A blue and white logo

Description automatically generated

**VIETNAM NATIONAL UNIVERSITY OF HOCHIMINH CITY**

**THE INTERNATIONAL UNIVERSITY**

**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING**

**Software Engineer Final Report**

**ONLINE ART GALLERY**

**Lecturer: N.T.T.Loan**

**N.Q.Phu**

Group: 5

Class: Software Engineering\_S1\_2023-24\_G01

Submitting Date: 21/12/2023

Member:

|  |  |
| --- | --- |
| ITITIU20282 | Đỗ Minh Quân |
| ITCSIU21034 | Huỳnh Ngọc Anh Thư |
| ITITIU20278 | Nguyễn Bá Phúc |
| ITITIU20259 | Trần Tuấn Nghiệp |
| ITITIU20140 | Trương Tấn Phát |
| ITITIU20214 | Nguyễn Quang Huy |
| ITITIU20171 | Nguyễn Cần |
| ITITIU20226 | Tạ Vĩ Khang |
| ITITIU21107 | Nguyễn Minh Thông |

TABLE OF CONTENTS

[1 EXECUTIVE SUMMARY 8](#_Toc156138787)

[**1.1** **Overview of the Project: Online Art Gallery:** 8](#_Toc156138788)

[**1.2** **Objectives and Scopes:** 8](#_Toc156138789)

[1.2.1 Objectives: 8](#_Toc156138790)

[1.2.2 Scopes: 8](#_Toc156138791)

[**1.3** **Key Achievements and Milestones:** 9](#_Toc156138792)

[**1.4** **Final Outcomes:** 9](#_Toc156138793)

[2 CHAPTER I: INTRODUCTION 11](#_Toc156138794)

[**2.1** **Background of the Project:** 11](#_Toc156138795)

[**2.2** **Context of the Project:** 11](#_Toc156138796)

[**2.3** **Goals and Objectives:** 11](#_Toc156138797)

[2.3.1 Goals: 11](#_Toc156138798)

[2.3.2 Objectives: 12](#_Toc156138799)

[**2.4** **Importance and Relevance:** 12](#_Toc156138800)

[**2.5** **Report’s Structure Overview:** 13](#_Toc156138801)

[3 CHAPTER II: PROJECT PLANNING 15](#_Toc156138802)

[**3.1** **Timeline and Milestones:** 15](#_Toc156138803)

[3.1.1 Project Kick-off: 15](#_Toc156138804)

[3.1.2 Requirements Gathering: 15](#_Toc156138805)

[3.1.3 Design and Architecture: 15](#_Toc156138806)

[3.1.4 Development and Implementation: 15](#_Toc156138807)

[3.1.5 Testing and Quality Assurance: 15](#_Toc156138808)

[3.1.6 Deployment and Implementation: 15](#_Toc156138809)

[**3.2** **Members’ Roles and Responsibilities:** 15](#_Toc156138810)

[**3.3** **Resource Allocation:** 16](#_Toc156138811)

[3.3.1 Budget: 16](#_Toc156138812)

[3.3.2 Tools: 17](#_Toc156138813)

[3.3.3 Software: 17](#_Toc156138814)

[3.3.4 Hardware: 18](#_Toc156138815)

[**3.4** **Risk Assessment and Mitigation Strategies:** 18](#_Toc156138816)

[4 CHAPTER III: REQUIREMENT ANALYSIS 21](#_Toc156138817)

[**4.1** **Requirements Detailed Description:** 21](#_Toc156138818)

[4.1.1 User-Friendly Online Platform: 21](#_Toc156138819)

[4.1.2 Artist Profile Management: 21](#_Toc156138820)

[4.1.3 Foundational Website Structure: 21](#_Toc156138821)

[4.1.4 User Interface Design: 21](#_Toc156138822)

[4.1.5 Continuous Evolution and Adaptation: 21](#_Toc156138823)

[4.1.6 Technical Resilience: 21](#_Toc156138824)

[4.1.7 Resource Optimisation: 21](#_Toc156138825)

[4.1.8 Security and Privacy Measures: 21](#_Toc156138826)

[4.1.9 Usability and Accessibility: 21](#_Toc156138827)

[4.1.10 Advanced Search and Filtering: 21](#_Toc156138828)

[4.1.11 Marketing and Outreach Integration: 22](#_Toc156138829)

[**4.2** **User’s Stories, Use Cases:** 22](#_Toc156138830)

[4.2.1 User’s Stories: 22](#_Toc156138831)

[4.2.2 Use Cases: 22](#_Toc156138832)

[**4.3** **Requirements’ Changes or Updates during the Progress:** 29](#_Toc156138833)

[4.3.1 Enhanced Search Functionality: 29](#_Toc156138834)

[4.3.2 Extended Artist Profile Features: 29](#_Toc156138835)

[4.3.3 Introduction of Virtual Art Exhibitions: 29](#_Toc156138836)

[4.3.4 Incorporation of Accessibility Features: 29](#_Toc156138837)

[5 CHAPTER IV: DESIGN AND ARCHITECTURE 30](#_Toc156138838)

[**5.1** **System’s Architecture Overview:** 30](#_Toc156138839)

[5.1.1 Backend Overview: 30](#_Toc156138840)

[5.1.2 Frontend Overview: 33](#_Toc156138841)

[**5.2** **Diagram or Schematics Illustrations:** 36](#_Toc156138842)

[5.2.1 UML Diagrams: 36](#_Toc156138843)

[5.2.2 System Diagram: 49](#_Toc156138844)

[**5.3** **Key Design Decisions and Justifications:** 54](#_Toc156138845)

[5.3.1 Technology Diagram: 54](#_Toc156138846)

[5.3.2 Deployment Diagram: 54](#_Toc156138847)

[5.3.3 Architecture Diagram: 55](#_Toc156138848)

[6 CHAPTER V: DEVELOPMENT 56](#_Toc156138849)

[**6.1** **Development Process’s Description:** 56](#_Toc156138850)

[6.1.1 Project Initiation: 56](#_Toc156138851)

[6.1.2 Planning: 56](#_Toc156138852)

[6.1.3 Design: 56](#_Toc156138853)

[6.1.4 Development/Implementation: 56](#_Toc156138854)

[6.1.5 Testing: 56](#_Toc156138855)

[6.1.6 User Feedback and Iteration: 56](#_Toc156138856)

[6.1.7 Deployment/Release: 56](#_Toc156138857)

[6.1.8 Post-Launch Support and Maintenance: 56](#_Toc156138858)

[6.1.9 Legal Compliance: 56](#_Toc156138859)

[**6.2** **Programming Languages, Frameworks and Tools:** 57](#_Toc156138860)

[6.2.1 IDE : 57](#_Toc156138861)

[6.2.2 Language: 57](#_Toc156138862)

[6.2.3 Support: 57](#_Toc156138863)

[**6.3** **Challenges:** 57](#_Toc156138864)

[6.3.1 Integration Issues: 57](#_Toc156138865)

[6.3.2 Scalability Concerns: 57](#_Toc156138866)

[6.3.3 Security Hole: 57](#_Toc156138867)

[6.3.4 Cooperation Issues: 57](#_Toc156138868)

[6.3.5 Version Control Conflict: 57](#_Toc156138869)

[6.3.6 User Experience (UX) Challenges: 57](#_Toc156138870)

[6.3.7 Performance Bottlenecks: 58](#_Toc156138871)

[6.3.8 Data Management Issues: 58](#_Toc156138872)

[6.3.9 Legal and Compliance Issues: 58](#_Toc156138873)

[**6.4** **Code Management and Version Control:** 58](#_Toc156138874)

[6.4.1 GitHub: Version Control and Collaboration 58](#_Toc156138875)

[6.4.2 Google Drive: Backup and Image Storage 58](#_Toc156138876)

[6.4.3 Zalo: Team Communication 59](#_Toc156138877)

[6.4.4 Conclusion: 59](#_Toc156138878)

[7 CHAPTER VI: TESTING AND QUALITY ASSURANCE 60](#_Toc156138879)

[**7.1** **Testing Methodologies:** 60](#_Toc156138880)

[**7.2** **Results and Bug Tracking:** 60](#_Toc156138881)

[7.2.1 Test Case 1: 60](#_Toc156138882)

[7.2.2 Test Case 2: 62](#_Toc156138883)

[7.2.3 Test Case 3: 63](#_Toc156138884)

[7.2.4 Test Case 4: 65](#_Toc156138885)

[7.2.5 Test Case 5: 66](#_Toc156138886)

[**7.3** **Quality Assurance Measures:** 67](#_Toc156138887)

