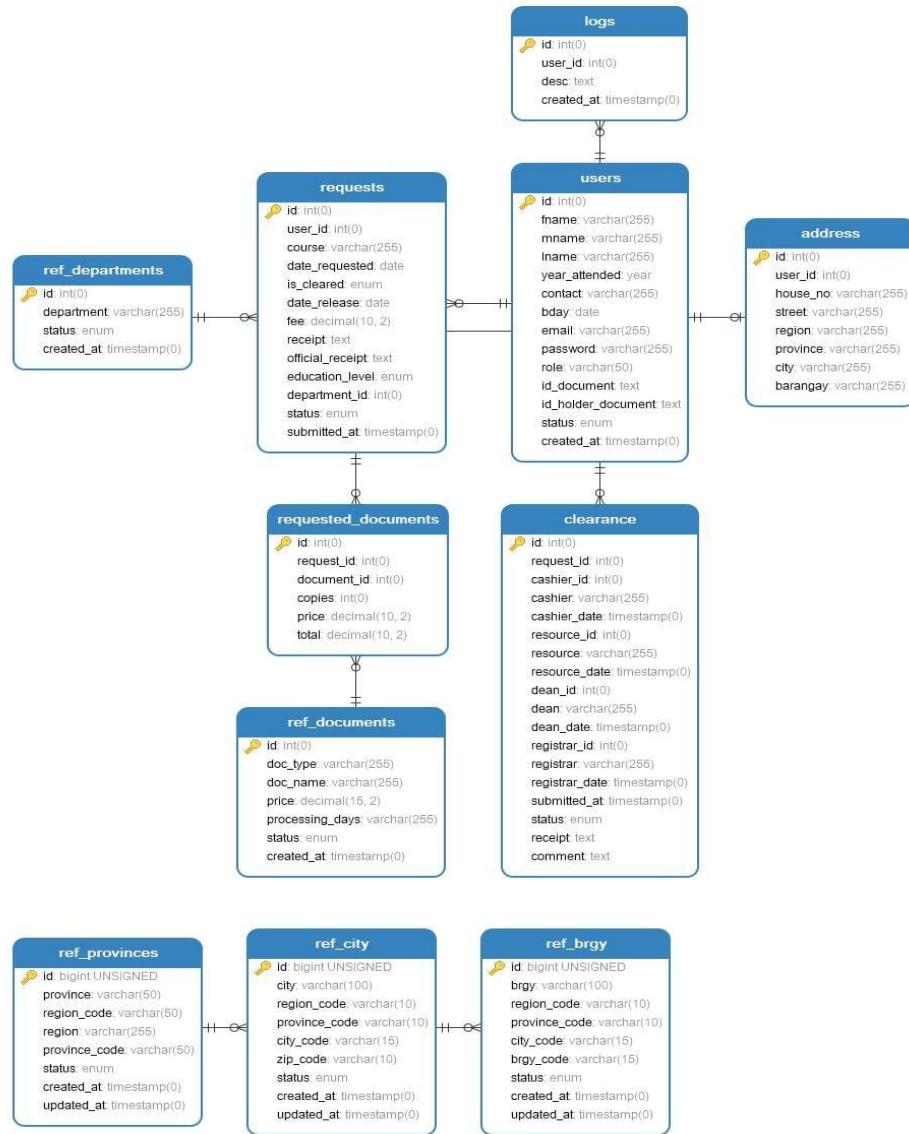


Figure 12. Sequence Diagram – Registrar Staff

Figure 12 illustrates the procedural steps available to users. The user can access the login interface to complete the login process, which an account provided by the administrator facilitates. Once logged in successfully, the user can access the dashboard to view the list of pending clearances and approve the requests listed. The registrar staff reviews the requested credentials, processes the document type requests made by students or alumni, and sets a schedule for the release date along with a tracking number. Finally, the registrar sends an email to inform the user that their document will be available for pickup soon.



### 3.6 ERD (Database Design)

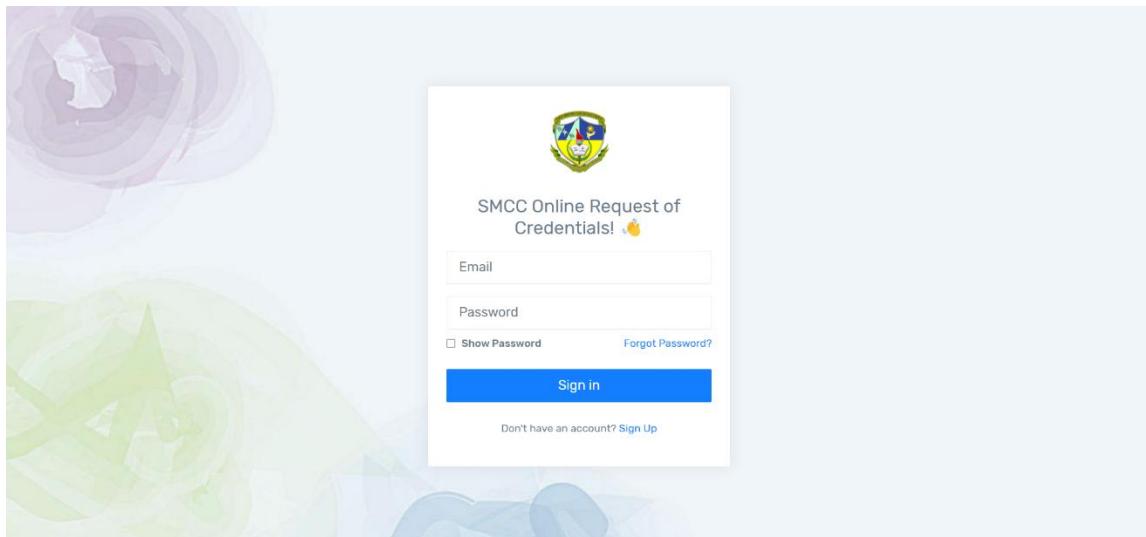


*Figure 13. Entity Relationship Diagram – Database Design*

Figure 13 shows each data processing step and how the data is processed and stored. The attached file is the ERD, which illustrates the contents of each table, as well as the primary key and foreign key. It also illustrates how the tables of the system's database are connected.



### 3.7. User Interface Design (Prototype)



*Figure 14. Admin/All Users - Login Page (User Interface Design)*

Figure 14 displays the login page of the Online Request of Credential System (ORCS), designed for secure and user-friendly access. It includes fields for username and password and a Forgot Password option for account recovery. The system supports role-based access: students and alumni must create their accounts, while staff users, such as Cashiers, Head of LRC, Registrar Staff, and Deans/Principals, receive accounts from the administrator. Upon login, the system automatically identifies the user's role and displays the corresponding interface.

A screenshot of the SMCC Online Request of Credentials sign-up page. The page has a light gray background with a decorative purple and green floral watermark on the left. At the top center is the college's crest. Below it, the title "SMCC Online Request of Credentials!" is displayed with a small hand icon. The form consists of several input fields: "First Name \*" (Ricel Ada), "Middle Name" (Salutis) with a checked "Has Middle Name" checkbox, "Last Name \*" (Villarin), and "Ext \*" (N/A). There are dropdown menus for "Birthdate" (01/11/2003) and "Last School Year Attended" (2025). Below these are address fields: "House No. & Street" (D-6), "Municipality / City" (NASIPIT), "Province" (AGUSAN DEL NORTE), and "Barangay" (Camagong). Two file upload fields are present: "Upload Valid ID \*" with a "Choose File" button (Valid ID.jpg) and "Upload a photo holding your valid ID \*" with another "Choose File" button (Holding with valid ID.jpg). Both fields specify allowed file types: png, jpg, jpeg, pdf. A note at the bottom of each field says "Please upload any valid government ID (Driver's License, Passport, etc.)." A large blue "Sign Up" button is centered at the bottom of the form. Below it, a link says "Already have an account? Sign In instead".

*Figure 15. Students/Alumni - Sign-Up Page (User Interface Design)*

Figure 15 presents the sign-up page enabling new users (students and alumni) to create an account. The interface includes fields for essential information such as Name (with separate inputs for last, first, and middle names), school year, course, contact number, address, and username, with a confirmation field for added security. The password will be emailed after the admin validates the user's information. A prominent 'Sign Up' button completes the form, leading users to the next step in the registration process.

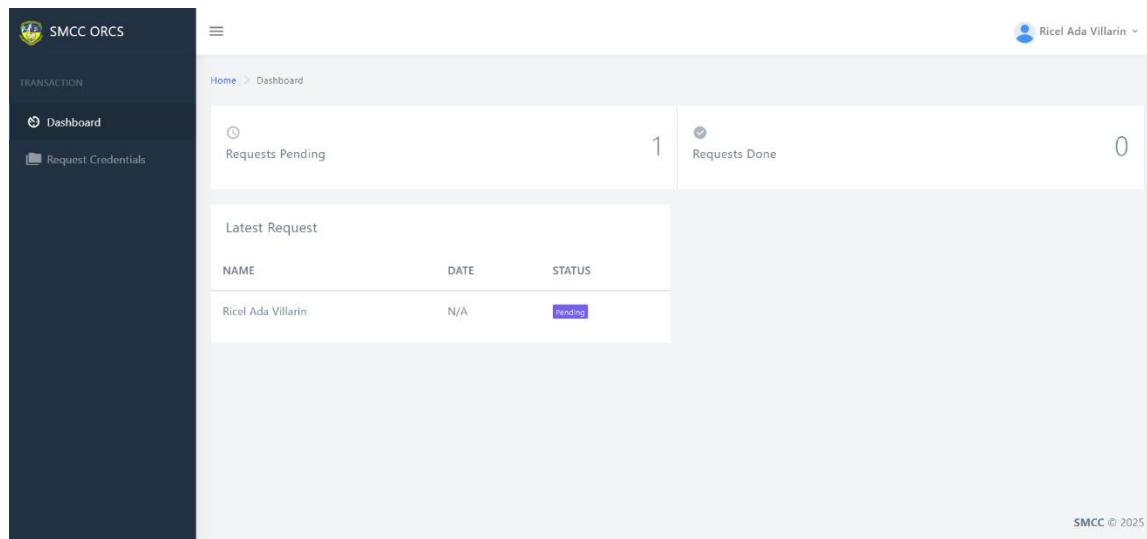


Figure 16. Students/Alumni- Dashboard (User Interface Design)

Figure 16 displays the dashboard of the SMCC Credentials system, where students and alumni can view the status of their credential requests. The dashboard shows the number of requests pending and completed. It also lists the latest requests with details such as the requester's name, date of request, and current status (e.g., *For Processing*, *Released*). Users must fulfill all request requirements to proceed to the following processing stage.



The screenshot shows a user interface for requesting clearance. At the top right, there is a profile picture and the name "Ricel Ada Villarin". Below the header, a button labeled "Request Clearance" is visible. A modal window titled "Clearance Form" is open in the center. Inside the modal, the name "Ricel Ada Villarin" is displayed above a search bar. The search bar has dropdown menus for "Department" (set to CCIS) and "Course" (set to BSIT). There are "Close" and "Submit" buttons at the bottom of the modal. Below the modal, a table is shown with the message "No data available in table". At the bottom of the page, there is a footer with the text "SMCC © 2025".

*Figure 17. Students/Alumni – Request Clearance (User Interface Design)*

Figure 17 displays the ‘Request Clearance’ interface page. The Request Clearance button opens a clearance form, where the student can request clearance by selecting their department and course, then submitting the form.

The screenshot shows a user interface for payment. At the top right, there is a profile picture and the name "Ricel Ada Villarin". Below the header, a button labeled "Request Clearance" is visible. A modal window titled "Upload Receipt" is open in the center. It asks for a "Payment Method" and lists three options: "gcash" (with a yellow logo), "BDO" (with a blue logo), and "GCash" (with a blue logo). Below each method, there is account information: "Account Name: Saint Michael College of Caraga" and "Account Number: 040558000114" for BDO; and "Account Name: Saint Michael College of Caraga" and "Account Number: 09098549424" for GCash. Underneath the payment methods, there is a section for "Proof of Payment" with a "Choose File" button and a preview area showing "gcash image.png". The allowed file types are ".png, .jpg, .jpeg". At the bottom of the modal, there are "Close" and "Save changes" buttons. Below the modal, a table is shown with the message "No data available in table". At the bottom of the page, there is a footer with the text "SMCC © 2025".

*Figure 18. Students/Alumni - Payment for Clearance (User Interface Design)*

Figure 18 shows the interface page for payment of clearance requirements. This interface is essential for requesters, especially those with outstanding balances at the school. Requesters



must complete the payment for clearance requirements before their request can be processed.

The system also provides an option to upload proof of payment.

The screenshot shows the 'Payment Details' section of the SMCC ORCS system. It displays three payment method options: BPI (Bank of the Philippine Islands), BDO, and GCash. Each method has its account name, account number, and account number. Below these, the 'Total Price' is listed as ₱1855.00. A 'Proof of Payment\*' section includes a 'Choose File' button with 'No file chosen' and 'Allowed file types: png, jpg, jpeg'. At the bottom right are 'Close' and 'Save changes' buttons.

*Figure 19. Students/Alumni - Payment for Requested Credentials (User Interface Design)*

Figure 19 illustrates the interface page for the payment of requested documents. Students and alumni must complete the payment for requested credentials before processing can begin. The system allows users to upload proof of payment as part of their application.

The screenshot shows the 'Clearance Status - No. 1' section. It displays a message from 'Head, Learning Resource Center' stating: 'You have an unreturned library item. You are advised to return it.' Below this, there is a link: 'Please send your message here for further assistance: Click here'. On the left, there is a message from 'Cashier' dated 'May 4, 2025'. On the right, there are messages from 'Dean / Principal' and 'School Registrar'. At the bottom right are 'Close', 'Previous', 'Next', and 'Action' buttons.

*Figure 20. Students/Alumni - Clearance Status (User Interface Design)*



Figure 20 displays the 'Clearance Status' page, part of the credential request system for students and alumni at Saint Michael College of Caraga. This interface allows users to track the approval status from various departments for their requested documents. If a section is marked as "DISAPPROVED," a comment box appears with additional requirements the requester must comply with. The system uses color-coded indicators, green for "Approved," blue for "Pending," and red for "Disapproved", for easy visual tracking of progress.

The screenshot shows the SMCC ORCS application interface. On the left, a sidebar lists 'TRANSACTION' options: Dashboard and Request Credentials. The main area shows a 'My Request List' table with one entry (ID 1, Date Requested 04/05/2025). To the right is the 'Application Form' for Rigel Ada Villarin. The form fields include:

- Name: Rigel Ada Villarin
- Are you a graduate? Yes (radio button selected)
- Date Requested: 04/05/2025
- Year Graduated: 2025
- Mailing Address: House No. & Street: D-1, Province: SURIGAO DEL NORTE, Municipality / City: ALEGRIA, Barangay: Alipao
- Certificates selected: Certificate - General Weighted Average, Certificate of Honors, Transcript of Record - Employment Purpose
- TOR selected: Transcript of Record - Board Exam Purpose
- Others: Honorable Dismissal, Reissuance of Diploma, School Form 10

At the bottom are 'Close' and 'Submit' buttons. The top right shows a user profile for Rigel Ada Villarin and a 'Request Clearance' button. The bottom right corner has the text 'SMCC © 2025'.

*Figure 21. Students/Alumni – Application Form (User Interface Design)*

Figure 21 displays the 'Application Form' interface that students or alumni use to request credentials. Before submitting the form, requesters can specify whether they are graduates, enter their personal and address details, and select the type of credentials they need, such as certificates or transcripts, along with the purpose for each request.



The screenshot shows a feedback form titled 'Feedback Form' with instructions: 'Read each statement below. Check the appropriate column that corresponds to your rating. [Ganit ang kasangkuran; susha ang angay nga kolum ngakatumbas sa imong rating.]'. It includes a scale from 'VS-Very Satisfied' to 'VD-Very Dissatisfied' for five categories: Staff Appearance, Staff Helpfulness, Speed/Efficiency, Job Knowledge, and Quality of Service. Below the form is a text area for comments with placeholder text: 'I already received the credentials. Thank you for your service.' At the bottom, there is a footer with form details: Form Code No. FM-OCM-SMCC-HOF-01, Issue Month 02, Revision No. 01, Date Effective 14 September 2023, Approved by President, and buttons for Close and Submit.

*Figure 22. Students/Alumni – Feedback Form (User Interface Design)*

Figure 22 displays the ‘Feedback Form’ page. This form allows students or alumni to provide feedback on the quality of services received from the institution. Users are asked to rate five key service areas: Staff Appearance, Staff Helpfulness, Speed/Efficiency, Job Knowledge, and Overall Quality of Service. Each criterion can be rated using a four-point scale: Very Satisfied (VS), Satisfied (S), Dissatisfied (D), and Very Dissatisfied (VD) with corresponding radio buttons for easy selection.

The form also includes a text box where users can add comments, suggestions, or complaints for more detailed feedback. Located at the bottom are ‘Submit’ and ‘Close’ buttons, enabling users to either finalize and send their feedback or exit the form without submitting.



The screenshot shows the 'Activity Logs' section of the SMCC ORCS application. At the top, there's a navigation bar with 'Home > Activity Logs'. Below it is a table titled 'Activity Logs' with columns 'ID', 'Description', and 'TimeStamp'. The table contains five rows of data:

ID	Description	TimeStamp
1017	Registration added: Riel Ada Villarin	2025-05-04 16:57:54
1021	Logged In	2025-05-04 17:38:08
1022	Changed Password user #7	2025-05-04 17:38:21
1023	Added clearance request # 1	2025-05-04 17:42:54

At the bottom of the table, it says 'Showing 21 to 24 of 24 entries' and has navigation buttons for 'Previous', '1', '2', '3', and 'Next'. The footer of the page includes the text 'SMCC © 2025'.

*Figure 23. All Users – Activity Logs (User Interface Design)*

Figure 23 displays the profile interface, where all users can view activity logs showing each action's ID, description, and timestamp. Through this interface, users can identify the actions they have taken along with the corresponding timestamps.

The screenshot shows the 'Dashboard' section of the SMCC ORCS application for an admin user. At the top, there's a navigation bar with 'Home > Dashboard'. The dashboard features four summary boxes: 'Users' (6), 'Students' (1), 'Requests Pending' (0), and 'Requests Done' (0). Below these boxes is a section titled 'Latest Request' with a table header 'NAME', 'DATE', and 'STATUS'.

NAME	DATE	STATUS

The footer of the page includes the text 'SMCC © 2025'.

*Figure 24. Admin - Dashboard (User Interface Design)*

Figure 24 displays the 'Dashboard' interface, where the admin can view summary statistics, including the total number of users, students, pending requests, and completed



requests. The 'Latest Request' section shows recent transactions, including the requester's name, date, and current status, such as 'For Processing' or 'Released'.

The screenshot shows the 'Requesters' list page in the SMCC ORCS system. The left sidebar contains navigation links for Dashboard, Requesters, Users, References, Department, Course, Document Type, Payment Account, and ISO Form Code. The main content area displays a table titled 'Student List' with one entry:

ID	Name	Year Attended	Contact	Email	Status
7	Villarin, Richel Ada	2025	09705366534	adangvillarin@gmail.com	Pending

Below the table, it says 'Showing 1 to 1 of 1 entries'. To the right of the table is a vertical sidebar with actions: View Upload, Activate Account, Disapprove Account, and Reset Password. At the bottom right of the page is the copyright notice 'SMCC © 2025'.

*Figure 25. Admin – Requesters List (User Interface Design)*

Figure 25 displays a list of requesters who have created accounts, along with details such as their names, year attended, department, contact information, educational level, and account status. This interface also provides actions for viewing the requirements uploaded by students or alumni during sign-up. Additionally, administrators have the option to activate or disapprove accounts and to reset passwords.



The screenshot shows the 'User List' page. The table contains the following data:

ID	First Name	Middle Name	Last Name	Email	Role	Department	Status	Action
2	Darlene	Registrar	Alag	ricelada_villarin@smccnasipit.edu.ph	Registrar	Registrar	Active	Action
3	Noel	Cashier	Baja	popohinautan45@gmail.com	Cashier	Cashier	Active	Action
4	Renel	Librarian	Maneda	rjsgwapof23@gmail.com	Librarian	Learning Resource Center	Active	Action
5	Daisa	DeanCCIS	Gupit	angelmaebaguio16@gmail.com	Dean	CCIS	Active	Action
6	Rene	Principal JHS	Japitana	angelo_hinautan@smccnasipit.edu.ph	Principal	JHS	Active	Action

Showing 1 to 5 of 5 entries

*Figure 26. Admin – Users List (User Interface Design)*

Figure 26 displays the Users List interface, showing roles such as Cashier, Head of LRC, Deans, and Principals responsible for clearance approvals. The admin creates their accounts and can edit users via the “Action” button or add new users using the “Add Users” button.

The screenshot shows the 'Department List' page. The table contains the following data:

ID	Department	Facebook Link	Status	Action
1	Elementary	<a href="https://www.facebook.com/smccelementarydepartment">https://www.facebook.com/smccelementarydepartment</a>	Active	Edit
2	JHS	<a href="https://www.facebook.com/SMCCJuniorHighSchool">https://www.facebook.com/SMCCJuniorHighSchool</a>	Active	Edit
3	SHS	<a href="https://www.facebook.com/SMCCSHSOfficial">https://www.facebook.com/SMCCSHSOfficial</a>	Active	Edit
4	CCIS	<a href="https://www.facebook.com/CCISofSMCCNasipit">https://www.facebook.com/CCISofSMCCNasipit</a>	Active	Edit
5	CCIE	<a href="https://www.facebook.com/smccorim">https://www.facebook.com/smccorim</a>	Active	Edit
6	CBM	<a href="https://www.facebook.com/SMCCCMBM">https://www.facebook.com/SMCCCMBM</a>	Active	Edit
7	CTHM	<a href="https://www.facebook.com/profile.php?id=100095316962172">https://www.facebook.com/profile.php?id=100095316962172</a>	Active	Edit
8	CTE	<a href="https://www.facebook.com/smcccollegeofteachereducation">https://www.facebook.com/smcccollegeofteachereducation</a>	Active	Edit
9	GAS	<a href="https://www.facebook.com/SMCCCCAS">https://www.facebook.com/SMCCCCAS</a>	Active	Edit
10	Learning Resource Center	<a href="https://www.facebook.com/LRCSMCC">https://www.facebook.com/LRCSMCC</a>	Active	Edit

Showing 1 to 10 of 13 entries

*Figure 27. Admin – Department List (User Interface Design)*

Figure 27 displays the ‘Department List’ page for administrators in the ORCS. This interface allows admins to view and manage department records, including the department name and



associated Facebook link. Each department entry is marked with a status (e.g., Active) and includes an Edit button for making changes. An Add Department button at the top-right corner enables the addition of new departments. The page also has a Search bar and pagination controls for easy navigation.

The screenshot shows a web-based administrative interface for managing courses. On the left is a dark sidebar menu with options like Dashboard, Requests, Users, Department, Course, Document Type, Payment Account, and ISO Form Code. The main content area has a header with 'Home', 'Reference', and 'Course' tabs, and a search bar. A blue 'Add Course' button is in the top right. Below is a table titled 'Course List' with columns: ID, Department, Course, Status, and Action (with an 'Edit' button). The table contains 10 entries, with 34 total entries shown. The data includes courses like BSIT, BSIS, DIT, and Grade 1-5, categorized under CCIS and Elementary departments, all marked as 'Active'.

ID	Department	Course	Status	Action
1	CCIS	BSIT	Active	Edit
2	CCIS	BSCS	Active	Edit
3	CCIS	BLIS	Active	Edit
4	CCIS	BSIS	Active	Edit
5	CCIS	DIT	Active	Edit
6	Elementary	Grade 1	Active	Edit
7	Elementary	Grade 2	Active	Edit
8	Elementary	Grade 3	Active	Edit
9	Elementary	Grade 4	Active	Edit
10	Elementary	Grade 5	Active	Edit

Figure 28. Admin – Course List (User Interface Design)

Figure 28 displays the ‘Course List’ page for administrators in the ORCS. This interface allows admins to manage courses under various departments. Each entry shows the department, course name, status (e.g., Active), and an Edit button for modifications. A Search bar and an Add Course button are also available for quick access and updates.



The screenshot shows a table titled "Document Type List" with 10 entries. The columns are: ID, Document Type, Document Name, Price, Processing Days, With Doc Stamp?, Status, and Action. The entries are:

ID	Document Type	Document Name	Price	Processing Days	With Doc Stamp?	Status	Action
1	TOR	Transcript of Record - Board Exam Purpose	1050.00	3-5 working days	Yes	Active	Edit
2	TOR	Transcript of Record - Employment Purpose	1050.00	3-5 working days	Yes	Active	Edit
3	TOR	Transcript of Records - Transfer Credential	1050.00	3-5 working days	Yes	Active	Edit
4	Certificate	Certificate - General Weighted Average	200.00	1-2 working days	Yes	Active	Edit
5	Certificate	Certificate of Honors	200.00	1-2 working days	Yes	Active	Edit
6	Certificate	Certification of Enrollment	200.00	1-2 working days	Yes	Active	Edit
7	Certificate	Certificate (English as a medium of instruction)	200.00	1-2 working days	No	Inactive	Edit
8	Certificate	Certificate of Units Earned	100.00	1-2 working days	Yes	Active	Edit
9	Certificate	Certification of Graduation	100.00	1-2 working days	Yes	Active	Edit
10	Others	School Form 10	300.00	3-5 working days	Yes	Active	Edit

Showing 1 to 10 of 14 entries

*Figure 29. Admin – Document Type List (User Interface Design)*

Figure 29 displays a list of document types available for student-requested credentials.

The admin can manage document types in this section by editing existing entries. The list includes key details such as the document type, name, price, processing time (in days), and status. The 'Add Document Type' button allows admins to add new document categories as needed.

The screenshot shows a table titled "Payment Account List" with 3 entries. The columns are: ID, Account Name, Account Number, Picture, Status, and Action. The entries are:

ID	Account Name	Account Number	Picture	Status	Action
1	Saint Michael College of Caraga	500000098291		Active	Edit
2	Saint Michael College of Caraga	040558000114		Active	Edit
3	Saint Michael College of Caraga	09098549424		Active	Edit

Showing 1 to 3 of 3 entries

SMCC © 2025

*Figure 30. Admin – Payment Account (User Interface Design)*

Figure 30 displays a list of available payment account methods within the system. This



interface allows the administrator to manage various payment options efficiently. It includes key details such as the payment name, account number, payment logo, status, and an action button for modifications.

The administrator can use the 'Add Payment Account' button to register new payment methods, providing flexibility in handling transactions. The 'Edit' button enables updates to existing payment details, including account information, status changes, and payment logo replacements. Additionally, the system supports search, sorting, and pagination features to enhance navigation and management of payment accounts.

The screenshot shows the 'ISO Form Code List' page. The table has the following data:

ID	Form Name	Form Code No.	Issue Status	Revision No.	Date Effective	Approved	Status	Action
1	Feedback Form	FM-OCM-SMCC-HCF-01	02	03	2023-09-14	President	Active	<button>Edit</button>

Below the table, it says 'Showing 1 to 1 of 1 entries'. At the top right, there is a 'Search:' input field and a blue 'Add ISO Form Code' button. On the far left, a sidebar menu lists: TRANSACTION (Dashboard, Requesters, Users), REFERENCES (Department, Course, Document Type, Payment Account, ISO Form Code), and SMCC ORCS.

Figure 31. Admin- ISO Form Code (User Interface Design)

Figure 31 shows the 'ISO Form Code' interface page, where the admin can add and edit ISO form codes for reports or other forms provided by the ISO. Each form entry includes its corresponding form number, issue status, revision number, effective date, and the name of the approving authority.



The screenshot shows the SMCC ORCS dashboard. On the left sidebar, there are links for Dashboard, Clearance (Clearance List, Clearance Disapproved List), Requests (Payment Verification), and Reports (Clearance Report, Credentials Report). The main content area shows a summary of pending requests: 'Clearance Pending' (1) and 'Requests Pending' (0). Below this, a table titled 'Latest Request' displays a single row: NAME (Ricel Ada Villarin), DATE (N/A), and STATUS (Pending).

*Figure 32. User (Cashier) - Dashboard (User Interface Design)*

Figure 32 shows the ‘Dashboard’ interface, which enables the Cashier to view credential requests, including pending clearances, completed requests, and recent request transactions, along with their statuses.

The screenshot shows the SMCC ORCS Clearance List page. The sidebar includes links for Dashboard, Clearance (Clearance List, Clearance Disapproved List), Requests (Payment Verification), and Reports (Clearance Report, Credentials Report). The main content area shows a table titled 'Clearance List' with one entry. The table columns are ID, Name, Date Submitted, Is Cleared?, Comment, Status, and Action. The entry shows: ID 1, Name Ricel Ada Villarin, Date Submitted May 4, 2025, Is Cleared? No, Comment You have an outstanding balance of 5,000. Please pay it first., Status For Cashier, and Action buttons (Approve, Disapprove, View Receipt). Navigation buttons for Previous (1) and Next are at the bottom.

*Figure 33. User (Cashier) – Clearance List (User Interface Design)*

Figure 33 shows the clearance approval section for the cashier, which enables authorized users to review, approve, or disapprove clearance requests. Each request includes an action button that allows staff to take appropriate action. Upon approval, a message prompt notifies the



requester of any outstanding requirements, such as institutional balances, ensuring transparency and proper processing. Once a student fulfills the requirements, the interface provides an action button that allows the cashier to approve the request again. The system also features a searchable and sortable table for efficient management of clearance requests.

A screenshot of the SMCC ORCS software interface. The left sidebar has a dark theme with white text and icons. It includes sections for Dashboard, CLEARANCE (Clearance List, Clearance Disapproved List), REQUESTS (Payment Verification), and REPORTS (Clearance Report, Credentials Report). The main content area shows a table titled "Disapproved List" with one entry. The table columns are ID, Name, Date Submitted, Is Cleared?, Comment, Status, and Action. The entry shows ID 1, Name "Ricel Ada Villarin", Date Submitted "May 4, 2025", Is Cleared? "No", Comment "You have an outstanding balance of 5,000. Please pay it first.", Status "For Uploading", and Action (button). Navigation buttons at the bottom include "Previous", "1", and "Next". The top right shows a user profile for "Noel Baja".

ID	Name	Date Submitted	Is Cleared?	Comment	Status	Action
1	Ricel Ada Villarin	May 4, 2025	No	You have an outstanding balance of 5,000. Please pay it first.	For Uploading	

*Figure 34. User (Cashier) – Clearance Disapproved List (User Interface Design)*

Figure 34 shows the ‘Cashier Clearance Disapproved List’ interface. This page displays clearance requests that have been disapproved and are pending student compliance with specific requirements.



A screenshot of the SMCC ORCS software interface. The left sidebar shows navigation options: Dashboard, CLEARANCE (Clearance List, Clearance Disapproved List), REQUESTS (Payment Verification checked), and REPORTS (Clearance Report, Credentials Report). The main content area is titled 'Verify Payments' and shows a 'Payment List'. A table displays one entry: ID 1, Name Ricel Ada Villarin, Date Requested May 4, 2025, Is Cleared? Yes, Date Release (empty), Total 1855.00, Status For verification. Action buttons include View Receipt, Approve (highlighted in blue), and Disapprove. Navigation at the bottom shows 'Showing 1 to 1 of 1 entries' and buttons for Previous (1) and Next.

Figure 35. User (Cashier) – Payment Verification (User Interface Design)

Figure 35 shows the Payment List's 'Payment Verification' interface, which allows the cashier to manage and validate students' credential payments efficiently. This ensures students have paid the required amount by reviewing the uploaded proof of payment. The interface features a six-column table displaying each student's ID number, name, course and year, date requested, status, and actions. In the Action column, the 'View Payment' button lets the cashier verify the payment. Once validated, the cashier can send the official receipt to confirm and approve the payment of the student's credentials.





**SMCC ORCS**

- [Dashboard](#)
- [CLEARANCE](#)
  - [Clearance List](#)
  - [Clearance Disapproved List](#)
- [REQUESTS](#)
- [REPORTS](#)
  - [Payment Verification](#)
  - [Clearance Report](#)
  - [Credentials Report](#)

**PRINT**

Home > Report > Clearance Report

From \* To \*

01/05/2025 05/05/2025

Department \*

All Departments

**Print Report**

Show 10 entries

#	Name	Department	Course	Receipt	Date Approved	Status
1	Ricel Ada Villarin	CCIS	BSIT	<a href="#">View Receipt</a>	May 5, 2025	Approved

Showing 1 to 1 of 1 entries

Search:

Previous **1** Next

SMCC © 2025

**Saint Michael College of Caraga**  
 Brgy. 4, Nasipit, Agusan del Norte, Philippines  
 District 8, Brgy. Triangulo, Nasipit, Agusan del Norte, Philippines  
 Tel. Nos. +63 065 343-5231, 283-3113 | Fax: +63 065 898-0892  
[www.smccnasipit.edu.ph](http://www.smccnasipit.edu.ph)

**CLEARANCE REPORT**  
 Period: May 01, 2025 - May 05, 2025  
 Cashier Office

#	Name	Department	Course	Date Approved	Status
1	Ricel Ada Villarin	CCIS	BSIT	May 05, 2025	Approved

Prepared by:  
 \_\_\_\_\_  
 Noel Baja

*Figure 36. User (Cashier) – Report for Clearance (User Interface Design)*

Figure 36 shows the completed clearance report interface designed for the cashier. The interface allows the cashier to monitor and manage clearance requests that have been reviewed and approved. Users can filter the report based on a specific date range and department to locate particular records easily. Additionally, it features a “View Receipt” button, enabling the cashier to view the uploaded payment receipt submitted by the requester, especially when an outstanding balance needs verification.



The screenshot shows the SMCC ORCS dashboard. On the left, a sidebar menu includes 'Dashboard', 'CLEARANCE' (with 'Clearance List' and 'Clearance Disapproved List' options), and 'REPORTS' (with 'Clearance Report'). The main content area shows a 'Home > Dashboard' breadcrumb. A prominent message 'Clearance Pending' with a count of '1' is displayed. The bottom right corner of the page footer says 'SMCC © 2025'.

*Figure 37. User (Dean/Principal / Head of LRC) – Dashboard (User Interface Design)*

Figure 37 illustrates the dashboard for the Head of LRC, Dean, or Principal, which displays pending clearance requests and the number of students seeking clearance. This dashboard interface is consistent across all users in these roles and specifically highlights pending requests to notify them that students are awaiting clearance approval.

The screenshot shows the 'Clearance List' page. The sidebar menu is identical to Figure 37. The main content area shows a table titled 'Clearance List' with one entry:

ID	Name	Date Submitted	Is Cleared?	Comment	Status	Action
1	Ricel Ada Villarin	May 4, 2025	No		For Resource	Action

Below the table, it says 'Showing 1 to 1 of 1 entries'. The bottom right corner of the page footer says 'SMCC © 2025'.

*Figure 38. User (Dean/Principal / Head of LRC) – Clearance List (User Interface Design)*

Figure 38 shows the clearance list for the Head of LRC, Dean, or Principal, displaying a list of students requesting clearance. All users in these roles share the same interface, which includes



columns for ID, Name, Date Submitted, Is Cleared, Comments, Status, and an Action column with Approve and Disapprove buttons. If the student or alumni has completed all requirements and there are no issues, the clearance can be approved following validation. However, the clearance request will be disapproved if there are pending requirements.

ID	Name	Date Submitted	Is Cleared?	Comment	Status	Action
1	Ricel Ada Villarin	May 4, 2025	No	You have an unreturned library item. You are advised to return it.	Disapproved	<input type="button" value="Action"/> <input checked="" type="radio"/> Approve

*Figure 39. User (Dean/Principal / Head of LRC) – Clearance Disapprove List (User Interface Design)*

Figure 39 illustrates the ‘Disapproved Clearance’ requests interface, where the Head of LRC, Dean, or Principal can review clearance requests previously disapproved due to incomplete compliance with requirements. This interface provides a clear overview of requests that require further action, particularly for students or alumni who have chosen to fulfill their requirements online.

The Disapproved Clearance Requests list includes the same columns as the Approved Clearance Requests list: ID #, Name, Date Submitted, Is Cleared, Comment, and Status, but also features an additional Action column with an Approve button. Once the student or alumnus has fulfilled all requirements, the Head of LRC, Dean, or Principal can approve the clearance request following a validation process.



The screenshot shows the 'Clearance Report' section of the SMCC ORCS software. The report table displays the following data:

#	Name	Department	Course	Date Approved	Status
1	Rigel Ada Villarin	CCIS	BSIT	May 5, 2025	Approved

**Saint Michael College of Caraga**  
Brgy. 4, Nasipit, Agusan del Norte, Philippines  
District 8, Brgy. Triangulo, Nasipit, Agusan del Norte, Philippines  
Tel. Nos. +63 085 343-5231, 283-3113 | Fax: +63 085 898-0892  
[www.smccnaiso.edu.ph](http://www.smccnaiso.edu.ph)

**CLEARANCE REPORT**  
Period: May 01, 2025 - May 05, 2025  
Librarian Office

#	Name	Department	Course	Date Approved	Status
1	Rigel Ada Villarin	CCIS	BSIT	May 05, 2025	Approved

Prepared by:

Renel Maneda

*Figure 40. User (Dean/Principal / Head of LRC) – Report for Clearance (User Interface Design)*

Figure 40 shows the completed clearance report interface for Deans, Principals, or Heads of LRC. It allows users to filter approved clearance records by date and department, revealing key details like name, department, course, approval date, and status for easy monitoring.



The screenshot shows the SMCC ORCS dashboard. On the left is a sidebar with navigation links: SMCC ORCS, Dashboard, CLEARANCE (Clearance List, Clearance Disapproved List), REQUESTS (Requester, For Processing, Schedule Release Date, Feedback List), and REPORTS (Clearance Report, Credentials Report, Request History Report). The main area displays a summary of requests: Clearance Pending (1), Requests Pending (0), and Request for Processing (0). Below this is a 'Latest Request' table with columns NAME, DATE, and STATUS, showing a single entry for Riel Ada Villarin with N/A for DATE and Pending for STATUS. The bottom right corner shows 'SMCC © 2025'.

*Figure 41. User (Registrar Staff) – Dashboard (User Interface Design)*

Figure 41 shows the registrar's 'Dashboard' interface, providing an overview of credential requests, including pending clearances, submitted requests, and recent transaction activity.

The screenshot shows the SMCC ORCS Clearance List page. The sidebar is identical to Figure 41. The main area shows a table titled 'Clearance List' with columns ID, Name, Date Submitted, Is Cleared?, Comment, Status, and Action. One entry is listed: ID 1, Name Riel Ada Villarin, Date Submitted May 4, 2025, Is Cleared? No, Comment (empty), Status For Review, and Action (button). To the right of the table is a search bar and buttons for Approve (radio button) and Disapprove (checkbox). Navigation buttons Previous, Next, and a dropdown menu are also present. The bottom right corner shows 'SMCC © 2025'.

*Figure 42. User (Registrar Staff) – Clearance List (User Interface Design)*

Figure 42 illustrates the registrar's clearance list, displaying students requesting clearance along with their ID, name, date submitted, clearance status, comments, and overall status. The Action column includes Approve and Disapprove buttons. Clearances can be approved after validation if all requirements are met; otherwise, they will be disapproved.



The screenshot shows the 'Disapproved List' page of the SMCC ORCS software. The left sidebar contains navigation links for Dashboard, Clearance (Clearance List, Clearance Disapproved List), Requests (Requester, For Processing, Schedule Release Date, Feedback List), and Reports (Clearance Report, Credentials Report, Request History Report). The main content area has a header 'Disapproved List' with a search bar. Below it is a table with columns: ID, Name, Date Submitted, Is Cleared?, Comment, Status, and Action. A single entry is listed: ID 1, Name Rigel Ada Villarin, Date Submitted May 4, 2025, Is Cleared? No, Comment 'You lack credentials in the registrars office, which is a School Form 10. You are advised to comply with it first.', Status 'Disapproved', and Action buttons for 'Action' (dropdown) and 'Approve'. Navigation buttons for Previous and Next are at the bottom.

ID	Name	Date Submitted	Is Cleared?	Comment	Status	Action
1	Rigel Ada Villarin	May 4, 2025	No	You lack credentials in the registrars office, which is a School Form 10. You are advised to comply with it first.	Disapproved	Action Approve

*Figure 43. User (Registrar Staff) – Clearance Disapprove List (User Interface Design)*

Figure 43 illustrates the ‘Disapproved Clearance’ requests interface, where the registrar reviews requests initially disapproved due to incomplete requirements. The list displays the same columns as the Approved Requests: ID #, Name, Date Submitted, Is Cleared, Comment, and Status, with an additional Action column for approval. Once students or alumni fulfill the requirements, the registrar can validate and approve the request.




**SMCC ORCS**  

- [Dashboard](#)
- [CLEARANCE](#)
  - [Clearance List](#)
  - [X Clearance Disapproved List](#)
- [REQUESTS](#)
  - [Requester](#)
  - [For Processing](#)
  - [Schedule Release Date](#)
  - [Feedback List](#)
- [REPORTS](#)
  - [Clearance Report](#)
  - [Credentials Report](#)
  - [Request History Report](#)

Home > Requested Documents

### List of Requested Documents

Show 10 entries

ID	Name	Department	Course	Date Requested	Graduate?	Year Attended / Graduated	Status	Action
1	Ricel Ada Villarin	CCIS	BSIT	May 4, 2025	Yes	2025	Pending	<a href="#">Details</a> <a href="#">Deny Request</a>

Showing 1 to 1 of 1 entries

Search:

Previous [1](#) Next

SMCC © 2025


**SMCC ORCS**  

- [Dashboard](#)
- [CLEARANCE](#)
  - [Clearance List](#)
  - [X Clearance Disapproved List](#)
- [REQUESTS](#)
  - [Requester](#)
  - [For Processing](#)
  - [Schedule Release Date](#)
  - [Feedback List](#)
- [REPORTS](#)
  - [Clearance Report](#)
  - [Credentials Report](#)
  - [Request History Report](#)

Home > Requested Documents

### Request Details

#	Document Type	Document Name	Number of Copies	Doc Stamp	Price	Total
1	Certificate	Certificate - General Weighted Average	<input type="text" value="1"/>	<input type="text" value="35.00"/>	<input type="text" value="200.00"/>	235.00
2	Certificate	Certificate of Honors	<input type="text" value="1"/>	<input type="text" value="0.00"/>	<input type="text" value="200.00"/>	200.00
3	Certificate	Certification of Graduation	<input type="text" value="1"/>	<input type="text" value="35.00"/>	<input type="text" value="100.00"/>	135.00
4	TOR	Transcript of Record - Employment Purpose	<input type="text" value="1"/>	<input type="text" value="35.00"/>	<input type="text" value="1050.00"/>	1085.00

Search:

Action [Action](#)

Previous [1](#) Next

Shipping Fee:

Total:

[Close](#) [Save Changes](#)

SMCC © 2025

*Figure 44. User (Registrar Staff) – Request Details (User Interface Design)*

Figure 44 illustrates the ‘Request Details’ interface, where the registrar can view the details and types of documents requested by students or alumni. By clicking the ‘Save’ button, the registrar notifies them to proceed with the payment for the total amount of the requested credentials. The registrar also has the right to deny the request if the credential is unavailable in the registrar's office.



The screenshot shows the 'Documents Awaiting Processing' section of the SMCC ORCS system. A modal dialog box is open, titled 'For Processing', with the instruction 'Please enter how many days'. A text input field contains '3-5 working days'. Below the input field are 'Submit' and 'Cancel' buttons. In the background, there is a table with one entry: ID 1, Name Ricel Ada Villarin, Department CCIS. The table has columns for ID, Name, Department, Year Attended / Graduated, Status, and Action. The 'Action' column for the first row shows a button labeled 'For Processing'.

*Figure 45. User (Registrar Staff) – Processing of Documents (User Interface Design)*

Figure 45 illustrates the ‘Processing Documents’ interface, where the registrar can view document requests from students or alumni after payment has been made. The interface displays the request details and student information. The registrar can use the action button to notify the requester that their credentials are being processed.

The screenshot shows the 'Schedule Release Date' section of the SMCC ORCS system. A modal dialog box is open, titled 'Schedule Release Date', containing fields for 'Release Date\*' (05/05/2025), 'Estimated Delivery\*' (5 - 7 days), 'Courier Name\*' (J&T), 'Reference Number\*' (930000123456), and 'Mailing Address\*' (D-1, Alipao, ALEGRIA, SURIGAO DEL NORTE). Below these fields are 'Close' and 'Save changes' buttons. In the background, there is a table with one entry: ID 1, Name Ricel Ada Villarin. The table has columns for ID, Name, Status, and Action. The 'Status' column for the first row shows a button labeled 'Processing'.

*Figure 46. User (Registrar Staff) – Schedule Release Date (User Interface Design)*

Figure 46 illustrates the list of scheduled release dates for the documents interface, where the registrar can view the details and types of documents requested. The registrar can also check



the official receipt to verify that the student or alumni has completed the payment for the requested credentials. A modal input form opens by clicking the action button, allowing the registrar to inform the student or alumni that their request is in transit. Once the process is complete, the registrar can email the tracking number and courier details.

The screenshot shows the 'Feedback List' page of the SMCC Online Credential System. On the left is a sidebar with navigation links: Dashboard, CLEARANCE (Clearance List, Clearance Disapproved List), REQUESTS (Requester, For Processing, Schedule Release Date, Feedback List), and REPORTS (Clearance Report, Credentials Report, Request History Report). The main area displays a table titled 'Feedback List' with one entry:

ID	Date Requested	Date Approved	Date Released	Name	Feedback	Status
1	May 4, 2025	May 4, 2025	May 4, 2025	Ricel Ada Villarin	<a href="#">View</a>	<a href="#">Done</a>

Below the table, it says 'Showing 1 to 1 of 1 entries'. At the bottom right are 'Previous' and 'Next' buttons. The top right shows a user profile for Darlene Alag. The bottom right corner of the main window says 'SMCC © 2025'.

On the right side of the screenshot, a 'Print' dialog box is open. It shows the 'Clients Feedback' form with various rating scales and a checkbox for receiving a certificate. The dialog includes fields for Destination (EPSON L120 Series), Pages (All), Copies (1), Layout (Portrait), and Color (Color). There are also 'More settings' and 'Print' (blue) and 'Cancel' buttons.

**Instructions: [Panudio]**  
Read each statement below. Using the scale, check [Gamit ang kasangkuran, gamit ang kasangkaran].

**Scale: [Kasangkaran]**  
VS-Very Satisfied: [Nagagbibigay]  
D-Dissatisfied: [Wala matac]

**How would you rate our services?**  
Ginunsila nimo pag-rate ang among mga:

**Staff Appearance** [Hitsura sa mga K:  
✓  
Staff Helpfulness [Pagtabang sa Ke:  
✓  
Speed/Efficiency [Kapaspasan/kasi:  
✓  
Job Knowledge [Kahibalo sa trabah:  
✓  
Quality of Service [Kaliad sa serbi:  
✓  
Additional comment/suggestion/complaint: [Paghinggap ng komento/sugyo/reklamo]:

Add comment/ suggestions/complaint: [Paghinggap ng komento/sugyo/reklamo]:

Figure 47. User (Registrar Staff) – Feedback List (User Interface Design)

Figure 47 illustrates the 'Feedback List' interface, where the registrar can view and print the feedback form of the requesters after the credentials have been released.



SMCC ORCS

Dashboard

CLEARANCE

- Clearance List
- Clearance Disapproved List

REQUESTS

- Requester
- For Processing
- Schedule Release Date
- Feedback List

REPORTS

- Clearance Report
- Credentials Report
- Request History Report

Home > Report > Clearance Report

From \* To \*

01/05/2025 04/05/2025

Department \*

All Departments Print Report

Show 10 entries Search:

#	Name	Department	Course	Date Approved	Status
1	Ricel Ada Villarin	CCIS	BSIT	May 4, 2025	Approved

Showing 1 to 1 of 1 entries Previous 1 Next

SMCC © 2025

**PRINT**

Saint Michael College of Caraga  
 Brgy. 4, Nasipit, Agusan del Norte, Philippines  
 District 8, Brgy. Triangulo, Nasipit, Agusan del Norte, Philippines  
 Tel. Nos. +63 085 343-5231, 283-3113 | Fax: +63 085 898-0892  
[www.smccnasipit.edu.ph](http://www.smccnasipit.edu.ph)

**CLEARANCE REPORT**  
 Period: May 01, 2025 - May 04, 2025  
 Registrar Office

#	Name	Department	Course	Date Approved	Status
1	Ricel Ada Villarin	CCIS	BSIT	May 04, 2025	Approved

Prepared by:  
 Darlene Alag

*Figure 48. User (Registrar Staff) – Report for Clearance (User Interface Design)*

Figure 48 displays the ‘Clearance Report’ page for registrar staff. The registrar can filter clearance records by date and department and generate a printable report. The table shows relevant details such as Name, Department, Course, Date Approved, and Status. A search bar and pagination controls support easy navigation.



SMCC ORCS

Dashboard

CLEARANCE

Clearance List

Clearance Disapproved List

REQUESTS

Requester

For Processing

Schedule Release Date

Feedback List

REPORTS

Clearance Report

Credentials Report

Request History Report

From \* 05/05/2025 To \* 05/05/2025

Department \* All Departments Document Type \* All Documents Search Name \*

Print Report

Show 10 entries

#	Name	Department	Course	Document Type	Mailing Address	Date Released	Status
1	Ricel Ada Villarin	CCIS	BSIT	Certificate - General Weighted Average	D-1, Alipao, ALEGRIA, SURIGAO DEL NORTE	May 5, 2025	Done
2	Ricel Ada Villarin	CCIS	BSIT	Certificate of Honors	D-1, Alipao, ALEGRIA, SURIGAO DEL NORTE	May 5, 2025	Done
3	Ricel Ada Villarin	CCIS	BSIT	Certification of Graduation	D-1, Alipao, ALEGRIA, SURIGAO DEL NORTE	May 5, 2025	Done
4	Ricel Ada Villarin	CCIS	BSIT	Transcript of Record - Employment Purpose	D-1, Alipao, ALEGRIA, SURIGAO DEL NORTE	May 5, 2025	Done

Showing 1 to 4 of 4 entries

Previous 1 Next

**PRINT**

**Saint Michael College of Caraga**

Brgy. 4, Nasipit, Agusan del Norte, Philippines  
 District 8, Brgy. Triangulo Nasipit, Agusan del Norte, Philippines  
 Tel. Nos. +63 085 343-5231 / 283-3113 | Fax: +63 085 898-0892  
[www.smcncaoit.edu.ph](http://www.smcncaoit.edu.ph)

**CREDENTIALS REPORT**

Period: May 05, 2025 - May 05, 2025

Registrar Office

#	Name	Department	Course	Document Type	Mailing Address	Date Released	Status
1	Ricel Ada Villarin	CCIS	BSIT	Certificate - General Weighted Average	D-1, Alipao, ALEGRIA, SURIGAO DEL NORTE	May 05, 2025	Done
2	Ricel Ada Villarin	CCIS	BSIT	Certificate of Honors	D-1, Alipao, ALEGRIA, SURIGAO DEL NORTE	May 05, 2025	Done
3	Ricel Ada Villarin	CCIS	BSIT	Certification of Graduation	D-1, Alipao, ALEGRIA, SURIGAO DEL NORTE	May 05, 2025	Done
4	Ricel Ada Villarin	CCIS	BSIT	Transcript of Record - Employment Purpose	D-1, Alipao, ALEGRIA, SURIGAO DEL NORTE	May 05, 2025	Done

Prepared by:

Darlene Alag

Approved by:

Cecilio C. Ariolla

*Figure 49. User (Registrar Staff) – Report for Requested Credentials (User Interface Design)*

Figure 49 shows the ‘Requested Credentials Report’ page used by registrar staff. It enables users to generate reports by filtering requests through date range, department, document type, and student name. The table displays key information, including the requester’s name, course, type of document requested, mailing address, approval date, and current status. A “Print Report” button is provided for quick printing, search, and pagination features for efficient navigation.



**SMCC ORCS**

- [Dashboard](#)
- [CLEARANCE](#)
  - [Clearance List](#)
  - [Clearance Disapproved List](#)
- [REQUESTS](#)
  - [Requester](#)
  - [For Processing](#)
  - [Schedule Release Date](#)
  - [Feedback List](#)
- [REPORTS](#)
  - [Clearance Report](#)
  - [Credentials Report](#)
  - [Request History Report](#)

Home > Report > Request History Report

From \*

To \*

Department \*

Search Name \*

#	Name	Application Form	Mailing Address	Date Released	Status
1	Ricel Ada Villarin	<a href="#">View</a>	D-1, Alipao, ALEGRIA, SURIGAO DEL NORTE	May 5, 2025	Done

Showing 1 to 1 of 1 entries

Previous
1
Next

SMCC © 2025

<div style="position: absolute; top: 0; left: 0; width: 100%; height: 100



### 3.8 Software Platforms, Development Environment, and Tools

**Table 1**

#### Software Requirements

Components	Specification	Usage
Internet Browser	Any	The internet browser simplifies access to web pages and enables users to navigate the World Wide Web. It connects to the web server, which links users to the GCMS and grants access to its features after a successful login.
Front-end	HTML	HTML serves as the structural foundation of web pages, enabling the organization and arrangement of content components such as text, images, and forms.
	CSS	CSS enhances the visual design of HTML elements, allowing developers to customize layouts, colors, fonts, and spacing, which results in aesthetically pleasing and responsive designs.
	JavaScript	JavaScript adds functionality and interactivity to web pages, empowering developers to implement features such as animations and quick updates that respond to user input.
	Bootstrap	Bootstrap is an HTML, CSS, and JS library that focuses on simplifying the development of informative web pages (as opposed to web applications).
	jQuery	jQuery is a fast, lightweight JavaScript library that simplifies HTML document traversal, event handling, animation, and AJAX interactions to make web development easier and more efficient.



Back-end	MySQL (Database)  PHP	MySQL is an open-source software that offers various features for efficient data management. phpMyAdmin provides a user-friendly interface for tasks like managing databases and running queries.  PHP, a server-side programming language, enables dynamic content generation from static web pages and connects the front end to the backend (database) through an MVC architectural design.
Server	Apache	The XAMPP Control Panel manages XAMPP components, ensuring smooth operation and easy access to Apache and MySQL servers.

### 3.9 Hardware Requirements

**Table 2**

#### Hardware Requirements

Components	Specification	Usage
Device	RAM  Processor  HDD SSD Flash Storage	Electronic devices like computers, laptops, smartphones, and tablets use Random Access Memory (RAM) to quickly access stored information and serve as temporary storage.  The Central Processing Unit (CPU) is the brain of a computing device, responsible for processing data and executing instructions. It has multiple cores for handling different tasks simultaneously.  Computers and laptops use Hard Disk Drives (HDD) or Solid State Drives (SSD) for storage, while mobile devices rely on flash storage for applications, multimedia, and operating systems.
Printer	Any Ink Jet Printer Units	An inkjet printer is a device that prints text, images, and graphics on paper using ink. It is commonly used and usually more affordable than other types of printers. This reliable option is perfect for everyday printing tasks, as it can produce high-quality prints for a variety of documents and images.



---

### 3.10 Ethical Standard

This research, entitled "Online Request of Credentials for Saint Michael College of Caraga," addressed several ethical considerations to ensure that the study met the required standards for research integrity and participant protection.

#### A. Protection of Intellectual Property Rights (IPR)

The "Online Request of Credentials for Saint Michael College of Caraga" system incorporated innovative processes to streamline the credential request procedure. All materials produced during this project, including system documentation, source code, and technical designs, were safeguarded through copyright protection to ensure proper attribution and prevent unauthorized use. The logo of Saint Michael College of Caraga was used in the system to represent the institution and was protected under applicable trademark and copyright laws to prevent misuse. Proper permissions were secured to ensure compliance with intellectual property standards.

#### B. Informed Consent

The capstone project involved conducting interviews with key stakeholders including office heads such as the Registrar, Cashier, LRC personnel, Deans, and Principals, as well as students and alumni, to gather relevant feedback for the development of a web-based system aimed at enhancing the online credential request process at Saint Michael College of Caraga. The interview guide was carefully designed based on the study's objectives and the specific data required to achieve them. It focused on key themes such as existing workflows, encountered challenges, user experiences, and confidentiality practices. Participants were informed about the study's objectives, procedures, minimal risks, and potential benefits, particularly how their feedback would improve the system's functionality. Before conducting the interviews, a letter of consent was provided and secured to ensure that participants fully understood the purpose of the study and voluntarily agreed to participate. Participants were also informed of their right to withdraw from the study at any time



---

without facing any consequences. All information collected during the interviews was treated with strict confidentiality and used solely for system development and academic research.

#### **C. Data Privacy and Confidentiality**

The system ensured data protection by implementing strong security measures, including encryption and Role-Based Access Control (RBAC). Access to sensitive information, such as student credentials and request statuses, was restricted to authorized personnel only. User data was safeguarded and not disclosed to any unauthorized individuals. For performance analysis, anonymized and aggregated data were utilized while maintaining strict confidentiality for all users, upholding privacy and ethical standards throughout system operations.

#### **D. Voluntary Participation and Freedom to Withdraw**

Participation in developing and testing the Online Request of Credentials System was entirely voluntary. Participants were fully informed of their right to withdraw from the project without impacting their professional roles or standing. To uphold the integrity of the research process, no coercion, undue influence, or financial incentives were employed, ensuring that participation remained completely voluntary and free from bias. Clear and transparent communication regarding the study's objectives, procedures, and participants' rights was consistently provided. This approach fostered mutual respect and understanding, promoting ethical engagement and cooperation throughout the project.

#### **E. Minimization of Harm and Risk Management**

The research team remained committed to ensuring the safety and well-being of all participants by minimizing potential risks throughout the development and implementation of the Online Request of Credentials System. Careful consideration was given to addressing technical, emotional, or social concerns that could arise during system testing or user feedback collection. If participants encountered difficulties, the research team and designated support personnel were

---



readily available to provide assistance and user guidance to help them navigate the system effectively. Furthermore, all collected data, including user inputs and records, were securely stored and handled with strict confidentiality to prevent unauthorized access. These measures ensured that no participant experienced adverse consequences due to their involvement in the project. As the study did not involve animal subjects, applying the 3Rs principle (Replacement, Reduction, and Refinement) was not applicable.

#### **F. Beneficence and Contribution to Knowledge**

The development of the Online Credential Request System aimed to improve the efficiency of requesting and processing academic records at Saint Michael College of Caraga, benefiting students, alumni, and the Office of the Registrar. By reducing processing time and minimizing errors, the system enhanced service delivery. It provided a more user-friendly and accurate experience, aligning with the institution's commitment to technological innovation and modernized administrative processes.

Upon request, participants were provided with a summary of the findings and improvements to the system. The project was presented through a comprehensive dissemination strategy, including demonstrations and informational materials, to raise awareness of the system's purpose and benefits and address concerns.

#### **G. Justice and Fair Participant Selection**

Participants were selected through a fair and transparent process that promoted inclusiveness and diversity. Selection was based on predefined criteria and focused on individuals from key stakeholder groups, including students, alumni, and staff from the Office of the Registrar, Office of the Learning Resource Center, Office of the Cashier, and the offices of the Deans or Principals. This ensured that participants could provide valuable insights for system development.



---

The process avoided discrimination and included individuals from various backgrounds, giving all stakeholders an equal opportunity to contribute meaningfully.

#### **H. Data Integrity and Accuracy**

All data related to the design, testing, and evaluation of the Online Request of Credentials System were collected with high standards of accuracy and integrity. Standardized procedures ensured that feedback and performance data were documented correctly and interpreted. ChatGPT was used during the preliminary analysis to organize ideas and summarize insights, while Grammarly helped ensure clarity and grammatical accuracy. All AI-generated content was reviewed and validated by the research team. No potential biases, errors, or limitations were identified or documented, as the tools were used only to support the study's accuracy and reliability.

#### **I. Transparency and Honesty in Reporting**

All findings, including system performance data and user feedback, were accurately reported without manipulation to preserve the study's credibility. Any potential conflicts of interest were openly disclosed to ensure transparency. Third-party sources, such as research papers, tools, and frameworks, were properly cited to maintain academic integrity and uphold ethical standards in reporting. Furthermore, the documentation process was supported by using artificial intelligence tools to enhance clarity, consistency, and accuracy. This comprehensive approach helped ensure the results remained objective, reliable, and informative for all stakeholders.

#### **J. Use of Patented or Copyrighted Materials**

The project did not involve using any patented technologies, so no clearance from patent holders was required. It focused on developing a web-based system to enhance the credential request process using open-source and standard technologies that did not require proprietary rights. Additionally, the study utilized copyrighted materials, such as literature and previous research, all of which were cited correctly to ensure due credit was given to the original authors,

---



---

thereby upholding intellectual property rights and ethical research standards.

#### **K. Ethical Considerations for Animal and Human Trials**

The project involved human participants, specifically teachers, students, and alumni, who were fully informed about their rights, the minimal risks involved, and the benefits of their participation. Ethical clearance for the use of animals was not applicable, as no animals were involved in the study. To ensure confidentiality and protect the professional standing of the participants, all feedback provided by teachers, students, and alumni was anonymized, upholding ethical standards throughout the research process.

#### **L. Responsible Use of AI and Other Related Technologies**

AI tools like Grammarly, paraphrasing tools, and ChatGPT were used to improve the quality of the written content. Grammarly helped correct grammar and syntax errors, ensuring accuracy throughout the text. Paraphrasing tools enhance readability by rephrasing sentences for better clarity. ChatGPT contributed by generating ideas, refining wording, and suggesting alternative content presentations. Together, these tools ensured the paper was clear, professional, and well-organized, allowing the research team to focus more on the technical aspects of the study.

#### **M. Ethical Clearance and Institutional Approval**

The capstone project was granted ethical approval by the Institutional Review Board of Saint Michael College of Caraga. This approval process ensured that the study complied with recognized ethical standards by safeguarding the rights of participants, maintaining data confidentiality, and protecting intellectual property. The necessary clearance was secured before the project began, ensuring that all research activities were conducted responsibly and complied with institutional policies and ethical guidelines.



## CHAPTER 4

### SOFTWARE DEVELOPMENT AND TESTING

This chapter describes the main activities regarding the design and testing of the Online Request of Credentials System (ORCS). It sets forth the strategies used in the system's developmental process, defines the specific steps taken during its construction, and mentions activities done to preserve the system throughout its lifecycle.

#### 4.1 Development Process

The systematic software development process of the Online Request of Credentials System (ORCS) is illustrated through the input-process-output diagram shown in Figure 51.

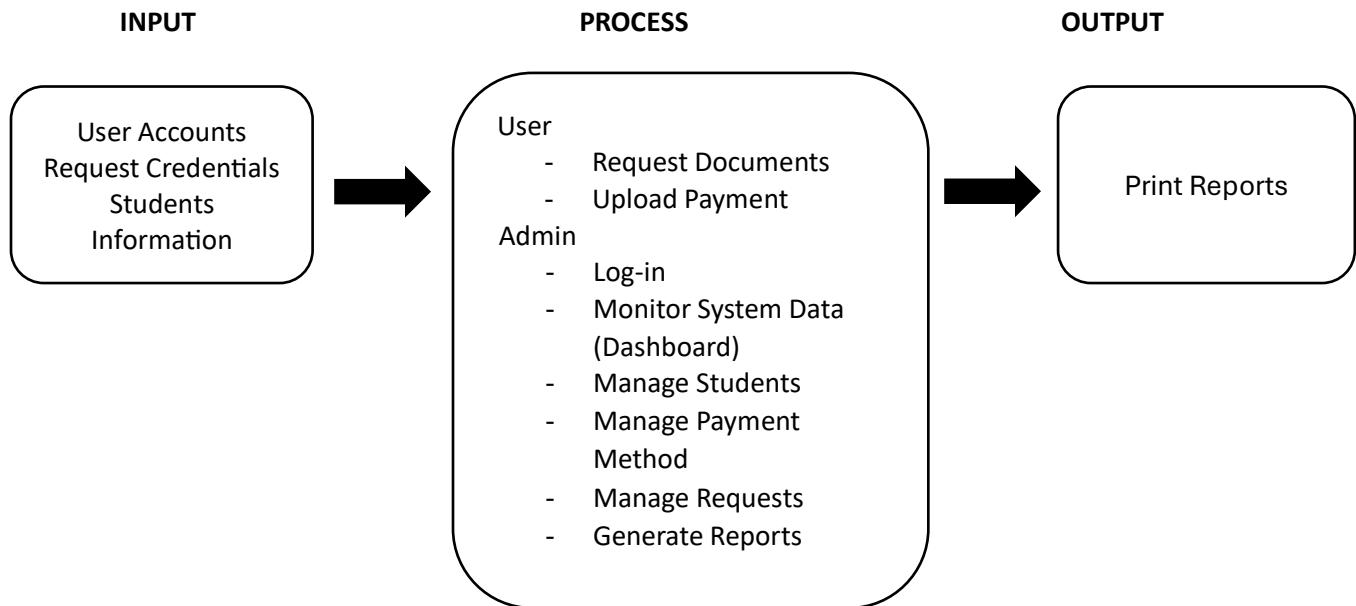


Figure 51. Input-Process-Output Diagram

#### Input

The input "User Accounts," "Requested Credentials," and "Students Information" represent the data that is provided to a system for processing. User Accounts refer to unique profiles or identities created for individual users in the system. The requested credentials are the



---

authentication details, usernames, and passwords that should be user-specific and would verify and allow entry to the system. Alongside the data about students, such as names, grades, or courses specific to students, they are entered or retrieved for many purposes. The above inputs thus authenticate users, manage access, and process student-related data for the system.

### **Process**

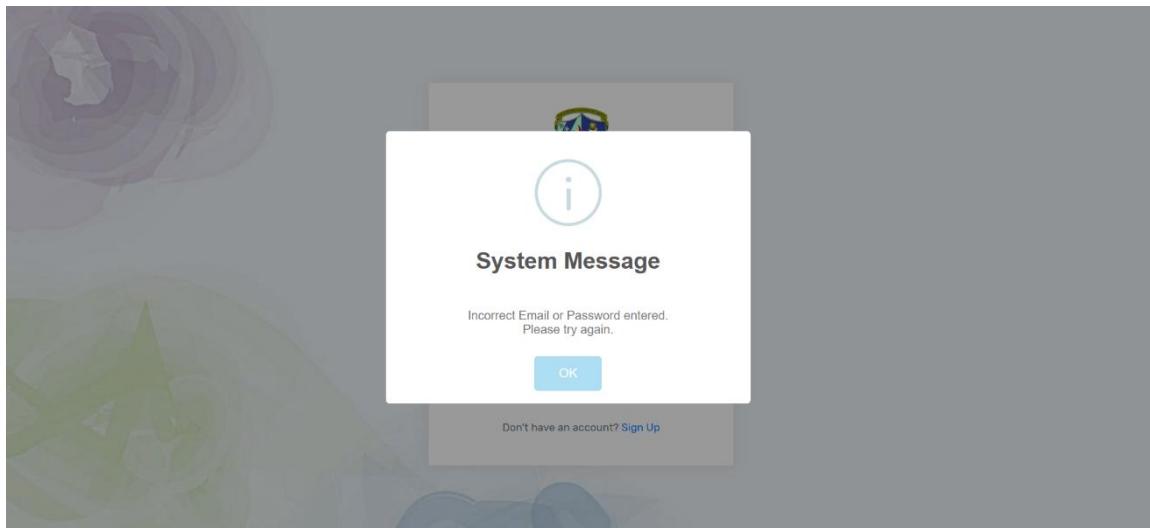
Various actions are involved in the system's processes depending on user roles. For the User, the process consists of Requesting Documents, where he/she can ask for necessary files, and Upload Payment, wherein he/she inputs payment particulars. Admin processes include several functions: logging in to verify access rights, monitoring system data on a dashboard to view system performance, managing students by viewing their information, managing payment methods to make payment options available, managing requests to respond to user requests, and generating reports on the performance and insights from the data thereof. These processes help the system function smoothly and manage itself.

Pop-up alert for success or a different message for failure. If an error occurs, it logs the details to the console for debugging.

A screenshot of a login page titled "SMCC Online Request of Credentials!". The page features a logo at the top left, followed by two input fields for "Email" and "Password". Below the password field are "Show Password" and "Forgot Password?" links. A large blue "Sign in" button is centered below the fields. At the bottom of the form, there is a link "Don't have an account? Sign Up". The background of the page has a faint floral watermark.

*Figure 52. Admin/All Users – Login Page*

Figure 52 shows the first thing Users see when they access the system. To access the system, users enter the username and password of their respective accounts.



*Figure 53. Admin/All Users – Login Error Notification*

Figure 53 displays a modal dialog box (pop-up window) over a login or sign-in screen. It alerts users when they input incorrect login credentials.



```

dbassets > login.php
1 <?php
2     require_once ('connection.php');
3     include 'functions.php';
4     session_start();
5
6     $email = mysqli_real_escape_string($connection, $_GET["email"]);
7     $password = mysqli_real_escape_string($connection, $_GET["password"]);
8
9     // Query to get the user record based on the email
10    $query = mysqli_query($connection, "SELECT u.*, d.department FROM users u
11                                LEFT JOIN ref_departments d ON u.department_id = d.id
12                                WHERE email = '$email' AND u.`status` != 'Verify Email';") or die(mysqli_error($connection));
13
14    $row = mysqli_fetch_array($query);
15
16    // If user exists
17    if ($row) {
18        $user_status = $row['status'];
19        $id = $row['id'];
20        $_SESSION['id'] = $row['id'];
21        $_SESSION['fname'] = $row['fname'];
22        $_SESSION['mname'] = $row['mname'];
23        $_SESSION['lname'] = $row['lname'];
24        $_SESSION['name'] = $row['fname'] . ' ' . $row['lname'];
25        $_SESSION['email'] = $row['email'];
26        $_SESSION['role'] = $row['role'];
27
28        if ($row['role'] != 'Client') {
29            $department_id = $row['department_id'];
30            $_SESSION['department_id'] = $row['department_id'];
31
32            $queryDepartment = mysqli_query($connection, "SELECT * FROM ref_departments WHERE id = '$department_id';") or die(mysqli_error($connection));
33            $departmentRow = mysqli_fetch_array($queryDepartment);
34            $_SESSION['department'] = $departmentRow['department'];
35        }
36
37        // First, check if the user account is active
38        if ($user_status == 'Active') {
39            // Then verify the password using password_verify()
40            if (password_verify($password, $row['password'])) {
41                log_action($id, 'Logged In', $connection);
42                echo 'success';
43            } else {
44                echo "Incorrect Email or Password entered. \nPlease try again.";
45            }
46        } else {
47            echo 'User account deactivated! Please contact system administrator for further assistance.';
48        }
49
50    } else {
51        echo "Incorrect Email or Password entered. \nPlease try again.";
52    }
53 ?>

```

Figure 54. Code Snippet for Admin/Users – login.php

Figure 54 illustrates a PHP backend code snippet that handles user authentication by validating email and password credentials provided via GET parameters. This script is initiated by including two essential external files: connection.php for database connectivity and functions.php, likely for utility functions such as log\_action. The session\_start() function initializes a session, enabling the script to store and retrieve user-specific data throughout the session lifecycle.