[8 CHAPTER VII: DEPLOYMENT AND IMPLEMENTATION 69](#_Toc156138888)

[**8.1** **Deployment Process Details:** 69](#_Toc156138889)

[**8.2** **Rollout Strategy and Timeline:** 69](#_Toc156138890)

[8.2.1 Preparation: 69](#_Toc156138891)

[8.2.2 Soft Launch: 69](#_Toc156138892)

[8.2.3 Public Launch: 69](#_Toc156138893)

[**8.3** **User Training and Onboarding:** 69](#_Toc156138894)

[**8.4** **Post-deployment Support Plan:** 70](#_Toc156138895)

[9 CHAPTER VIII: USER DOCUMENTATION 71](#_Toc156138896)

[**9.1** **User’s Manuals or Guides:** 71](#_Toc156138897)

[9.1.1 Getting Started: 71](#_Toc156138898)

[9.1.2 User Profiles: 71](#_Toc156138899)

[**9.2** **System Requirements for End-users:** 72](#_Toc156138900)

[**9.3** **Troubleshooting Tips and FAQs:** 72](#_Toc156138901)

[9.3.1 Troubleshooting Tips: 72](#_Toc156138902)

[9.3.2 Frequently Asked Questions (FAQs): 72](#_Toc156138903)

[10 CHAPTER IX: PROJECT EVALUATION: 74](#_Toc156138904)

[**10.1** **Evaluation Criteria:** 74](#_Toc156138905)

[10.1.1 Scope: 74](#_Toc156138906)

[10.1.2 Quality: 74](#_Toc156138907)

[10.1.3 Time: 74](#_Toc156138908)

[10.1.4 Cost: 74](#_Toc156138909)

[**10.2** **Goals-checking Process:** 74](#_Toc156138910)

[**10.3** **Users’ Feedback and Satisfaction:** 75](#_Toc156138911)

[10.3.1 Introduction: 75](#_Toc156138912)

[10.3.2 Overall Satisfaction: 75](#_Toc156138913)

[10.3.3 Login Interface Experience: 76](#_Toc156138914)

[10.3.4 User’s Website Experience: 77](#_Toc156138915)

[10.3.5 Artist’s Website Experience: 77](#_Toc156138916)

[10.3.6 Overall Assessment: 78](#_Toc156138917)

[10.3.7 General Assessment from Users: 78](#_Toc156138918)

[11 CHAPTER X: LESSONS LEARNED: 80](#_Toc156138919)

[**11.1** **Reflection:** 80](#_Toc156138920)

[11.1.1 Outcome: 80](#_Toc156138921)

[11.1.2 Time: 80](#_Toc156138922)

[11.1.3 Budget: 80](#_Toc156138923)

[11.1.4 Value: 80](#_Toc156138924)

[**11.2** **Further Improvements:** 80](#_Toc156138925)

[**11.3** **Insights:** 80](#_Toc156138926)

[**11.4** **Recommendations:** 80](#_Toc156138927)

[12 CHAPTER XI: CONCLUSION: 82](#_Toc156138928)

[**12.1** **Summary of the Outcomes:** 82](#_Toc156138929)

[**12.2** **Reiteration of Key Achievements:** 82](#_Toc156138930)

[**12.3** **Closing Remarks:** 82](#_Toc156138931)

[13 CHAPTER XII: APPENDICES 83](#_Toc156138932)

[**13.1** **GitHub:** 83](#_Toc156138933)

[**13.2** **Glossary of Terms:** 83](#_Toc156138934)

[**13.3** **Survey Questionnaire:** 87](#_Toc156138935)

[**13.4** **Stakeholder Communication Plan (Draft):** 87](#_Toc156138936)

[**13.5** **Goals and Objectives Review:** 88](#_Toc156138937)

[14 CHAPTER XIII: REFERENCES 89](#_Toc156138938)

[15 ACKNOWLEDGEMENTS 90](#_Toc156138939)

**LIST OF FIGURES**

[*Figure 1: Use Case Diagram 21*](file:////Users/huynhngocanhthu/Desktop/SE/SE%20Final%20Report.docx#_Toc154084691)

[*Figure 2: Express.js Framework 28*](#_Toc154084692)

[*Figure 3: Database Connection 28*](#_Toc154084693)

[*Figure 4: Middleware for User Authentication 28*](#_Toc154084694)

[*Figure 5: CRUD Operations for Category Management 29*](#_Toc154084695)

[*Figure 6: Middleware for User Management 29*](#_Toc154084696)

[*Figure 7: RESTful Interface 30*](#_Toc154084697)

[*Figure 8: Multer for File Uploading 30*](#_Toc154084698)

[*Figure 9: Session Storage 31*](#_Toc154084699)

[*Figure 10: Error Handling 31*](#_Toc154084700)

[*Figure 11: Use Case Diagram 34*](#_Toc154084701)

[*Figure 12: Artist Registration and Artwork Uploading Sequence Diagram 35*](#_Toc154084702)

[*Figure 13: Artwork Purchase Sequence Diagram 37*](#_Toc154084703)

[*Figure 14: Admin Management Sequence Diagram 39*](#_Toc154084704)

[*Figure 15: ER Diagram 41*](#_Toc154084705)

[*Figure 16: Class Diagram 43*](#_Toc154084706)

[*Figure 17: Deployment Diagram 45*](#_Toc154084707)

[*Figure 18: Architecture Diagram 47*](#_Toc154084708)

[*Figure 19: Technology Diagram 48*](#_Toc154084709)

[*Figure 20: Admin Data Flow Diagram 49*](#_Toc154084710)

[*Figure 21: Artist Data Flow Diagram 50*](#_Toc154084711)

[*Figure 22: Buyer Data Flow Diagram 51*](#_Toc154084712)

[*Figure 23: Overall Assessment of the website's user interface 71*](#_Toc154084713)

[*Figure 24: Login Interface Experience 71*](#_Toc154084714)

[*Figure 25: User's Website Experience 72*](#_Toc154084715)

[*Figure 26: Artist's Website Experience 73*](#_Toc154084716)

[*Figure 27: Overall Website's User Interface Assessment 73*](#_Toc154084717)

**LIST OF TABLES**

[*Table 1: Roles and Responsibilities 15*](#_Toc154084649)

[*Table 2: Budget Allocation 16*](#_Toc154084650)

[*Table 3: Risk and Mitigation Strategy 18*](#_Toc154084651)

[*Table 4: UC1 Basic Course 21*](#_Toc154084652)

[*Table 5: UC2 Basic Course 22*](#_Toc154084653)

[*Table 6: UC3 Basic Course 22*](#_Toc154084654)

[*Table 7: UC4 Basic Course 22*](#_Toc154084655)

[*Table 8: UC4 Alternative Course 23*](#_Toc154084656)

[*Table 9: UC5 Basic Course 23*](#_Toc154084657)

[*Table 10: UC6 Basic Course 23*](#_Toc154084658)

[*Table 11: UC7 Basic Course 24*](#_Toc154084659)

[*Table 12: UC8 Basic Course 24*](#_Toc154084660)

[*Table 13: UC9 Basic Course 24*](#_Toc154084661)

[*Table 14: UC10 Basic Course 25*](#_Toc154084662)

[*Table 15: UC10 Alternative Course 25*](#_Toc154084663)

[*Table 16: UC11 Basic Course 25*](#_Toc154084664)

[*Table 17: UC11 Alternative Course 26*](#_Toc154084665)

[*Table 18: UC12 Basic Course 26*](#_Toc154084666)

[*Table 19: UC13 Basic Course 26*](#_Toc154084667)

[*Table 20: UC14 Basic Course 26*](#_Toc154084668)

[*Table 21: Test Case 1 Information 57*](#_Toc154084669)

[*Table 22: Test Case 1 Pre-conditions 57*](#_Toc154084670)

[*Table 23: Test Case 1 Description 59*](#_Toc154084671)

[*Table 24: Test Case 1 Post-conditions 59*](#_Toc154084672)

[*Table 25: Test Case 2 Information 59*](#_Toc154084673)

[*Table 26: Test Case 2 Pre-conditions 59*](#_Toc154084674)

[*Table 27: Test Case 2 Description 60*](#_Toc154084675)

[*Table 28: Test Case 2 Post-conditions 60*](#_Toc154084676)

[*Table 29: Test Case 3 Information 60*](#_Toc154084677)

[*Table 30: Test Case 3 Pre-conditions 61*](#_Toc154084678)

[*Table 31: Test Case 3 Description 62*](#_Toc154084679)

[*Table 32: Test Case 3 Post-conditions 62*](#_Toc154084680)

[*Table 33: Test Case 4 Information 62*](#_Toc154084681)

[*Table 34: Test Case 4 Pre-conditions 62*](#_Toc154084682)

[*Table 35: Test Case 4 Description 63*](#_Toc154084683)

[*Table 36: Test Case 4 Post-conditions 63*](#_Toc154084684)

[*Table 37: Test Case 5 Information 63*](#_Toc154084685)

[*Table 38: Test Case 5 Pre-conditions 63*](#_Toc154084686)

[*Table 39: Test Case 5 Description 64*](#_Toc154084687)

[*Table 40: Test Case 5 Post-conditions 64*](#_Toc154084688)

[*Table 41: Issues and Troubleshooting Tips 68*](#_Toc154084689)

[*Table 42: Glossary of Terms 81*](#_Toc154084690)

# **EXECUTIVE SUMMARY**

## **Overview of the Project: Online Art Gallery:**

The Art Online Gallery project seeks to streamline the art industry by creating a user-friendly platform connecting artists and buyers. Midterm achievements include successfully implementing foundational features, adapting to challenges, and receiving valuable stakeholder feedback. Future goals involve refining the user interface, expanding marketing efforts, and implementing advanced features for a more accessible and enriched art experience.