Key to the login mechanism is the data sanitization step using mysqli\_real\_escape\_string() on both the email and password inputs (\$\_GET["email"] and \$\_GET["password"]), protecting against SQL injection



---

vulnerabilities. These sanitized inputs are then used in a `SELECT SQL` query, which fetches the user details from the `users` table and joins the `ref_departments` table to retrieve associated department data. Notably, the condition `u.status != 'Verify Email'` ensures that users without email verification are excluded from the login process.

If a matching user record is found, the script proceeds to extract user information into session variables such as `$_SESSION["id"]`, `$_SESSION["fname"]`, `$_SESSION["lname"]`, and more, which are essential for maintaining state across the application. An additional check determines if the user's role is not 'Client', and if so, it performs a secondary query to retrieve the department name from the `ref_departments` table using the `department_id`, then stores it in the session as `$_SESSION["department"]`.

Following session initialization, the script performs two critical validations: first, it checks if the user's status is 'Active', ensuring that only active accounts can proceed. Second, it uses `password_verify($password, $row['password'])` — a crucial security function — to compare the entered password with the hashed password stored in the database. If successful, the `log_action()` function logs the login activity, and the string 'success' is echoed back, signaling the frontend that the login was successful.

If any stage fails, such as an incorrect password, inactive account, or no matching user, the script responds with a meaningful error message like "Incorrect Email or Password entered" or "User account deactivated!", which is likely displayed to the user, as shown in the modal alert from the earlier image.




SMCC Online Request of Credentials! 🌟

First Name *	Middle Name	Last Name *	Ext *
Ricel Ada	Salutis <input checked="" type="checkbox"/> Has Middle Name	Villarin	N/A
Birthday *	Last School Year Attended *	Contact *	Email *
01/11/2003	2025	09705366534	adangvillarin@gmail.com
House No. & Street *	Province *		
D-6	AGUSAN DEL NORTE		
Municipality / City *	Barangay *		
NASIPIT	Carmagong		
Upload Valid ID *	Upload a photo holding your valid ID *		
<input type="button" value="Choose File"/> Valid ID.jpg	<input type="button" value="Choose File"/> Holding with valid ID.jpg		
<small>Allowed file types: png, jpg, jpeg, pdf. Please upload any valid government ID (Driver's License, Passport, etc.)</small>			
<a href="#" style="background-color: #007bff; color: white; padding: 5px 20px; text-decoration: none; border-radius: 5px;">Sign Up</a>			
<small>Already have an account? <a href="#">Sign In instead</a></small>			

Figure 55. Students/Alumni – Sign Up Page

Figure 55 shows the signup page, where students can input all the necessary information to create an account.

```

46 iv class="form-group row mb-3">
47   <div class="col-md-4">
48     <label for="fname">First Name <span class="text-danger">*</span></label>
49     <input type="text" class="form-control form-control-lg" id="fname" name="fname" placeholder="Enter your first name" required autofocus>
50   </div>
51   <div class="col-md-3">
52     <label for="mname">Middle Name</label>
53     <input type="text" class="form-control form-control-lg" id="mname" name="mname" placeholder="Enter your middle name" readonly />
54   </div>
55     | <input type="checkbox" id="editMname" /> <label for="editMname">Has Middle Name</label>
56   </div>
57 </div>
58   <div class="col-md-3">
59     <label for="lname">Last Name <span class="text-danger">*</span></label>
60     <input type="text" class="form-control form-control-lg" id="lname" name="lname" placeholder="Enter your last name" required />
61   </div>
62   <div class="col-md-2">
63     <label for="ext">Ext <span class="text-danger">*</span></label>
64     <select id="ext" name="ext" class="select2 form-control form-control-lg" required>
65       | <option value="N/A">N/A</option>
66       | <option value="JR">JR</option>
67       | <option value="SR">SR</option>
68       | <option value="I">I</option>
69       | <option value="II">II</option>
70       | <option value="III">III</option>
71       | <option value="IV">IV</option>
72       | <option value="V">V</option>
73     </select>
74   </div>
75 
```



```

12   <title>Register | SMCC Online Credentials System</title>
13   <!-- Custom CSS -->
14   <link href="assets/dist/css/style.min.css" rel="stylesheet">
15   <link href="assets/images/smcc-logo.png" rel="icon" type="image/png" sizes="96x96">
16   <!-- <link href="assets/css/sweetalert/sweetalert.css" rel="stylesheet" -->
17   <!-- Sweet Alert -->
18   <link href="assets/css/sweetalert2/sweetalert2.min.css" rel="stylesheet">
19 </head>
20
21 <body>
22   <div class="main-wrapper">
23     <!-- <div class="preloader">
24       <div class="lds-ripple">
25         <div class="lds-pos"></div>
26         <div class="lds-pos"></div>
27       </div>
28     </div> -->
29
30   <div class="auth-wrapper d-flex no-block justify-content-center align-items-center"
31     style="background: url(assets/images/auth-bg2.jpg) no-repeat center center; background-size: cover;">
32
33     <div class="auth-box p-4 bg-white rounded" style="max-width: 1100px;">
34       <div id="loginform">
35         <div class="logo">
36           <span class="db"></span>
37           <br>
38           <br>
39           <h3 class="box-title mb-3">SMCC Online Request of Credentials! 🌟</h3>
40
41         <!-- Form -->
42         <div class="row">
43           <div class="col-12">
44             <form class="form-horizontal m-t-20" id="register-form" autocomplete="off">
45               <!-- First row -->
46
47               <script>
48                 // Set the current year dynamically as max
49                 document.getElementById('yearAttended').max = new Date().getFullYear();
50
51                 document.getElementById('editMname').addEventListener('change', function () {
52                   const mnameInput = document.getElementById('mname');
53                   if (this.checked) {
54                     mnameInput.removeAttribute('readonly'); // Make the input editable
55                   } else {
56                     mnameInput.setAttribute('readonly', true); // Make the input readonly
57                     mnameInput.value = '';
58                   }
59                 });
60
61                 $('#select[name="province"]').on('change', function() {
62                   var province_id = $(this).val();
63                   $.get('dbassets/reference/get_city_by_province.php', { province_id }, function(response) {
64                     var data = JSON.parse(response);
65
66                     var $citySelect = $('select[name="city"]');
67                     $citySelect.empty();
68                     $citySelect.append(
69                       $(<option>).val('').text('Please select')
70                     );
71
72                     // Populate the second select with new options
73                     $.each(data, function(index, item) {
74                       $citySelect.append(
75                         $(<option>).val(item.city_code).text(item.city)
76                       );
77
78                     });
79
80                     // Only proceed if the user clicked "Submit" (i.e., isConfirmed is true)
81                     if (isConfirmed) {
82                       if (userInput) {
83                         validInput = await checkInput(userInput);
84                         console.log('OTP validation result:', validInput);
85
86                         if (validInput) {
87                           // Success - OTP is correct
88                           await Swal.fire({
89                             title: '🌟 OTP Verified!',
90                             text: 'Your OTP is correct! Your account is under review.',
91                             icon: 'success',
92                             confirmButtonText: 'Great, Thanks!',
93                             showCancelButton: true,
94                             customClass: {
95                               popup: 'swal-popup',
96                               title: 'swal-title',
97                               content: 'swal-content',
98                               confirmButton: 'swal-confirm-btn'
99                             },
100                            background: '#f0f0f0'
101                          });
102
103                     }
104
105                   });
106
107                 });
108
109               </script>
110             </div>
111           </div>
112         </div>
113       </div>
114     </div>
115   </div>
116
117   <div class="text-center">
118     <p>Don't have an account? <a href="#">Create one</a></p>
119     <p>Forgot your password? <a href="#">Reset it</a></p>
120   </div>
121
122 </body>
123
124 <!-- Scripts -->
125 <script src="https://cdn.jsdelivr.net/npm/sweetalert2@11">
126 <script src="assets/js/register.js">
127
128 </html>

```

Figure 56. Code Snippet for Students/Alumni – register.php



---

Figure 56 illustrates a comprehensive user registration form developed using HTML, PHP, JavaScript (with jQuery), and SweetAlert2 for interaction feedback. This form is structured using Bootstrap's grid system, ensuring responsive behavior across devices, and is divided into multiple sections to collect detailed user information.

The form begins with a series of personal input fields such as First Name, Middle Name, Last Name, and Name Extension. A notable feature is the `<input type="checkbox" id="editMname" />` paired with JavaScript logic that toggles the `readonly` attribute of the middle name field (`mname`), ensuring users only provide a middle name if applicable — a small but impactful enhancement to usability and data accuracy.

The dynamic population of address fields demonstrates robust interaction with backend data. Upon changing the Province dropdown (`select[name="province"]`), an **AJAX** call (`$.get`) fetches related City/Municipality options via `get_city_by_province.php`. Similarly, choosing a city triggers another **AJAX** call to populate the Barangay options through `get_barangay_by_city.php`. This technique ensures location consistency by only displaying options relevant to the user's previous selections.

Another significant portion of the code is handling file uploads, where users are required to submit both a valid ID and a photo of themselves holding it. Each file input is constrained to specific **MIME** types (e.g., .png, .jpg, .jpeg, .pdf) using the `accept` attribute, enforcing file type compliance at the client level.

The JavaScript code block at the bottom of the form adds interactivity and asynchronous functionality. The form submission is handled using `$('#register-form').on('submit')`, where a `Swal.fire` loading modal is shown while the form data using `FormData` is sent via **AJAX POST** to `save.php`. This decouples the form from a traditional page reload and provides a modern, user-friendly interface.



Upon successful form submission, the user is prompted to enter a One-Time Password (OTP) through a **SweetAlert2** modal. The function `askForInput()` encapsulates this loop, where the user's OTP input is validated via the `checkInput()` function, which performs a **POST** request to `verify_email.php`. This OTP-based verification adds a crucial security layer before the account is finalized.

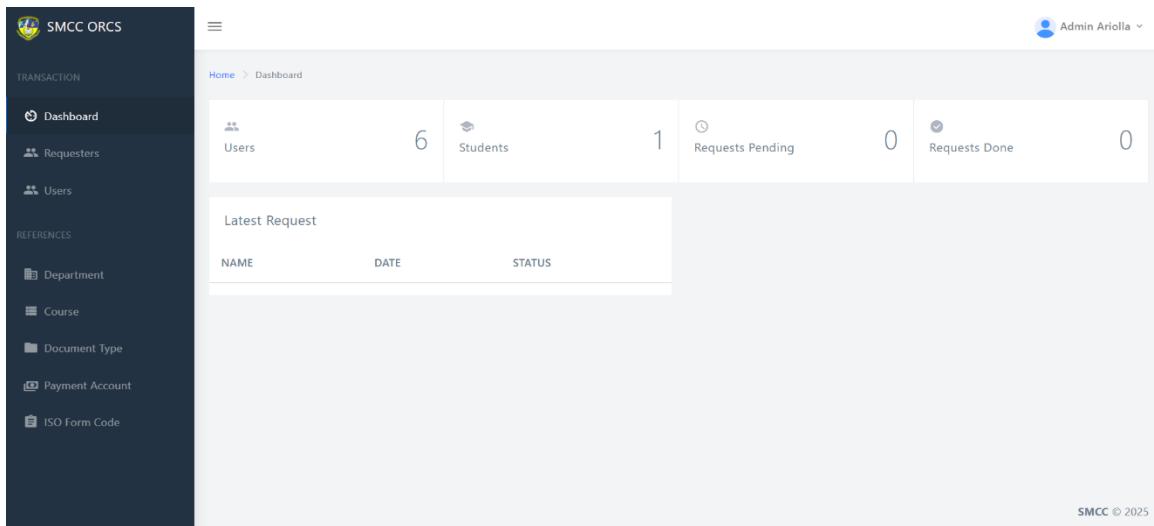


Figure 57. Admin – Dashboard

Figure 57 shows the Dashboard page, which provides an overview of key data in the system.

```

117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
    <?php
        require_once('../dbassets/connection.php');
        $query = "SELECT r.*, u.fname, u.lname FROM requests r
                  JOIN users u ON r.user_id = u.id
                  ORDER BY r.submitted_at DESC
                  LIMIT 5;";
        $result = mysqli_query($connection, $query);
        while($rs = mysqli_fetch_array($result)) {
            // Handle NULL for date_requested
            $date_requested = !is_null($rs['date_requested']) ? date('F j, Y', strtotime($rs['date_requested'])) : 'N/A';

            // Handle NULL for status
            $status = !is_null($rs['status']) ? $rs['status'] : 'Unknown';
            $status_class = ($status == 'Active') ? 'success' : 'primary';
        }
    <?>
    <tr>
        <td><?php echo $rs['fname'].' '.$rs['lname']; ?></td>
        <td><?php echo $date_requested; ?></td>
        <td data-target="status">
            | <span class="badge badge-<?php echo $status_class; ?>"><?php echo $status; ?></span>
        </td>
    </tr>
<?php } ?>
```

Figure 58. Code Snippet for Admin – dashboard\_admin.php



Figure 58 illustrates a PHP code snippet in a dashboard interface to display dynamic data about user accounts and service requests within a civil registry management system. The script establishes a database connection using the `require_once('..../dbassets/connection.php')` statement, which grants access to essential MySQL functions. A central component of this functionality is the SQL query executed through `mysqli_query()`, which performs multiple subqueries in a single call to retrieve statistical summaries from the database. Specifically, it counts the total number of users, excluding those with the 'Client' role, assigning the result to the `$users` variable. It then calculates the number of users with the 'Client' role and assigns that to `$students`. Additionally, it gathers the number of ongoing service requests, those whose status is not marked as 'Done', into the `$ongoing` variable, and counts completed service requests (status marked as 'Done') in `$done`.

These retrieved values are rendered dynamically into four Bootstrap card components using `<?php echo $variable; ?>` to inject the data directly into the HTML. Each card is visually represented with appropriate icons and labels, giving administrators a quick overview of current system statistics, such as the total number of system users, students, ongoing requests, and completed requests.

Another portion of the dashboard displays the five most recent service requests. This section begins with a separate SQL query that joins the requests table with the users table using the statement `SELECT r.*, u.fname, u.lname FROM requests r JOIN users u ON r.user_id = u.id ORDER BY rsubmitted_at DESC LIMIT 5`. This **JOIN** operation is essential as it enables the system to pull together request details and user information in a single, efficient operation. The retrieved data is processed inside a while loop, which dynamically creates table rows displaying the requester's full name, the date the request was submitted, and the current status of the request.



To ensure robustness, the code includes conditional checks using `is_null()` to handle any missing values gracefully. For instance, if `date_requested` or `status` fields are null, the code substitutes them with user-friendly defaults such as 'N/A' or 'Unknown'. Additionally, the request status is visually emphasized using Bootstrap badge classes, where the `$status_class` is conditionally set to either 'success' or 'primary', depending on whether the request status is labeled 'Active' or not. This provides a clear and immediate visual cue for the dashboard administrator.

ID	Name	Year Attended	Contact	Email	Status	Action
7	Villarin, Richel Ada	2025	09705366534	adangvillarin@gmail.com	Active	Action

Figure 59. Admin – Requesters List

Figure 59 shows the requester's account list; this interface manages the requester's accounts.

```

89
90 <div class="btn-group">
91   <button type="button" class="btn btn-info">Action</button>
92   <button type="button" class="btn btn-info dropdown-toggle dropdown-toggle-split" data-toggle="dropdown" aria-haspopup="true" aria-ex
93     <span class="sr-only">Toggle Dropdown</span>
94   </button>
95   <div class="dropdown-menu">
96     <a class="dropdown-item" href="javascript:void(0)" data-role='view_image' data-image=<?php echo $rs['id_document']; ?>" data-i
97       <i class="mdi mdi-upload"></i> View Upload</a>
98
99     <?php if ($rs['status'] == 'Pending') { ?>
100      <a class="dropdown-item text-primary" href="javascript:void(0)" data-role='activate' data-id=<?php echo $id; ?>">
101        <i class="mdi mdi-account"></i> Activate Account</a>
102      <a class="dropdown-item text-danger" href="javascript:void(0)" data-role='disapprove' data-id=<?php echo $id; ?>">
103        <i class="mdi mdi-account-minus"></i> Disapprove Account</a>
104    <?php } ?>
105
106      <a class="dropdown-item text-info" href="javascript:void(0)" data-role='reset' data-id=<?php echo $id; ?>">
107        <i class="mdi mdi-lock"></i> Reset Password</a>
108    </div>
109  </div>
110

```



```

157 <script>
158 $(document).ready(function(){
159
160     $(document).on('click', 'button[data-role=edit]', function(){
161         edit = true;
162         var id = $(this).data('id');
163         var fname = $('#' + id).children('td[data-target=fname]').text();
164         var mname = $('#' + id).children('td[data-target=mname]').text();
165         var lname = $('#' + id).children('td[data-target=lname]').text();
166         var email = $('#' + id).children('td[data-target=email]').text();
167         var role = $('#' + id).children('td[data-target=role]').text();
168         var status = $.trim($('#' + id).children('td[data-target=status]').text());
169         $('input[name="fname"]').val(fname);
170         $('input[name="mname"]').val(mname);
171         $('input[name="lname"]').val(lname);
172         $('input[name="email"]').val(email);
173         $('#role').val(role).change();
174         $('#status').val(status).change();
175
176         $('input[name="id"]').val(id);
177         $('#add-modal').modal('toggle');
178     });
179
180     $(document).on('click', 'a[data-role=view_image]', function(){
181         var imagePath = $(this).data('image');
182         var imagePath2 = $(this).data('image2');
183         $('#modalImage').attr('src', './assets' + imagePath);
184         $('#modalImage2').attr('src', './assets' + imagePath2);
185         $('#view-modal').modal('toggle');
186     });
187
188     $(document).on('click', 'a[data-role=activate]', function() {
189         var id = $(this).data('id');
190
191     });
192
193     const url = '../dbassets/user/reset-password.php';
194
195     // Perform the AJAX POST request
196     $.post(url, { id }, (response) => {
197         // Close the loading spinner
198         Swal.close();
199
200         if ($.trim(response) === 'success') {
201             toastrOptions();
202             toastr.success('Reset Successfully!', 'System Message');
203         } else {
204             console.log(response);
205             toastr.error('Failed to reset the password', 'System Message');
206         }
207     }).fail(function() {
208         // In case of any errors with the AJAX request
209         Swal.close();
210         toastr.error('An error occurred while resetting the password', 'System Message');
211     });
212 });

```

*Figure 60. Code Snippet for Admin – students.php*

Figure 60 presents the Student Account Management interface, a dynamic and user-friendly system component designed to display and manage student accounts. The process starts with establishing a database connection through `require_once ('../dbassets/connection.php');`, followed by an SQL query that fetches all student accounts with a role of "Client" whose status is not "Verify Email". Each user's data is rendered within a table, where `mysqli_fetch_array()` iterates through each row to display key information such as name, year attended, contact, email, and status. The status is



visually marked with dynamic badge colors—green for "Active" and red for others—providing immediate insight into each account's current state.

Administrative controls are provided via an Action dropdown in each row, offering options to view ID uploads, activate or disapprove pending accounts, and reset passwords. These options are powered by data-role attributes that trigger specific JavaScript functions. For example, selecting “View Upload” opens a modal that dynamically loads the student's submitted ID and a photo of them holding it. Account activation and disapproval actions invoke **SweetAlert2** confirmation prompts, followed by **AJAX POST** requests to `activate.php` or `disapprove.php`, with the system responding through success messages and page refreshes. Similarly, password resets prompt confirmation and send a request to `reset-password.php`, with feedback delivered via **Toastr** notifications, ensuring a seamless and responsive admin experience.

The screenshot shows the 'Add / Edit User Information' form. The form fields include Firstname, Middlename, Lastname, Email, Department (set to Elementary), Role (dropdown menu), and Status (set to Active). Below the form is a table listing users with columns for ID, Firstname, Middlename, Lastname, Email, Department, Role, Status, and Action. Each user row has a green 'Active' badge and a blue 'Action' dropdown menu. The sidebar on the left shows navigation links for TRANSACTIONS (Dashboard, Requesters, Users) and REFERENCES (Department, Course, Document type, Payment Account, ISO Form Code).

ID	Firstname	Middlename	Lastname	Email	Department	Role	Status	Action
1	Enter data here	Enter data here	Enter data here	Enter data here	Elementary	Select a role	Active	Action
2	Enter data here	Enter data here	Enter data here	Enter data here	Elementary	Select a role	Active	Action
3	Enter data here	Enter data here	Enter data here	Enter data here	Elementary	Select a role	Active	Action
4	Enter data here	Enter data here	Enter data here	Enter data here	Elementary	Select a role	Active	Action
5	Enter data here	Enter data here	Enter data here	Enter data here	Elementary	Select a role	Active	Action
6	Enter data here	Enter data here	Enter data here	Enter data here	Elementary	Select a role	Active	Action

Figure 61. Admin – Add User

Figure 61 illustrates the Add User interface, which enables administrators to register new users or staff members who will participate in managing and processing credential requests submitted by students and alumni.



The screenshot shows the 'User List' page of the SMCC ORCS system. The left sidebar includes links for Dashboard, Requesters, Users, Department, Course, Document Type, Payment Account, and ISO Form Code. The main area displays a table with the following data:

ID	First Name	Middle Name	Last Name	Email	Role	Department	Status	Action
2	Darlene	Registrar	Alag	ricelada_villarin@smccnasipit.edu.ph	Registrar	Registrar	Active	Action
3	Noel	Cashier	Baja	popohinautan45@gmail.com	Cashier	Cashier	Active	Action
4	Renel	Librarian	Maneda	rjsgwapo123@gmail.com	Librarian	Learning Resource Center	Active	<span>Edit User</span>
5	Daisa	DeanCCIS	Gupit	angelmaebaguio16@gmail.com	Dean	CCIS	Active	Action
6	Rene	Principal JHS	Japitana	angelo_hinautan@smccnasipit.edu.ph	Principal	JHS	Active	Action

Showing 1 to 5 of 5 entries

Figure 62. Admin – User List

Figure 62 presents the User List interface, where all registered users are displayed for administrative management. The Action button allows the administrator to modify user information when updates are necessary and reset user passwords.

The screenshot shows the 'Add / Edit User Information' modal dialog. The form fields include:

- Firstname: Noel
- Middlename: Cashier
- Lastname: Baja
- Email: popohinautan45@gmail.com
- Department: Cashier
- Role: Cashier
- Status: Active

At the bottom right of the modal are 'Close' and 'Save changes' buttons.

Figure 63. Admin – Edit User

Figure 63 illustrates the Edit User interface, which enables administrators to update user information when modifications are required.



The screenshot shows a user list table with columns: ID, First Name, Middle Name, Last Name, Department, Status, and Action. A modal dialog box is overlaid on the table, containing a large exclamation mark icon at the top, followed by the text "Are you sure you want to reset the password?", and two buttons: "Yes" (red) and "No" (gray). The background table shows several users with their details and roles.

ID	First Name	Middle Name	Last Name	Department	Status	Action
2	Darlene	Regis		Registrar	Active	Action
3	Noel	Cashier		Cashier	Active	Action
4	Renel	Librarian		Librarian	Active	Action
5	Daisa	DeanCCIS	Gupit	Dean	CCIS	Action
6	Rene	Principal JHS	Japitana	Principal	JHS	Action

Figure 64. Admin – Reset User Password

Figure 64 shows a SweetAlert confirmation dialog that allows the administrator to reset a user's password in cases where it has been forgotten.

```

73 <?php
74     require_once('../dbassets/connection.php');
75     $session_id = $_SESSION['id'];
76     $query = "SELECT u.*, d.department FROM users u
77             LEFT JOIN ref_departments d ON u.department_id = d.id
78             WHERE u.`role` != 'client' AND u.id != '$session_id';";
79     $result = mysqli_query($connection,$query);
80     while($rs = mysqli_fetch_array($result)) {
81         $id = $rs['id'];
82     }
83     <tr id=<?php echo $id; ?>">
84         <td><?php echo $id; ?></td>
85         <td data-target="fname"><?php echo $rs['fname'];?></td>
86         <td data-target="mname"><?php echo $rs['mname'];?></td>
87         <td data-target="lname"><?php echo $rs['lname'];?></td>
88         <td data-target="email"><?php echo $rs['email'];?></td>
89         <td data-target="role"><?php echo $rs['role'];?></td>
90         <td data-target="department"><?php echo $rs['department'];?></td>
91         <td data-target="status">
92             <span class="badge badge-<?php if ($rs['status'] == 'Active') echo 'success'; else echo 'danger';?>"><?php echo $rs['status'];?>
93         </td>
94     <td>
95         <div class="btn-group">
96             <button type="button" class="btn btn-info">Action</button>
97             <button type="button" class="btn btn-info dropdown-toggle dropdown-toggle-split" data-toggle="dropdown" aria-haspopup="true"
98                 <span class="sr-only">Toggle Dropdown</span>
99             </button>
100            <div class="dropdown-menu">
101                <a class="dropdown-item" href="javascript:void(0)" data-role='edit' data-id=<?php echo $id; ?>">
```



```
122 <!-- ADD MODAL -->
123 <div id="add-modal" class="modal fade" tabindex="-1" role="dialog" aria-labelledby="myModalLabel" aria-hidden="true" data-backdrop="stat
124   <div class="modal-dialog modal-lg">
125     <div class="modal-content">
126       <div class="modal-header">
127         <h4 class="modal-title font-weight-bold" id="myModalLabel">Add / Edit User Information</h4>
128       </div>
129       <form id="user-form" class="form-horizontal m-t-30" autocomplete="off">
130         <input type="hidden" name="id">
131         <div class="modal-body">
132           <div class="form-group">
133             <label>firstname <span class="text-danger">*</span> </label>
134             <input type="text" name="fname" class="form-control" placeholder="Enter data here" required>
135           </div>
136           <div class="form-group">
137             <label>Middlename </label>
138             <input type="text" name="mname" class="form-control" placeholder="Enter data here">
139           </div>
140           <div class="form-group">
141             <label>Lastname <span class="text-danger">*</span></label>
142             <input type="text" name="lname" class="form-control" placeholder="Enter data here" required>
143           </div>
144           <div class="form-group">
145             <label>Email <span class="text-danger">*</span></label>
146             <input type="text" name="email" class="form-control" placeholder="Enter data here" required>
147           </div>
148
149           <div class="form-group row">
150             <div class="col-md-12">
151               <label for="department">Department <span class="text-danger">*</span></label>
152               <select id="department" name="department" class="form-control">
153                 <?php
154
237 // SweetAlert2 confirmation dialog
238 Swal.fire({
239   title: "Are you sure you want to reset the password?",
240   icon: "warning",
241   showCancelButton: true,
242   confirmButtonColor: "#DD6B55",
243   confirmButtonText: "Yes",
244   cancelButtonText: "No"
245 }).then(result) => {
246   if (result.isConfirmed) {
247
248     // Show loading state
249     Swal.fire({
250       title: 'Please wait...',
251       text: 'Resetting the password...',
252       icon: 'info',
253       allowOutsideClick: false, // Disable closing by clicking outside
254       allowEscapeKey: false, // Disable closing by pressing ESC
255       didOpen: () => {
256         Swal.showLoading(); // Show loading spinner
257       }
258     });
259
260     const url = '../dbassets/user/reset-password.php';
261
262     // Perform the AJAX POST request
263     $.post(url, { id }, (response) => {
264       // Close the loading spinner
265       Swal.close();
266     });
267   }
268 }
269 
```

*Figure 65. Code Snippet for Admin – users.php*

Figure 65 illustrates the code implementation of the User Management Interface, an all-encompassing administrative panel designed to enable system administrators to oversee user accounts involved in managing student and alumni credential requests. This interface combines front-end and back-end technologies like HTML, PHP, MySQL, jQuery, SweetAlert2, and AJAX to provide a dynamic and responsive user experience. The script begins by starting a session with `session_start()` and securing page access using `if (!isset($_SESSION['id']))`,



redirecting unauthenticated users to the login page. A key component of the functionality is the

```
SQL query $query = "SELECT u.*, d.department FROM users u LEFT JOIN
ref_departments d ON u.department_id = d.id WHERE u.role != 'Client' AND u.id != '$session_id';"; This query ensures that only non-client
users, excluding the currently logged-in user, are fetched and displayed in a table. Each user's
record is shown with editable fields and status labels, where the class badge-<?php if
($rs['status'] == 'Active') echo 'success'; else echo 'danger'; ?>
visually represents the account status.
```

A notable feature is the Action Dropdown Menu, which enables administrators to perform actions such as "Edit User" and "Reset Password." The "Edit User" option uses jQuery to extract the existing data from the selected row and populate it into a modal form, enabling real-time updates without page reload. Similarly, the "Reset Password" option triggers a **SweetAlert2** confirmation prompt, and, upon approval, sends an **AJAX POST** request to `reset-password.php`, simplifying the password recovery process while providing immediate feedback via the **toastr library**.

Finally, the modal form for adding or editing user features validated input fields and dynamically populated `<select>` elements for assigning departments and roles. Upon submission, another **AJAX POST** request is sent to `submit.php`, which either creates a new user or updates an existing one. Success or failure messages are returned and displayed using **SweetAlert** dialogs, ensuring a transparent and responsive user interaction flow.



The screenshot shows the 'Add / Edit Department' modal open over a list of existing departments. The modal fields include 'Department\*' (Elementary), 'Facebook Link' (Enter data here), and 'Status\*' (Active). Below the modal is a table of departments with columns for ID, Department, Facebook Link, Status, and Action (Edit). The table lists six departments: Elementary, JHS, SHS, CCIS, CCJE, and CBM.

ID	Department	Facebook Link	Status	Action
1	Elementary		Active	Edit
2	JHS		Active	Edit
3	SHS	<a href="https://www.facebook.com/SMCCSHSOfficial">https://www.facebook.com/SMCCSHSOfficial</a>	Active	Edit
4	CCIS	<a href="https://www.facebook.com/CCIsOfSMCCNasipit">https://www.facebook.com/CCIsOfSMCCNasipit</a>	Active	Edit
5	CCJE	<a href="https://www.facebook.com/smcccrim">https://www.facebook.com/smcccrim</a>	Active	Edit
6	CBM	<a href="https://www.facebook.com/SMCCCCBM">https://www.facebook.com/SMCCCCBM</a>	Active	Edit

Figure 66. Admin – Add Department

Figure 66 presents the interface that enables the administrator to add all departments within the institution. This functionality is essential for the clearance process, as it allows the system to identify the specific department to which a student belongs. The designated department will then review and approve the student's request as part of the clearance procedure.

The screenshot shows a table of 10 departments. Each row includes the department name, its Facebook link, status (Active or Inactive), and an 'Edit' button. The departments listed are Elementary, JHS, SHS, CCIS, CCJE, CBM, CTHM, CTE, CAS, and Learning Resource Center.

ID	Department	Facebook Link	Status	Action
1	Elementary	<a href="https://www.facebook.com/smccelementarydepartment">https://www.facebook.com/smccelementarydepartment</a>	Active	Edit
2	JHS	<a href="https://www.facebook.com/SMCCJuniorHighSchool">https://www.facebook.com/SMCCJuniorHighSchool</a>	Active	Edit
3	SHS	<a href="https://www.facebook.com/SMCCSHSOfficial">https://www.facebook.com/SMCCSHSOfficial</a>	Active	Edit
4	CCIS	<a href="https://www.facebook.com/CCIsOfSMCCNasipit">https://www.facebook.com/CCIsOfSMCCNasipit</a>	Active	Edit
5	CCJE	<a href="https://www.facebook.com/smcccrim">https://www.facebook.com/smcccrim</a>	Inactive	Edit
6	CBM	<a href="https://www.facebook.com/SMCCCCBM">https://www.facebook.com/SMCCCCBM</a>	Inactive	Edit
7	CTHM	<a href="https://www.facebook.com/profile.php?id=10009316962172">https://www.facebook.com/profile.php?id=10009316962172</a>	Active	Edit
8	CTE	<a href="https://www.facebook.com/smcccollegeofteachereducation">https://www.facebook.com/smcccollegeofteachereducation</a>	Active	Edit
9	CAS	<a href="https://www.facebook.com/SMCCCAS">https://www.facebook.com/SMCCCAS</a>	Active	Edit
10	Learning Resource Center	<a href="https://www.facebook.com/LRCSMCC">https://www.facebook.com/LRCSMCC</a>	Active	Edit

Figure 67. Admin – Department List

Figure 67 displays the Department List interface, which presents all departments added by the administrator. This interface includes an edit button for modifying department information if updates or changes are required.



ID	Department	Facebook Link	Status	Action
1	Elementary	https://www.facebook.com/smcelementarydepartment	Active	<button>Edit</button>
2	JHS		Active	<button>Edit</button>
3	SHS	https://www.facebook.com/SMCCSHSOfficial	Active	<button>Edit</button>
4	CCIS	https://www.facebook.com/CCISOfSMCCNasipit	Active	<button>Edit</button>
5	CCJE	https://www.facebook.com/smccrim	Active	<button>Edit</button>
6	CBM	https://www.facebook.com/SMCCCBM	Active	<button>Edit</button>

Figure 68. Admin – Edit Department

Figure 68 illustrates the Edit Department interface, which is accessed through the edit button in the Department List. This interface allows the administrator to update or modify information related to the department within the institution.

```

59 <h3>Department List</h3>
60 <div class="table-responsive">
61   <table id="zero_config" class="table table-hover table-bordered">
62     <thead>
63       <tr>
64         <th>ID</th>
65         <th>Department</th>
66         <th>Facebook Link</th>
67         <th>Status</th>
68         <th>Action</th>
69       </tr>
70     </thead>
71     <tbody>
72       <?php
73         require_once('../dbassets/connection.php');
74         $query = "SELECT * FROM ref_departments";
75         $result = mysqli_query($connection,$query);
76         while($rs = mysqli_fetch_array($result)) {
77           $id = $rs['id'];
78         ?>
79         <tr id="php echo $id; ?&gt;"&gt;
80           &lt;td&gt;<?php echo $id; ?&gt;&lt;/td&gt;
81           &lt;td data-target="department"&gt;&lt;?php echo $rs['department'];?&gt;&lt;/td&gt;
82           &lt;td data-target="fb_link"&gt;&lt;?php echo $rs['fb_link'];?&gt;&lt;/td&gt;
83           &lt;td data-target="status"&gt;
84             &lt;span class="badge badge-&lt;?php if ($rs['status'] == 'Active') echo 'success'; else echo 'danger';?&gt;&gt;&lt;?php echo
85             &lt;/td&gt;
86             &lt;td&gt;&lt;button type='button' data-role='edit' data-id="<?php echo $id; ?&gt;" class="btn waves-effect waves-light btn-infor
87           &lt;/tr&gt;
88         &lt;?php } ?&gt;
</pre

```



```

100   <!-- ADD MODAL -->
101   <div id="add-modal" class="modal fade" tabindex="-1" role="dialog" aria-labelledby="myModalLabel" data-backdrop="static">
102     <div class="modal-dialog">
103       <div class="modal-content">
104         <div class="modal-header">
105           |   <h4 class="modal-title font-weight-bold" id="myModalLabel">Add / Edit Department</h4>
106         </div>
107         <form id="department-form" class="form-horizontal m-t-30" autocomplete="off">
108           <input type="hidden" name="id">
109           <div class="modal-body">
110             <div class="form-group">
111               |   <label>Department<span class="text-danger">*</span></label>
112               |   <input type="text" name="department" class="form-control" placeholder="Enter data here" required>
113             </div>
114             <div class="form-group">
115               |   <label>Facebook Link</label>
116               |   <input type="text" name="fb_link" class="form-control" placeholder="Enter data here">
117             </div>
118             <div class="form-group">
119               |   <label>Status<span class="text-danger">*</span></label>
120               |   <select class="form-control col-12" name="status" id="status" required>
121                 <option value="Active">Active</option>
122                 <option value="Inactive">Inactive</option>
123               </select>
124             </div>
125           </div>
126           <div class="modal-footer">
127             <button type="button" class="btn btn-secondary waves-effect" data-dismiss="modal">Close</button>
128             <button type="submit" class="btn btn-info">Save changes</button>
129           </div>
130         </form>
131     </div>

```

```

166   $('#department-form').submit(function(e){
167     e.preventDefault();
168
169     var id = $('input[name="id"]').val();
170     var department = $('input[name="department"]').val();
171     var fb_link = $('input[name="fb_link"]').val();
172     var status = $('select[name="status"] :selected').val();
173
174     const url = '../dbassets/department/submit.php';
175     $.post(url, {id, department, fb_link, status}, (response) => {
176       if($.trim(response) == 'success') {
177         Swal.fire({
178           title: 'System Message',
179           text: 'Data saved successfully!',
180           icon: 'success',
181           confirmButtonText: 'Okay'
182         }).then((result) => {
183           if (result.isConfirmed) {
184             location.reload();
185           }
186         });
187       } else {
188         Swal.fire({
189           title: 'System Message',
190           text: response,
191           icon: 'info',
192           confirmButtonText: 'Okay'
193         });
194       }

```

*Figure 69. Code Snippet for Admin – departments.php*

Figure 69 presents the Department Management Interface, which allows administrators to view, add, and edit departmental records within the system. This interface integrates PHP for server-side processing, MySQL for data retrieval, HTML and Bootstrap for structuring and styling, and jQuery with AJAX for dynamic interaction and asynchronous data submission.

The script begins by establishing a session using `session_start()` and restricting unauthorized access through `if (!isset($_SESSION['id']))`, which redirects



---

unauthenticated users to the login page. This ensures that only logged-in administrators can access and manipulate departmental data.

The central component of the interface is a responsive HTML table dynamically populated by fetching records from the `ref_departments` table via a MySQL query `$query = "SELECT * FROM ref_departments"`. Each row of the table displays the department ID, name, Facebook link, status, and an Edit button. Notably, the status field uses conditional classes such as `badge-success` or `badge-danger` to differentiate between active and inactive departments visually.

When clicking the Edit button, a **jQuery** event handler captures the data from the selected row and pre-fills a modal form. The form fields include inputs for the department name, Facebook link, and a dropdown for status. This interaction enables real-time data editing without requiring a page reload.

The modal form submission is handled by an **AJAX POST** request to `submit.php`, as shown in this javascript CopyEdit `$.post(url, {id, department, fb_link, status}, (response) => { ... })`; If the request is successful, a **SweetAlert** dialog provides user feedback and reloads the page to reflect the changes. Otherwise, it displays an appropriate message to the administrator, maintaining a user-friendly experience. In addition, clicking the Add Department button triggers the same modal form in reset mode, allowing the administrator to insert a new department entry. The logic is managed by toggling an edit flag and resetting the form's input fields.



The screenshot shows the 'Add or Edit Course' modal. It has fields for 'Department' (set to 'Elementary'), 'Course' (text input field), and 'Status' (set to 'Active'). Below the modal is a table of course entries:

ID	Department	Course	Status	Action
1	CCIS		Active	Edit
2	CCIS		Active	Edit
3	CCIS	BUS	Active	Edit
4	CCIS	BSIS	Active	Edit
5	CCIS	DIT	Active	Edit
6	Elementary	Grade 1	Active	Edit

Figure 70. Admin – Add Course

Figure 70 presents the Add Course modal, which enables the administrator to add and manage all courses offered within the institution.

The screenshot shows the 'Course List' interface. It displays a table of courses with the following data:

ID	Department	Course	Status	Action
1	CCS	BSIT	Active	Edit
2	COS	BSCS	Active	Edit
3	CCS	BUS	Active	Edit
4	CCS	BSIS	Active	Edit
5	COS	DIT	Active	Edit
6	Elementary	Grade 1	Active	Edit
7	Elementary	Grade 2	Active	Edit
8	Elementary	Grade 3	Active	Edit
9	Elementary	Grade 4	Active	Edit
10	Elementary	Grade 5	Active	Edit

Figure 71. Admin – Course List

Figure 71 displays the Course List interface, where the administrator can view all courses added to the system. These courses play a vital role in identifying the requester's academic program. The interface also includes an edit button, allowing the administrator to update course information when necessary.



ID	Department	Course	Status	Action
1	CCIS	BSIT	Active	<button>Edit</button>
2	CCIS		Active	<button>Edit</button>
3	CCIS	BLIS	Active	<button>Edit</button>
4	CCIS	BSIS	Active	<button>Edit</button>
5	CCIS	DIT	Active	<button>Edit</button>
6	Elementary	Grade 1	Active	<button>Edit</button>

Figure 72. Admin – Edit Course

Figure 72 illustrates the edit course interface, which is accessed through the edit button in the Course List. This interface allows the administrator to update or modify information related to the course within the institution.

```

101 <!-- ADD MODAL -->
102 <div id="add-modal" class="modal fade" tabindex="-1" role="dialog" aria-labelledby="myModalLabel" data-backdrop="static">
103   <div class="modal-dialog">
104     <div class="modal-content">
105       <div class="modal-header">
106         <h4 class="modal-title font-weight-bold" id="myModalLabel">Add or Edit Course</h4>
107       </div>
108       <form id="course-form" class="form-horizontal m-t-30" autocomplete="off">
109         <input type="hidden" name="id">
110         <div class="modal-body">
111           <div class="form-group">
112             <label for="department">Department <span class="text-danger">*</span></label>
113             <select id="department" name="department" class="form-control">
114               <?php
115                 require_once('../dbassets/connection.php');
116                 $query = "SELECT * FROM ref_departments
117                               WHERE `status` = 'Active'
118                               AND department != 'Administrative'
119                               AND department != 'Cashier'
120                               AND department != 'Registrar'
121                               AND department != 'Learning Resource Center' ; ";
122                 $result = mysqli_query($connection, $query);
123                 while($row = mysqli_fetch_array($result)) {
124                   $id = $row['id'];
125                   $department = $row['department'];
126                 ?>
127                   <option value="<?php echo $id?><?php echo $department?>"></option>
128                 <?php } ?>
129             </select>
130           </div>
131           <div class="form-group">
132             <label>Course <span class="text-danger">*</span> </label>
133           </div>

```



```

161 <script>
162     $(document).ready(function(){
163         var edit = false;
164
165         $(document).on('click', 'button[data-role=edit]', function(){
166             edit = true;
167             var id = $(this).data('id');
168             var department = $('#'+id).children('td[data-target=department]').text();
169             var course = $('#'+id).children('td[data-target=course]').text();
170             var status = $.trim($('#'+id).children('td[data-target=status]').text());
171
172             $('select[name="department"] option:contains(' + department + '')').prop('selected', true);
173             $('input[name="course"]').val(course);
174             $('#status').val(status).change();
175             $('input[name="id"]').val(id);
176             $('#add-modal').modal('toggle');
177         });
178
179         $(document).on('click', '#add, #close', function(){
180             edit = false;
181             $('#course-form').trigger("reset");
182             $('#add-modal').modal('toggle');
183         });
184
185         $('#course-form').submit(function(e){
186             e.preventDefault();
187
188             var id = $('input[name="id"]').val();
189             var course = $('input[name="course"]').val();
190             var department_id = $('select[name="department"]').val();
191             var status = $('select[name="status"] :selected').val();
192
193             const url = '../dbassets/course/submit.php';
194             $.post(url, {id, department_id, course, status}, (response) => {
195                 if($.trim(response) == 'success') {
196                     $.post(url, {id, department_id, course, status}, (response) => {
197                         if($.trim(response) == 'success') {
198                             Swal.fire({
199                                 title: 'System Message',
200                                 text: 'Data saved successfully!',
201                                 icon: 'success',
202                                 confirmButtonText: 'Okay'
203                             }).then((result) => {
204                                 if (result.isConfirmed) {
205                                     location.reload();
206                                 }
207                             });
208                         } else {
209                             Swal.fire({
210                                 title: 'System Message',
211                                 text: response,
212                                 icon: 'info',
213                                 confirmButtonText: 'Okay'
214                             });
215                         }
216                     });
217                 });
218             });
219         </script>

```

*Figure 73. Code Snippet for Admin – course.php*

Figure 73 illustrates the interface and functionality for managing courses within the institution's system. This interface allows administrators to add, view, and edit course records, ensuring they are correctly associated with their respective departments and statuses.

At the beginning of the script, a session is initiated using `session_start()` to maintain access control, ensuring only logged-in users can proceed. The system checks if a session ID exists; if not, the user is redirected to the login page `index.php`, securing the page.



---

The core of the user interface displays a table populated from the database `ref_courses` using an SQL query that joins department data `ref_departments` to each course using a **LEFT JOIN**. This enables displaying not only the course name but also its affiliated department (`$rs['department']`), a critical feature in identifying each course's academic classification.

An “Add Course” button triggers a modal form (#add-modal) for adding and editing courses. When a course is edited, jQuery dynamically populates the modal with the selected course’s existing data by targeting specific table cells (`data-target="department"`, `data-target="course"`, and `data-target="status"`). This pre-filled data ensures a smooth and error-free editing process.

Inside the modal, a dropdown menu fetches all *active* departments, excluding administrative ones, through a filtered SQL query (excluding departments like ‘Administrative’, ‘Cashier’, and ‘Registrar’). This is done to limit course associations to academic departments only.

Upon form submission, an **AJAX `$.post()`** request sends course details—including course name, selected department ID, and status—to a backend script (`submit.php`) for database insertion or update. If successful, the system uses SweetAlert (`Swal.fire`) to notify the user of the successful operation and refreshes the page to reflect changes.

The conditional badge styling in the status column (`badge-success` for active, `badge-danger` for inactive) provides quick visual feedback about each course’s availability, improving the overall user experience and administrative clarity.



The screenshot shows the 'Add / Edit Document Type Information' modal. It includes fields for 'Document Type\*' (with a placeholder 'Enter data here'), 'Document Name\*' (placeholder 'Enter data here'), 'Price\*' (checkbox 'With Documentary Stamp?'), 'Processing Days\*' (placeholder 'Enter data here'), and 'Status \*' (dropdown menu showing 'Active'). At the bottom right of the modal are 'Close' and 'Save changes' buttons.

ID	Document Type	Document Name	Price	Processing Days	Status	Action
1	TOR	Transcript of Record - Board Exam Purpose	1050.00	3-5 working days	Yes	<span>Active</span>
2	TOR	Transcript of Record - Employment Purpose	1050.00	3-5 working days	Yes	<span>Active</span>
3	TOR	Transcript of Records - Transfer Credential	1050.00	3-5 working days	Yes	<span>Active</span>
4	Certificate	Certificate - General Weighted Average	200.00	1-2 working days	Yes	<span>Active</span>
5	Certificate	Certificate of Honors	200.00	1-2 working days	Yes	<span>Active</span>
6	Certificate	Certification of Enrollment	200.00	1-2 working days	Yes	<span>Active</span>
7	Certificate	Certificate (English as a medium of instruction)	200.00	1-2 working days	No	<span>Active</span>
8	Certificate	Certificate of Units Earned	100.00	1-2 working days	Yes	<span>Active</span>
9	Certificate	Certification of Graduation	100.00	1-2 working days	Yes	<span>Active</span>
10	Others	School Form 10	300.00	3-5 working days	Yes	<span>Active</span>

Figure 74. Admin – Add Document Type

Figure 74 displays the Add Document Type modal, enabling the administrator to add various documents that students or alumni may request.

The screenshot shows the 'Document Type List' page. It features a table with 10 entries, each representing a different document type. The columns are: ID, Document Type, Document Name, Price, Processing Days, With Doc Stamp?, Status, and Action. The 'Action' column contains 'Edit' buttons for each row. At the bottom of the table, there is a message 'Showing 1 to 10 of 14 entries' and navigation links 'Previous', '1', '2', and 'Next'.

Figure 75. Admin – Document Type List

Figure 75 illustrates the list of document types added by the administrator. Through this interface, the administrator can manage all document types requested by students or alumni.



The screenshot shows the SMCC ORCS software interface. On the left, there is a sidebar with navigation links: SMCC ORCS, TRANSACTION (Dashboard, Students, Users), REFERENCES (Department, Course, Document Type, Payment Account, ISO Form Code). The main area displays a table of document types with columns: ID, Status, Action. A modal window titled "Add / Edit Document Type Information" is open over the table. The modal contains the following fields:

Document Type*	TOR
Document Name*	Transcript of Record - Board Exam Purpose
Price*	350.00
With Documentary Stamp?	<input checked="" type="checkbox"/> 35
Processing Days*	3-5 working days
Status *	Active

At the bottom right of the modal are "Close" and "Save changes" buttons.

Figure 76. Admin - Edit Document Type Information

Figure 76 illustrates the edit document type interface, which is accessed through the edit button in the document type list. This interface allows the administrator to update or modify information related to the course within the institution.





```

214     $('#type-form').submit(function(e){
215         e.preventDefault();
216
217         var id = $('input[name="id"]').val();
218         var document_type = $('input[name="document_type"]').val();
219         var document_name = $('input[name="document_name"]').val();
220         var price = $('input[name="price"]').val();
221         var with_stamp = $('input[name="with_stamp"]').prop('checked');
222         var stamp_price = $('input[name="stamp_price"]').val();
223         var processing_days = $('input[name="processing_days"]').val();
224         var status = $('select[name="status"] :selected').val();
225
226         const url = '../dbassets/document_type/submit.php';
227         $.post(url, {id, document_type, document_name, price, with_stamp, stamp_price, processing_days, status}, (response) => {
228             if($.trim(response) == 'success') {
229                 Swal.fire({
230                     title: 'System Message',
231                     text: 'Data saved successfully!',
232                     icon: 'success',
233                     confirmButtonText: 'Okay'
234                 }).then((result) => {
235                     if (result.isConfirmed) {
236                         location.reload();
237                     }
238                 });
239             } else {
240                 Swal.fire({
241                     title: 'System Message',
242                     text: response,
243                     icon: 'info',
244                     confirmButtonText: 'Okay'
245                 });
246             }
247         });
248     });
249 
```

*Figure 77. Code Snippet for Admin – document\_types.php*

Figure 77 illustrates the code implementation of the Document Type Management page, which allows administrators to view, add, and modify different document types that users may request, such as students or alumni. At the top of the file, a session validation check ensures that only authenticated users can access the page (`if(!isset($_SESSION['id'])) { header("Location: ../index.php"); }`), thereby enforcing security. The interface contains a button labeled "Add Document Type" (`id="add"`), which triggers a Bootstrap modal allowing data input.

Necessary fields inside the modal form (`id="type-form"`) include `document_type`, `document_name`, `price`, `processing_days`, and `status`. A notable feature is the "With Documentary Stamp?" checkbox (`name="with_stamp"`), which, when selected, activates an input field for the stamp's price (`name="stamp_price"`) and sets it to a default value of 35. This conditional logic is implemented through a `jQuery.change()` event listener that toggles the read-only state of the `stamp_price` input.



The existing document types are displayed using a responsive HTML table, where each row contains an Edit button. When clicked, this button retrieves data from the table row using the assigned `data-target` attributes (`data-target="type"`, `data-target="price"`, etc.), populates the modal form, and toggles its visibility (`$('#add-modal').modal('toggle');`). This functionality enables seamless editing of entries. When the form is submitted, it sends a **POST** request to `submit.php` using jQuery's `$.post()` method. This payload includes all form values, including conditionally rendered ones like `with_stamp` and `stamp_price`. If the response is "success", a **SweetAlert2** modal confirms the operation and reloads the page; otherwise, an error alert is shown.

ID	Account Name	Account Number	Picture	Status	Action
1	Saint Michael College of Caraga	500000098291		Active	Edit
2	Saint Michael College of Caraga	040558000114		Active	Edit
3	Saint Michael College of Caraga	09098549424		Active	Edit

Figure 78. Admin – Payment Account List

Figure 78 presents the Payment List interface, which enables administrators to add and manage school accounts related to student payments. Through this interface, administrators are granted the authority to input all necessary school account details. Additionally, the interface provides functionality for editing existing records, allowing modifications when updates or corrections are required.



```

85     $id = mysqli_real_escape_string($connection, $_POST['request_id']);
86     $imageaccept = ["jpg", "png", "jpeg"];
87     $directory = "upload/payment_receipts";
88     $session_id = $_SESSION["id"];
89
90     // Upload the file and get the file path (assuming the file is always present)
91     $image_path = uploadfile('file', 'file', $imageaccept, $directory);
92
93     // Prepare the query to update the record
94     if ($id) {
95         $query = "SELECT email FROM users WHERE `role` = 'Cashier' AND `status` = 'Active'";
96         $stmt1 = $connection->prepare($query);
97         $stmt1->execute();
98         $result = $stmt1->get_result();
99
100        while ($row = $result->fetch_assoc()) {
101            if (sendEmail($row['email'], $subject, $body)) {
102                // echo "Email successfully sent to: " . $row['email'] . "<br>";
103            }
104        }
105
106        // Update the record in the clearance table
107        $update_query = "UPDATE requests SET receipt = ?, `status` = 'For Verification' WHERE id = ?";
108
109        // Prepare the statement
110        $stmt = $connection->prepare($update_query);
111
112        // Bind the parameters to the placeholders in the query
113        $stmt->bind_param("si", $image_path, $id);
114
115        // Execute the query and check if it was successful
116        if ($stmt && $stmt->execute()) {
117            // Log the action

```

*Figure 79. Code Snippet for Admin – payment.php*

Figure 79 illustrates the backend PHP script responsible for uploading and processing a payment receipt related to a specific request. This snippet demonstrates several key functionalities that support secure file handling, database updates, and automated email notifications within a student payment system.

The script begins by sanitizing the incoming request\_id using mysqli\_real\_escape\_string, ensuring protection against **SQL injection**. This step is crucial as it ensures that only valid, safe data enters the system. It also defines a whitelist of acceptable image file extensions—jpg, png, and jpeg—enforced by the uploadFile() function to prevent malicious file uploads. The receipt image is then saved to the upload/payment\_receipts directory, and the file path is stored in the \$image\_path variable.

Next, the script checks if a valid request ID exists. If so, it queries the database for all active users with the role of "Cashier". For each cashier found, an email notification is sent using the



sendEmail() function. This notifies them that a receipt has been uploaded and that further verification is needed. The core update operation is handled using a prepared SQL statement, which updates the receipt field and changes the request status to 'For Verification' in the requests table. The line `stmt->bind_param("si", $image_path, $id);` is critical as it binds the file path and request ID to the SQL query, ensuring both data integrity and security through parameterized inputs.

Upon successful execution, the script logs the action via `log_action()`, capturing the session user ID and the nature of the update for audit purposes and returning a success message. If the execution fails or the request ID is missing, appropriate error messages are echoed back to the client.

ID	Form Name	Form Code No.	Issue Status	Revision No.	Date Effective	Approved	Status	Action
1	Feedback Form	FM-OCM-SMCC-HCF-01	02	03	2023-09-14	President	Active	<a href="#">Edit</a>

Figure 80. Admin – ISO Form Code List

Figure 80 shows the ISO Form Code interface page, where the admin can add and edit ISO form codes for reports or other forms provided by the ISO. Each form entry includes its corresponding form number, issue status, revision number, effective date, and the name of the approving authority.



*Figure 81. Code Snippet for Admin – iso\_form\_code.php*

Figure 81 presents a complete implementation of an ISO Form Code management interface, combining PHP for server-side logic with HTML and JavaScript (primarily jQuery) for frontend interaction. This web page enables administrators to manage reference data associated with ISO form documentation, such as form names, codes, revision details, and approval information.



---

At the beginning of the script, session handling is initiated using `session_start()`, and the user's authentication status is verified via `$_SESSION['id']`. Unauthorized users are redirected to the login page (`../index.php`), ensuring that only authenticated users can access the ISO form code management module. The layout is modular, with headers, navigation bars, and sidebars included through PHP `include()` statements to maintain clean and reusable code.

The page's core content is a dynamic table (#zero\_config) that fetches ISO form data from the `ref_iso_codes` database table using a SQL query (`SELECT * FROM ref_iso_codes`). Each row in the table displays a single ISO form entry and an "Edit" button containing attributes like `data-role='edit'` and `data-id` for reference. When the Edit button is clicked, JavaScript extracts the current row's content and fills it into a modal form for editing.

A modal dialog (#add-modal) is triggered to add new entries and edit existing ones. This modal contains a form (#code-form) with input fields such as `form_name`, `form_code`, `issue_status`, `revision_no`, `date_effective`, and `approved_by`. A hidden input field named `id` determines whether the operation is an insertion or an update. When the form is submitted, JavaScript collects the form data and sends it to the server via a jQuery `$.post()` function targeting `../dbassets/iso_code/submit.php`. The server-side script processes the submission, and upon receiving a 'success' response, the frontend shows a **SweetAlert** confirmation message and reloads the page.



The screenshot shows the SMCC ORCS Dashboard. On the left sidebar, there are sections for CLEARANCE (Clearance List, Clearance Disapproved List), REQUESTS (Payment Verification), and REPORTS (Clearance Report, Credentials Report). The main content area shows a summary of pending requests: 'Clearance Pending' (1) and 'Requests Pending' (0). Below this, a 'Latest Request' table is displayed with one row for 'Ricel Ada Villarin' with status 'Pending'. The footer of the page includes the text 'SMCC © 2025'.

*Figure 82. Cashier – Dashboard*

Figure 82 shows the Dashboard interface, which enables the Cashier to view credential requests, including pending clearances, completed requests, and recent request transactions, along with their statuses.

```

79 <?php
80     require_once('../dbassets/connection.php');
81     $query = "SELECT r.*, u.fname, u.lname FROM requests r
82             JOIN users u ON r.user_id = u.id
83             WHERE r.status = 'Pending'
84             ORDER BY r.submitted_at DESC
85             LIMIT 5;";
86     $result = mysqli_query($connection, $query);
87
88     // Check if there are any rows returned
89     if (mysqli_num_rows($result) > 0) {
90         // Loop through the result and display rows
91         while ($rs = mysqli_fetch_array($result)) {
92             // Check if date_requested exists and calculate if it's late
93             $date_requested = $rs['date_requested'];
94             $current_date = date('Y-m-d'); // Current date in Y-m-d format
95             $date_diff = 0; // Default value for date_diff
96             $late_class = ''; // Default class for row
97
98             // Only calculate if date_requested is not null
99             if (!is_null($date_requested)) {
100                 $date_diff = strtotime($current_date) - strtotime($date_requested);
101
102                 // Check if the date_requested is more than 3 days late
103                 if ($date_diff > (3 * 86400)) { // 3 days * 86400 seconds per day
104                     $late_class = 'table-danger'; // Add 'table-danger' class for red row
105                 }
106             }
107         }
108     ?>

```

*Figure 83. Code Snippet for Cashier – Dashboard\_cashier.php*

Figure 83 illustrates a dashboard component that provides real-time insights into system activities, particularly on clearance and request monitoring. This query utilizes two subqueries

```
SELECT COUNT(id) FROM `clearance` WHERE `status` = 'For Cashier'
```



and SELECT COUNT(id) FROM `requests` WHERE `status` = 'For Verification' These subqueries respectively count the number of pending clearance records that require the Cashier's attention and the number of requests awaiting verification. The results are fetched and stored in the variables \$clearance and \$requests, which are then dynamically echoed inside two card components to display the total counts on the dashboard visually.

The screenshot shows a web-based application interface titled "Clearance List". At the top right, there is a user profile icon for "Noel Baja". Below the title, there is a navigation breadcrumb: "Home > Clearance List". The main content area has a heading "Clearance List" and a table with the following columns: ID, Name, Date Submitted, Is Cleared?, Comment, Status, and Action. The table displays the message "No data available in table". Above the table, there is a "Show: 10 entries" dropdown and a "Search:" input field. Below the table, it says "Showing 0 to 0 of 0 entries" and has "Previous" and "Next" buttons. At the bottom right of the page, it says "SMCC © 2025".

*Figure 84. Cashier/Head of LRC/Dean/Principal/Registrar – Clearance List*

Figure 84 presents the clearance list interface utilized by the Cashier, Head of the Learning Resource Center (LRC), Dean/Principal, and Registrar. Although this is a shared interface among multiple roles, it is designed to ensure role-specific functionality. Requester information is displayed only when it is the corresponding role's turn to take action in the clearance process. Since the system is role-based, the requester's name will appear sequentially according to the designated approval flow. This ensures that users interact only with the requests assigned to their role at the appropriate stage.



```

198     <script>
199     $(document).ready(function(){
200         var edit = false;
201
202         $(document).on('click', 'a[data-role=approve]', function() {
203             var id = $(this).data('id');
204             var email = $(this).data('email');
205             var deanid = $(this).data('deanid');
206             const formData = { id: id, email: email, dean_id: deanid };
207
208             Swal.fire({
209                 title: "Are you sure?",
210                 text: "You are about to approve this transaction.",
211                 icon: "warning",
212                 showCancelButton: true,
213                 confirmButtonColor: "#DD6B55",
214                 confirmButtonText: "Yes",
215             }).then((result) => {
216                 if (result.isConfirmed) {
217                     // Show the loading state
218                     Swal.fire({
219                         title: 'Please wait...',
220                         text: 'Approving transaction...',
221                         icon: 'info',
222                         allowOutsideClick: false, // Prevent clicking outside to close
223                         didOpen: () => {
224                             Swal.showLoading(); // Show the loading spinner
225                         }
226                     });
227
228                     $.post('../dbassets/request/approve.php', formData, (response) => {
229                         console.log(response);
230                         if ($.trim(response) === 'success') {
231                             Swal.fire({
232                                 title: 'System Message',
233                                 text: "The transaction has been approved successfully.",
234                                 icon: 'success',
235                                 confirmButtonText: "Okay"
236                             }).then(() => location.reload());
237                         } else {
238                             Swal.fire('Error', 'There was a problem approving the transaction.', 'error');
239                         }
240                     });
241
242                 }
243             });
244
245             $(document).on('click', 'a[data-role=disapprove]', function() {
246                 var id = $(this).data('id');
247                 var email = $(this).data('email'); // Optional if email is needed
248                 const formData = { id: id, email: email };
249
250                 // Show SweetAlert2 input prompt for feedback
251                 Swal.fire({
252                     title: "Feedback / Remarks",
253                     text: "Please enter reason for disapproval:",
254                     input: 'textarea', // Using a textarea for multi-line input
255                     showCancelButton: true,
256                     inputPlaceholder: "Write something",
257                     preConfirm: (inputValue) => {
258                         // Check if the input is empty and prevent proceeding if it is
259                         if (!inputValue) {
260                             Swal.showValidationMessage("You need to write something!");
261                             return false;
262                         }
263                     }
264                 })
265             });
266
267             $.then((result) => {
268                 if (result.isConfirmed) {
269                     // If the user entered a reason, prepare the form data
270                     formData.input_value = result.value;
271
272                     // Show the loading state
273                     Swal.fire({
274                         title: 'Please wait...',
275                         text: 'Disapproving transaction...',
276                         icon: 'info',
277                         allowOutsideClick: false, // Prevent clicking outside to close
278                         didOpen: () => {
279                             Swal.showLoading(); // Show the loading spinner
280                         }
281                     });
282
283                     $.post('../dbassets/request/disapprove.php', formData, (response) => {
284                         console.log(response);
285                         if ($.trim(response) === 'success') {
286                             Swal.fire({
287                                 title: 'System Message',
288                                 text: "The transaction has been disapproved successfully.",
289                                 icon: 'success',
290                                 confirmButtonText: "Okay"
291                             }).then(() => location.reload()); // Reload the page after success
292                         } else {
293                             Swal.fire('Error', 'There was a problem disapproving the transaction.', 'error');
294                         }
295                     });
296                 }
297             });
298
299         });
300     });
301 
```

Figure 85. Code Snippet for Cashier/Head of LRC/Dean/Principal/Registrar – Clearance List



---

Figure 85 illustrates the server-side and client-side implementation of the Clearance List interface, which serves multiple roles, including Cashier, Librarian, Dean/Principal, and Registrar. The PHP logic dynamically generates the SQL query based on the logged-in user's role, as defined in the `$statusConditions` associative array. For example, if the user has a Cashier role, the condition `c.status = 'For Cashier'` filters the clearance records relevant to the Cashier only. Similarly, the Dean and Principal roles include an additional constraint `c.dean_id = '$sess_id'`, ensuring that only clearances assigned to the specific dean or principal are shown.

The role-based logic ensures users only interact with requests appropriate to their current step in the clearance process. The frontend further enhances role-specific actions with conditionally rendered dropdowns, such as Approve, Disapprove, and View Receipt, based on the clearance's current status. For instance, the receipt view option appears only if the status is For Cashier and a receipt is uploaded (`$rs['receipt']`). The embedded **jQuery** and **SweetAlert2 JavaScript** enhance the user experience by providing confirmation dialogs for approval/disapproval, feedback input on rejection, and modal views for uploaded receipts.

This role-driven workflow not only strengthens access control and accountability but also enforces the sequential approval structure of the clearance process, ensuring each stakeholder fulfills their responsibility at the appropriate stage.

A screenshot of a web-based application interface titled "Disapproved List". The page includes a header with the college's name and logo, a user profile for "Noel Baja", and a navigation breadcrumb "Home > Disapproved List". Below the title is a search bar and a table header with columns: ID, Name, Date Submitted, Is Cleared?, Comment, Status, and Action. A message "No data available in table" is displayed below the table. At the bottom, it shows "Showing 0 to 0 of 0 entries" and navigation links for "Previous" and "Next".

SMCC © 2025

*Figure 86. Cashier/Head of LRC/Dean/Principal/Registrar – Disapproved List*

Figure 86 displays the disapproved list for the Cashier, Head of the LRC, Dean/Principal, and Registrar. Although this is a shared interface, it is designed to provide different functionalities based on user roles. A requester will appear on the disapproved list of a specific role (e.g., Cashier) if there are deficiencies in that particular process. However, suppose there are no deficiencies in the processes under the LRC, Dean/Principal, or Registrar. In that case, the requester will not appear on their respective disapproved lists, only on the list of the role where the issue exists.



```

83 // construct the query dynamically
84 $query = "SELECT r.*, u.fname, u.lname, u.email, c.id AS clearance_id, c.dean_id, c.status AS clearance_status, c.receipt, c.comment
85     FROM `requests` r
86     JOIN clearance c ON r.id = c.request_id
87     JOIN users u ON r.user_id = u.id
88     WHERE (c.status = 'For Uploading' OR c.status = 'Disapproved')
89         AND c.disapproved_by_id = '$sess_id';";
90
91 $result = mysqli_query($connection,$query);
92 while($rs = mysqli_fetch_array($result)) {
93     $id = $rs['id'];
94     $email = $rs['email'];
95     $dean_id = $rs['dean_id'];
96     $clearance_status = $rs['clearance_status'];
97     $statusClass = isset($statusClasses[$clearance_status]) ? $statusClasses[$clearance_status] : 'info';
98 }
99 >?>
100 <tr id=<?php echo $id; ?>>
101 <td><?php echo $id; ?></td>
102 <td data-target="name"><?php echo $rs['fname'].' '.$rs['lname']; ?></td>
103 <td data-target="submitted_at">
104 | <?php echo $rs['submitted_at'] ? date('F j, Y', strtotime($rs['submitted_at'])) : '' ; ?>
105 </td>
106 <td data-target="is_cleared">
107 | <span class="badge badge-<?php echo $rs['is_cleared'] == 'Yes' ? 'badge-primary' : 'badge-warning'; ?>">
108 | | <?php echo $rs['is_cleared']; ?>
109 </span>
110 </td>
111 <td data-target="comment"><?php echo $rs['comment']; ?></td>
112 <td data-target="clearance_status">
113 | <span class="badge badge-<?php echo $statusClass; ?>"><?php echo $rs['clearance_status']; ?></span>

```

```

151 <script>
152     $(document).ready(function(){
153
154         $(document).on('click', '[data-role=approve]', function() {
155             var id = $(this).data('id');
156             var email = $(this).data('email');
157             var deanid = $(this).data('deanid');
158             const formData = { id: id, email: email, dean_id: deanid };
159
160             Swal.fire({
161                 title: "Are you sure?",
162                 text: "You are about to approve this transaction.",
163                 icon: "warning",
164                 showCancelButton: true,
165                 confirmButtonColor: "#006B55",
166                 confirmButtonText: "Yes!",
167             }).then((result) => {
168                 if (result.isConfirmed) {
169                     // Show the loading state
170                     Swal.fire({
171                         title: 'Please wait...',
172                         text: 'Approving transaction...',
173                         icon: 'info',
174                         allowOutsideClick: false, // Prevent clicking outside to close
175                         didOpen: () => {
176                             Swal.showLoading(); // Show the loading spinner
177                         }
178                     });
179
180                     $.post('../dbassets/request/approve.php', formData, (response) => {
181                         console.log(response);
182                         if ($.trim(response) === 'success') {
183                             Swal.fire({
184                             title: 'System Message',
185                             text: "The transaction has been approved successfully."
186                         });
187                     });
188
189                 }
190             });
191         });
192     });
193 
```

Figure 87. Code Snippet for Cashier/Head of LRC/Dean/Principal/Registrar – Disapproved List

Figure 87 presents the implementation of the Disapproved List page in a web-based clearance system. It dynamically displays users whose clearance requests have been disapproved by specific personnel, based on their roles such as Cashier, Head of LRC, Dean/Principal, or Registrar.

The key functionality of role-based display is emphasized in the SQL query section \$query

= "SELECT ... WHERE (c.status = 'For Uploading' OR c.status = 'Disapproved') AND c.disapproved\_by\_id = '\$sess\_id';"; This query retrieves records only if they were disapproved by the currently logged-in staff (\$sess\_id), ensuring



---

that each role sees only the entries they specifically disapproved. This is vital in a role-based system where, for instance, the Cashier should not see disapprovals from the Dean/Principal unless they were the disapproving party.

The table dynamically generates rows with key details such as ID, Name, Date Submitted, Is Cleared, Comment, and Status. Based on the value of \$clearance\_status, status labels use conditional styling with classes like 'badge-success', 'badge-warning', and 'badge-danger'. Another role-sensitive condition appears within this block <?php if (\$role != 'Cashier') { ?>

This condition hides the "Approve" action button from Cashiers, emphasizing that only roles such as Dean or LRC Head can approve specific requests. This reinforces role integrity and prevents unauthorized actions within the system.

On the frontend, a JavaScript function listens for the "Approve" action button (for non-Cashier roles). A **SweetAlert** confirmation is triggered upon clicking, followed by a **POST** request to approve.php with the necessary data. Successful approval refreshes the page and updates the table.

This implementation ensures that each user role only interacts with their respective disapproved requests, maintaining a modular, secure, and role-specific workflow in the clearance approval process.