GitHub link: https://github.com/panadolextra91/Software-Engineering.git

## **Objectives and Scopes:**

### *Objectives:*

* **Develop a User-Friendly Platform:** Create an intuitive and aesthetically appealing online gallery interface that is accessible and engaging for both artists and art purchasers.
* **Efficient Transaction Mechanism:** Implement a streamlined transaction process that facilitates easy management of profits for artists, ensuring a smooth and secure financial interaction.
* **Artist Profile Management:** Provide artists with a user-friendly dashboard to manage their profiles effectively, allowing them to showcase artwork, set prices, and provide detailed descriptions.
* **Continuous Evolution:** Dynamically adapt the project to evolving requirements and stakeholder feedback, fostering improvements in functionalities, user experience, and overall platform performance.
* **Address Technical Challenges:** Overcome technical complexities, including system integrations and compatibility issues, to maintain project momentum and ensure a robust and functional platform.
* **Resource Optimisation:** Address resource and timeline constraints through prioritisation and optimisation, ensuring essential project objectives are met within the stipulated timeframe and budget.
* **Quality Assurance Measures:** Implement robust testing procedures, including code cross-checks, stakeholder discussions, and flexible resource allocation, to ensure optimal program performance and user satisfaction.
* **Enhance Art Discovery:** Implement features such as an advanced search function and artist profile pages to enhance art discovery, providing users with a comprehensive and enjoyable exploration experience.
* **Expand Marketing Efforts:** Increase platform visibility through marketing strategies, including social media campaigns, search engine optimisation, and collaborations with influencers or art bloggers.
* **Usability Testing:** Conduct ongoing usability testing to gather direct user feedback, leading to continuous improvements based on user preferences and expectations.

### *Scopes:*

* **Platform Development:** Create a centralised online hub for art display, discovery, and transactions.
* **User-Centric Design:** Ensure an aesthetically pleasing and user-friendly interface.
* **Artist Profiles:** Implement a system for artists to effectively showcase their work.
* **Secure Transactions:** Establish a safe and efficient process for art transactions.
* **Adaptability:** Dynamically respond to changing requirements and technologies.
* **Technical Resilience:** Address technical challenges for a robust platform.
* **Resource Optimisation:** Efficiently allocate resources for project execution.
* **Quality Assurance:** Implement rigorous testing procedures for high software quality.
* **Art Exploration:** Integrate features for enhanced discovery of artworks.
* **Marketing and Visibility:** Expand efforts to increase platform visibility.
* **Usability Testing:** Conduct ongoing tests for continuous user-driven improvements.

## **Key Achievements and Milestones:**

* **Foundational Website Structure:** Successfully laid the groundwork for the online gallery with a robust website structure. This achievement ensures a smooth and cohesive user experience. The milestone marks the completion of the foundational architecture, setting the stage for further development.
* **Artists Profile Management:** Introduced a user-friendly dashboard empowering artists to manage profiles, upload artwork, and set prices with ease. The effective deployment of artist profile management enhances the platform's functionality and overall appeal, creating a seamless experience for both artists and buyers.
* **User Interface Development:** Meticulously crafted an engaging user interface, including modules like "Home," "User Profile," and "Art Listings." The completed user interface development prioritises navigational ease and aesthetic appeal, contributing to a visually pleasing and intuitive platform.
* **Continuous Evolution:** Dynamically adapted the project to meet emerging requirements and stakeholder feedback. This ongoing evolution ensures improved functionalities and an enhanced user experience. This milestone signifies the project's agility in responding to changing needs.
* **Adaptation and Adjustments:** Proactively made strategic deviations from the initial plan to align the project with evolving needs. This adjustment process reflects the project's commitment to continuous improvement and responsiveness to stakeholder expectations, ensuring a more refined outcome.
* **Progress Schedule:** Developed and adhered to a progress schedule outlining key milestones such as UI/UX development, database connection, and user interaction modules. The adherence to this schedule signifies the project's disciplined approach, resulting in the timely completion of essential development stages.
* **Challenges Addressed:** Effectively navigated technical hurdles, including system integrations and compatibility issues. This achievement ensures sustained project momentum and resilience in the face of challenges, demonstrating the team's problem-solving capabilities.
* **Resource and Timeline Optimisation:** Addressed limited resources and timeline constraints through strategic prioritisation and optimisation. This milestone showcases the team's ability to efficiently allocate and utilise resources within the specified timeframe, ensuring the project's successful progression.

## **Final Outcomes:**

The Art Online Gallery project has achieved significant milestones, delivering a user-friendly platform for artists and art enthusiasts. The outcomes include a robust website structure, streamlined artist profile management, and an engaging user interface. The project's adaptability and timely progress, as depicted in the outlined schedule, demonstrate its disciplined approach. Overcoming technical challenges and optimising resources ensured a resilient and efficient development process.

The impact of the project is evident in its tangible contributions to the art industry—providing a cost-effective platform for artists and a seamless, enjoyable experience for art buyers. With enhanced features for art discovery and secure transactions, the Art Online Gallery has successfully connected artists with a broader audience, achieving its initial objectives and scopes.

# **CHAPTER I: INTRODUCTION**

## **Background of the Project:**

The Art Online Gallery project emerged from the recognition of prevalent challenges within the traditional art industry. In the contemporary landscape, artists faced financial barriers to showcasing their work, resorting to costly exhibitions or museum displays. Simultaneously, art buyers encountered difficulties navigating numerous physical locations in search of desired artworks.

To address these issues, the project aimed to create a centralised online platform that would serve as a dynamic hub for artists and art purchasers. The goal was to streamline the art discovery and transaction process, making it more accessible and cost-effective for both parties. The envisioned platform sought to eliminate financial burdens on artists, provide an engaging space for art enthusiasts and foster a convenient and enjoyable art-buying experience.

The background of the project reflects a commitment to democratising the art world, offering artists an affordable and efficient way to showcase their creations while simplifying the art-buying journey for consumers. This initiative aimed to bridge the gap between artists and buyers in the digital age, leveraging technology to make the art experience more inclusive and enjoyable for a broader audience.

## **Context of the Project:**

The context of the Art Online Gallery project lies in the evolving landscape of the art industry and the transformative power of technology in shaping how art is created, discovered, and acquired. In a traditional setting, artists often face challenges in reaching a wider audience due to the constraints of physical galleries and exhibitions. Simultaneously, art enthusiasts grappled with the time-consuming and often limited nature of in-person art exploration.

Against this backdrop, the project sought to capitalise on the digital era's opportunities, where connectivity and accessibility are paramount. The rise of online platforms and the increasing prevalence of e-commerce presented a unique opportunity to revolutionise the art ecosystem. The project aimed to create a virtual space that transcended geographical limitations, providing artists with a cost-effective means to showcase their work, and offering buyers a diverse and convenient platform for art exploration.

In the context of an increasingly digital and interconnected world, the Art Online Gallery project aimed to redefine how art is experienced and acquired. By leveraging technology, the project aimed to democratise access to the art world, fostering a more inclusive environment where artists could thrive, and art enthusiasts could engage with a rich tapestry of creations from the comfort of their digital devices. The context underscored the project's commitment to bridging the gap between traditional art practices and the possibilities afforded by the digital age.

## **Goals and Objectives:**

### *Goals:*

The overarching goals centre around democratising access to the art world, streamlining art transactions, enhancing artist visibility, fostering adaptability and evolution, ensuring technical resilience, and optimising resources.