The screenshot shows the 'Payment List' section of the SMCC ORCS system. The left sidebar includes links for Dashboard, Clearance (Clearance List, Clearance Disapproved List), Requests (Payment Verification), and Reports (Clearance Report, Credentials Report). The main area displays a table with one entry:

ID	Name	Date Requested	Is Cleared?	Date Release	Total	Status	Action
1	Ricel Ada Villarin	May 4, 2025	Yes		1855.00	<span style="color: blue;">For verification</span>	<span style="color: blue;">View Receipt</span>   <span style="color: red;">Approve</span>   <span style="color: red;">Disapprove</span>

Showing 1 to 1 of 1 entries

SMCC © 2025

Figure 88. Cashier – Payment List

Figure 88 shows the Payment Verification interface for the Payment List, which allows the cashier to manage and validate students' credential payments efficiently. This ensures students have paid the required amount by reviewing the uploaded proof of payment. The interface features a six-column table displaying each student's ID number, name, course and year, date requested, status, and actions. In the Action column, the 'View Payment' button lets the cashier verify the payment. Once validated, the cashier can send the official receipt to confirm and approve the payment of the student's credentials.



```
80 require_once('../dbassets/connection.php');
81 $query = "SELECT
82     r.*,
83     u.fname,
84     u.lname,
85     u.email,
86     COALESCE(SUM(CASE WHEN r.status = 'For Verification' THEN rd.total ELSE 0 END), 0) + COALESCE(r.fee, 0) AS total_price
87     FROM
88     requests r
89     LEFT JOIN
90     requested_documents rd ON r.id = rd.request_id
91     JOIN
92     users u ON r.user_id = u.id
93     WHERE r.status = 'For Verification'
94     GROUP BY
95     r.id; ";
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
789
789
790
791
792
793
794
795
796
797
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
889
889
890
891
892
893
894
895
896
897
898
899
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
918
919
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
949
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
979
979
980
981
982
983
984
985
986
987
987
988
989
989
990
991
992
993
994
995
996
997
998
999
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1089
1090
1091
1092
1093
1094
1095
1096
1097
1097
1098
1099
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1197
1198
1199
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1297
1298
1299
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1397
1398
1399
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1497
1498
1499
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1597
1598
1599
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1697
1698
1699
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1797
1798
1799
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1897
1898
1899
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2097
2098
2099
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2197
2198
2199
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2288
2289
2290
2291
2292
2293
2294
2295
2296
2296
2297
2298
2299
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2388
2389
2390
2391
2392
2393
2394
2395
2396
2396
2397
2398
2399
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2488
2489
2490
2491
2492
2493
2494
2495
2496
2496
2497
2498
2499
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2509
2510
2511
2512
2513
2
```

*Figure 89. Code Snippet for Cashier – Request\_verification.php*

Figure 89 shows a PHP-embedded HTML page that serves as the *Request Approval* module of a payment verification system. The script begins with `session_start()` and includes a conditional redirection using `if (!isset($_SESSION['id']))` to ensure that only authenticated users can access the page. This initial PHP block is a security measure, preventing unauthorized access to the approval interface.

The main content is enclosed within a `<div class="page-wrapper">`, which contains a breadcrumb navigation for user orientation and a responsive HTML table identified by `<table id="zero_config">`. This table dynamically displays payment-related records retrieved through a MySQL query stored in `$query = "SELECT ... FROM requests r LEFT JOIN requested_documents rd ..."`, a statement that joins the requests,



---

requested\_documents, and users tables. This setup consolidates data such as the request ID, user name, date requested, total price, and status. The COALESCE() function is used within the query to handle NULL values in the price fields, ensuring accurate total price calculations even when some values are missing.

The table rows are dynamically generated using a while loop that calls mysqli\_fetch\_array(\$result). Each row features action buttons grouped with Bootstrap's dropdown components. These buttons are conditionally displayed based on the request's status and may include options for viewing the receipt (data-role='receipt'), approving the request (data-role='upload'), or disapproving it (data-role='disapprove').

When triggered, the *View Receipt Modal* displays an image using the #modalImage element, whose src attribute is dynamically assigned via **jQuery**. The Upload Modal allows users to upload an official receipt through an HTML form, which is submitted asynchronously using jQuery's \$.ajax({ url: '../dbassets/request/upload\_official\_receipt.php', ... }). This non-blocking process ensures a smooth and responsive user experience.

**SweetAlert2** is used to gather a reason from the user through a text area input for the disapproval process. A key segment of the logic is the use of Swal.fire({ title: "Feedback / Remarks", input: 'textarea', ... }).then((result) => { if (result.isConfirmed) { ... \$.post('../dbassets/request/deny\_payment.php', formData, ...); } })



} ); . This ensures that user input is validated before sending a POST request to the server, enabling the system to handle disapprovals securely and responsibly.

The screenshot shows the 'Clearance Report' page of the SMCC ORCS system. The left sidebar has navigation links for Dashboard, Clearance (Clearance List, Clearance Disapproved List), Requests (Payment Verification), Reports (Clearance Report, Credentials Report), and a Print button. The main area shows a search form with 'From' (01/05/2025) and 'To' (05/05/2025) dates, a 'Department' dropdown set to 'All Departments', and a 'Print Report' button. Below is a table with one entry:

#	Name	Department	Course	Receipt	Date Approved	Status
1	Richel Ada Villarin	CCIS	BSIT	<a href="#">View Receipt</a>	May 5, 2025	Approved

At the bottom, it says 'Showing 1 to 1 of 1 entries' and has 'Previous' and 'Next' buttons. The footer says 'SMCC © 2025'.

**Saint Michael College of Caraga**  
 Brgy. 4, Nasipit, Agripina, Misamis Oriental, Philippines  
 District 6, Brgy. Triangulo, Nasipit, Agripina, Misamis Oriental, Philippines  
 Tel. Nos. +63 86 343-5201, 283-3113 | Fax: +63 86 898-0892  
[www.smccnasiol.edu.ph](http://www.smccnasiol.edu.ph)  
**CLEARANCE REPORT**  
 Period: May 01, 2025 - May 05, 2025  
 Cashier Office

#	Name	Department	Course	Date Approved	Status
1	Richel Ada Villarin	CCIS	BSIT	May 05, 2025	Approved

Prepared by:  
 \_\_\_\_\_  
 Noel Baja

*Figure 90. Cashier – Clearance Report*

Figure 90 shows the Completed Clearance Report interface designed for the cashier. The interface allows the cashier to monitor and manage clearance requests that have been reviewed and approved. Users can filter the report based on a specific date range and department to locate particular records easily. Additionally, it features a “View Receipt” button, enabling the cashier to view the uploaded payment receipt submitted by the requester, especially when an outstanding balance needs verification.



```
235 <script>
236   $(document).ready(function(){
237
238     $('#fromDate, #toDate, #department').change(function() {
239       var fromDate = $('#fromDate').val();
240       var toDate = $('#toDate').val();
241       var department = $('#department').val();
242
243       if (new Date(fromDate) > new Date(toDate)) {
244         Swal.fire({
245           title: 'Error',
246           text: 'The "From" date must be earlier than the "To" date.',
247           icon: 'error',
248           confirmButtonText: 'OK'
249         });
250
251         // Optionally, clear the "From" date or reset it
252         $('#fromDate').val('');
253       }
254       else {
255         getData(fromDate, toDate, department);
256       }
257     });
258
259     function getData(fromDate, toDate, department) {
260       const url = './dbassets/report/clearance_cashier.php';
261
262       var table = $('.report-table').DataTable();
263       table.clear().draw();
264       $.get(url, { fromDate, toDate, department }, (response) => {
265         console.log(response);
266         const rows = JSON.parse(response);
267         if (rows.message) {
268           alert('No data available');
269         }
270       });
271     }
272   });
273 
```

```
300 $('#printReport').click(function(event) {  
301     event.preventDefault(); // Prevent form submission  
302  
303     var fromDate = $('#fromDate').val();  
304     var toDate = $('#toDate').val();  
305     var department = $('#department').val();  
306  
307     if (!fromDate || !toDate) {  
308         Swal.fire('System Message', "Please enter both From Date and To Date.", 'info');  
309         return;  
310     }  
311  
312     if (new Date(fromDate) > new Date(toDate)) {  
313         Swal.fire('System Message', "From Date cannot be after To Date.", 'info');  
314         return;  
315     }  
316  
317     $('#periodText').text(moment(fromDate).format('MMMM DD, YYYY') + ' - ' + moment(toDate).format('MMMM DD, YYYY'));  
318     fetchReportData(fromDate, toDate, department);  
319  
320     setTimeout(() => {  
321         // $('#report-modal').modal('toggle');  
322         let css = `  
323             @media print {  
324                 @page {  
325                     size: A4 landscape; /* Ensure page size is A4 in landscape orientation */  
326                 }  
327             }  
328         `;
```

*Figure 91. Code Snippet for Cashier – report\_clearance\_cashier.php*

The code in Figure 91 is a complete PHP and HTML web page for generating a Clearance Report used in an institutional setting, such as Saint Michael College of Caraga. The page displays breadcrumbs to help users track their location in the app and contains a form for filtering reports by From Date, To Date, and Department. These filters are essential as they determine the data scope for the clearance report.



---

A key part of the functionality is the dropdown menu for department selection, which dynamically loads data from the database (`ref_departments`) while excluding specific departments like *Administrative*, *Cashier*, *Registrar*, and *Learning Resource Center*. This is done through a SQL query inside a PHP block and is critical for filtering relevant clearance data.

Below the filters is a `DataTable (#zero_config)` that is initially empty and later populated dynamically using **jQuery** and **AJAX**. When the date range or department is changed, the `getData()` function is triggered. This function sends a **GET** request to `clearance_cashier.php` to fetch filtered data and populates the table with the resulting **JSON**. Each row contains the student's name, department, course, approval date, and a conditional button to view receipts using a modal (`#view-modal`).

Another necessary functionality is the Print Report button. When clicked, it validates date fields and calls the `fetchReportData()` function, which builds a printable report table using the selected filters. The report is styled for landscape A4 paper using inline CSS within a dynamically opened browser window. This window loads the report's content and a linked CSS stylesheet for formatting, then automatically prints it.

Lastly, the modals serve interactive purposes: `#report-modal` displays the formatted clearance report with institutional branding and preparer's name (`<?php echo $_SESSION['name']; ?>`), while `#view-modal` shows an image of the receipt. The script also handles validation using **SweetAlert** for user-friendly alerts when date inputs are incorrect.



A screenshot of the SMCC ORCS (Online Registration and Clearance System) dashboard. The left sidebar has a dark blue background with white text and icons. It includes sections for 'CLEARANCE' (with 'Clearance List' and 'Clearance Disapproved List') and 'REPORTS' (with 'Clearance Report'). The main content area has a light gray header with 'Home &gt; Dashboard'. Below this is a white card with a small icon and the text 'Clearance Pending' followed by the number '1'. In the top right corner, there is a user profile for 'Renel Maneda' with a dropdown arrow. At the bottom right of the main area, it says 'SMCC © 2025'.

*Figure 92. Head of LRC/ Dean/Principal – Dashboard*

Figure 92 illustrates the dashboard for the Head of LRC, Dean, or Principal, which displays pending clearance requests and the number of students seeking clearance. This dashboard interface is consistent across all users in these roles and specifically highlights pending requests to notify them that students are awaiting clearance approval.



```
1 <?php
2     require_once ('../dbassets/connection.php');
3     $session_id = $_SESSION['id'];
4     $role = $_SESSION['role'];
5
6     if ($role == 'Librarian') {
7         $query = mysqli_query($connection, "SELECT
8             `clearance`.*,
9             (SELECT COUNT(id) FROM `clearance` WHERE `status` = 'For Resource') AS `clearance`;
10        OR die(mysqli_error($connection));
11    } elseif ($role == 'Dean') {
12        $query = mysqli_query($connection, "SELECT
13            `clearance`.*,
14            (SELECT COUNT(id) FROM `clearance` WHERE `status` = 'For Dean' AND `dean_id` = '$session_id')
15        OR die(mysqli_error($connection));
16    } elseif ($role == 'Principal') {
17        $query = mysqli_query($connection, "SELECT
18            `clearance`.*,
19            (SELECT COUNT(id) FROM `clearance` WHERE `status` = 'For Principal' AND `dean_id` = '$ses
20        OR die(mysqli_error($connection));
21
22        $row = mysqli_fetch_array($query);
23        $clearance = $row['clearance'];
24    }
25
26    <div class="card-group">
27        <!-- column -->
28        <div class="card">
29            <div class="card-body">
30                <div class="row">
31                    <div class="col-md-12">
32                        <div class="d-flex no-block align-items-center">
33                            <div>
34                                <i class="mdi mdi-file-check font-20 text-muted"></i>
35                                <p class="font-16 m-b-5">Clearance Pending</p>
36                            </div>
37                            <div class="ml-auto">
38                                <a href="#" class="font-light text-right">?php echo $clearance; ?</a>
39                            </div>
40                        </div>
41                    </div>
42                </div>
43            </div>
44        </div>
45    </div>
46
```

Figure 93. Code Snippet for Head of LRC / Dean/Principal – dashboard\_others.php

Figure 93 illustrates a PHP code snippet that dynamically displays the number of pending clearance requests based on the user's role within an academic system. The code begins by including a database connection file (require\_once ('.. /dbassets/connection.php')) to establish access to the MySQL database. Then, it retrieves the currently logged-in user's session ID and role using \$\_SESSION['id'] and \$\_SESSION['role'], which serve as the basis for conditional logic.

Depending on the user's role—Librarian, Dean, or Principal—the code executes a specific SQL query to count the number of clearance records with a matching status and role-specific criteria. For instance, if the user is a Librarian, the subquery (SELECT COUNT(id) FROM clearance WHERE status = 'For Resource'), Line 7, counts all clearances awaiting resource approval. For a Dean, the query becomes more restrictive, filtering not only by status = 'For Dean' but also ensuring the dean\_id matches the current user's session ID Line 10,



thus personalizing the data. Similarly, for a Principal, the condition changes to status = 'For Principal' with the same dean\_id filter on Line 14.

After executing the query with `mysqli_query()`, the result is fetched using `mysqli_fetch_array($query)` and stored in the \$clearance variable on Line 18. This value is rendered in the HTML structure inside a styled card component. Within the `<h1>` tag (Figure 1, Line 27), `<?php echo $clearance; ?>` dynamically inserts the clearance count, enabling real-time display of pending requests specific to the user's role. The card is visually enhanced with a Material Design icon (`mdi-file-check`) and a descriptive label ("Clearance Pending"), providing a straightforward and user-friendly interface.

#	Name	Department	Course	Date Approved	Status
1	Ricel Ada Villarin	CCIS	BSIT	May 5, 2025	Approved

*Figure 94. Head of LRC/ Dean/Principal – Clearance Report*

Figure 94 presents the clearance report interface designed for the Head of the Learning Resource Center (LRC) and the Dean or Principal. Although this interface is shared, it remains role-based, meaning the clearance report will only be displayed to the user once the assigned personnel or staff have completed the corresponding process or responsibility.



```

238     function getData(fromDate, toDate, department) {
239         const url = '../dbassets/report/clearance_librarian.php';
240
241         var table = $('.report-table').DataTable();
242         table.clear().draw();
243         $.get(url, { fromDate, toDate, department }, (response) => {
244             console.log(response);
245             const rows = JSON.parse(response);
246             if (rows.message) {
247                 alert('No data available');
248             }
249             else {
250                 rows.forEach((row, index) => {
251                     table.row.add($('
252                         <tr>
253                             <td>' + (index + 1) + '</td>
254                             <td>' + row.fname + ' ' + row.lname + '</td>
255                             <td>' + row.department + '</td>
256                             <td>' + (row.course || '-') + '</td>
257                             <td>' + moment(row.resource_date).format('MMMM D, YYYY') + '</td>
258                             <td>' + row.status + '</td>
259                         </tr>')).draw();
260                 });
261             }
262         });
263
264         $(document).on('click', 'a[data-role=receipt]', function(){
265             var imagePath = $(this).data('image');
266             $('#modalImage').attr('src', '..assets' + imagePath);
267             $('#view-modal').modal('toggle');
268         });
269
270         $('#printReport').click(function(event) {
271             event.preventDefault(); // Prevent form submission
272
273             var fromDate = $('#fromDate').val();
274             var toDate = $('#toDate').val();
275             var department = $('#department').val();
276
277             if (!fromDate || !toDate) {
278                 Swal.fire('System Message', "Please enter both From Date and To Date.", 'info');
279                 return;
280             }
281
282             if (new Date(fromDate) > new Date(toDate)) {
283                 Swal.fire('System Message', "From Date cannot be after To Date.", 'info');
284                 return;
285             }
286
287             $('#periodText').text(moment(fromDate).format('MMMM DD, YYYY') + ' - ' + moment(toDate).format('MMMM DD, YYYY'));
288             fetchReportData(fromDate, toDate, department);
289
290             setTimeout(() => {
291                 // $('#report-modal').modal('toggle');
292                 let css =
293                     `@media print {
294                         @page {
295                             size: A4 landscape; /* Ensure page size is A4 in landscape orientation */
296                         }
297                     }`;
298
299             });
300

```

*Figure 95. Code Snippet for Head of LRC – report\_clearance\_librarian.php*

Figure 95 illustrates a comprehensive PHP-based web application page designed to generate a "Clearance Report" for the librarian's office of Saint Michael College of Caraga. The PHP script begins by starting a session using `session_start()` and checking if a user session exists via `if (!isset($_SESSION['id']))`, redirecting unauthenticated users to the login page (`index.php`)—ensuring secure access. Within the main content, the user interface allows report filtering using HTML `<input type="date">` fields for the “From” and “To” date



---

ranges, and a <select> dropdown for choosing a department. This dropdown is dynamically populated by querying the database with `SELECT * FROM ref_departments WHERE status = 'Active'...`, which excludes specific departments (e.g., 'Administrative', 'Cashier') to refine the report scope.

Client-side interactivity is implemented using jQuery. When any filter input changes (`($('#fromDate, #toDate, #department').change())`), the script validates the date logic (if `(new Date(fromDate) > new Date(toDate))`) and either alerts the user or proceeds to fetch data via `getData(fromDate, toDate, department)`. The `getData` function sends a GET request to `../dbassets/report/clearance_librarian.php`, receives JSON data, and dynamically populates a `DataTable` (`($('.report-table').DataTable())`) with rows rendered inside a `<tbody>` element.

The "Print Report" functionality is handled by the `#printReport` click event, which first validates input dates and sets the report period text with `($('#periodText').text(...))`. It then constructs a printable version of the report by opening a new browser window and writing the content of the `#report-form` div. A key styling enhancement is the addition of a media print rule: `@media print { @page { size: A4 landscape; } }` to ensure proper formatting during printing. Lastly, `fetchReportData(fromDate, toDate, department)` prepares the table content inside the modal by resetting the header and body, and appending new `<tr>` and `<td>` elements based on the fetched data. A fallback message, "No data available!" is inserted if no records are found. Overall, this snippet integrates authentication, data filtering, dynamic UI rendering, and printable reporting in a cohesive and modular way.



```
214
215     <script>
216         $(document).ready(function(){
217
218             $('#fromDate', '#toDate', '#department').change(function() {
219                 var fromDate = $('#fromDate').val();
220                 var toDate = $('#toDate').val();
221                 var department = $('#department').val();
222
223                 if (new Date(fromDate) > new Date(toDate)) {
224                     Swal.fire({
225                         title: "Error",
226                         text: 'The "From" date must be earlier than the "To" date.',
227                         icon: "error",
228                         confirmButtonText: 'OK'
229                     });
230
231                     // Optionally, clear the "From" date or reset it
232                     $('#fromDate').val('');
233                 } else {
234                     getData(fromDate, toDate, department);
235                 }
236             });
237
238             function getData(fromDate, toDate, department) {
239                 const url = '../dbassets/report/clearance_dean.php';
240
241                 var table = $('.report-table').DataTable();
242                 table.clear().draw();
243                 $.get(url, { fromDate, toDate, department }, (response) => {
244                     console.log(response);
245                     const rows = JSON.parse(response);
246                     if (rows.message) {
247                         alert('No data available');
248                     }
249                 });
250             }
251
252             $.ajaxSetup({async: false});
253             $.get(url, { fromDate, toDate, department }, (response) => {
254                 console.log(response);
255                 const rows = JSON.parse(response);
256                 if (rows.length === 0) {
257                     template += `<tr>`;
258                     template += `<td colspan=6>No data available!</td>`;
259                     template += `</tr>`;
260                 }
261                 else {
262                     rows.forEach((row, index) => {
263                         const { fname, lname, department, course, resource_date, status } = row;
264
265                         template += `<tr>`;
266                         template += `<td>${index + 1}</td>`;
267                         template += `<td>${fname} ${lname}</td>`;
268                         template += `<td>${department}</td>`;
269                         template += `<td>${course} || '-'</td>`;
270                         template += `<td>${moment(resource_date).format('MMMM DD, YYYY')}</td>`;
271                         template += `<td>${status}</td>`;
272                         template += `</tr>`;
273                     });
274                 }
275             });
276         });
277     </script>
```

*Figure 96. Code Snippet for Dean/Principal – report\_clearance\_dean.php*

Figure 96 demonstrates a dynamic web page built with PHP, HTML, and JavaScript that serves as a *Clearance Report* interface for authenticated users. The script begins with `session_start()` to initialize session handling, followed by a conditional redirect using `if(!isset($_SESSION['id']))` to ensure only logged-in users can access the page. This is a critical security step to protect the report data.

The user interface contains date input fields and a dropdown for selecting a department. The department dropdown is dynamically populated from a MySQL database using a query inside



a PHP while loop, which excludes non-academic departments such as 'Administrative', 'Cashier', and others using the condition WHERE status = 'Active' AND department != ... When users change the date or department selections (#fromDate, #toDate, #department), the getData() function is triggered via **jQuery**. This function uses an **HTTP GET** request to clearance\_dean.php, sending the selected filters and populating a DataTable dynamically with the result. Each row in the table displays details like name, department, course, approval date, and status.

The Print Report button, defined as <button id="printReport">, validates input and displays the filtered report inside a modal. This modal (#report-modal) is styled to resemble an official document, including a logo, institutional address, and preparer's name pulled from \$\_SESSION['name']. The fetchReportData() function loads this data, while a setTimeout() opens a print-friendly tab containing styled HTML generated with JavaScript.

JavaScript libraries such as **jQuery**, **DataTables**, **Moment.js**, and **SweetAlert** are used for AJAX handling, table management, date formatting, and user-friendly alerts, respectively. For instance, moment(row.resource\_date).format('MMMM D, YYYY') ensures readable date output in the report.

This integrated code effectively manages back-end and front-end operations for clearance reporting, offering print-ready outputs, dynamic filtering, and secure access.

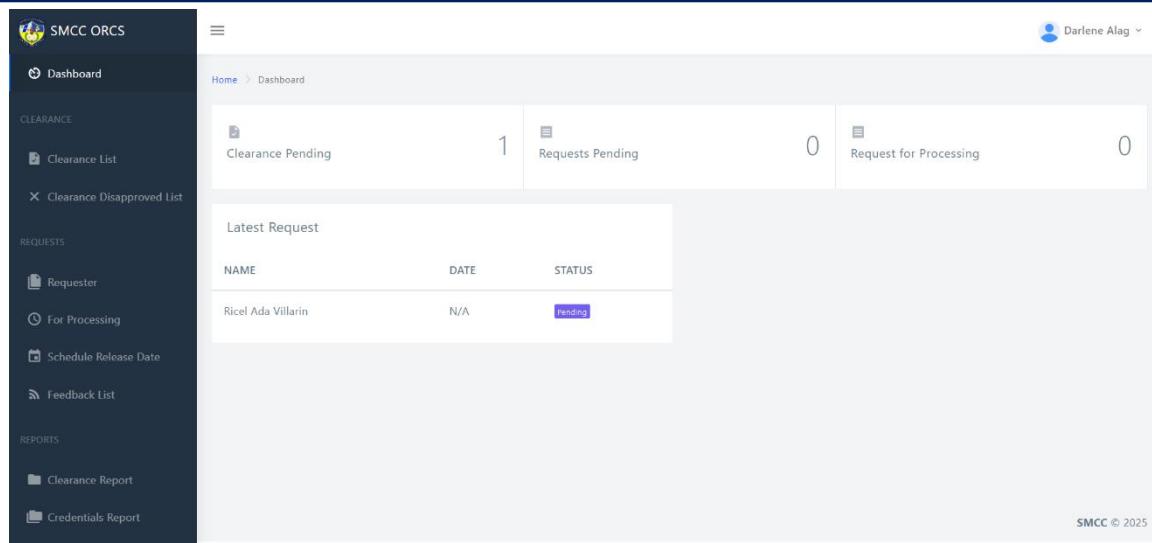


Figure 97. Registrar – Dashboard

Figure 97 shows the Registrar Dashboard interface, providing an overview of credential requests, including pending clearances, submitted requests, and recent transaction activity.



```

1 <?php
2     require_once ('../dbassets/connection.php');
3     $session_id = $_SESSION['id'];
4     $role = $_SESSION['role'];
5
6     if ($role == 'Cashier') {
7         $query = mysqli_query($connection, "SELECT
8             (SELECT COUNT(id) FROM `clearance` WHERE `status` = 'For Cashier') AS `clearance`,
9             (SELECT COUNT(id) FROM `requests` WHERE `status` = 'For Verification') AS `requests`;");
10        or die(mysqli_error($connection));
11    } elseif ($role == 'Registrar') {
12        $query = mysqli_query($connection, "SELECT
13            (SELECT COUNT(id) FROM `clearance` WHERE `status` = 'For Registrar') AS `clearance`,
14            (SELECT COUNT(id) FROM `requests` WHERE `status` = 'Pending' AND `is_cleared` = 'Yes') AS
15            (SELECT COUNT(id) FROM `requests` WHERE `status` = 'For Processing') AS `processing`;
16        or die(mysqli_error($connection));
17    }
18
19    $row = mysqli_fetch_array($query);
20    $clearance = $row['clearance'];
21    $requests = $row['requests'];
22    $processing = $row['processing'];
23 ?>

```

```

116 // check if there are any rows returned
117 if (mysqli_num_rows($result) > 0) {
118     // Loop through the result and display rows
119     while ($rs = mysqli_fetch_array($result)) {
120         // Check if date_requested exists and calculate if it's late
121         $date_requested = $rs['date_requested'];
122         $current_date = date('Y-m-d'); // Current date in Y-m-d format
123         $date_diff = 0; // Default value for date_diff
124         $late_class = ''; // Default class for row
125
126         // Only calculate if date_requested is not null
127         if (!is_null($date_requested)) {
128             $date_diff = strtotime($current_date) - strtotime($date_requested);
129
130             // Check if the date_requested is more than 3 days late
131             if ($date_diff > (3 * 86400)) { // 3 days * 86400 seconds per day
132                 $late_class = 'table-danger'; // Add 'table-danger' class for red row
133             }
134         }
135     }
136 ?>
137     <tr class="<?php echo $late_class; ?>">
138         <td><?php echo $rs['fname']. ' ' . $rs['lname']; ?></td>
139         <td><?php echo !is_null($date_requested) ? date('F j, Y', strtotime($rs['date_requested'])) : 'N/A'; ?></td>
140         <td data-target="status">
141             <span class="badge badge-<?php echo ($rs['status'] == 'Active') ? 'success' : 'primary';?>">
142                 <?php echo $rs['status']; ?>
143             </span>
144         </td>
145     </tr>

```

Figure 98. Code Snippet for Registrar – dashboard\_registar.php

Figure 98 presents a PHP-driven dashboard that displays real-time summaries of clearance and request statuses based on user roles (e.g., *Cashier* or *Registrar*). The script starts by accessing the session variables using `$_SESSION['id']` and `$_SESSION['role']`, which determine the authenticated user's identity and access level. Depending on the role, different SQL queries are executed using `mysqli_query()` to fetch count values from the clearance and requests tables. For instance, Cashiers retrieve the number of clearances with the status 'For



---

*Cashier*', while Registrars obtain data for '*For Registrar*', '*Pending*' (with clearance), and '*For Processing*' statuses.

These count values—\$clearance, \$requests, and \$processing—are stored after fetching results with `mysqli_fetch_array($query)`. They are then embedded dynamically into visually styled cards using `<?php echo $clearance; ?>`, showcasing real-time numerical summaries in the dashboard (e.g., “Clearance Pending” and “Requests Pending”).

The following section introduces a data table under the “Latest Request” card. This is constructed using a query that joins the requests and users tables via `JOIN users u ON r.user_id = u.id`, and limits the output to the five most recent pending requests using `LIMIT 5`. Each row of the table displays the requester's name, the date of request, and the status. A notable functionality is the highlighting of delayed requests. The script calculates the time difference between the current and the previous date and flags rows with a delay of over three days by assigning a Bootstrap class `table-danger`, creating a red background. This visual alert draws attention to overdue items, enhancing user awareness.

To ensure a user-friendly status display, the status column uses dynamic badge classes: `badge-success` for “Active” statuses and `badge-primary` otherwise. This makes it easy to identify the progress of each request at a glance.



ID	Name	Department	Course	Date Requested	Graduate?	Year Attended / Graduated	Status	Action
1	Richel Ada Villarin	CCIS	BSIT	May 4, 2025	Yes	2025	Pending	<button>Action</button>

*Figure 99. Registrar -List of Requested Documents*

Figure 99 shows the list of requested documents, where users can view all requesters along with the details of their requested documents.

Document #	Type	Document Name	Number of Copies	Doc Stamp	Price	Total	
1	Certificate	Certificate - General Weighted Average	1	35.00	200.00	235.00	
2	Certificate	Certificate of Honors	1	0.00	200.00	200.00	
3	Certificate	Certification of Graduation	1	35.00	100.00	135.00	
4	TOR	Transcript of Record - Employment Purpose	1	35.00	1050.00	1085.00	
						Shipping Fee:	200.00
						Total:	1855.00

*Figure 100. Registrar – Request Details*

Figure 100 shows the request details, where the user or registrar staff can set or modify the price of a specific document requested by the requester.



```

187 <script>
188 $(document).ready(function(){
189
190     $(document).on('click', '[data-role=denied]', function() {
191         var requestId = $(this).data('id');
192         var email = $(this).data('email');
193         const formData = { request_id: requestId, email: email };
194
195         // Show SweetAlert2 input prompt for feedback
196         Swal.fire({
197             title: "Deny Request",
198             text: "Please enter reason for denial:",
199             input: 'textarea', // Using a textarea for multi-line input
200             showCancelButton: true,
201             inputPlaceholder: "Write something",
202             preConfirm: (inputValue) => {
203                 // Check if the input is empty and prevent proceeding if it is
204                 if (!inputValue) {
205                     Swal.showValidationMessage("You need to write something!");
206                     return false;
207                 }
208                 return inputValue;
209             }
210         }).then(result) => {
211             if (result.isConfirmed) {
212                 // If the user entered a reason, prepare the form data
213                 formData.input_value = result.value;
214
215                 // Show the loading state
216                 Swal.fire({
217                     title: 'Please wait...',
218                     text: 'Denying transaction...',
219                     icon: 'info',
220                     allowOutsideClick: false, // Prevent clicking outside to close
221                     didOpen: () => {
222
223                         // Send the disapproval request via $.post()
224                         $.post('../dbassets/request/denied.php', formData, (response) => {
225                             console.log(response);
226                             if ($.trim(response) === 'success') {
227                                 Swal.fire({
228                                     title: 'System Message',
229                                     text: "The transaction has been disapproved successfully.",
230                                     icon: 'success',
231                                     confirmButtonText: "Okay"
232                                 }).then(() => location.reload()); // Reload the page after success
233                             } else {
234                                 Swal.fire('Error', 'There was a problem disapproving the transaction.', 'error');
235                             }
236                         });
237                     });
238                 }
239             }
240         });
241     });
242 });
243
244 $('#saveButton').on('click', async function(e) {
245     e.preventDefault();
246
247     var requestId = $('input[name="request_id"]').val();
248     var email = $('input[name="email"]').val();
249     var shippingFee = parseFloat($('.shipping-fee-input').val());
250
251     // Create an array to hold all the data we want to save
252     var dataToSend = [];
253
254     $('#detailsData').on('click', async function(e) {
255         e.preventDefault();
256
257         var requestId = $('input[name="request_id"]').val();
258         var email = $('input[name="email"]').val();
259         var shippingFee = parseFloat($('.shipping-fee-input').val());
260
261         // Create an array to hold all the data we want to save
262         var dataToSend = [];
263
264         // Loop through each row in the table
265         $('#detailsData tr[data-row-index]').each(function() {
266             var rowIndex = $(this).data('id');
267             var copies = parseFloat($(this).find('.copies-input').val());
268             var stamp = parseFloat($(this).find('.stamp-input').val());
269             var price = parseFloat($(this).find('.price-input').val());
270             var total = parseFloat($(this).find('.total').text());
271
272             // Check if copies, price, and total are valid numbers
273             if (!isNaN(copies) && !isNaN(price) && !isNaN(total)) {
274                 // Add the row data to the array, including the total
275                 dataToSend.push({
276                     index: rowIndex, // Add any identifier if needed (e.g., document ID)
277                     copies: copies,
278                     stamp: stamp,
279                     price: price,
280                     total: total // Include the total for this row
281                 });
282             }
283         });
284     });
285 });

```



```
276 const result = await Swal.fire({
277   title: "Are you sure?",
278   text: "You are about to save this transaction.",
279   icon: "warning",
280   showCancelButton: true,
281   confirmButtonColor: "#DD6B55",
282   confirmButtonText: "Yes!",
283 });
284
285 if (result.isConfirmed) {
286   // Show the loading state and ensure it's shown before continuing
287   Swal.fire({
288     title: 'Please wait...',
289     text: 'Saving transaction...',
290     icon: 'info',
291     allowOutsideClick: false, // Prevent clicking outside to close
292     didOpen: () => {
293       Swal.showLoading(); // Show the loading spinner
294     }
295   });
296
297   // Prepare the data to send
298   const formData = {
299     data: JSON.stringify(dataToSend),
300     shipping_fee: shippingFee,
301     request_id: requestId,
302     email: email
303   };
304
305   // Fetch the shipping fee from the API
306   $get('../databases/request/get_shipping_fee.php', (shippingResponse) => {
307     var data = JSON.parse(shippingResponse)[0];
308     var shippingFee = parseFloat(data.price); // Assuming the API returns a number or a string that can be parsed into a number
309     if (!isNaN(shippingFee)) {
310
311       // Append a row for the shipping fee with an editable input field
312       $('#detailsData').append(
313         '<tr id="shippingRow">
314           <td colspan="6" class="text-right"><strong>Shipping Fee:</strong></td>
315           <td>
316             <input type="number" value="${shippingFee.toFixed(2)}" min="0" class="form-control shipping-fee-input">
317           </td>
318         </tr>
319       ');
320       // Update the grand total to include the shipping fee
321       updateGrandTotal(shippingFee);
322     } else {
323       console.log('else');
324       // Handle the case where the shipping fee isn't valid (optional)
325       $('#detailsData').append(
326         '<tr id="shippingRow">
327           <td colspan="6" class="text-right"><strong>Shipping Fee:</strong></td>
328           <td>
329             <input type="number" value="0.00" min="0" class="form-control shipping-fee-input">
330           </td>
331         </tr>
332       ');
333     }
334   });
335 }
```

*Figure 101. Code Snippet for Registrar – request\_document.php*

Figure 101 represents a dynamic PHP and JavaScript-based system that manages and displays student document requests within a registrar's web portal. It integrates database queries, status handling, modal data manipulation, and AJAX-based form submission for a smooth user experience.

At the top of the PHP section, the associative array `$statusClasses` maps status values such as 'Done', 'Pending', and 'Inactive' to Bootstrap badge classes



('success', 'warning', and 'danger') to dynamically style the status badge for each request row. This ensures users can easily identify the state of each request via visual color cues. The SQL query inside `mysqli_query()` joins the requests, users, and `ref_departments` tables to retrieve detailed request data, such as student names, departments, and courses. However, it only selects records with 'Pending' status and a non-null request date. Each result is looped through and displayed in a table, with columns like name, department, course, and dynamically formatted date using PHP's `date()` function.

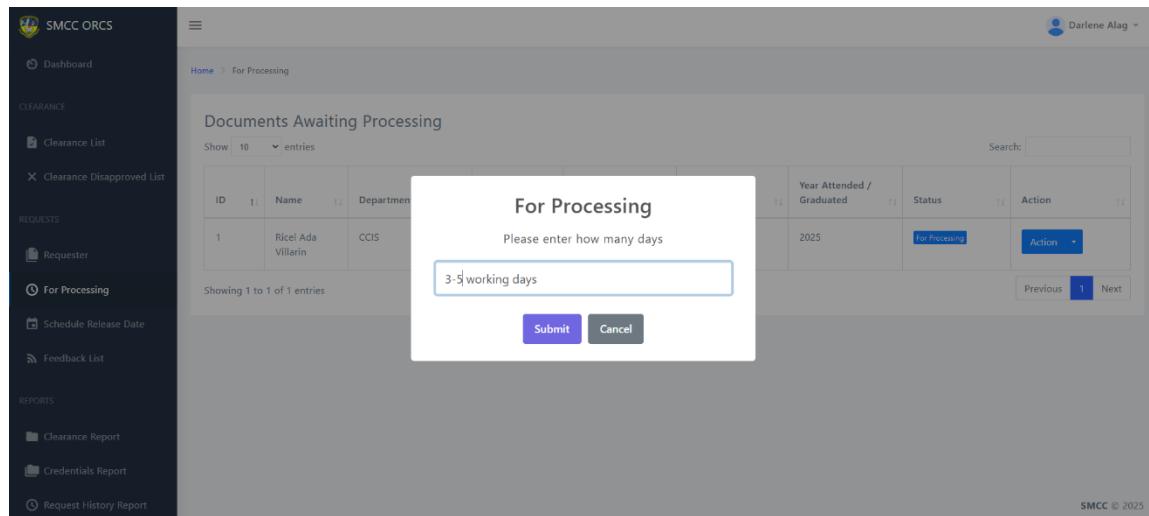
A noteworthy UI functionality is handled in the JavaScript section, particularly with the event listeners such as `$(document).on('click', 'a[data-role=denied]', ...)`. This function uses SweetAlert2 to prompt the user for a reason before denying a request. It then submits this data to a PHP endpoint (`denied.php`) via `$.post()` and gives feedback on the result, ensuring proper validation using `preConfirm`.

Additionally, the Details Modal allows administrators to view and update request specifics. When the Details button is clicked (`a[data-role=details]`), the script makes a GET request to retrieve the request's associated documents. These are populated into a dynamic table using `$('#detailsData').append(...)`, each row including fields like document type, name, price, copies, and total.

The `#saveButton` click event gathers these fields, calculates totals, and sends them as a **JSON** string to `save_details.php` using an asynchronous `$.post()` request. Before submission, the action is confirmed with a SweetAlert dialog. The logic ensures form integrity by validating numeric inputs and updating the total, including document costs and a fetched shipping fee (`get_shipping_fee.php`).



Functions like `updateGrandTotal(shippingFee)` recalculate the final amount based on the document details and shipping fee, ensuring real-time feedback as users adjust values.



*Figure 102. Registrar – For Processing*

Figure 102 illustrates the Processing Documents interface, where the registrar can view document requests from students or alumni after payment. The interface displays the request details and student information. The registrar can use the action button to notify the requester that their credentials are being processed.

```

197 | <script>
198 | $(document).ready(function(){
199 |
200 |     $(document).on('click', '[data-role=process]', function() {
201 |         var id = $(this).data('id');
202 |         var email = $(this).data('email');
203 |         var days = $(this).data('days');
204 |         const formData = { id: id, email: email };
205 |
206 |         var defaultValue = days;
207 |
208 |         // Show SweetAlert2 input prompt for feedback (as a text input instead of textarea)
209 |         swal.fire({
210 |             title: "For Processing",
211 |             text: "Please enter how many days",
212 |             input: 'text',
213 |             inputPlaceholder: "Write something",
214 |             inputValue: defaultValue,
215 |             showCancelButton: true,
216 |             confirmButtonText: 'Submit',
217 |             cancelButtonText: 'Cancel',
218 |             preConfirm: (inputValue) => {
219 |                 if (!inputValue) {
220 |                     Swal.showValidationMessage('You need to enter something!');
221 |                     return false;
222 |                 }
223 |                 return inputValue;
224 |             }
225 |         })
226 |     });
227 | });
228 |
229 | 
```



```

250
251     $(document).on('click', 'a[data-role-details]', function() {
252         var id = $(this).data('id');
253         var email = $(this).data('email');
254         $('input[name="request_id"]').val(id);
255         $('input[name="email"]').val(email);
256         var status = $(this).data('status');
257         var saveButton = document.getElementById('saveButton');
258
259         // Check the status and hide the button if it's 'No Upload'
260         if (status != 'Pending') {
261             saveButton.style.display = 'none';
262         } else {
263             saveButton.style.display = 'inline'; // or 'block' depending on your layout
264         }
265
266         $.ajaxSetup({ async: false });
267         $.get('../dbassets/request/get_details.php', {id}, (response) => {
268             $('#detailsData').empty();
269             var rows = JSON.parse(response);
270             var shippingFee = parseFloat(rows[0].fee);
271
272             rows.forEach((row, index) => {
273                 $('#detailsData').append(
274                     <tr data-row-index="${index}" data-id="${row.id}">
275                         <td>${index + 1}</td>
276                         <td>${row.doc_type}</td>
277                         <td>${row.doc_name}</td>
278                         <td>
279                             <input type="number" value="${row.copies}" min="1" class="form-control copies-input" data-index="${index}" required>
280                         </td>
281                         <td>
282                             <input type="number" value="${row.stamp_price}" class="form-control stamp-input" data-index="${index}" readonly>
283                         </td>
284                     </tr>
285                 );
286             });
287         });
288     });

```

*Figure 103. Code Snippet for Registar – for\_processing.php*

Figure 103 illustrates the implementation of a document processing system where pending document requests are displayed in a structured table format and processed through interactive modal prompts and **AJAX** transactions. The system leverages a combination of PHP, MySQL, JavaScript, and SweetAlert2 to allow dynamic interaction and secure handling of user-submitted data.

At the beginning of the script, the session management block (`session_start()` and `session checking`) ensures that only authenticated users can access the page, redirecting unauthenticated users to the login page (`../index.php`) — a crucial security measure. The page also includes reusable components (navbar, sidebar, footer) through include statements for consistent UI rendering.

The core of this snippet lies in the table element with ID `zero_config`, which dynamically lists pending document requests using data fetched from the database (`SELECT r.* , ... FROM requests r JOIN ... WHERE r.status = 'For Processing'`). Notably,



it ensures one row per request, and ORDER BY max\_processing\_days DESC prioritizes items with the longest processing time, emphasizing urgency.

One crucial functionality occurs in the <button> under the “Action” column, which includes two dropdown items: Details – triggers a modal showing document breakdown (#details-modal) via the **AJAX** request to get\_details.php. Process Documents – opens a SweetAlert2 prompt to input processing days and sends data to processing.php.

Another vital feature is the JavaScript function updateGrandTotal(), which calculates and updates the total cost of requested documents, including stamp fees and shipping. It iterates over the total cells and combines them with the shipping fee to compute a real-time display of the total.

*Figure 104. Registrar – Schedule Release Date*

Figure 104 illustrates the List of Scheduled Release Dates for Documents interface, where the registrar can view the details and types of documents requested. The registrar can also check the official receipt to verify that the student or alumni has completed the payment for the requested credentials. A modal input form opens by clicking the action button, allowing the registrar to inform the student or alumni that their request is in transit.



```

221 | <script>
222 | $(document).ready(function(){
223 |   var edit = false;
224 |   $('#input[name="date_release"]').val(new Date().toISOString().split('T')[0]);
225 |
226 |   $(document).on('click', 'button[data-role=receipt]', function(){
227 |     var imagePath = $(this).data('image');
228 |     $('#modalImage').attr('src', '../assets' + imagePath);
229 |     $('#view-modal').modal('toggle');
230 |   });
231 |
232 |   $(document).on('click', 'a[data-role=schedule]', function(){
233 |     var id = $(this).data('id');
234 |     var email = $(this).data('email');
235 |     var address = $(this).data('address');
236 |     $('#input[name="id"]').val(id);
237 |     $('#input[name="email"]').val(email);
238 |     $('#input[name="address"]').val(address);
239 |     $('#schedule-modal').modal('toggle');
240 |   });
241 |
242 |   $('#schedule-form').submit(function(e){
243 |     e.preventDefault();
244 |
245 |     var id = $('#input[name="id"]').val();
246 |     var email = $('#input[name="email"]').val();
247 |     var courier = $('#input[name="courier"]').val();
248 |     var referenceNo = $('#input[name="reference_no"]').val();
249 |     var dateRelease = $('#input[name="date_release"]').val();
250 |     var expectedDelivery = $('#input[name="expected_delivery"]').val();
251 |     var address = $('#input[name="address"]').val();
252 |
253 |     const url = '../dbassets/request/release.php';
  
```

*Figure 105. Code Snippet for Registrar – schedule\_release.php*

Figure 105 presents a web interface designed to schedule the release date of requested academic credentials. The main interface is a responsive table (<table id="zero\_config">) that displays credential requests with the status “Processing.” Each table row is dynamically populated from the database using a complex SQL JOIN query that pulls user information from related tables (users, address, ref\_city, ref\_provinces, ref\_brgy) and merges it with the requests table. This multi-table query ensures that each request includes full user details such as name, email, and complete mailing address.

Each record includes an Action column containing a dropdown menu with a “Schedule Date” option. This item triggers a JavaScript function bound to data-role="schedule" which captures important attributes (id, email, address) and opens the #schedule-modal form for inputting shipment details.

The schedule modal form (#schedule-modal) is prefilled with the current date (\$('input[name="date\_release"]').val(new



Date().toISOString().split('T')[0]);) and collects additional release information including estimated delivery days, courier name, reference number, and complete address. When submitted, the form sends the data via **AJAX POST** to ../dbassets/request/release.php. During the process, a SweetAlert2 loading modal appears (Swal.fire({ ... showLoading(); })), and upon success, another SweetAlert confirms the data was saved (Swal.fire(... 'Data saved successfully' ...)), followed by a page refresh.

Additionally, the “View Receipt” functionality enhances administrative verification. When a user clicks the receipt button (data-role="receipt"), it fetches the uploaded image path and displays the official receipt in a modal (#view-modal) using jQuery's \$( '#modalImage' ).attr('src', ...).

The screenshot shows the 'Feedback List' section of the SMCC ORCS application. The left sidebar contains navigation links for Dashboard, Clearance (Clearance List, Clearance Disapproved List), Requests (Requester, For Processing, Schedule Release Date), and Reports (Clearance Report, Credentials Report, Request History Report). The main content area is titled 'Feedback List' and shows a table with one row of data. The table columns are ID, Date Requested, Date Approved, Date Released, Name, Feedback, and Status. The data row is: ID 1, Date Requested May 4, 2025, Date Approved May 4, 2025, Date Released May 4, 2025, Name Riel Ada Villarin, Feedback (button labeled 'View'), and Status (button labeled 'Done'). The top right corner shows the user Darlene Alag.

ID	Date Requested	Date Approved	Date Released	Name	Feedback	Status
1	May 4, 2025	May 4, 2025	May 4, 2025	Riel Ada Villarin	<a href="#">View</a>	<a href="#">Done</a>

Figure 106. Registrar – Feedback List

Figure 106 presents the feedback list, where the staff or registrar can view responses submitted by requesters. This allows them to address any concerns or inconveniences and use the feedback to improve future services.



```

213 <script>
214   $(document).ready(function(){
215     $(document).on('click', 'button[data-role=view]', function(){
216       var request_id = $(this).data('id');
217       fetchFeedbackData(request_id);
218       setTimeout(() => {
219         var toPrint = document.getElementById('report-form');
220
221         // if ($('#request-table tbody').children().length === 0) {
222         //   alert('No data available to generate report.');
223         //   return; // Stop the printing if no data is available
224         // }
225
226         var newTab = window.open('', '_blank');
227         newTab.document.write('<html><head><title>' + document.title + '</title>');
228
229         // Link to an external CSS file
230         newTab.document.write('<link rel="stylesheet" type="text/css" href="../assets/css/feedback_report.css?v=' + new Date(
231
232         newTab.document.write('</head><body>');
233         newTab.document.write(toPrint.innerHTML);
234         newTab.document.write('</body></html>');
235
236         newTab.document.close();
237         // Wait for the CSS and other resources to fully load before printing
238         newTab.onload = function() {
239           newTab.print();
240         };
241
242         // Focus on the new tab
243         newTab.focus();
244       }, 500); // Delay of 500 milliseconds (0.5 seconds)
245     });
246   });

```

*Figure 107. Code Snippet for Registrar – request.php*

Figure 107 illustrates the process of retrieving feedback from metadata from the database, displaying the form details, and dynamically generating a printable report based on user interaction. The first PHP block begins by including a database connection file using `require_once('../dbassets/connection.php')`, ensuring the script can securely access the database. The SQL query `SELECT * FROM ref_iso_codes WHERE form_name = 'Feedback Form'`; fetches all details related to the feedback form from the `ref_iso_codes` table. The while loop then extracts specific fields like `form_code`, `issue_status`, `revision_no`, `date_effective`, and `approved_by`, echoing in the subsequent HTML `<table>` structure to render the form metadata visibly on the webpage.

The JavaScript section plays a pivotal role in the interactivity of the report. It binds a click event to a button with the `data-role="view"` attribute. Upon clicking, the function retrieves the `request_id` and invokes `fetchFeedbackData(request_id)` to fetch detailed feedback data via an **AJAX GET** request to `../dbassets/request/get_feedback.php`.



Inside this function, the response is parsed to build an HTML template showing instructions, a rating scale, and responses for five service criteria: Staff Appearance, Staff Helpfulness, Speed/Efficiency, Job Knowledge, and Quality of Service. The user's ratings are visually displayed using checkmarks based on the returned data values (e.g., VS, S, D, or VD), highlighting satisfaction levels.

An additional feature is the ability to print the feedback report. After populating the data, a new browser tab is opened using `window.open()`, and the complete report content is injected into the new document. The `print()` function is then triggered within `newTab.onload` to allow printing after all resources, including a linked stylesheet (`feedback_report.css`), are loaded.

A screenshot of the SMCC ORCS application interface. The left sidebar has a dark theme with white icons and text. It includes sections for CLEARANCE (Clearance List, Clearance Disapproved List), REQUESTS (Requester, For Processing, Schedule Release Date, Feedback List), and REPORTS (Clearance Report, Credentials Report, Request History Report). The main content area shows a "Clearance Report" page with a header "Home > Report > Clearance Report". It has date filters ("From: 01/05/2025" and "To: 05/05/2025"), a department dropdown ("All Departments"), and a "Print Report" button. Below these are search and filter options ("Show: 10 entries", "Search:"). A table displays one entry: #1, Name: Ricel Ada Villarin, Department: CCIS, Course: BSIT, Date Approved: May 5, 2025, Status: Approved. At the bottom, it says "Showing 1 to 1 of 1 entries" and has navigation buttons for "Previous" and "Next". The footer of the page says "SMCC © 2025".

#	Name	Department	Course	Date Approved	Status
1	Ricel Ada Villarin	CCIS	BSIT	May 5, 2025	Approved



PRINT

Saint Michael College of Caraga					
District 8, Brgy. Triangulo, Nasipit, Agusan del Norte, Philippines					
Tel. Nos. +63 085 343-5231; 283-3113   Fax: +63 085 898-0892					
<a href="http://www.smcncaagp.edu.ph">www.smcncaagp.edu.ph</a>					
CLEARANCE REPORT					
Period: May 01, 2024 - May 05, 2024					
Registrar Office					
#	Name	Department	Course	Date Approved	Status
1	Ricel Ada Villarin	CCIS	BSIT	May 05, 2024	Approved

Prepared by:

Darlene Alag

Figure 108. Registrar – Clearance Report

Figure 108 displays the ‘Clearance Report’ page for registrar staff. The registrar can filter clearance records by date and department and generate a printable report. The table shows relevant details such as Name, Department, Course, Date Approved, and Status. A search bar and pagination controls support easy navigation.

```

214 <script>
215   $(document).ready(function(){
216
217     $('#fromDate, #toDate, #department').change(function() {
218       var fromDate = $('#fromDate').val();
219       var toDate = $('#toDate').val();
220       var department = $('#department').val();
221
222       if (new Date(fromDate) > new Date(toDate)) {
223         Swal.fire({
224           title: 'Error',
225           text: 'The "From" date must be earlier than the "To" date.',
226           icon: 'error',
227           confirmButtonText: 'OK'
228         });
229
230         // Optionally, clear the "From" date or reset it
231         $('#fromDate').val('');
232       }
233       else {
234         getData(fromDate, toDate, department);
235       }
236     });
237
238     function getData(fromDate, toDate, department) {
239       const url = '../dbassets/report/clearance_registrar.php';
240
241       var table = $('.report-table').DataTable();
242       table.clear().draw();
243       $.get(url, { fromDate, toDate, department }, (response) => {
244         console.log(response);
245         const rows = JSON.parse(response);
246         if (rows.message) {
247           alert('No data available');
248         }
249       });
250     }
251   });
252 
```



```

270
271     $('#printReport').click(function(event) {
272         event.preventDefault(); // Prevent form submission
273
274         var fromDate = $('#fromDate').val();
275         var toDate = $('#toDate').val();
276         var department = $('#department').val();
277
278         if (!fromDate || !toDate) {
279             Swal.fire('System Message', "Please enter both From Date and To Date.", 'info');
280             return;
281         }
282
283         if (new Date(fromDate) > new Date(toDate)) {
284             Swal.fire('System Message', "From Date cannot be after To Date.", 'info');
285             return;
286         }
287
288         $('#periodText').text(moment(fromDate).format('MMMM DD, YYYY') + ' - ' + moment(toDate).format('MMMM DD, YYYY'));
289         fetchReportData(fromDate, toDate, department);
290
291         setTimeout(() => {
292             // $('#report-modal').modal('toggle');
293             let css =
294                 '@media print {
295                     @page {
296                         size: A4 landscape; /* Ensure page size is A4 in landscape orientation */
297                     }
298                 }';
299

```

*Figure 109. Code snippet for Registrar – report\_clearance\_registrar.php*

Figure 109 presents a PHP and JavaScript-driven web interface for generating a Clearance Report within a student information system. It begins with PHP session handling (`session_start()` and `$_SESSION['id']`) to ensure the user is authenticated before accessing the page. If the session ID is missing, the script redirects the user to the login page, enforcing access control (`if (!isset($_SESSION['id'])) { header("Location: ../../index.php"); }`).

The form section allows users to filter report data by date range (`fromDate`, `toDate`) and department (`department`). These filters trigger a jQuery change event, which validates the date range and calls the `getData()` function. This function uses an **AJAX GET** request to `clearance_registrar.php`, retrieving JSON-formatted clearance data based on the selected filters. A conditional block inside `getData()` processes the data and dynamically populates the `.report-table` using DataTables, enabling search, sort, and pagination capabilities.

A key feature is the printable modal report. When the "Print Report" button is clicked (`#printReport`), the script validates inputs again and uses `fetchReportData()` to fill the



printable report table (#report-table) . It then opens a new browser tab, dynamically injects styles including the A4 landscape print layout, and automatically triggers the print dialog using `window.print()` after a short timeout.

Additionally, PHP is used within the HTML modal template to personalize the “Prepared by” section (<?php echo \$\_SESSION['name']; ?>) , ensuring accountability.

#	Name	Department	Course	Document Type	Mailing Address	Date Released	Status
1	Rigel Ada Villarin	CCIS	BSIT	Certificate - General Weighted Average	D-1, Alipao, ALEGRIA, SURIGAO DEL NORTE	May 5, 2025	Done
2	Rigel Ada Villarin	CCIS	BSIT	Certificate of Honors	D-1, Alipao, ALEGRIA, SURIGAO DEL NORTE	May 5, 2025	Done
3	Rigel Ada Villarin	CCIS	BSIT	Certification of Graduation	D-1, Alipao, ALEGRIA, SURIGAO DEL NORTE	May 5, 2025	Done
4	Rigel Ada Villarin	CCIS	BSIT	Transcript of Record - Employment Purpose	D-1, Alipao, ALEGRIA, SURIGAO DEL NORTE	May 5, 2025	Done

#	Name	Department	Course	Document Type	Mailing Address	Date Released	Status
1	Rigel Ada Villarin	CCIS	BSIT	Certificate - General Weighted Average	D-1, Alipao, ALEGRIA, SURIGAO DEL NORTE	May 05, 2025	Done
2	Rigel Ada Villarin	CCIS	BSIT	Certificate of Honors	D-1, Alipao, ALEGRIA, SURIGAO DEL NORTE	May 05, 2025	Done
3	Rigel Ada Villarin	CCIS	BSIT	Certification of Graduation	D-1, Alipao, ALEGRIA, SURIGAO DEL NORTE	May 05, 2025	Done
4	Rigel Ada Villarin	CCIS	BSIT	Transcript of Record - Employment Purpose	D-1, Alipao, ALEGRIA, SURIGAO DEL NORTE	May 05, 2025	Done

Prepared by:  
Darlene Alag

Approved by:  
Cecilio C. Ariolla

*Figure 110. Registrar – Report Requested Credentials*

Figure 110 shows the ‘Requested Credentials Report’ page used by registrar staff. It enables users to generate reports by filtering requests through date range, department,



document type, and student name. The table displays key information, including the requester's name, course, type of document requested, mailing address, approval date, and current status. A "Print Report" button is provided for quick printing, search, and pagination features for efficient navigation.

```
258 <script>
259     $(document).ready(function(){
260
261         $('#search_name').on('keypress', function(e) {
262             if (e.which === 13) { // 13 is the Enter key
263                 e.preventDefault(); // optional: prevents form submission if inside a form
264
265                 var fromDate = $('#fromDate').val();
266                 var toDate = $('#toDate').val();
267                 var department = $('#department').val();
268                 var document_type = $('#document_type').val();
269                 var search_name = $('#search_name').val();
270
271                 // You may want to add some validation or further checks for search_name if necessary
272                 getData(fromDate, toDate, department, document_type, search_name);
273
274             }
275         });
276
277         $('#fromDate, #toDate, #department, #document_type').change(function() {
278             var fromDate = $('#fromDate').val();
279             var toDate = $('#toDate').val();
280             var department = $('#department').val();
281             var document_type = $('#document_type').val();
282             var search_name = $('#search_name').val();
283
284             if (new Date(fromDate) > new Date(toDate)) {
285                 Swal.fire({
286                     title: 'Error',
287                     text: 'The "From" date must be earlier than the "To" date.',
288                     icon: 'error',
289                     confirmButtonText: 'OK'
290                 });
291             }
292         });
293     });
294 
```

```
295     else {
296         getData(fromDate, toDate, department, document_type, search_name);
297     }
298 });
299
300 $('#search_name').on('blur', function() {
301     var fromDate = $('#fromDate').val();
302     var toDate = $('#toDate').val();
303     var department = $('#department').val();
304     var document_type = $('#document_type').val();
305     var search_name = $('#search_name').val();
306
307     // You may want to add some validation or further checks for search_name if necessary
308     getData(fromDate, toDate, department, document_type, search_name);
309 });
310
311 function getData(fromDate, toDate, department, document_type, search_name) {
312     const url = '../dbassets/report/credentials_registrar.php';
313
314     var table = $('.report-table').DataTable();
315     table.clear().draw();
316     $.get(url, { fromDate, toDate, department, document_type, search_name }, (response) =>
317         console.log(response);
318         const rows = JSON.parse(response);
319         if (rows.message) {
320             alert('No data available');
321         }
322         else {
323             rows.forEach((row, index) => {
324                 table.row.add(`\n<tr>\n    <td>${row[0]}</td>\n    <td>${row[1]}</td>\n    <td>${row[2]}</td>\n    <td>${row[3]}</td>\n    <td>${row[4]}</td>\n</tr>`)
```



Figure 111. Code Snippet for Registrar – report\_credential\_registar.php

Figure 111 presents a full-featured web interface that integrates HTML, PHP, and JavaScript to handle a dynamic credential reporting system. At the core is the `<div id="report-modal" class="modal fade">`, which structures a large modal dialog containing institutional information and a credential report section. Within it, a custom-styled button (`<button class="btnPrint" ... onclick="window.print()">`) allows users to print the displayed report content immediately. The modal includes the official logo and details of the Saint Michael College of Caraga and dynamically shows the report period through `<span id="periodText"></span>`. At the bottom of the modal, PHP displays the preparer's name via `<?php echo $_SESSION['name']; ?>`, personalizing the document output. Another modal, identified by `<div id="view-modal" class="modal fade">`, provides a simple image preview feature for user-uploaded receipts using `$( '#modalImage' ).attr('src', '../assets' + imagePath);`. The interactivity of this system relies on jQuery, particularly in event listeners attached to input fields like



#search\_name, #fromDate, #toDate, and select dropdowns. When any of these inputs change or receive an Enter keypress, the getData() function sends a request to credentials\_registrar.php using \$.get() and then populates a DataTable (\$.('.report-table').DataTable()) with the returned results. For printing, the #printReport button validates the input dates, then calls fetchReportData() to structure the report. A temporary print window is created with window.open(), and the formatted content from #report-form is inserted into this new document. The line newTab.document.write('<link rel="stylesheet" type="text/css" href="../assets/css/report.css?v=' + new Date().getTime() + '">'); ensures that updated print styles are applied. The report contains a table with columns such as Name, Department, Course, Document Type, Mailing Address, Date Released, and Status, all generated through a template string filled with values parsed from the **JSON** response. This figure efficiently demonstrates dynamic data retrieval, user filtering, document previewing, and print formatting, all essential for a digital registrar system.



The screenshot shows the Request History Report page of the SMCC ORCS system. The left sidebar contains navigation links for Dashboard, Clearance (Clearance List, Clearance Disapproved List), Requests (Requester, For Processing, Schedule Release Date, Feedback List), and Reports (Clearance Report, Credentials Report, Request History Report). The main content area shows a table of request history. A single entry is displayed: Richel Ada Villarin requested an Application Form on May 5, 2025, from D-1, Alipao, ALEGRIA, SURIGAO DEL NORTE, status Done. There is a "Print Report" button at the top right of the table.

This screenshot shows the same Request History Report interface, but with a modal window overlaid. The modal displays detailed information about the application form for Richel Ada Villarin, including her name, department (CCIS), SY: 2025, course (BSIT), and a table of document names and amounts. The modal also includes the college's address, contact number (0917 707 6630), email (registrar@smcc.nasipit.edu.ph), and logos for SUCOPAC and VAB.

Document Name	Amount
Certificate - General Weighted Average	200.00
Certificate of Honors	200.00
Certification of Graduation	100.00
Transcript of Record - Employment Purpose	1050.00
<b>Total</b>	<b>1550.00</b>

Figure 112. Registrar – Request History Report

Figure 112 shows the Request History Report interface, where the registrar can filter and view records of document requests submitted by students or alumni. The list displays the requester's name, mailing address, date released, and status. Each entry provides an option to view the corresponding application form containing the requester's information and request details.



```
326 <script>
327     $(document).ready(function(){
328
329         $('#search_name').on('keypress', function(e) {
330             if (e.which === 13) { // 13 is the Enter key
331                 e.preventDefault(); // optional: prevents form submission if inside a form
332
333                 var fromDate = $('#fromDate').val();
334                 var toDate = $('#toDate').val();
335                 var department = $('#department').val();
336                 var search_name = $('#search_name').val();
337
338                 getdata(fromDate, toDate, department, search_name);
339             }
340         });
341
342         $('#fromDate, #toDate, #department').change(function() {
343             var fromDate = $('#fromDate').val();
344             var toDate = $('#toDate').val();
345             var department = $('#department').val();
346             var search_name = $('#search_name').val();
347
348             if (new Date(fromDate) > new Date(toDate)) {
349                 Swal.fire({
350                     title: 'Error',
351                     text: 'The "From" date must be earlier than the "To" date.',
352                     icon: 'error',
353                     confirmButtonText: 'OK'
354                 });
355
356                 // Optionally, clear the "From" date or reset it
357                 $('#fromDate').val('');
358             }
359         });
360     });
361
362     // If the user presses the Esc key, clear the search input field
363     $(document).on('keyup', function(e) {
364         if (e.keyCode === 27) { // Esc key
365             $('#search_name').val('');
366         }
367     });
368
369     // If the user presses the Esc key, clear the search input field
370     $(document).on('keydown', function(e) {
371         if (e.keyCode === 27) { // Esc key
372             $('#search_name').val('');
373         }
374     });
375
376     // If the user presses the Esc key, clear the search input field
377     $(document).on('keypress', function(e) {
378         if (e.keyCode === 27) { // Esc key
379             $('#search_name').val('');
380         }
381     });
382
383     // If the user presses the Esc key, clear the search input field
384     $(document).on('click', function(e) {
385         if (e.target === document.getElementById('search_name')) {
386             $('#search_name').val('');
387         }
388     });
389
390     // If the user presses the Esc key, clear the search input field
391     $(document).on('blur', function(e) {
392         if (e.target === document.getElementById('search_name')) {
393             $('#search_name').val('');
394         }
395     });
396
397     // If the user presses the Esc key, clear the search input field
398     $(document).on('focusout', function(e) {
399         if (e.target === document.getElementById('search_name')) {
400             $('#search_name').val('');
401         }
402     });
403
404     // If the user presses the Esc key, clear the search input field
405     $(document).on('focusin', function(e) {
406         if (e.target === document.getElementById('search_name')) {
407             $('#search_name').val('');
408         }
409     });
410
411     // If the user presses the Esc key, clear the search input field
412     $(document).on('focus', function(e) {
413         if (e.target === document.getElementById('search_name')) {
414             $('#search_name').val('');
415         }
416     });
417
418     // If the user presses the Esc key, clear the search input field
419     $(document).on('focus', function(e) {
420         if (e.target === document.getElementById('search_name')) {
421             $('#search_name').val('');
422         }
423     });
424
425     // If the user presses the Esc key, clear the search input field
426     $(document).on('focus', function(e) {
427         if (e.target === document.getElementById('search_name')) {
428             $('#search_name').val('');
429         }
430     });
431
432     // If the user presses the Esc key, clear the search input field
433     $(document).on('focus', function(e) {
434         if (e.target === document.getElementById('search_name')) {
435             $('#search_name').val('');
436         }
437     });
438
439     // If the user presses the Esc key, clear the search input field
440     $(document).on('focus', function(e) {
441         if (e.target === document.getElementById('search_name')) {
442             $('#search_name').val('');
443         }
444     });
445
446     // If the user presses the Esc key, clear the search input field
447     $(document).on('focus', function(e) {
448         if (e.target === document.getElementById('search_name')) {
449             $('#search_name').val('');
450         }
451     });
452
453     // If the user presses the Esc key, clear the search input field
454     $(document).on('focus', function(e) {
455         if (e.target === document.getElementById('search_name')) {
456             $('#search_name').val('');
457         }
458     });
459
460     // If the user presses the Esc key, clear the search input field
461     $(document).on('focus', function(e) {
462         if (e.target === document.getElementById('search_name')) {
463             $('#search_name').val('');
464         }
465     });
466
467     // If the user presses the Esc key, clear the search input field
468     $(document).on('focus', function(e) {
469         if (e.target === document.getElementById('search_name')) {
470             $('#search_name').val('');
471         }
472     });
473
474     // If the user presses the Esc key, clear the search input field
475     $(document).on('focus', function(e) {
476         if (e.target === document.getElementById('search_name')) {
477             $('#search_name').val('');
478         }
479     });
480
481     // If the user presses the Esc key, clear the search input field
482     $(document).on('focus', function(e) {
483         if (e.target === document.getElementById('search_name')) {
484             $('#search_name').val('');
485         }
486     });
487
488     // If the user presses the Esc key, clear the search input field
489     $(document).on('focus', function(e) {
490         if (e.target === document.getElementById('search_name')) {
491             $('#search_name').val('');
492         }
493     });
494
495     // If the user presses the Esc key, clear the search input field
496     $(document).on('focus', function(e) {
497         if (e.target === document.getElementById('search_name')) {
498             $('#search_name').val('');
499         }
500     });
501
502     // If the user presses the Esc key, clear the search input field
503     $(document).on('focus', function(e) {
504         if (e.target === document.getElementById('search_name')) {
505             $('#search_name').val('');
506         }
507     });
508
509     // If the user presses the Esc key, clear the search input field
510     $(document).on('focus', function(e) {
511         if (e.target === document.getElementById('search_name')) {
512             $('#search_name').val('');
513         }
514     });
515
516     // If the user presses the Esc key, clear the search input field
517     $(document).on('focus', function(e) {
518         if (e.target === document.getElementById('search_name')) {
519             $('#search_name').val('');
520         }
521     });
522
523     // If the user presses the Esc key, clear the search input field
524     $(document).on('focus', function(e) {
525         if (e.target === document.getElementById('search_name')) {
526             $('#search_name').val('');
527         }
528     });
529
530     // If the user presses the Esc key, clear the search input field
531     $(document).on('focus', function(e) {
532         if (e.target === document.getElementById('search_name')) {
533             $('#search_name').val('');
534         }
535     });
536
537     // If the user presses the Esc key, clear the search input field
538     $(document).on('focus', function(e) {
539         if (e.target === document.getElementById('search_name')) {
540             $('#search_name').val('');
541         }
542     });
543
544     // If the user presses the Esc key, clear the search input field
545     $(document).on('focus', function(e) {
546         if (e.target === document.getElementById('search_name')) {
547             $('#search_name').val('');
548         }
549     });
550
551     // If the user presses the Esc key, clear the search input field
552     $(document).on('focus', function(e) {
553         if (e.target === document.getElementById('search_name')) {
554             $('#search_name').val('');
555         }
556     });
557
558     // If the user presses the Esc key, clear the search input field
559     $(document).on('focus', function(e) {
560         if (e.target === document.getElementById('search_name')) {
561             $('#search_name').val('');
562         }
563     });
564
565     // If the user presses the Esc key, clear the search input field
566     $(document).on('focus', function(e) {
567         if (e.target === document.getElementById('search_name')) {
568             $('#search_name').val('');
569         }
570     });
571
572     // If the user presses the Esc key, clear the search input field
573     $(document).on('focus', function(e) {
574         if (e.target === document.getElementById('search_name')) {
575             $('#search_name').val('');
576         }
577     });
578
579     // If the user presses the Esc key, clear the search input field
580     $(document).on('focus', function(e) {
581         if (e.target === document.getElementById('search_name')) {
582             $('#search_name').val('');
583         }
584     });
585
586     // If the user presses the Esc key, clear the search input field
587     $(document).on('focus', function(e) {
588         if (e.target === document.getElementById('search_name')) {
589             $('#search_name').val('');
590         }
591     });
592
593     // If the user presses the Esc key, clear the search input field
594     $(document).on('focus', function(e) {
595         if (e.target === document.getElementById('search_name')) {
596             $('#search_name').val('');
597         }
598     });
599
599 </script>
```

*Figure 113. Code Snippet for Registrar – Report\_request\_registrar.php*

Figure 113 illustrates a JavaScript-driven reporting system that enhances user interactivity through dynamic filtering, data validation, modal display, and print functionality for credential records. The core script begins by binding a keypress event to the #search\_name input field. Pressing the Enter key (`e.which === 13`) triggers the `getData()` function using the input values for fromDate, toDate, department, and search\_name. This allows for quick searches without the need for a submit button.

A vital part of the code is the conditional validation embedded in the `change()` event listener attached to `#fromDate`, `#toDate`, and `#department`. The script checks if the "From Date" is later than the "To Date" and uses `SweetAlert (Swal.fire(...))` to show a user-friendly error message. If dates are valid, the `getData()` function calls an endpoint (`request_registrar.php`) via `$.get()` to retrieve and parse data into a `DataTable`, with the row template built dynamically using values like `row.fname`, `row.lname`, `row.status`, and formatted dates (`moment(row.date_release).format(...)`). Further interactivity is provided through a delegated click handler on buttons with `data-role="application"`.