* **Democratise Art Access:** Create an inclusive online platform for artists and art enthusiasts, making art more accessible and affordable for everyone.
* **Streamline Art Transactions:** Simplify and secure the art-buying process for consumers while ensuring artists can manage profits seamlessly.
* **Enhance Artist Visibility:** Empower artists with a user-friendly dashboard for effective profile management, showcasing their work effortlessly.
* **Adaptability and Evolution:** Foster a culture of continuous improvement, adapting dynamically to changing needs and feedback.
* **Technical Resilience:** Overcome technical challenges to maintain project momentum and deliver a reliable online gallery.
* **Resource Optimisation:** Efficiently use budget, tools, and resources to maximise impact within specified constraints.

### *Objectives:*

The foundational objective centres around establishing a robust website structure. Aligned with the goal of democratising art access, this objective involves developing the foundational architecture of the online gallery. The intention is to create an environment that ensures a cohesive user experience, setting the stage for further development and user engagement.

* **Establish a Robust Website Structure:** Develop a strong foundation for the online gallery, ensuring a smooth user experience.
* **Implement Artist Profile Management:** Provide artists with a user-friendly dashboard for effective profile management and artwork showcasing.
* **Craft an Engaging User Interface:** Design an intuitive interface for an enjoyable art exploration experience.
* **Ensure Continuous Evolution:** Dynamically adapt to meet emerging requirements and stakeholder feedback for ongoing improvement.
* **Address Technical Hurdles:** Overcome technical challenges, ensuring a robust and reliable online gallery.
* **Optimise Resources and Timelines:** Efficiently allocate resources, staying within the specified timeframe and budget for a successful project outcome.

## **Importance and Relevance:**

* **Democratising Art Access:** By providing an inclusive online platform, the project democratises access to the art world. It breaks down financial barriers for artists, allowing them to showcase their work affordably. Simultaneously, it offers a diverse audience the chance to explore art conveniently, fostering a more inclusive cultural environment.
* **Streamlining Art Transactions:** The project streamlines art transactions, simplifying the buying process for art enthusiasts. This not only benefits consumers but also ensures that artists can efficiently manage their profits. The secure transaction mechanism adds a layer of trust, contributing to a positive and reliable online art marketplace.
* **Empowering Artists:** The emphasis on enhancing artist visibility through a user-friendly dashboard empowers artists to control their narratives. This tool allows them to showcase their creations, set prices, and provide detailed descriptions, giving artists agency in promoting their work and building a digital presence.
* **Adaptability and Evolution:** In a rapidly changing digital landscape, the project's commitment to adaptability and continuous evolution is highly relevant. This approach ensures that the online platform remains responsive to emerging trends, user preferences, and technological advancements, sustaining its long-term relevance.
* **Technical Resilience:** In addressing technical challenges, the project demonstrates its relevance in a technology-driven era. Overcoming hurdles such as system integrations and compatibility issues ensures the development of a reliable, cutting-edge online gallery that aligns with contemporary technological standards.
* **Resource Optimisation:** Efficient resource allocation is not just a practical necessity but also a relevant approach to demonstrating responsible project management. By optimising budget, tools, and resources, the project maximises impact while remaining fiscally responsible and sustainable.

## **Report’s Structure Overview:**

**Title Page:**

* University Name
* School name
* Title of the Report.
* Project Name, ID
* Names, IDs of Team Members.
* Date.

**Table of Contents:**

List of Sections and Subsections with Page Numbers.

**List of Figures:**

List of Figures with Page Numbers.

**List of Tables:**

List of Tables with Page Numbers.

**Executive Summary:**

* Project’s Overview
* Objectives and Scopes
* Key Achievements and Milestones
* Project’s Final Outcomes
* Outstanding Issues or Recommendations

**Chapter I: Introduction:**

* Background and Context of the project.
* Goals and Objectives
* Importance and Relevance
* Project’s Structure Overview

**Chapter II: Project Planning:**

* Timeline and Milestones
* Members’ Roles and Responsibilities
* Resource Allocation
* Risk Assessment and Mitigation Strategies

**Chapter III: Requirement Analysis:**

* Requirements Detailed Description
* User’s Stories, Use Cases
* Requirements’ Changes or Updates during the Progress

**Chapter IV: Design and Architecture:**

* System’s Architecture Overview
* Diagram or Schematics Illustrations
* Key Design Decisions and Justifications

**Chapter V: Development:**

* Development Process’s Description
* Programming languages, Frameworks and Tools
* Challenges
* Code Management and Version Control

**Chapter VI: Testing and Quality Assurance:**

* Testing Methodologies
* Results and Bug Tracking
* Quality Assurance Measures

**Chapter VII: Deployment and Implementation:**

* Deployment Process Details
* Rollout Strategy and Timeline
* User Training and Onboarding
* Post-deployment Support Plan

**Chapter VIII: User Documentation:**

* User’s Manuals or Guides
* System Requirements for End-users
* Troubleshooting tips and FAQs

**Chapter IX: Project Evaluation:**

* Evaluation Criteria
* Goals-checking Process
* Users’ Feedback and Satisfaction

**Chapter X: Lessons Learned:**

* Reflection
* Insights
* Recommendations

**Chapter XI: Conclusion:**

* Summary of the Outcomes
* Reiteration of Key Achievements
* Closing Remarks

**Chapter XII: Appendices**

**Chapter XIII: References**

**Acknowledgements**

# **CHAPTER II: PROJECT PLANNING**

## **Timeline and Milestones:**

This section provides a high-level view of the timeline and critical milestones for your software project from October 1st to December 20th. It breaks down the project into specific phases and highlights key activities within each phase. Important milestones such as project kick-off, requirements gathering, design, development, testing, and deployment are mentioned. Having a clear timeline and milestones helps ensure project progress and effective project management.

### Project Kick-off:

* **Date:** October 1, 2023
* **Description:** The official start of the project. It involves assembling the project team, communicating project objectives, and defining roles and responsibilities.
* **Deliverable:** Project kick-off meeting minutes and project plan.

### Requirements Gathering:

* **Date:** October 1 - October 15, 2023
* **Description**: This phase focuses on gathering and documenting the project requirements from stakeholders. It involves conducting interviews, workshops, and surveys to understand the needs and expectations for the software.
* **Deliverable:** Requirements document.

### Design and Architecture:

* **Date:** October 16 - November 5, 2023
* **Description:** In this phase, the system architecture and software design are created. It includes defining the system components, database structure, and user interface. The design should align with the gathered requirements.
* **Deliverable:** System architecture and software design document.

### Development and Implementation:

* **Date:** November 6 - December 10, 2023
* **Description:** The development and coding phase begins, where the software is built according to the design specifications. The development team writes code, integrates necessary libraries or frameworks, and ensures proper functionality.
* **Deliverable:** Working software with implemented features.

### Testing and Quality Assurance:

* **Date:** December 11 - December 15, 2023
* **Description:** This phase involves rigorous testing of the software to identify and fix any defects or issues. Different testing techniques such as unit testing, integration testing, and system testing are employed to ensure the software meets the requirements.
* **Deliverable:** Test cases, test results, and resolved issue reports.

### Deployment and Implementation:

* **Date:** December 16 - December 20, 2023
* **Description:** The software is prepared for deployment to the production environment. This phase includes setting up the necessary infrastructure, configuring the software, and conducting final checks to ensure a smooth deployment.
* **Deliverable:** Deployed and operational software.

## **Members’ Roles and Responsibilities:**

|  |  |
| --- | --- |
| **Project Manager, Report Generator** | Huỳnh Ngọc Anh Thư |
| **Developer** | Đỗ Minh Quân |
| Developer, Report Generator | Nguyễn Bá Phúc |
| Developer | Trần Tuấn Nghiệp |
| Developer | Trương Tấn Phát |
|  | Huỳnh Thiên Tường |
| Developer | Nguyễn Cần |
| Developer | Nguyễn Minh Thông |
| **Report Generator** | Nguyễn Quang Huy |
| **Diagram Generator, Report Generator** | Tạ Vĩ Khang |

*Table 1: Roles and Responsibilities*

## **Resource Allocation:**

This section focuses on the allocation of resources for your software project. It covers various aspects such as budget, tools, software, and hardware required for successful project execution. The budget allocation specifies the total project budget and how it is distributed among different project activities. The tools and software allocation highlights the specific tools and software needed for development, design, testing, and deployment. The hardware allocation refers to the necessary hardware equipment for development, testing, and deployment. Proper resource allocation ensures that the project has the necessary resources to meet its objectives effectively and efficiently.