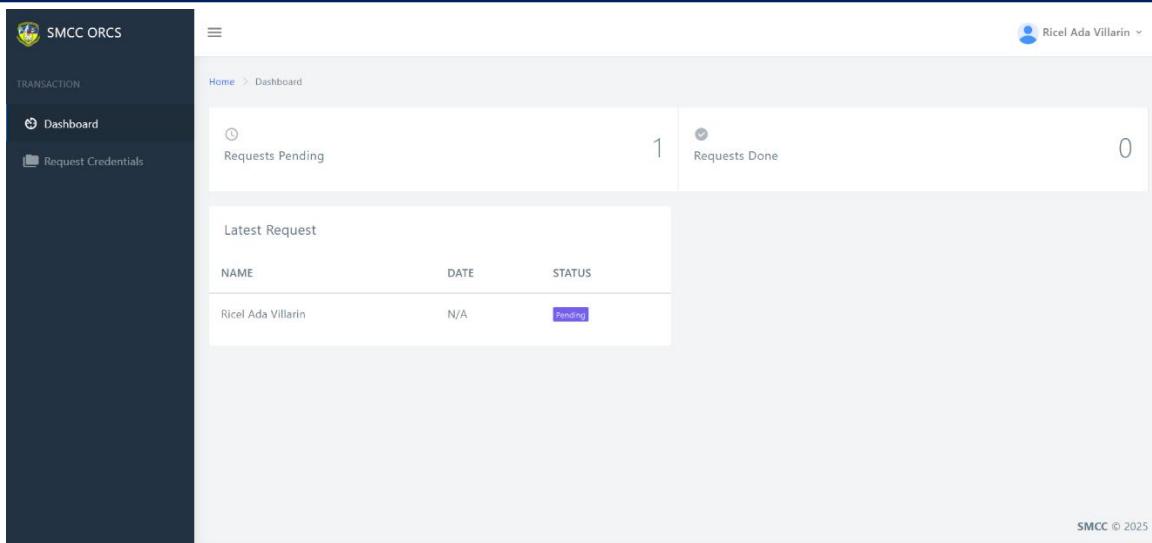
---

When clicked, this extracts specific data-\* attributes from the button—such as data-id, data-name, and data-graduated—and populates the modal with corresponding request data. It also fetches related documents from get\_requested\_documents.php using the request\_id.

Printing functionalities are split into two sections: for the main report and individual application forms. The #printReport button validates inputs before generating a printable view of #report-form. A temporary window is created via window.open(), and CSS is applied to enforce an A4 landscape layout, ensuring compatibility across browsers. It uses the line newTab.document.write('<link rel="stylesheet" type="text/css" href="../../assets/css/report.css?v=' + new Date().getTime() + '">'); This ensures that the latest version of the report stylesheet is applied without cache issues.

The fetchReportData() function structures the content for the report table with key fields such as Name, Documents, Mailing Address, Date Released, and Status. It dynamically builds the header (#report-table thead) and populates the body (#report-table tbody) based on the fetched JSON response.

Finally, the .btnPrintApplication click event mirrors the print logic but targets #view-form, tailored for individual credential application records. Both printing methods utilize a delay (setTimeout(..., 500)) to ensure modal rendering completes before triggering the print dialog.



*Figure 114. Students/Alumni – Dashboard*

Figure 114 displays the students and alumni dashboard. This interface allows students and alumni to view the status of their requests, enhancing management efficiency.

```

1  <?php
2      require_once ('../dbassets/connection.php');
3      $session_id = $_SESSION['id'];
4      $query = mysqli_query($connection, "SELECT
5          (SELECT COUNT(id) FROM `requests` WHERE user_id = '$session_id' AND `status` != 'Done') AS `c`,
6          (SELECT COUNT(id) FROM `requests` WHERE user_id = '$session_id' AND `status` = 'Done') AS `d`,
7      or die(mysqli_error($connection));
8      $row = mysqli_fetch_array($query);
9      $ongoing = $row['ongoing'];
10     $done = $row['done'];
11  ?>
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

*Figure 115. Code Snippet for Students/Alumni – Dashboard\_students.php*



---

Figure 115 illustrates a dashboard section displaying student and alumni request data.

Initially, the PHP code establishes a connection to the database and retrieves the status of requests submitted by the currently logged-in user. The code utilizes the `$_SESSION['id']` to identify the user and runs two subqueries to count the number of ongoing and completed requests from the requests table. The query results are fetched and stored in the `$ongoing` and `$done` variables, which are then used in the HTML to display the respective request statuses.

The first part of the display is composed of two cards showing the counts of pending ("Requests Pending") and completed ("Requests Done") requests, which are dynamically populated with the values stored in `$ongoing` and `$done`. The card structure uses Bootstrap's grid system to create a visually appealing layout, and the data is shown with significant text elements (`<h1 class="font-light text-right">`) to highlight the numbers.

Additionally, the code includes another section displaying the user's latest five requests in a table format. This is done through another **SQL** query that joins the requests table with the users table to fetch the user's first and last name and the request details. The query results are processed in a loop, displaying the name, the date of the request, and the status of each request. In cases where the `date_requested` or `status` is null, default values such as 'N/A' and 'Unknown' are displayed. The status is highlighted with a color-coded badge using Bootstrap's badge class. The status of each request is dynamically determined, and the badge color changes based on whether the status is 'Active' (green) or another status (blue).



The screenshot shows a modal window titled "Clearance Form". Inside, the name "Ricel Ada Villarin" is displayed. Below it, there are two dropdown menus: "Department \*" set to "CCIS" and "Course" set to "BSIT". At the bottom of the modal are "Close" and "Submit" buttons. A message at the bottom center says "No data available in table". The background shows a dark sidebar with "TRANSACTION" and "Dashboard" options, and a main area with a table header for "My Rec". The footer of the page includes the copyright notice "SMCC © 2025".

*Figure 116. Students/Alumni – Clearance Form (Department and Course Identification)*

Figure 116 illustrates the modal used for department and course identification for clearance purposes. This process assists the system in determining which department or administrative staff will be responsible for approving students' clearance.

The screenshot shows a modal window titled "Clearance Status - No. 1". It displays a single entry for "Name: Ricel Ada Villarin". Below the name are four status items, each with a green circular icon containing a number: 1. Cashier (May 4, 2025), 2. Head, Learning Resource Center (May 4, 2025), 3. Dean / Principal (May 4, 2025), and 4. School Registrar (May 4, 2025). At the bottom of the modal are "Action" and "Close" buttons. The footer of the page includes the copyright notice "SMCC © 2025".

*Figure 117. Students/Alumni – Clearance Status*

Figure 117 displays the clearance status interface, accessible through the action button. This feature allows students and alumni to track the progress of their requests and view their clearance status.



SMCC ORCS

TRANSACTION

Dashboard Request Credentials

Home > Request Credentials

Request Clearance

### My Request List

Show 10 entries

ID	Date Requested	Date Approved	Is Cleared?	Requested Documents	Date Release	Reference	Comment	Total	Status	Action
1			<input checked="" type="checkbox"/>					0.00	Pending	<button>Action</button>

Showing 1 to 1 of 1 entries

Previous 1 Next

Search:

SMCC © 2025

SMCC ORCS

TRANSACTION

Dashboard Request Credentials

Home > Request Credentials

Request Clearance

### Application Form

Name: Riel Ada Villarin

Are you a graduate? \*

Yes  
 No

Date Requested \*

04/05/2025

Year Graduated

2025

Mailing Address:

House No. & Street \*

D-1

Province \*

SURIGAO DEL NORTE

Municipality / City \*

ALEGRIA

Barangay \*

Alipao

Certificate

- Certificate (English as a medium of instruction)
- Certificate - General Weighted Average
- Certificate - GWA for ELEMENTARY
- Certificate of Honors
- Certificate of Honors (ELEMENTARY)
- Certificate of Units Earned
- Certification of Enrollment
- Certification of Graduation

TOR

- Transcript of Record - Board Exam Purpose
- Transcript of Record - Employment Purpose
- Transcript of Records - Transfer Credential

Others

- Honorable Dismissal
- Relissance of Diploma
- School Form 10

Close Submit

Search:

Previous 1 Next

SMCC © 2025

Figure 118. Students/Alumni – Request Form

Figure 118 illustrates the request form interface, which becomes accessible through the “Action” button once the student or alumnus has completed the clearance process. This form is designed to collect essential personal information from the requester and specify the type of credential or document they want to request.



The screenshot shows the 'Payment Details' section of the SMCC ORCS system. It displays three payment method options: BDO, GCash, and another unnamed account. The total price is listed as ₱1855.00. A file upload input field is present for proof of payment, with a placeholder 'Choose file' and a note about allowed file types (png, jpg, jpeg). Below the form are buttons for 'Close', 'Save changes', and navigation links for 'Previous' and 'Next'.

*Figure 119. Students/ Alumni – Payment Details*

Figure 119 presents the payment details interface, where students and alumni can view the available payment methods provided by the school. Additionally, this interface allows users to upload their payment receipts for verification by the staff or cashier.

The screenshot shows the 'Feedback Form' section of the SMCC ORCS system. It includes a satisfaction survey with five-point Likert scale options (VS, S, D, VS) for various service aspects like Staff Appearance, Staff Helpfulness, Speed/Efficiency, Job Knowledge, and Quality of Service. Below the survey is a text area for comments/complaints, which contains the message: 'I already received the credentials. Thank you for your service.' At the bottom left is a footer with form details: Form Code No. FM-OCM-SMCC-HCF-01, Issue Status 02, Record No. 03, Date Effective 14 September 2023, and Approved by President. On the right, there is a summary table showing a total amount of 1855.00, a status of Received, and an action button labeled 'Action'.

*Figure 120. Students/Alumni – Feedback Form*

Figure 120 displays the feedback form interface, which appears after the successful delivery of a document when the user clicks the "Received Document" action button to confirm



receipt. This feedback form serves as a medium for collecting user input regarding system performance and identifying areas for future improvement.

```

353 <script>
354 $(document).ready(function(){
355     var edit = false;
356
357     function getType() {
358         var template = "";
359         var doc_type = "";
360         var counter = 0;
361
362         $.ajaxSetup({async: false });
363         $.get('../dbassets/request/get_type.php', {response} => {
364             console.log(response);
365             var rows = JSON.parse(response);
366
367             rows.forEach((row, index) => {
368                 // Add the doc_type checkbox
369                 if (counter % 2 === 0) {
370                     template += "<div class='form-group row mb-3'>"; // open new row every 2nd group
371                 }
372
373                 template += `<div class='col-md-6'>
374                     <div class='form-check'>
375                         <input type='checkbox' class='form-check-input' id='selectionType${(row.doc_type)}' data-doc-type='${row.doc_type}'>
376                         <label class='form-check-label' for='selectionType${(row.doc_type)}'>${row.doc_type}</label>
377                     </div>
378                 `;
379
380                 doc_type = row.doc_type;
381
382                 // Get the documents for this doc_type
383                 $.get('../dbassets/request/get_document_by_type.php', { doc_type }, {response} => {
384                     var rows = JSON.parse(response);
385                     rows.forEach((row, index) => {
386                         // Add the document checkboxes under this doc_type with indentation
387                         template += `
```

```

434     getType();
435
436     $('input[name="has_graduated"]').change(function() {
437         if($('input[name="has_graduated"]').is(':checked')) {
438             $('.attendedText').text('Year Graduated');
439             $('input[name="year_attended"]').val('');
440         } else {
441             $('.attendedText').text('Year Attended');
442             $('input[name="year_attended"]').val('');
443         }
444     });
445
446     $('select[name="clearance_department"]').on('change', function() {
447         var department_id = $(this).val();
448         $.get('../dbassets/reference/get_course_by_department.php', { department_id }, function(response) {
449             console.log(response);
450             // Parse the JSON response
451             var data = JSON.parse(response);
452
453             var $courseSelect = $('select[name="course"]');
454             $courseSelect.empty();
455             $courseSelect.append(
456                 $(<option>).val('').text('Please select')
457             );
458
459             // Populate the second select with new options
460             $.each(data, function(index, item) {
461                 $courseSelect.append(
462                     $(<option>).val(item.course).text(item.course)
463                 );
464             });
465         });

```

```

466     $('select[name="clearance_department"]').trigger('change');
467
468     $(document).on('click', '[data-role=add], #close', function(){
469         var id = $(this).data('id');
470         $('#request-form').trigger("reset");
471         $('#input[name="id"]').val(id);
472         $('#input[name="date_requested"]').val(new Date().toISOString().split('T')[0]);
473         $('#select[name="course"]').trigger('change');
474         $('#add-modal').modal('toggle');
475     });
476
477     $('#request_clearance').click(function() {
478
479         $.get('../dbassets/request/checker.php', function(response) {
480             var data = JSON.parse(response);
481             console.log(data);
482             if (data.length != 0) {
483                 if (data[0]['status'] == 'Released' && data[0]['with_feedback'] == 'No') {
484                     $('#input[name="request_id"]').val(data[0]['id']);
485                     $('#feedback-modal').modal('toggle');
486                 }
487                 else {
488                     $('#clearance-form-modal').modal('toggle');
489                 }
490             }
491             else {
492                 $('#clearance-form-modal').modal('toggle');
493             }
494         });
495     });

```



```

500  $('#clearance-form').submit(function(e){
501      e.preventDefault();
502
503      var clearance_department = $('select[name="clearance_department"]').val();
504      var course = $('select[name="course"]').val();
505
506      const url = './dbassets/request/submit_clearance.php';
507      $.post(url, { department: clearance_department, course }, (response) => {
508          if ($.trim(response) == 'success') {
509              Swal.fire({
510                  title: 'System Message',
511                  text: "Request submitted successfully. Await clearance validation. You'll receive further instructions via email.",
512                  icon: 'success',
513                  confirmButtonText: "Okay"
514              }).then((result) => {
515                  if (result.isConfirmed) {
516                      location.reload();
517                  }
518              });
519          } else {
520              Swal.fire('System Message', response, 'info');
521          }
522      });
523  });
524
525
526  $('#request-form').submit(function(e){
527      e.preventDefault();
528
529      var id = $('input[name="id"]').val();
530      var has_graduated = $('input[name="has_graduated"]:checked').val();
531      var year_attended = $('input[name="year_attended"]').val();
532      var date_requested = $('input[name="date_requested"]').val();
533      var street = $('input[name="street"]').val();
534      var province = $('select[name="province"]').val();
535      var city = $('select[name="city"]').val();
536      var brgy = $('select[name="brgy"]').val();
537
538      var selectedDocs = [];
539
540      $('input[type="checkbox"]:checked').each(function() {
541          if ($(this).prop('id').startsWith('selectionDoc')) {
542              selectedDocs.push($(this).val());
543          }
544      });
545
546      if (selectedDocs.length === 0) {
547          alert('Please select documents!');
548          return;
549      }
550
551      const url = './dbassets/request/submit.php';
552      $.post(url, {id, has_graduated, year_attended, date_requested, selectedDocs, street, province, city, brgy}, (data) => {
553          const response = JSON.parse(data);
554          if ($.trim(response.status) == 'success') {
555              Swal.fire({
556                  title: 'System Message',
557                  text: "Request submitted successfully. Await clearance validation. You'll receive further instructions via email."
558              });
559          } else {
560              Swal.fire('System Message', response.message, 'info');
561          }
562      });
563  });
564
565
566
567
568
569
570

```

Figure 121. Code Snippet for Students/Alumni – request.php



---

Figure 121 presents a jQuery-based JavaScript script for handling dynamic interactions and form submissions within the SMCC Online Request and Clearance System interface. One of the core functions, `getType()`, initiates an asynchronous retrieval of document types via a GET request to `get_type.php`. It dynamically builds checkboxes for each document type and their respective options by sending subsequent requests to `get_document_by_type.php`. The logic includes grouping inputs in rows, enabling or disabling child checkboxes based on the selected document type, and appending the final HTML template to the container with the ID `containerID`.

A vital part of the form logic lies in the checkbox change event listener. This manages the relationship between the parent (document type) and the child (individual documents) checkboxes. Specifically, when a document type is selected, its associated documents are enabled for selection; if it is unchecked, all related documents are disabled and unchecked for clarity and validation control.

Another notable functionality is the dynamic population of course options based on the selected department using a **GET** request to `get_course_by_department.php`. This is triggered when the `clearance_department` dropdown changes and is crucial for ensuring that the course list is relevant to the selected department.

When clicking the Request Clearance button (`#request_clearance`), the system first verifies if the user has a pending feedback requirement using `checker.php`. If feedback is pending (i.e., the last request was "Released" but no feedback was given), the feedback modal is shown. Otherwise, the clearance form modal is displayed to proceed with a new request.

The `#clearance-form` submission is handled using an **AJAX POST** request to `submit_clearance.php`, which includes department and course data. A successful submission triggers a **SweetAlert** confirmation and page reload. Similarly, the `#request-form`



---

submission processes a more detailed data set, including graduation status, year attended, complete address, and selected document IDs. It validates that at least one document is selected and sends the request via `submit.php`. Upon a successful response, it notifies the user with another **SweetAlert** dialog and reloads the page for a clean user state.



## 4.2 Testing Process

The testing process outlines the various procedures to assess the system's accuracy, functionality, and overall performance. To evaluate its effectiveness, efficiency, and reliability, the researchers utilized a survey instrument based on the system usability, functionality, and efficiency scale test. This instrument was adapted from the ISO 25010 Software Product Quality Standards. The test scale provided respondents with four rating options, where 4 represented the highest score and one the lowest.

The collected data were analyzed and interpreted according to the following parameters:

**Table 3**

**Functional Suitability**

<b>Adjectival Rating</b>	<b>Scale Range (Mean)</b>	<b>Verbal Interpretation</b>
4	3.50 – 4.0	Very Functional
3	2.50 – 3.49	Functional
2	1.50 – 2.49	Moderately Functional
1	1.0 – 1.49	Poorly Functional

Table 3 represents the parameters to which a product or system provides functions that meet stated and implied needs when used under specified conditions.

**Table 4**

**Performance Efficiency**

<b>Adjectival Rating</b>	<b>Scale Range (Mean)</b>	<b>Verbal Interpretation</b>
4	3.50 – 4.0	Very Efficient
3	2.50 – 3.49	Efficient




---

2	1.50 – 2.49	Moderately Efficient
1	1.0 – 1.49	Poorly Efficient

---

Table 4 represents the performance relative to the resources used under stated conditions.

**Table 5****Usability**


---

<b>Adjectival Rating</b>	<b>Scale Range (Mean)</b>	<b>Verbal Interpretation</b>
4	3.50 – 4.0	Very Usable
3	2.50 – 3.49	Usable
2	1.50 – 2.49	Moderately Usable
1	1.0 – 1.49	Poorly Usable

---

Table 5 represents the parameters to which specified users can use a product or system to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use.

**Table 6****Respondent's Distribution**


---

<b>Online Request of Credentials for Saint Michael College of Caraga</b>		
<b>Position</b>	<b>N (Population)</b>	<b>Percentage (%)</b>
Student	1	10
Alumni	1	10
Registrar Staff	3	30
Dean	1	10
Cashier Staff	1	10
Principal	2	20
Librarian	1	10

---



<b>Total</b>	<b>10</b>	<b>100</b>
--------------	-----------	------------

The respondents included students, alumni, registrar and cashier staff, the dean, the principal, and the librarian of Saint Michael College of Caraga. The researchers employed purposive sampling techniques to conduct the survey.

The primary data was gathered directly from the respondents. Survey instruments were distributed, completed, collected, and recorded for analysis. The collected data was then analyzed and interpreted to support the project's development.

To ensure the validity of the results, the researchers also conducted a focus group discussion and a project demonstration. Participants tested and evaluated the system to assess its effectiveness.

### **Descriptive Statistics Result**

**Table 7**

#### **Functional Suitability**

<b>A. Functional Suitability</b>	<b>MEAN</b>	<b>VERBAL INTERPRETATION</b>
1. Functional completeness - Degree to which the set of functions covers all the specified tasks and user objectives	3.60	VF
2. Functional correctness - Degree to which a product or system provides the correct results with the needed degree of precision.	3.60	VF
3. Functional appropriateness - Degree to which the functions facilitate the accomplishment of specified tasks and objectives.	3.60	VF
	<b>Weighted Mean</b>	<b>3.60</b>
		<b>VF</b>

#### **Legend:**

**VF** – Very Functional (3.50 – 4.0)

**F** – Functional (2.50 – 3.49)

**MF** – Moderately Functional (1.50 – 2.49)

**PF** – Poorly Functional (1.0 – 1.49)



Table 7 presents the mean distribution and verbal interpretation of Functional Suitability.

With a weighted mean of 3.60, it is categorized as "Very Functional" (VF). This indicates that users perceive the system as highly effective in fulfilling its intended functions.

Each functional aspect—completeness, correctness, and appropriateness—received an equal mean score of 3.60, signifying that the system comprehensively meets user objectives, provides accurate results, and efficiently facilitates task accomplishment.

The results suggest that the system delivers a seamless and user-friendly experience, making it highly accessible and reliable for users to perform tasks, manage data, and generate reports. The high rating in Functional Correctness further reinforces that the system consistently produces precise and accurate outputs, ensuring trust in its performance.

**Table 8**

**Performance Efficiency**

B. Performance Efficiency	MEAN	VERBAL INTERPRETATION
1. Time behaviour - Degree to which the response and processing times and throughput rates of a system, when performing its functions, meet requirements.	3.60	VE
2. Resource utilization - Degree to which the amounts and types of resources used by a system, when performing its functions, meet requirements.	3.60	VE
3. Capacity - Degree to which the maximum limits of a product or system parameter meet requirements.	3.50	VE
<b>Weighted Mean</b>	<b>3.57</b>	<b>VE</b>

**Legend:**

**VE** – Very Efficient (3.50 – 4.0)  
**E** – Efficient (2.50 – 3.49)

**ME** – Moderately Efficient (1.50 – 2.49)  
**PE** – Poorly Efficient (1.0 – 1.49)



Table 8 presents the mean distribution and verbal description of "Performance Efficiency," with a total mean of 3.57, classified as "Very Efficient." "Time Behaviour" and "Resource Utilization" achieved the highest mean of 3.60, indicating that the system efficiently delivers responses, processes data, handles workloads within the required time constraints, and optimally utilizes resources.

**Table 9****Usability**

C. Usability	MEAN	VERBAL INTERPRETATION
1. Appropriateness recognizability - Degree to which users can recognize whether a system is appropriate for their needs.	3.70	VU
2. Learnability - Degree to which a system can be used by specified users to achieve specified goals of learning to use the product or system with effectiveness, efficiency, freedom from risk and satisfaction in a specified context of use.	3.70	VU
3. Operability - Degree to which a system has attributes that make it easy to operate and control.	3.70	VU
4. User error protection - Degree to which a system protects users against making errors.	3.60	U
5. User interface aesthetics - Degree to which a user interface enables pleasing and satisfying interaction for the user.	3.60	VU
6. Accessibility - Degree to which a system can be used by people with the widest range of characteristics and capabilities to achieve a specified goal in a specified context of use.	3.70	VU
<b>Weighted Mean</b>	<b>3.67</b>	<b>VU</b>



## Legend:

**VU** – Very Usable (3.50 – 4.0)  
**U** – Usable (2.50 – 3.49)

**MU** – Moderately Usable (1.50 – 2.49)  
**PU** – Poorly Usable (1.0 – 1.49)

Table 9 presents the mean distribution and verbal description of "Usability," with a total mean of 3.67, classified as "Very Usable." The highest scores were in "Appropriateness Recognizability," "Learnability," "Operability," and "Accessibility," all with a mean of 3.70. This indicates that the system is highly usable, allowing users to recognize its suitability, learn and operate it efficiently, and access it regardless of their characteristics and capabilities.

**Table 10****Summary Table of the Overall Mean and Grand Distribution of the Acceptability Level**

Acceptability Level of the System in terms of:	Over-all Mean	Rating
Functional Suitability	3.60	SA
Performance Efficiency	3.57	SA
Usability	3.67	SA
<b>Grand Mean</b>	<b>3.61</b>	<b>SA</b>

## Legend:

**SA** – Strongly Acceptable (3.50 – 4.0)  
**A** – Acceptable (2.50 – 3.49)

**U** – Unacceptable (1.50 – 2.49)  
**SU** – Strongly Unacceptable (1.0 – 1.49)

The system demonstrates strong overall acceptance, particularly in Usability, which received the highest rating of "Strongly Acceptable." Functional Suitability and Performance Efficiency also fall under "Strongly Acceptable," reflecting the system's effectiveness and efficiency. With a Grand Mean of 3.61, the system is well-received, though there may still be room for further refinements to enhance user experience.



---

## CHAPTER 5

### SUMMARY, CONCLUSION, AND RECOMMENDATION

This chapter summarizes the research's key findings, draws conclusions from the results, and suggests recommendations for future enhancements or further studies. It highlights the project's accomplishments, considers its implications, and offers practical insights into improving the system and expanding its potential applications.

#### 5.1. Summary of Findings

Based on the testing process, the Online Request of Credentials for Saint Michael College of Caraga was evaluated for its accuracy, functionality, and overall performance using a survey instrument adapted from the ISO 25010 Software Product Quality Standards. The evaluation focused on three significant aspects: functional suitability, performance efficiency, and usability.

The evaluation results revealed that the system was "Very Functional" with a mean score of 3.60. This indicated that it effectively met the specific tasks and objectives required for its implementation. The system scored highly in functional completeness, correctness, and appropriateness, ensuring users received accurate results and a seamless operational experience.

Regarding performance efficiency, the system was rated as "Very Efficient," with a mean score of 3.57. It excelled in time behavior and resource utilization, demonstrating its capability to process requests promptly while efficiently utilizing system resources. However, there was room for improvement in capacity management to ensure optimal performance during peak usage periods.

The system's usability was rated as "Very Usable," with a mean score of 3.67. It performed best in appropriateness, recognizability, learnability, and accessibility, indicating that users could easily recognize its purpose, learn how to use it, and easily access it. However, user error protection and interface aesthetics could have been further refined to improve user experience.



---

Overall, the system received a "Strongly Acceptable" rating with a grand mean of 3.61, indicating that it was well-received by its users and aligned effectively with their functionality, efficiency, and usability needs.

### 5.2. Conclusion

In conclusion, the Online Request of Credentials system for Saint Michael College of Caraga demonstrated strong functional suitability, high performance efficiency, and excellent usability, making it a highly valuable tool for its intended purpose. The system effectively delivered accurate results, handled requests promptly, and provided a user-friendly interface that supported students and staff in processing credential requests. Furthermore, all the objectives set for the development and implementation of the system were successfully met, confirming its effectiveness and alignment with institutional goals.

The evaluation results affirmed that the system met its functional requirements, utilized resources efficiently, and was easy to learn and operate. These strengths made the system highly reliable and convenient for users. However, despite these strengths, areas could be improved further to enhance the system's overall quality and user satisfaction.

Specifically, capacity management needed to be optimized to ensure stable performance during peak usage. Additionally, usability enhancements such as improving operability, interface aesthetics, and user error protection would have contributed to a smoother and more intuitive experience. Despite these minor areas for improvement, the system remained highly effective, dependable, and well-suited to meet the needs of its users.



### 5.3. Recommendation/s

To further improve the system's overall performance and user satisfaction, it is recommended that capacity optimization be focused on. This involves enhancing the system's ability to handle more users and requests during peak times, ensuring consistent performance and responsiveness without resource strain.

Additionally, efforts should be made to enhance the user experience by refining error-handling mechanisms and improving interface aesthetics. Providing users with better visual cues, streamlined navigation, and error feedback will make the system more intuitive and reduce the likelihood of mistakes during use.

Finally, integrating advanced features such as real-time request tracking, automated notifications, and analytics dashboards can add value to the user experience and administrative oversight. These enhancements would improve operational efficiency and support data-driven decision-making, ensuring the system remains relevant, sustainable, and scalable in the long term.

For the SMCC Registrar's Office, it is recommended to continue collaborating closely with IT staff to monitor system usage trends, address user feedback promptly, and plan for future updates based on institutional needs. Additionally, future researchers are encouraged to explore areas such as mobile optimization, integration with other academic systems (e.g., student information systems), and the application of artificial intelligence for predictive analytics. These directions will help ensure the system's continuous growth, relevance, and adaptability in supporting academic operations.



---

## REFERENCES

- [1] K. N. Devika and R. Bhakthavatchalu, "Modified blockchain based hardware paradigm for data provenance in academia," *Advances in Science, Technology and Engineering Systems*, vol. 6, no. 1, pp. 66–77, 2021, doi: <https://doi.org/10.25046/aj060108>. [Accessed: June 2021]
- [2] A. Capeloto, "Agency perspectives on online public records request portals," *The Journal of Civic Information*, vol. 1, no. 1, pp. 59–74, Sep. 2019, doi: <https://doi.org/10.32473/joci.v1i1.115659>. [September 2019]
- [3] I. E. Khairuddin, N. A. Safian, M. K. Zaini, and N. Ahmad Uzir, "Navigating academic credentials in the digital age: Insights, challenges, and implications for graduates," *Journal of Ecohumanism*, vol. 3, no. 4, pp. 642–651, 2024, [Online]. Available: <https://www.ceeol.com/search/article-detail?id=1272306> [Accessed: Jan. 11, 2025]
- [4] K. I. R. Abang, D. L. V. Gatmaitan, F. R. Manalo, M. R. Torcelino, R. L. Rodriguez, and E. A. Serrano, "CCT online request of students credentials: A document management system for private HIEs in the Philippines," in Proceedings - 2022 2nd International Conference in Information and Computing Research, iCORE 2022, 2022. doi: <https://doi.org/10.1109/iCORE58172.2022.00024>. [Accessed: Dec. 2022]
- [5] M. Lilibeth and L. Wong, "Technology-aided document management system of the registrar's office in the division of city of San Jose del Monte, Bulacan towards efficient student service," [Online]. Available: <https://www.multijournals.org> [Jan. 2022]
- [6] B., Chakroun and J. Keevy, "Digital Credentialing Implications for the recognition of learning across borders," doi: <https://doi.org/10.54675/SABO8911>. [Accessed: 2018]
- [7] M. Chase, G. Fuchsbauer, E. Ghosh, and A. Plouviez, "Credential transparency system," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2022. doi: [https://doi.org/10.1007/978-3-031-14791-3\\_14](https://doi.org/10.1007/978-3-031-14791-3_14). [Accessed: Sep. 2022]
- [8] R. Mohanraj and Dr. C. Meenakshi, "Integration of secured authentication mechanisms to ensure integrity of user identities and assessment data for online quiz design system," *International Journal of Advanced Research in Science, Communication and Technology*, pp. 54–58, doi: <https://doi.org/10.48175/ijarsct-18610>. [Accessed: May 2024]
- [9] G. W. Matkin, "The challenge of digital credentials: How should universities respond?," in Lecture Notes in Educational Technology, 2020. doi: [https://doi.org/10.1007/978-981-15-4952-6\\_4](https://doi.org/10.1007/978-981-15-4952-6_4). [Accessed: May 2020]
- [10] R. T. Alegado, R. G. Luciano, G. M. Alcantara, A. L. Laureta, B. Ch. Y. Abando, and J.P. S. Makidang, "Design and development of chatbot request system," *International Journal of Science and Applied Information Technology*, vol. 11, no. 3, pp. 1–5, doi: <https://doi.org/10.30534/ijasait/2022/011132022>. [Accessed: June 2022]



- 
- [11] H. Flanagan, "Government-issued digital credentials and the privacy landscape," [Online]. Available: <https://openid.net/wpcontent/uploads/2023/08/Government-issued-Digital-Credentials-and-thePrivacy-Landscape-v1-1-final.pdf> , [Accessed: May 2023]
  - [12] D. A. Starosielsky, W. D. Thomas, and A. W. Brzczko, "Online identity and credential verification systems and methods protecting user data," 2019, [Online]. Available: <https://patents.google.com/patent/US20190342096A1/en> [Accessed: Nov. 06 2024]
  - [13] I. Keck, M.-E. Vidal, and L. Heller, "Digital transformation of education credential processes and life cycles - a structured overview on main challenges and research questions." [Online]. Available: [https://www.thinkmind.org/articles/elml\\_2020\\_1\\_60\\_58006.pdf](https://www.thinkmind.org/articles/elml_2020_1_60_58006.pdf) [Accessed Dec. 16, 2024]
  - [14] I. Rassameeroj, P. Jomkhamisri, and N. Thaithaweeewattana, "Student request system prototype using low-code development platform," in Proceedings - 2022 International Conference on Algorithms, Data Mining, and Information Technology, ADMIT 2022, 2022. doi: <https://doi.org/10.1109/ADMIT57209.2022.00038>. [Accessed: Sep. 2022]
  - [15] R. A. Mishra, A. Kalla, A. Braeken, and M. Liyanage, "Privacy protected blockchain based architecture and implementation for sharing of students' credentials," *Inf Process Manag*, vol. 58, no. 3, doi: <https://doi.org/10.1016/j.ipm.2021.102512>. [Accessed: May 2021]
  - [16] H. Gaikwad, N. D'Souza, R. Gupta, and A. K. Tripathy, "A blockchain-based verification system for academic certificates," in 2021 International Conference on System, Computation, Automation and Networking, ICSCAN 2021, 2021. doi: <https://doi.org/10.1109/ICSCAN53069.2021.9526377>. [Accessed: Jul. 30, 2021]
  - [17] P. Gune, A. Deighton, R. Kuo, W. Walcott, and M. Chen, "Defining an approval process for requests for approval," Oct. 2003, [Online]. Available: <https://patentimages.storage.googleapis.com/dd/a0/6c/d8219133ba0e0f/WO2003085503A1.pdf> [Accessed: Nov. 06, 2024]
  - [18] L. I. Chiahsin and L. Wu, "Credential verification and issuance through credential service providers," [Online]. Available: <https://patents.google.com/patent/US11997205B2/en>. [Accessed: May 2024]
  - [19] V. Gujar, "Enhancing verification & authenticity through digital certificate generation with facial recognition for educational institutions: A case study of IndoAI app," *Int J Res Appl Sci Eng Technol*, vol. 11, no. 12, pp. 2173–2179, doi: <https://doi.org/10.22214/ijraset.2023.57818>. [Accessed: Dec. 2023]
  - [20] C. Delgado-von-Eitzen, L. Anido-Rifón, and M. J. Fernández-Iglesias, "NFTs for the issuance and validation of academic information that complies with the GDPR," *Applied Sciences (Switzerland)*, vol. 14, no. 2, Jan. 2024, doi: <https://doi.org/10.3390/app14020706>. [Accessed: May 2021]
  - [21] J. Kaneria and H. Patel, "A secure and privacy-preserving student credential verification system using blockchain technology," *International Journal of Information and Education*
-



- Technology*, vol. 13, no. 8, pp. 1251–1260, doi: <https://doi.org/10.18178/ijiet.2023.13.8.1927>. [Accessed: Aug. 2023]
- [22] N. Terentieva, Y. Levchenko, A. Kotova, and L. Tan, “Enhancing professional competency validation for educational workers (leveraging international experience for Ukraine),” *SOCIETY. INTEGRATION. EDUCATION. Proceedings of the International Scientific Conference*, vol. 2, pp. 288–297, doi: <https://doi.org/10.17770/sie2024vol2.7824>. [Accessed: May 2024]
- [23] K. Revathi, S. Priyadarshini, R. Rahul, and V. Sriram, “*Distributed ledger credential verification system based on design thinking approach*,” doi: <https://doi.org/10.1109/ICOECA62351.2024.00055>. [Accessed: Apr. 2024]
- [24] C. Espinosa, “*Effective file upload validation techniques for web security - blue goat cyber*.” [Online]. Available: <https://bluegoatcyber.com/blog/effective-file-upload-validation-techniques-for-web-security/>. [Accessed: Dec. 12, 2024]
- [25] R. Frisch, Dobák, Dóra Éva, and J. Udvaros, “*Blockchain diploma authenticity verification system using smart contract technology - Intézményi Publikációk*.” 2023, [Online]. Available: <https://publikacio.unieszterhazy.hu/7698/>. [Accessed: Dec. 12, 2023]
- [26] W. Chen, Z. Xu, S. Shi, Y. Zhao, and J. Zhao, “A survey of blockchain applications in different domains,” *ACM International Conference Proceeding Series*, pp. 17–21, doi: <https://doi.org/10.1145/3301403.3301407>. [Accessed: Dec. 2018]
- [27] A. Askhatuly, D. Berdysheva, D. Yedilkhan, and A. Berdyshev, “Security risks of ML models: adversarial machine learning,” *SIST 2024 - 2024 IEEE 4th International Conference on Smart Information Systems and Technologies*, Proceedings, pp. 440– 446, 2024, doi: <https://doi.org/10.1109/SIST61555.2024.10629452>. [Accessed: May 2024]
- [28] H. Hamidi, “An approach to develop the smart health using internet of things and authentication based on biometric technology,” *Future Generation Computer Systems*, vol. 91, pp. 434–449, doi: <https://doi.org/10.1016/J.FUTURE.2018.09.024>. [Accessed: Feb. 2019]
- [29] C. Avram, J. Machado, and A. Aştilean, “Hardware passwords manager based on biometric authentication,” *MDPI AG*, p. 31. doi: <https://doi.org/10.3390/i3s2021dresden-10085>. [Accessed: August 2021]
- [30] F. Martín-Rodriguez and J. M. Fernandez-Montenegro, “SWAD: Secure watermarking for authentication of scanned documents,” in *2011 50th FITCE Congress - “ICT: Bridging an ever shifting digital divide”*, *FITCE 2011*, 2011. doi: <https://doi.org/10.1109/FITCE.2011.6133439>. [Accessed: 2011]
- [31] A. A. Al-Sulaiman, “*An integrated user-centric service platform to promote sustainable behaviour*,” 2014. [Online]. Available: <https://orca.cardiff.ac.uk/id/eprint/70637/> [Accessed: Nov. 06, 2024]



- 
- [32] M. Schmitt, J. Redi, and P. Cesar, "Towards context-aware interactive quality of experience evaluation for audiovisual multiparty conferencing," *International Speech Communication Association*, pp. 54–58. doi: <https://doi.org/10.21437/pqs.2016-14>. [Accessed: August 06, 2016]



## APPENDICES

### A. System Source Code

#### Database Connection