### *Budget*:

|  |  |  |
| --- | --- | --- |
| **Budget Category** | **Description** | **Allocation** |
| Development and Implementation | This includes the cost of hiring developers, software engineers, and any external resources required for coding and implementation. | $200,000 |
| Design and Architecture | This covers the expenses for graphic designers, UI/UX experts, and software architects involved in creating the system architecture and designing the user interface. | $25,000 |
| Testing and Quality Assurance | This budget is allocated for testing activities, such as hiring QA testers, implementing automated testing tools, and conducting performance and security testing. | $20,000 |
| Deployment and Implementation | This includes the cost of setting up the necessary infrastructure, configuring servers or cloud services, and ensuring a smooth deployment process. | $30,000 |
| User Training and Onboarding | This budget is allocated for training end-users on how to use the software effectively and providing necessary documentation or resources for onboarding. | $10,000 |
| Post-deployment Support | This covers ongoing maintenance and support activities after the software is deployed, including bug fixes, updates, and user support. | $15,000 |
| **Total:** | | **$300,000** |

Table 2: Budget Allocation

### *Tools:*

#### a. Development Tools:

* Integrated Development Environment (IDE): Visual Studio Code, IntelliJ IDEA, Eclipse.
* Code editors: Sublime Text, Atom, Notepad++.
* Version control systems: Git, SVN, Mercurial.
* Collaboration tools: Slack, Microsoft Teams, Trello.

#### b. Design Tools:

* Graphic design software: Adobe Creative Suite (Photoshop, Illustrator), Sketch, Figma.
* Wireframing tools: Balsamiq, Adobe XD.
* Prototyping tools: InVision, Adobe XD.

#### c. Testing Tools:

Automated testing frameworks: Selenium, Cypress, Appium.

Bug tracking systems: Jira, Bugzilla, Redmine.

Test management tools: TestRail, Zephyr, qTest.

#### d. Deployment Tools:

* Server infrastructure: AWS, Azure, Google Cloud Platform.
* Containerization tools: Docker, Kubernetes.
* Continuous Integration/Continuous Deployment (CI/CD) tools: Jenkins, CircleCI, GitLab CI/CD.

### *Software:*

#### a. Development Software:

* Programming languages: Python, JavaScript, Java, C#, Ruby.
* Frameworks: Django, React, Angular, Spring Boot, Ruby on Rails.
* Libraries: jQuery, Bootstrap, TensorFlow, PyTorch.
* Integrated Development Environments (IDEs): Visual Studio Code, PyCharm, Eclipse, IntelliJ IDEA.

#### b. Design Software:

* Graphic design software: Adobe Creative Suite (Photoshop, Illustrator), Sketch, Figma.
* Prototyping software: InVision, Adobe XD.

#### e. Testing Software:

* Automated testing frameworks: Selenium, Cypress, Appium.
* Unit testing frameworks: JUnit, NUnit, Jest.
* Performance testing tools: Apache JMeter, Gatling, Locust.

#### f. Deployment Software:

* Containerization tools: Docker, Kubernetes.
* Infrastructure as Code (IaC) tools: Terraform, AWS CloudFormation.
* Deployment automation tools: Jenkins, CircleCI, GitLab CI/CD.

### *Hardware:*

#### a. Development Hardware:

* Computers or laptops with sufficient processing power and memory.
* High-resolution displays for efficient coding and design work.

#### b. Testing Hardware:

Devices for cross-browser and cross-platform testing, such as smartphones, tablets, and various operating systems.

#### c. Deployment Hardware:

Servers or cloud infrastructure as per the deployment strategy, such as virtual machines or containers.

## **Risk Assessment and Mitigation Strategies:**

|  |  |  |
| --- | --- | --- |
| **Risk** | **Description** | **Mitigation Strategy** |
| Technical Challenges | Unforeseen technical complexities may arise during the implementation phase, impacting the project's timeline and functionality. | * Regular Technical Reviews: Conduct regular technical reviews to identify and address potential challenges early in the development process. * Prototyping: Develop prototypes for critical functionalities to uncover and resolve technical issues before full implementation. * Collaboration with Experts: Engage external technical experts for consultation on complex integration or compatibility issues. |
| Resource Limitations | Limited availability of skilled team members, budget constraints, or inadequate infrastructure may hinder the project's progress. | * Prioritised Resource Allocation: Allocate resources based on project priorities, ensuring that essential roles and functions are adequately supported. * Continuous Monitoring of Budget: Regularly monitor the budget to identify and address any deviations promptly. * Capacity Building: Provide training sessions to enhance the skill sets of team members and fill any knowledge gaps. |
| Timeline Constraints | Delays in project timelines can result from unforeseen challenges, impacting the overall project schedule. | * Buffer Time: Incorporate buffer time in the project timeline to accommodate unexpected delays without affecting critical milestones. * Regular Progress Reviews: Conduct frequent progress reviews to identify potential delays early, allowing for timely adjustments. * Agile Methodology: Adopt an agile development approach to adapt to changes efficiently and iteratively. |
| Security | Security vulnerabilities in the platform may compromise user data or lead to unauthorised access. | * Regular Security Audits: Conduct regular security audits to identify and address vulnerabilities promptly. * Encryption Protocols: Implement robust encryption protocols to safeguard user data during transmission and storage. * User Education: Educate users on security best practices, such as using strong passwords and enabling two-factor authentication. |
| User’s Experience | Users may face challenges in adopting and navigating the new platform, affecting user satisfaction and engagement. | * User-Friendly Design: Prioritise a user-friendly interface through effective UI/UX design to enhance user adoption. * User Training Resources: Provide comprehensive user manuals, guides, and tutorials to assist users in navigating the platform. * Feedback Mechanism: Implement a feedback mechanism to gather user input, allowing for continuous improvement based on user experiences. |
| Market Competition | Intense competition in the online art gallery market may impact the platform's ability to attract and retain users. | * Unique Selling Proposition (USP): Clearly define and promote the unique features and advantages of the Art Online Gallery to differentiate it from competitors. * Market Research: Continuously monitor market trends and user preferences to adapt the platform to changing demands. * Strategic Partnerships: Explore partnerships with art-related influencers, organisations, or events to enhance visibility and credibility. |

Table 3: Risk and Mitigation Strategy

# **CHAPTER III: REQUIREMENT ANALYSIS**

## **Requirements Detailed Description:**

### *User-Friendly Online Platform:*

Develop an interface that ensures intuitive navigation for both artists and art enthusiasts, enhancing the overall user experience. Incorporate user-centric design principles, focusing on accessibility and ease of use.

### *Artist Profile Management:*

Implement a comprehensive artist profile management system, allowing artists to create, edit, and manage profiles seamlessly. Enable artists to upload, curate, and present their artwork, set pricing, and provide detailed descriptions to showcase their unique styles.

Efficient Transaction Mechanism:

Create a secure and efficient transaction system for art purchases. Integrate payment processing, order management, and transparent communication channels between buyers and artists. Prioritise user-friendly interactions and build trust through transparent transaction processes.

### *Foundational Website Structure:*

Establish a robust foundational architecture for the online art gallery. Develop core components such as the homepage, artist profile pages, and art listing functionalities. Ensure a cohesive and visually appealing structure that aligns with the overall design principles.

### *User Interface Design:*

Craft an engaging and visually appealing user interface with specific modules such as "Home," "User Profile," and "Art Listings." Prioritise both navigational ease and aesthetic appeal, taking user feedback and design principles into consideration for an optimal user experience.

### *Continuous Evolution and Adaptation:*

Implement an agile approach, allowing for continuous evolution and adaptation to changing requirements and stakeholder feedback. Proactively respond to emerging trends, technology advancements, and user preferences throughout the project lifecycle.

### *Technical Resilience:*

Ensure technical resilience by addressing system integrations and compatibility issues. Overcome technical challenges promptly to maintain project momentum and deliver a robust, reliable, and technologically sound online gallery.

### *Resource Optimisation:*

Efficiently allocate and utilise project resources, including budget, tools, software, and hardware. Implement resource optimisation strategies to ensure a balance between maximising impact and staying within specified constraints.

### *Security and Privacy Measures:*

Implement robust security measures to safeguard user data, financial transactions, and sensitive information. This includes encryption protocols, secure payment gateways, and measures to prevent unauthorised access, ensuring a secure online environment.

### *Usability and Accessibility:*

Prioritise usability and accessibility, ensuring the online gallery is accessible to users with diverse needs and preferences. Implement features that enhance inclusivity, providing a user-friendly environment for a broad audience.

### *Advanced Search and Filtering:*

Implement advanced search and filtering functionalities to enhance art discovery. Develop features such as keyword search, artist-specific searches, and a variety of filters to cater to diverse user preferences and optimise the art exploration experience.

### *Marketing and Outreach Integration:*

Integrate marketing and outreach features to promote the online gallery. Develop tools for social media sharing, search engine optimisation, and collaborations with influencers or art bloggers to increase platform visibility and attract a wider audience.

## **User’s Stories, Use Cases:**

### *User’s Stories:*

#### a. As an Administrator:

* I want to manage user accounts and profiles to ensure a secure and regulated platform.
* I want to monitor and track transactions for transparency and security purposes.
* I want to receive reports on platform analytics to assess user engagement and popular artwork.

#### b. As an Artist:

* I want to easily create and manage my profile so that I can showcase my artwork effectively.
* I want to upload and curate my artwork with detailed descriptions and pricing options.
* I want to receive notifications for art purchases to stay informed about my sales.

#### c. As a Buyer:

* I want to browse and explore artwork based on various criteria like artist, style, genre, etc.
* I want to have a personalised user profile to keep track of my favourite artists and buy artwork.
* I want a secure and seamless transaction process when purchasing artwork online.

### Use Cases:

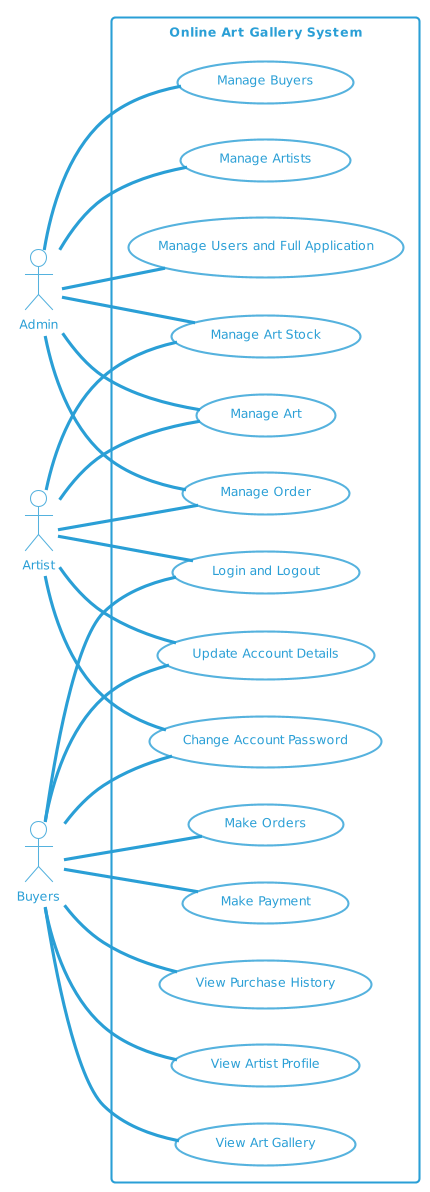


Figure 1: Use Case Diagram

#### a. Manage Buyers:

Identifier: UC1

Inputs: None

Outputs: Lists of registered buyers and their details

Basic Course:

|  |  |
| --- | --- |
| **Actor: Admin** | **System** |
| 1. Open the “Manage Buyers” section. | 1.1 Display the list of registered buyers with their details. |
| 2. Perform actions like viewing, editing, or removing buyer profiles, etc. | 2.1 Display corresponding dashboards for each action. |

*Table 4: UC1 Basic Course*

Preconditions: None

Postconditions: The admin has successfully managed the buyers’ information.

#### b. Manage Artists:

Identifier: UC2

Inputs: None

Outputs: Lists of registered artists and their details

Basic Course:

|  |  |
| --- | --- |
| **Actor: Admin** | **System** |
| 1. Open the “Manage Artists” section. | 1.1 Display the list of registered artists with their details. |
| 2. Perform actions like viewing, editing, or removing artist profiles, etc. | 2.1 Display corresponding dashboards for each action. |

*Table 5: UC2 Basic Course*

Preconditions: None

Postconditions: The admin has successfully managed the artists’ information.

#### c. Manage Users and Full Application:

Identifier: UC3

Inputs: None

Outputs: Full control over user profiles and application settings

Basic Course:

|  |  |
| --- | --- |
| **Actor: Admin** | **System** |
| 1. Open the “Manage User and Application” section. | 1.1 Display options to manage all user profiles and application settings. |
| 2. Perform actions like adding, removing, warning user, etc. | 2.1 Display corresponding dashboards for each action.  2.2 Send an email and a notification to the corresponding user to notify about the action of admin. |
| 3. Adjust the application settings. | 3.1 Display the setting dashboard.  3.2 After the adjustments are made, save it into the system’s database. |

*Table 6: UC3 Basic Course*

Preconditions: None

Postconditions: The admin has successfully managed users and adjusted in the application settings.

#### d. Manage Art Stock:

Identifier: UC4

Inputs: Art stock details (quantity, availability)

Output: Updated art stock information

Basic Course:

|  |  |
| --- | --- |
| **Actor: Admin/Artist** | **System** |
| 1. Open the “Manage Art Stock” section. | 1.1 Display the current art stock details. |
| 2. Update stock information, mark availability, etc. | 2.1 Display corresponding dashboards for each action.  2.2 Save the updated information into the system’s database. |

*Table 7: UC4 Basic Course*

Alternative Course: Art Stock Update Error.

|  |  |
| --- | --- |
| **Actor: Admin/Artist** | **System** |
| 1. Error occurs (e.g., database connection failure) | 1.1 Notify the admin/artist of the error.  Prompt them to try again or contact support. |

*Table 8: UC4 Alternative Course*

Preconditions: The artist is logged into the system.

Postconditions: The art stock information has been successfully updated.

#### e. Manage Art:

Identifier: UC5

Inputs: Art details (description, pricing, genre, etc.) and its image

Outputs: Updated art details and its image on the system

Basic Course:

|  |  |
| --- | --- |
| **Actor: Admin/Artist** | **System** |
| 1. Open the “Manage Art” section. | 1.1 Display the current art details. |
| 2. Update art information, pricing, and other relevant details. | 2.1 Display corresponding dashboards for each action.  2.2 Save the updated information into the system’s database. |

*Table 9: UC5 Basic Course*

Preconditions: The artist is logged into the system.

Postconditions: The art information and its image have been successfully updated.

#### f. Manage Order:

Identifier: UC6

Inputs: Order details

Outputs: Updated order status and payment confirmation

Basic Course:

|  |  |
| --- | --- |
| **Actor: Admin/Artist** | **System** |
| 1. Open the “Manage Order” section. | 1.1 Display the current order details. |
| 2. Update order status, confirm payment, etc. | 2.1 Display corresponding dashboards for each action.  2.2 Save the updated status and confirmation into the system’s database.  2.3 Send the confirmation notification to the buyer’s email and account. |

*Table 10: UC6 Basic Course*

Preconditions: The artist is logged into the system.

Postconditions: The order status and payment confirmation have been successfully updated.

#### g. Login/Logout:

Identifier: UC7

Inputs: Username, password

Outputs: Home page (if successful login), Login page (if unsuccessful login)

Basic Course:

|  |  |
| --- | --- |
| **Actor: Artist/Buyer** | **System** |
| 1. Open the login page. | 1.1 Display the login page. |
| 2. Enter username and password. |  |
| 3. Submit. | 3.1 Check the user’s information.  3.1.1 If successful: Display the home page.  3.1.2 If unsuccessful: Display the login page with an error description. |

*Table 11: UC7 Basic Course*

Preconditions: The artist/buyer is not logged into the system.

Postconditions: The artist/buyer is either successfully logged in, viewing the home page, or the login attempt failed, and they are returned to the login page with an error description.

#### h. Update Account Details:

Identifier: UC8

Inputs: New account details

Outputs: Updated account information

Basic Course:

|  |  |
| --- | --- |
| **Actor: Artist/Buyer** | **System** |
| 1. Open profile page | 1.1 Display the profile page. |
| 2. Open “Update Account Details” section | 2.1 Display the current profile information. |
| 3. Choose the information that they want to edit. | 3.1 Display the editor box. |
| 4. Edit information. |  |
| 5. Submit | 5.1 Save the changes into the system’s database.  5.2 Display the profile page with updated information. |

*Table 12: UC8 Basic Course*

Preconditions: The artist/buyer is logged into the system,

Postconditions: The account information has been successfully updated.