```

1 <?php
2
3
4 $server_name = "localhost";
5 $db_username = "root";
6 $db_password = "";
7 $db_name = "db_orc";
8
9 $connection = mysqli_connect($server_name,$db_username,$db_password,$db_name);
10 ?>

```

#### functions.php

```

1 <?php
2 define("BASE_DIR", '../assets'); // Define a base directory constant
3
4 // Ensure you have a valid database connection before using this function.
5 function log_action($user_id, $description, $connection) {
6     // Prepare the SQL query to insert the log entry
7     $query = "INSERT INTO logs ('user_id', 'desc', created_at) VALUES (?, ?, NOW())";
8
9     // Prepare the statement
10    $stmt = $connection->prepare($query);
11
12    // Bind the parameters: user_id is an integer, description is a string
13    $stmt->bind_param("is", $user_id, $description);
14
15    // Execute the query
16    $stmt->execute() or die($stmt->error);
17
18    // Close the statement
19    $stmt->close();
20 }
21
22 function uploadFile($fileKey, $prefix, $imageaccept, $directory) {
23     if (!isset($_FILES[$fileKey]) && $_FILES[$fileKey]['tmp_name'] != '') {
24         // Get the file extension and make it lowercase for comparison
25         $ext = strtolower(pathinfo($_FILES[$fileKey]['name'], PATHINFO_EXTENSION));
26
27         // Check if the file extension is in the accepted formats
28         if (in_array($ext, $imageaccept)) {
29             // Create a unique file name for the uploaded file
30             $file_path = '/' . $directory . '/' . uniqid($prefix . '_') . '.' . $ext;
31
32             // Ensure the full directory path exists
33             $fullDirPath = BASE_DIR . '/' . $directory;
34             if (!file_exists($fullDirPath)) {
35                 mkdir($fullDirPath, 0777, true); // Create the directory if it doesn't exist
36             }
37
38             // Move the uploaded file to the target directory
39             $move = move_uploaded_file($_FILES[$fileKey]['tmp_name'], BASE_DIR . $file_path);
40
41             // Check if the file move was successful
42             if (!$move) {
43                 echo "Error moving the file!";
44                 exit();
45             }
46             return $file_path; // Return the file path if successful
47         } else {
48             echo "File not accepted! Allowed formats: " . implode(", ", $imageaccept);
49             exit();
50         }
51     } else {
52         echo "File not accepted! Allowed formats: " . implode(", ", $imageaccept);
53         exit();
54     }
55 }
56
57 ?>

```



## login.php

```

1 <?php
2     require_once ('connection.php');
3     include 'Functions.php';
4     session_start();
5
6     $email = mysqli_real_escape_string($connection, $_GET["email"]);
7     $password = mysqli_real_escape_string($connection, $_GET["password"]);
8
9     // Query to get the user record based on the email
10    $query = mysqli_query($connection, "SELECT u.*, d.department FROM users
11                                LEFT JOIN ref_departments d ON u.department_id = d.id
12                                WHERE email = '$email' AND u.status != 'Verify Email';") or die(mysqli_error($connection));
13
14    $row = mysqli_fetch_array($query);
15
16    // If user exists
17    if ($row) {
18        $user_status = $row['status'];
19        $id = $row['id'];
20        $SESSION['id'] = $row['id'];
21        $SESSION['fname'] = $row['fname'];
22        $SESSION['mname'] = $row['mname'];
23        $SESSION['lname'] = $row['lname'];
24        $SESSION['name'] = $row['fname'] . ' ' . $row['lname'];
25        $SESSION['email'] = $row['email'];
26        $SESSION['role'] = $row['role'];
27
28        if ($row['role'] != 'Client') {
29            $department_id = $row['department_id'];
30            $SESSION['department_id'] = $row['department_id'];
31
32            $queryDepartment = mysqli_query($connection, "SELECT * FROM ref_departments WHERE id = '$department_id';") or die(mysqli_error($connection));
33            $departmentRow = mysqli_fetch_array($queryDepartment);
34            $SESSION['department'] = $departmentRow['department'];
35        }
36
37        // First, check if the user account is active
38        if ($user_status == 'Active') {
39            // Then verify the password using password_verify()
40            if (password_verify($password, $row['password'])) {
41                log_action($id, 'Logged In', $connection);
42                echo 'success';
43            } else {
44                echo "Incorrect Email or Password entered. \nPlease try again.";
45            }
46        } else {
47            echo 'User account deactivated! Please contact system administrator for further assistance.';
48        }
49
50    } else {
51        echo "Incorrect Email or Password entered. \nPlease try again.";
52    }
53 ?>

```

## logout.php

```

1 <?php
2     require_once ('connection.php');
3
4     session_start();
5     $id = $SESSION['id'];
6     $connection->query("INSERT INTO logs ('user_id', 'desc', created_at) VALUES( '$id', 'Logged Out', NOW() );"
7                         or die (mysqli_error($connection)));
8
9     session_destroy();
10    echo '<script type = "text/javascript">window.location = "../index.php";</script>';
11 ?>

```

## get\_requested\_documents.php

```

1 <?php
2     require_once ('../connection.php');
3
4     $id = $_GET['request_id'];
5
6     // Prepare and execute the query
7     $stmt = $connection->prepare("SELECT * FROM `requested_documents` rd
8                                 JOIN ref_documents d ON rd.document_id = d.id
9                                 WHERE rd.request_id = ?");
10    $stmt->bind_param("i", $id);
11    $stmt->execute();
12
13    // Get the result and output as JSON
14    echo json_encode($stmt->get_result()->fetch_all(MYSQLI_ASSOC));
15
16    // Close the statement and connection
17    $stmt->close();
18    $connection->close();
19
20 ?>

```



## received.php

```

1 <?php
2 require_once('../connection.php');
3 session_start();
4
5 $id = mysqli_real_escape_string($connection, $_POST['request_id']);
6 $serviceOne = mysqli_real_escape_string($connection, $_POST['serviceOne']);
7 $serviceTwo = mysqli_real_escape_string($connection, $_POST['serviceTwo']);
8 $serviceThree = mysqli_real_escape_string($connection, $_POST['serviceThree']);
9 $serviceFour = mysqli_real_escape_string($connection, $_POST['serviceFour']);
10 $serviceFive = mysqli_real_escape_string($connection, $_POST['serviceFive']);
11 $Suggestions = mysqli_real_escape_string($connection, $_POST['suggestions']);
12 $session_id = $_SESSION['id'];
13
14 $update_query =
15     "UPDATE `requests`"
16     . "SET"
17     . "    service_one = ?,
18     .     service_two = ?,
19     .     service_three = ?,
20     .     service_four = ?,
21     .     service_five = ?,
22     .     suggestions = ?,
23     .     with_feedback = 'Yes',
24     .     status = 'Done'
25     . WHERE id = ?"
26 ;
27
28 $stmt = mysqli_prepare($connection, $update_query);
29
30 // Assuming all fields are strings except 'rating' (int) and 'id' (int)
31 mysqli_stmt_bind_param(
32     $stmt,
33     'sssssi',
34     $serviceOne,
35     $serviceTwo,
36     $serviceThree,
37     $serviceFour,
38     $serviceFive,
39     $Suggestions,
40     $id
41 );
42
43 mysqli_stmt_execute($stmt) or die(mysqli_error($connection));
44 mysqli_stmt_close($stmt);
45
46 echo 'success';
47 ?>

```

## submit.php (payment)

```

1 <?php
2 require_once('../connection.php');
3 include '../functions.php';
4
5 session_start();
6
7 $id = mysqli_real_escape_string($connection, $_POST['id']);
8 $pay_name = mysqli_real_escape_string($connection, $_POST['pay_name']);
9 $pay_number = mysqli_real_escape_string($connection, $_POST['pay_number']);
10
11 $imageaccept = ["jpg", "png", "jpeg"];
12
13 $session_id = $_SESSION['id'];
14 $image_path = '';
15
16 $directory = "upload/payment_receipts";
17
18 // Handle file upload if present
19 if (isset($_FILES['file']) && $_FILES['file']['error'] === UPLOAD_ERR_OK) {
20     $image_path = uploadFile('file', 'file', $imageaccept, $directory);
21 }
22
23 // Prepare the query
24 if ($id) {
25     $status = mysqli_real_escape_string($connection, $_POST['status']);
26
27     // Update the record
28     $update_query = "UPDATE ref_payments SET pay_name = ?, pay_number = ?, status = ?" . ($image_path ? ", 'picture' = ?" : "") . " WHERE id = ?";
29     $stmt = $connection->prepare($update_query);
30
31     // Bind parameters
32     if ($image_path) {
33         $stmt->bind_param("ssssi", $pay_name, $pay_number, $status, $image_path, $id);
34     } else {
35         $stmt->bind_param("sssi", $pay_name, $pay_number, $status, $id);
36     }
37
38     $action = "Updated";
39 } else {
40     // Insert new record
41     $insert_query = "INSERT INTO ref_payments (pay_name, pay_number, picture, status) VALUES(?, ?, ?, 'Active')";
42     $stmt = $connection->prepare($insert_query);
43     $stmt->bind_param("sss", $pay_name, $pay_number, $image_path);
44     $action = "Added";
45 }
46
47 if ($stmt && $stmt->execute()) {
48     log_action($session_id, "$action Payment Account: " . $pay_name, $connection);
49     echo 'success';
50 } else {
51     echo 'Error, please try again.';
52 }
53
54 $stmt->close();
55 ?>

```



### get\_barangay\_by\_city.php

```

1 <?php
2
3     require_once ('../connection.php');
4
5     $city_id = mysqli_real_escape_string($connection, $_GET['city_id']);
6
7     $query = "SELECT brgy_code, brgy FROM ref_brgy WHERE city_code = '$city_id' ORDER BY brgy ASC";
8     $result = mysqli_query($connection, $query);
9
10    if(!$result) {
11        die('Query Failed'. mysqli_error($connection));
12    }
13
14    // Fetch all rows as an associative array
15    $rows = mysqli_fetch_all($result, MYSQLI_ASSOC);
16
17    // Encode the result into a JSON string
18    echo json_encode($rows);
19 ?>

```

### get\_city\_by\_province.php

```

1 <?php
2
3     require_once ('../connection.php');
4
5     $province_id = mysqli_real_escape_string($connection, $_GET['province_id']);
6
7     $query = "SELECT city_code, city FROM ref_city WHERE province_code = '$province_id' ORDER BY city ASC";
8     $result = mysqli_query($connection, $query);
9
10    if(!$result) {
11        die('Query Failed'. mysqli_error($connection));
12    }
13
14    // Fetch all rows as an associative array
15    $rows = mysqli_fetch_all($result, MYSQLI_ASSOC);
16
17    // Encode the result into a JSON string
18    echo json_encode($rows);
19 ?>

```

### get\_course\_by\_department.php

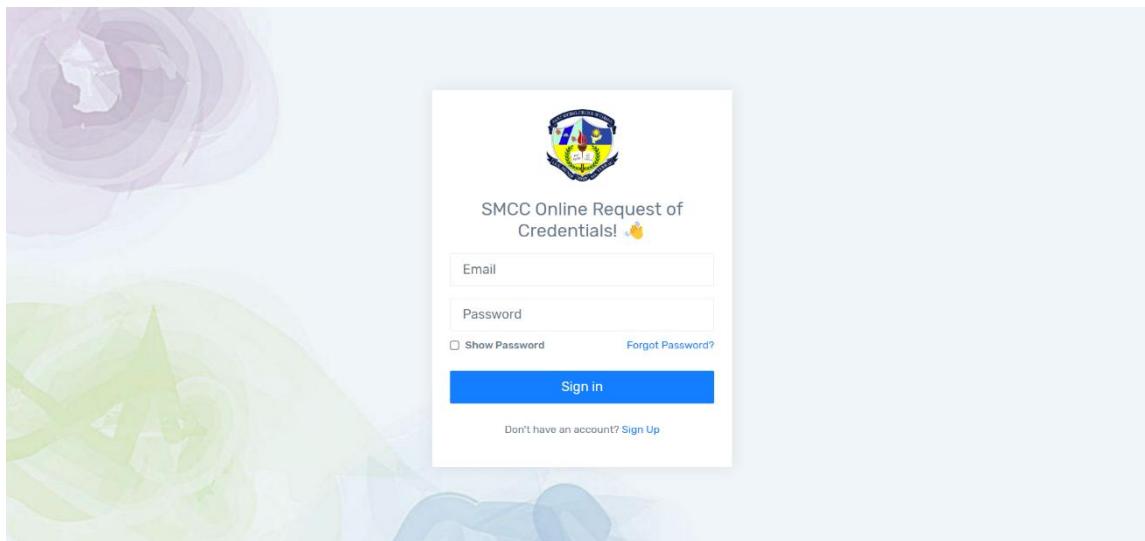
```

1 <?php
2
3     require_once ('../connection.php');
4
5     $department_id = mysqli_real_escape_string($connection, $_GET['department_id']);
6
7     $query = "SELECT course FROM ref_courses WHERE department_id = '$department_id' ";
8     $result = mysqli_query($connection, $query);
9
10    if(!$result) {
11        die('Query Failed'. mysqli_error($connection));
12    }
13
14    // Fetch all rows as an associative array
15    $rows = mysqli_fetch_all($result, MYSQLI_ASSOC);
16
17    // Encode the result into a JSON string
18    echo json_encode($rows);
19 ?>

```



## B. User's Manual



This is the login page for all users. Key personnel such as the cashier, head of LRC, dean/principal, and registrar staff can only access the system if the admin has created an account for them. This security measure helps prevent unauthorized users from accessing the organization or ORC system.

First Name *	Middle Name	Last Name *	Ext *
Ricel Ada	Salutis	Villarin	N/A

Has Middle Name

Birthdate *	Last School Year Attended *	Contact *	Email *
01/11/2003	2025	09705366534	adangvillarin@gmail.com

House No. & Street *	Province *
D-6	AGUASAN DEL NORTE

Municipality / City *	Barangay *
NASIPIT	Caramagong

Upload Valid ID \*

<input type="file"/> Valid ID.jpg	<input type="file"/> Holding with valid ID.jpg
-----------------------------------	--

Allowed file types: png, jpg, jpeg, pdf.  
Please upload any valid government ID (Driver's License, Passport, etc.)

[Sign Up](#)

Already have an account? [Sign In instead](#)

This sign-up page is designed for new users, specifically students and alumni, to create an account and gain access to the SMCC Online Request of Credentials system. To complete the registration, you must provide accurate personal information, including your first name, middle



name (optional), last name, and any applicable name extensions. Additional required details include your birthdate (selected via a date picker), the last school year attended, a valid contact number, and an active email address. You are also required to enter your complete address, including house number and street, province, municipality or city, and barangay. For identity verification, the system requires two uploads: a valid government-issued ID (such as a Driver's License or Passport) and a photo of yourself holding the same ID. Supported file formats for uploads are PNG, JPG, JPEG, and PDF. Once all required fields are completed and documents are uploaded, you can click the "Sign Up" button to submit your registration. If you already have an account, a "Sign In" link is available at the bottom of the page. This sign-up process ensures secure and verified access to the system for all users.

The screenshot shows the 'My Request List' page of the SMCC ORCS system. At the top right, there is a user profile for 'Ricel Ada Villarin'. Below it is a blue button labeled 'Request Clearance' with a red arrow pointing to it. On the far right, there is a dropdown menu labeled 'Action' with a red arrow pointing to it, and a status indicator 'Pending' with a red arrow pointing to it. The main table has columns for ID, Date Requested, Date Approved, Is Cleared?, Requested Documents, Date Release, Reference, Comment, Total, and Status. The first row shows an entry with 'Is Cleared?' set to 'No'. The bottom right corner of the page says 'SMCC © 2025'.

ID	Date Requested	Date Approved	Is Cleared?	Requested Documents	Date Release	Reference	Comment	Total	Status
1			No					0.00	Pending

The My Request List page provides you with a detailed overview of your submitted credential requests. Once you have signed in, this page displays a list of all your requests along with relevant information to help you track the progress and status of each one. (1) The Request Clearance button at the top right of the page allows you to request clearance, an essential step before documents can be processed. (2) The Clearance option in the Action dropdown button enables you to track the current status of your clearance.



This screenshot shows the 'Request Credentials' section of the SMCC ORCS system. The user, Riel Ada Villarin, has a pending clearance due to an outstanding institutional balance of 5,000. The 'Action' button includes options for 'Pay Balance' and 'Clearance'. A red arrow points to the 'Pay Balance' option.

ID	Date Requested	Date Approved	Is Cleared?	Requested Documents	Date Release	Reference	Comment	Total	Status	Action
1			No				You have an outstanding of 5,000. Please pay it first.	0.00	Pending	Action

This is the UI for balance payment when the Office of the Cashier disapproves a requester's clearance due to an outstanding institutional balance. (1) The Pay Balance option in the Action button allows you to pay their balance using their preferred payment method.

This screenshot shows the 'Request Credentials' section of the SMCC ORCS system. The user, Riel Ada Villarin, has cleared the pending status. The 'Action' button includes options for 'Request Credentials' and 'Clearance'. A red arrow points to the 'Request Credentials' option.

ID	Date Requested	Date Approved	Is Cleared?	Requested Documents	Date Release	Reference	Comment	Total	Status	Action
1			Yes					0.00	Pending	Action

This is the user interface for applying and submitting credential requests. (1) The Clearance option under the Action button must be thoroughly checked to determine whether you have completed the clearance process or still have any pending requirements that need to be fulfilled. (2) The Request Credentials option under the Action button allows you to fill out an application to request the necessary credentials.



SMCC ORCS

TRANSACTION

Dashboard Request Credentials

Home > Request Credentials

Request Clearance

My Request List

Show 10 entries

ID	Date Requested	Date Approved	Is Cleared?	Requested Documents	Date Release	Reference	Comment	Total	Status	Action
1	April 30, 2025	April 30, 2025	<span style="background-color: #007bff; color: white;">Issued</span>	Certificate - General Weighted Average, Certificate of Honors, Certification of Graduation, Transcript of Record - Employment Purpose				1855.00	<span style="background-color: #007bff; color: white;">For Payment</span>	Action

Showing 1 to 1 of 1 entries

1 → Total 1855.00

2 → Action button (Details, Clearance)

3 → Action button (Proceed to Payment)

Previous 1 Next

After the registrar confirms your request details, you can: (1) view the total price of all the documents you requested, (2) see the breakdown of your requested documents and their corresponding prices under the Details option, and (3) proceed with the payment for your requested credentials.

SMCC ORCS

TRANSACTION

Dashboard Request Credentials

Home > Request Credentials

Request Clearance

My Request List

Show 10 entries

ID	Date Requested	Date Approved	Is Cleared?	Requested Documents	Date Release	Reference	Comment	Total	Status	Action
1	April 30, 2025	April 30, 2025	<span style="background-color: #007bff; color: white;">Issued</span>	Certificate - General Weighted Average, Certificate of Honors, Certification of Graduation, Transcript of Record - Employment Purpose	April 30, 2025	1234567890		1855.00	<span style="background-color: #007bff; color: white;">Released</span>	Action

Showing 1 to 1 of 1 entries

1 → Date Release April 30, 2025

2 → Reference 1234567890

3 → Action button (Receive Documents, Details, Clearance)

Previous 1 Next

After the cashier verifies your payment and the registrar processes your documents and schedules a release date, you can: (1) view the release date and (2) reference or tracking number in this user interface. Then, (3) under the Received Documents option in the Action button, you must fill out a feedback form once your documents have successfully arrived at your location.



### C. Letter of Permission

December 11, 2024

**Mr. Cecilio C. Arriola**  
**Head of Registrar's Office**  
Saint Michael College of Caraga  
Nasipit, Agusan del Norte, 8602

Dear Mr. Arriola,

Greetings!

We hope this letter finds you well. We are third-year Bachelor of Science in Information Technology students at Saint Michael College of Caraga, currently working on our research study entitled "**Online Request of Credentials for Saint Michael College of Caraga**."

We are writing to formally request your permission and seek your collaboration for the development and implementation of this system. Our study aims to design and build an efficient online platform for requesting credentials, intended to streamline and modernize the current process.

As part of this initiative, we kindly request your support, particularly in the form of [access to relevant data, institutional policies, or feedback from your office/team], which are essential to the success of our research. We assure you that all information shared will be handled with the highest level of confidentiality and used strictly for academic purposes, in compliance with your office's guidelines.

We believe that this system will significantly improve the credential request process, aligning with the institution's commitment to innovation and academic excellence. Additionally, it may serve as a model for future digital transformation initiatives in similar institutions.

Thank you for considering our request. We look forward to your positive response and would be glad to discuss this project further at your convenience.

Sincerely Yours,  
Ricel Ada S. Villarin

Research Leader

Received by:

Cecilio C. Arriola  
Head, SMCC Registrar

Scanned with CamScanner



#### D. Documented Undertakings



The researchers successfully passed the title hearing and now working on the formulation of Chapters 1 and 2.



The researchers designing the system's user interface, creating diagrams, and specifying software and hardware requirements for Chapter 3, while also consulting with their adviser in preparation for the proposal defense.



Following the successful defense of Chapters 1 to 3 during the proposal defense, the researcher now proceeds with revising the manuscript in preparation for routing.



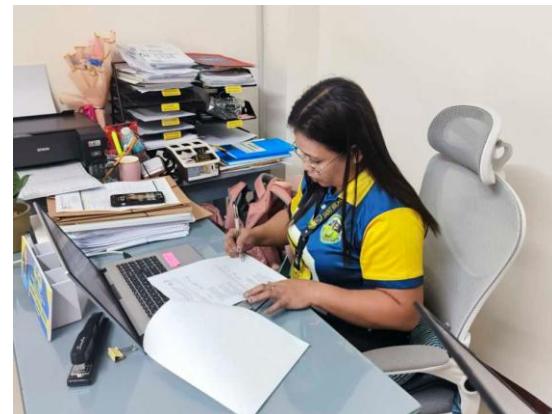
Researchers during the development of the system focused on implementing its core functionality by writing and refining the necessary code. They integrated key features, ensured smooth interactions, and conducted tests to identify and resolve any issues.



Researchers finalizing the system's user interface by refining layouts, visuals, and navigation based on initial checks and feedback from their beneficiaries. At the same time, they drafted the manuscript, detailing design decisions, implementation steps, and including annotated screenshots to explain the interface's functionality and its corresponding code snippets in Chapter 4.



Researchers consulting with their adviser and beneficiary to gather feedback on the system's progress. These discussions focused on identifying areas for improvement, refining features, and incorporating suggestions to better meet user needs. By integrating this feedback, they ensured the system's design aligned with both academic standards and practical requirements.



Researchers during the testing process worked closely with corresponding end users to evaluate the system's functionality and usability. They observed user interactions, gathered performance metrics, and collected feedback on any pain points or errors encountered. End users were provided with a questionnaire to evaluate the software's quality.



Researchers during the finalization of Chapters 4 and 5 included the code discussion, tallied end-user evaluation results, and synthesized the study's findings to draw conclusions and make clear recommendations. They also examined test data for key insights, suggested practical improvements or future research directions, and practiced debugging their code in preparation for the final defense.



Following the successful final oral defense, the researcher began revising the manuscript for routing, covering Chapters 1 to 5.



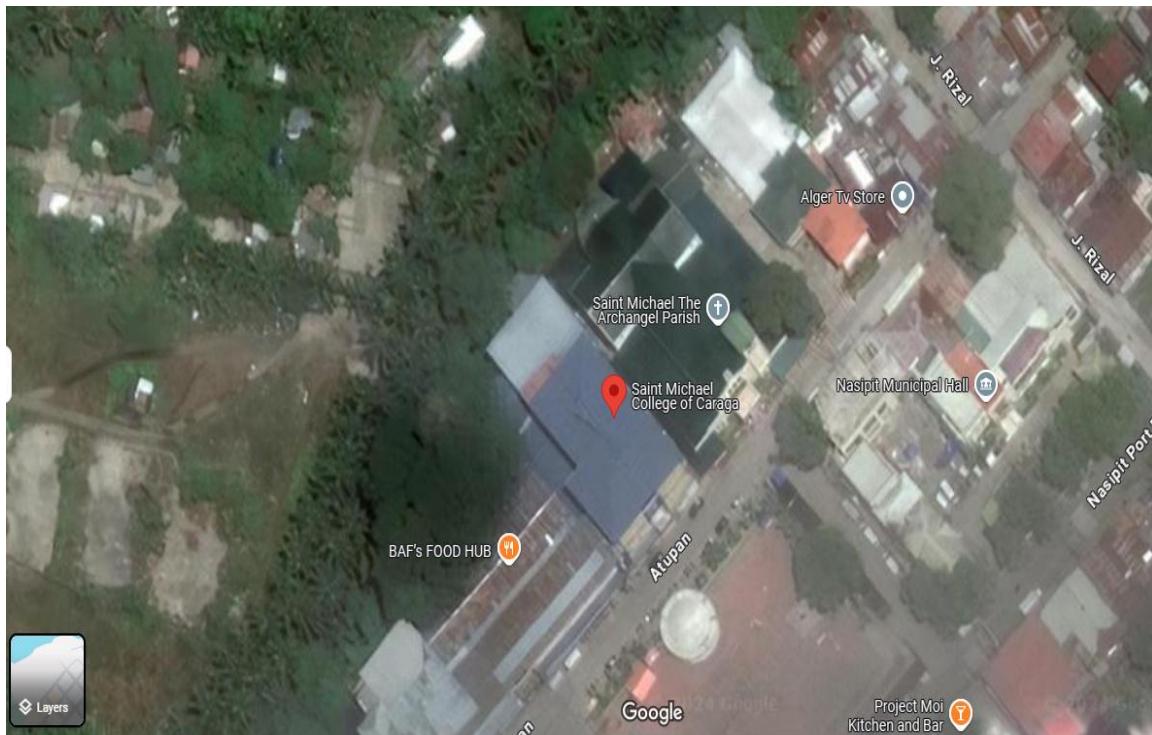
The researchers successfully deployed their system at the Electronic Data Processing (EDP) Office of Saint Michael College of Caraga, where it was formally acknowledged and signed by Mr. John Louie R. Abobo (MIS In-Charge) and Engr. Marisol S. Rosario (EDP Head).

**E. Certificate of Implementation**



#### F. Map of the Research Locale

SAINT MICHAEL COLLEGE OF CARAGA



Saint Michael College of Caraga, or SMCC, is a private Catholic school in Nasipit, Agusan del Norte, managed by the Roman Catholic Diocese of Butuan. It was founded in 1948 by the Missionaries of the Sacred Heart (MSC) fathers. The institution offers both basic and higher education, with its main campus located on Atupan Street, Barangay 4 Poblacion, Nasipit, Agusan del Norte. Its elementary and senior high school departments are housed on a separate campus in Barangay Triangulo.



### G. Curriculum Vitae

Name : Ricel Ada S. Villarin

Address : D-6 Camagong, Nasipit, Agusan Del Norte

Email : [adangvillarin@gmail.com](mailto:adangvillarin@gmail.com)



#### Personal Information

Sex : Female

Date of birth : November 1, 2003

Height : 155cm

Weight : 53kg

Civil Status : Single

#### Educational Attainment

Elementary : Camagong Elementary School

Secondary : Saint Michael College of Caraga

Tertiary : Saint Michael College of Caraga

#### Membership/Affiliations

ORGANIZATION	POSITIONS HELD
Michaelinian Cybersecurity Guild (2024-2025)	Member

**G. Curriculum Vitae**

Name : Angel Mae L. Baguio

Address : D-2 Camagong, Nasipit, Agusan Del Norte

Email : [angelmaebaguio16@gmail.com](mailto:angelmaebaguio16@gmail.com)

**Personal Information**

Sex : Female

Date of birth : February 21, 2004

Height : 160cm

Weight : 45kg

Civil Status : Single

**Educational Attainment**

Elementary : Camagong Elementary School

Secondary : Nasipit National Vocational School

Tertiary : Saint Michael College of Caraga

**Membership/Affiliations**

ORGANIZATION	POSITIONS HELD
N/A	

**G. Curriculum Vitae**

Name : Jay A. Bongado

Address : D-5 Punta, Nasipit, Agusan del Norte

Email : [jbongado5@gmail.com](mailto:jbongado5@gmail.com)

**Personal Information**

Sex : Male

Date of birth : January 25, 2004

Height : 168cm

Weight : 50kg

Civil Status : Single

**Educational Attainment**

Elementary : Nasipit Central Elementary School

Secondary : Northwestern Agusan Colleges

Tertiary : Saint Michael College of Caraga

**Membership/Affiliations**

ORGANIZATION	POSITIONS HELD
N/A	

**G. Curriculum Vitae**

Name : Angelo M. Hinautan

Address : Barangay 4, Nasipit, Agusan del Norte

Email : [angelohinautan45@gmail.com](mailto:angelohinautan45@gmail.com)

**Personal Information**

Sex : Male

Date of birth : November 23, 2000

Height : 170cm

Weight : 70kg

Civil Status : Single

**Educational Attainment**

Elementary : Libertad Central Elementary School

Secondary : Libertad National High School

Tertiary : Saint Michael College of Caraga

**Membership/Affiliations**

ORGANIZATION	POSITIONS HELD
N/A	

**G. Curriculum Vitae**

Name : Rom Jones Y. Salem

Address : Barangay 2, Nasipit, Agusan del Norte

Email : [rjsgwapo123@gmail.com](mailto:rjsgwapo123@gmail.com)

**Personal Information**

Sex : Male

Date of birth : May 20, 2003

Height : 163cm

Weight : 63 kg

Civil Status : Single

**Educational Attainment**

Elementary : Nasipit Central Elementary School

Secondary : Nasipit National Vocational School

Tertiary : Saint Michael College of Caraga

**Membership/Affiliations**

ORGANIZATION	POSITIONS HELD
N/A	



## CERTIFICATE OF TECHNOLOGY BASED ASSESSMENT



Saint Michael College of Caraga  
Brgy. 4, Nasipit, Agusan del Norte, Philippines  
Tel. Nos. +63 085 343-3251 / +63 085 283-3113  
[www.smccnasipit.edu.ph](http://www.smccnasipit.edu.ph)



## TEST RESULT

TECHNOLOGY-BASED QUALITY ASSURANCE

Date of Test	MAY 16, 2025	Control number	<b>RBSIT124004</b>
Research Title	<b>ONLINE REQUEST OF CREDENTIALS FOR SAINT MICHAEL COLLEGE OF CARAGA</b>		
Author (s)	1 RICEL ADA S. VILLARIN	ORCID No.	0009-0002-6584-0222
	2 ANGELO M. HINAUTAN	ORCID No.	0009-0007-1796-8930
	3 ANGEL MAE L. BAGUIO	ORCID No.	0009-0008-0448-7596
	4 ROM JONES Y. SALEM	ORCID No.	0009-0000-2929-6202
Co-author (s)	5 JAY A. BONGADO	ORCID No.	0009-0001-1581-3114
Co-author (s)	1 JESSIE S. MAHINAY	ORCID No.	0009-0005-4827-9033

Contact Person	RICEL ADA S. VILLARIN				
Address	D-6 CAMAGONG, NASIPIT, ADN				
Mobile Number	'09060088210				

Test subject	Chapters 1-5				
Total number of words	20,150	Sentences	1,215	Characters	141,710

	Test Area	Score	Required	Remarks
(1) Plagiarism	98	90%	PASSED	
(2) Grammar	98	95%	PASSED	
(3) Readability Score	25	25	PASSED	
(4) Reliability ( <i>Test Instrument only</i> )	N/A	N/A		

## FINDING(S) &amp; RECOMMENDATION(S)

SEE HUMAN GRAMMARIAN FOR THE REVISE SENTENCES

Evaluated by:

**MICHELLE ANN G. LUCINO**  
 Signature over Printed Name

Approved by:

**KENNETH IAN B. BARRERA, MA**  
 Head of Research

Member:





Saint Michael College of Caraga  
Atupan St., Nasipit, Agusan del Norte, Philippines



## Certificate of Human Grammarians

This is to certify that the research titled

Online Request of Credentials for Saint Michael College of Caraga

with authors

Villarin, Ridel Ada S., Bagwio, Angel Mae L., Hinautan, Angelo M., Bongado, Jay A.,  
Salem, Rom Jones Y.

had passed the evaluation by the undersigned as the **HUMAN GRAMMARIAN** and  
recommended for approval.

This certificate is given on the 20th day of May 2025 at Research and  
Instructional Innovation Department.

*Quinnayn*  
WENDELL B. GONZAGA

Human Grammarians  
Saint Michael College of Caraga

Member:



PHILIPPINE  
SOCIETY FOR  
QUALITY, INC.  
Since 1969

info@smccnasipit.edu.ph



---

### CERTIFICATE OF AUTHENTIC AUTHORSHIP

This is to certify that I/we, the undersigned author(s), have solely and honestly written the entire paper titled: "**Online Request of Credentials for Saint Michael College of Caraga**" All contents, ideas, data, and analysis presented in this paper are the results of my/our own diligent work and intellectual effort. The paper has been crafted to the best of my/our abilities, in full compliance with academic integrity standards and ethical research practices.

I/We further declare that:

1. This paper is free from any form of plagiarism or copyright infringement.
2. All referenced materials have been properly cited, and due credit has been given to the original authors and sources.
3. The work does not contain any falsified or fabricated data or information.

I/We acknowledge that if any rules or laws related to copyright, plagiarism, or other related legal or ethical standards are violated, Saint Michael College of Caraga shall not be held accountable for my/our actions. Any legal, academic, or ethical consequences arising from such violations shall solely be my/our responsibility.

Signed this

Date: May 5, 2025

Location: Saint Michael College of Caraga

Author(s):

RICEL ADA S. VILLARIN

Author

ANGEL MAE L. BAGUIO

Author

JAY A. BONGADO

Author

ANGELO M. HINAUTAN

Author

ROM JONES Y. SALEM

Author