#### i. Change Account Password:

Identifier: UC9

Inputs: Current password, new password

Outputs: Password changed successfully message and the new password is updated in the system’s database

Basic Course:

|  |  |
| --- | --- |
| **Actor: Artist/Buyer** | **System** |
| 1. Open the profile page | 1.1 Display the profile page. |
| 2. Open the “Change Password” section. | 2.1 Display the editor box of current password and new password. |
| 3. Enter current password and new password. | 3.1 Check the current password.  3.1.1 If valid: Change the password, update the system’s database, and send a successful message.  3.1.2 If invalid: Send an error message. |

*Table 13: UC9 Basic Course*

Preconditions: The artist/buyer is logged into the system.

Postconditions: The password has been successfully changed.

#### j. Make Order:

Identifier: UC10

Inputs: Artwork selection, payment details

Outputs: Order confirmation

Basic Course:

|  |  |
| --- | --- |
| **Actor: Buyer** | **System** |
| 1. Select artwork and add it to the cart. | 1.1 Add the artwork to the buyer’s cart.  Update the cart. |
| 2. Open the cart and select the artwork that they want to purchase. | 2.1 Calculate the total price.  2.2 Display the total price of selected artwork. |
| 3. Proceed to the checkout process | 3.1 Display payment details boxes. |
| 4. Enter payment details |  |
| 5. Submit. | 5.1 Check the payment details.  5.1.1 If valid: Confirm the order and send an order confirmation to the buyer.  5.1.2 If invalid: Send an error message. |

*Table 14: UC10 Basic Course*

Alternative Course: Order Cancellation

|  |  |
| --- | --- |
| **Actor: Buyer** | **System** |
| 1. Initiate the cancellation process. | 1.1 Display the list of orders. |
| 2. Select the order that they want to cancel. | 2.1 Verify if the order is eligible for cancellation (not shipped or completed).  2.1.1 If eligible: Update the order status to “Cancelled” and notify the relevant parties.  2.1.2 If ineligible: Send an unsuccessful message. |

*Table 15: UC10 Alternative Course*

Preconditions: The buyer is logged into the system; the selected artwork is available, and the payment method is working.

Postconditions: The order has been successfully placed and the buyer receives an order confirmation.

#### k. Make Payment:

Identifier: UC11

Inputs: Payment details

Outputs: Payment confirmation

Basic Course:

|  |  |
| --- | --- |
| **Actor: Buyer** | **System** |
| 1. Select the payment option during the checkout process. | 1.1 Display the selected payment option. |
| 2. Enter payment details. |  |
| 3. Submit | 3.1 Process the payment.  3.2 Confirms the successful transaction. |

*Table 16: UC11 Basic Course*

Alternative Course: Payment Failure.

|  |  |
| --- | --- |
| **Actor: Buyer** | **System** |
| 1. The payment transaction fails (e.g., declined card, insufficient funds) | 1.1 Notify the buyer of the failure.  1.2 Provide guidance on resolving issue or using an alternative method. |

*Table 17: UC11 Alternative Course*

Preconditions: The buyer is logged into the system and has initiated the payment process.

Postconditions: The payment has been successfully processed and the buyer receives payment confirmation.

#### l. View Purchase History:

Identifier: UC12

Inputs: None

Outputs: List of past purchase records

Basic Course:

|  |  |
| --- | --- |
| **Actor: Buyer** | **System** |
| 1. Open the “View Purchase History” section. | 1.1 Display the list of past purchase records. |

*Table 18: UC12 Basic Course*

Preconditions: The buyer is logged into the system.

Postconditions: The buyer successfully views their purchase history.

#### m. View Artist Profile:

Identifier: UC13

Inputs: Artist selection

Outputs: Artist profile details

Basic Course:

|  |  |
| --- | --- |
| **Actor: Buyer** | **System** |
| 1. Select an artist to view. | 1.1 Display the selected artist’s profiles with details like bio, artwork, etc. |

*Table 19: UC13 Basic Course*

Preconditions: The buyer is logged into the system.

Postconditions: The buyer successfully views the selected artist’s profile.

#### n. View Art Gallery:

Identifier: UC14

Inputs: None

Outputs: Visual representation of the art gallery

Basic Course:

|  |  |
| --- | --- |
| **Actor: Buyer** | **System** |
| 1. Open the “View Art Gallery” section. | 1.1 Display a visual representation of the available artworks. |

*Table 20: UC14 Basic Course*

Preconditions: The buyer is logged into the system.

Postconditions: The buyer successfully views the art gallery.

## **Requirements’ Changes or Updates during the Progress:**

### Enhanced Search Functionality:

* **Change:** Originally, the search functionality was designed to rely solely on keyword matching. However, during the progress of the project, user feedback highlighted the need for more advanced search filters, including filters for art styles, sizes, and prices.
* **Update:** The project team decided to enhance the search functionality to incorporate these additional filters, providing users with a more refined and personalised search experience.

### Extended Artist Profile Features:

* **Change:** Initial requirements outlined basic features for artist profiles, focusing on artwork management. As the project progressed, stakeholders expressed the importance of expanding artist profiles to include personal bios, social media links, and exhibition history.
* **Update:** The project scope was adjusted to accommodate these additional features, contributing to a richer and more informative artist profile experience.

### Introduction of Virtual Art Exhibitions:

* **Change:** Originally, the project focused on individual artwork transactions. However, during the process, the team identified an opportunity to create a more immersive experience by introducing virtual art exhibitions.
* **Update:** The project requirements were modified to incorporate the development of a virtual exhibition feature, allowing users to explore curated collections, and enhancing the overall platform's appeal.

### Incorporation of Accessibility Features:

* **Change:** Initial requirements did not explicitly address accessibility features. However, as the project progressed, there was a recognition of the importance of making the platform accessible to users with disabilities.
* **Update:** Project requirements were updated to include the implementation of accessibility features, such as alternative text for images and keyboard navigation, ensuring inclusivity and compliance with accessibility standards.

# **CHAPTER IV: DESIGN AND ARCHITECTURE**

## **System’s Architecture Overview:**

### Backend Overview:

The backend of the application is developed using Node.js and Express.js, utilising SQLite as the database for storing information. The codebase includes functionalities for user authentication, category management, and basic CRUD operations for categories and users. Additionally, it features API endpoints to interact with the data, providing a RESTful interface.

#### a. Server Setup:

The Express.js framework is used to set up the server, handle routes, and manage middleware.

EJS is employed as the view engine to dynamically render HTML content.



Figure 2: Express.js Framework

#### b. Database Connection:

SQLite3 is utilised as the database engine, and a connection to the 'demo.db' database is established.

A black screen with red green and blue text

Description automatically generated

Figure 3: Database Connection

#### c. User Authentication:

Middleware is implemented to check user authentication before accessing certain routes. Users are required to log in, except for the login and logout routes.

A computer screen with text on it

Description automatically generated

Figure 4: Middleware for User Authentication

#### d. Category Management:

CRUD operations are defined for managing categories, including listing, creating, editing, and deleting categories.

A computer code on a black background

Description automatically generated

Figure 5: CRUD Operations for Category Management

#### e. User Management:

Middleware is implemented to check user authentication before accessing certain routes. Users are required to log in, except for the login and logout routes.

A screen shot of a computer program

Description automatically generated

Figure 6: Middleware for User Management

#### f. API Endpoints:

API routes are defined for categories, users, and other entities. These endpoints return JSON responses and handle actions such as listing, creating, updating, and deleting data.

RESTful conventions are followed in the API design, such as using HTTP methods (GET, POST, PUT, DELETE) for different actions on resources.

A screen shot of a computer program

Description automatically generated

Figure 7: RESTful Interface

#### g. File Uploading:

Multer is utilised for handling file uploads, specifically for category images. The code defines a destination and filename for uploaded images.

A computer code on a black background

Description automatically generated

Figure 8: Multer for File Uploading

#### h. Session Storage:

The node-sessionstorage library is used to store session data, such as user login status and the shopping cart.

A black background with white text

Description automatically generated

Figure 9: Session Storage

#### i. Error Handling:

Basic error handling is implemented throughout the code to log errors and provide appropriate HTTP responses.

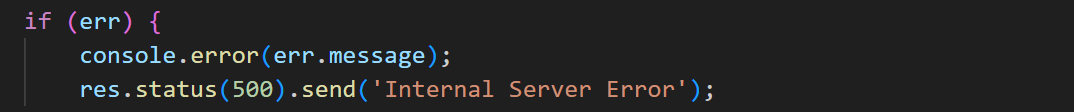


Figure 10: Error Handling

### Frontend Overview:

#### a. Starting Point:

This section will get to what you see first when you download and log in to our link. The starting page of our e-commerce Art Gallery serves as the entry point for users to access their accounts. This report outlines the design and functionality of the login page, implemented using VS Code and the EJS templating language.

i. Login Interface:

The login interface is a critical component of the starting page, allowing users to sign in with their respective credentials. The interface consists of two input fields:

* **Email:** Users are required to enter their email address associated with their accounts.
* **Password:** A secure password input field ensures the confidentiality of user credentials.

The login interface is designed for ease of use, providing a clear and concise layout for users to input their information.

ii. Role-based Authentication:

The system supports three distinct roles: Admin, Customer, and Artist. Upon entering their credentials and pressing the “Login” button, users are redirected to role-specific pages based on their account type.

iii. Implementation Details:

Technologies Used

* **VS Code:** The development environment for coding the project.
* **EJS (Embedded JavaScript):** The templating language used to generate dynamic content on the login page.

The login page is implemented using EJS templates, enhancing code readability and maintainability. The relevant code files are organised in a structured manner within the VS Code project.

iv. Conclusion:

Our website supply project start page is successfully presented for a user-friendly interface login based on role-based authentication. Using VS Code and EJS helps us improve project efficiency and maintainability.

#### b. Role-based Access and Functionalities:

Now, we will provide more detailed information about the differences between roles (Admin, Customers and Artists) so that everyone can better visualise the surface of our website.

##### i. Admin:

1. About:

The "About" section serves as a comprehensive repository of information, detailing the mission, vision, and core values of our e-commerce Art Gallery. Admins can refer to this section for a nuanced understanding of the platform's overarching goals.

2. Category Management:

In the Category Management section, administrators wield the power to curate and maintain the visual content available on the platform. Key functionalities include:

* **View:** Administrators can effortlessly navigate through the Art Gallery, gaining a holistic view of all images uploaded by the diverse community of artists. This overview enables admins to stay informed about the content available to users.
* **Edit and Delete:** To ensure content quality and relevance, admins possess the capability to edit the details or entirely remove any image from the gallery. This feature allows for the continuous refinement of the platform's visual aesthetics.
* **Add Button:** Admins actively contribute to the platform's growth by leveraging the "Add" button, facilitating the seamless inclusion of new and captivating artworks into the Art Gallery.

3. User Management:

The User Management section provides a bird's-eye view of all user accounts registered on the platform. Key features include:

* **User Details:** Admins can access crucial information such as user IDs, names, emails, and roles. This section acts as a centralised hub for overseeing user accounts.
* **No Password Display:** In adherence to stringent security protocols, the User Management section refrains from displaying user passwords, prioritising the confidentiality and privacy of user accounts.
* **Logout:** To conclude a secure session, administrators can effortlessly log out of the system by clicking the "Logout" button. This feature ensures the integrity of admin accounts and mitigates unauthorised access.

##### ii. Customer:

1. Art Gallery

The Art Gallery serves as a vibrant showcase of artistic expressions from talented contributors. Customers enjoy the following functionalities:

* **View:** Customers can leisurely peruse the Art Gallery, immersing themselves in the diverse array of images uploaded by passionate artists. This interactive experience enhances user engagement and appreciation for the arts.
* **Buttons:** Each artwork in the Art Gallery is equipped with two distinct buttons – "View" and "Add to Cart." These buttons empower customers to explore artworks in detail or seamlessly add them to their shopping carts for potential purchase.

2. About:

The "About" section caters to customers' curiosity, offering insights into the E-commerce Art Gallery's ethos, objectives, and the broader artistic community it supports.

3. Cart:

The Cart section plays a pivotal role in the customer journey, facilitating a smooth and intuitive shopping experience. Key features include:

* **Add to Cart:** Customers can curate their personalised collections by adding selected artworks to their cart. This feature enhances user convenience and supports the exploration of multiple artworks.
* **Buy Button:** Upon selecting artwork for purchase, customers can proceed to the checkout stage by clicking the "Buy" button. In cases where the cart is empty, a user-friendly message informs customers of their empty cart status.

4. Logout:

Customers can seamlessly end their browsing session by logging out through the dedicated "Logout" button. This ensures the security of customer accounts and maintains a user-centric approach to account management.

##### iii. Artist:

1. About

Like other roles, the "About" section caters to artists' curiosity, providing a detailed understanding of the E-commerce Art Gallery's mission, vision, and the supportive community they are part of.

2. Category Management

Artists enjoy specialised control over the Category Management section, where they can curate and showcase their artistic creations. Key functionalities include:

* **View:** Artists have the privilege of reviewing and managing all the images they have personally contributed to the platform. This feature empowers artists to maintain and curate their individual portfolios.
* **Edit and Delete:** Artists possess exclusive rights to edit or delete their own artworks. This level of autonomy ensures that artists can continuously refine and update their contributions to the Art Gallery.
* **Add Button:** The "Add" button within the Category Management section enables artists to seamlessly introduce new pieces of art to the platform, contributing to the diversity and richness of the overall gallery.

3. Logout:

Artists can gracefully conclude their session by logging out through the dedicated "Logout" button. This ensures the security and privacy of artist accounts while fostering a sense of control over their digital presence.

## **Diagram or Schematics Illustrations:**

### UML Diagrams:

#### a. Use Case Diagram:

**A screen shot of a diagram

Description automatically generated**

Figure 11: Use Case Diagram

#### b. Sequence Diagrams:

##### i. Artist Registration and Artwork Uploading:

**A screenshot of a computer

Description automatically generated**

Figure 12: Artist Registration and Artwork Uploading Sequence Diagram

This sequence diagram provides a clear overview of the interactions between the artist and the system in the context of an online art gallery.

1. Artist Registration:

* The artist initiates the process by sending a Registration Request to the system.
* The system then validates this request. This could involve checking if the artist’s details are complete and meet the platform’s requirements.
* Once validated, the system creates an Artist Account. This involves storing the artist’s details in the platform’s database.
* The system then sends an Artist Account Created message back to the artist, confirming that the registration has been successful.

2. Artist Login:

* After registration, the artist sends a Login Request to the system. This request would typically include the artist’s username and password.
* The system validates the Login request by checking the provided credentials against the stored details in the artist’s account.
* If the validation is successful, the system sends a Login Successful message back to the artist, granting them access to their account.

3. Account Update:

* At any point, the artist can send an Update Account Request to the system. This could involve changes to the artist’s personal details, password, or other account settings.
* The system validates and updates the account based on the received request.
* Once the updates are made, the system sends an Account Updated message back to the artist, confirming that the changes have been made successfully.

4. Artwork Upload:

* The artist can choose to Upload Artwork to the platform. This request would include the artwork file and any associated details or metadata.
* The system receives this as an Artwork Upload Request and validates it. This could involve checking the file format, size, and the provided metadata.
* Once validated, the system adds the artwork to the platform’s database.
* The system then sends an Artwork Upload Confirmation back to the artist, confirming that the artwork has been successfully uploaded and is now available on the platform.

5. Order Management:

* The system also has an Order Management process. This involves managing orders made for the artist’s artwork.
* The system retrieves Order Data from the database. This could include details of the artwork ordered, the customer’s details, and the order status.
* The system then processes the order. This could involve updating the order status, arranging for the artwork’s delivery, and handling payment.
* Any updates to the Order Status are communicated back to the relevant parties as needed.

##### ii. Artwork Purchase by Buyer/Customer:

**A screenshot of a computer

Description automatically generated**

Figure 13: Artwork Purchase Sequence Diagram

This sequence diagram provides a clear overview of the interactions between the buyer and the system in the context of an online art gallery.

1. Buyer Registration:

* The process begins with the buyer sending a Registration Request to the system.
* The system then validates this request. This could involve checking if the buyer’s details are complete and meet the platform’s requirements.
* Once validated, the system creates a Customer Account. This involves storing the buyer’s details in the platform’s database.
* The system then sends a Customer Account Created message back to the buyer, confirming that the registration has been successful.

2. Buyer Login:

* After registration, the buyer sends a Login Request to the system. This request would typically include the buyer’s username and password.
* The system validates the Login request by checking the provided credentials against the stored details in the buyer’s account.
* If the validation is successful, the system sends a Login Successful message back to the buyer, granting them access to their account.

3. Account Update:

* At any point, the buyer can send an Update Account Request to the system. This could involve changes to the buyer’s personal details, password, or other account settings.
* The system validates and updates the account based on the received request.
* Once the updates are made, the system sends an Account Updated message back to the buyer, confirming that the changes have been made successfully.

4. Browse Artworks:

The buyer can choose to Browse Artworks on the platform. The system retrieves the artwork details from the database and presents them to the buyer.

5. View Artist Profile:

The buyer can view the profile of an artist. The system retrieves the artist's profile data from the database and presents it to the buyer.

6. Add to Personal Gallery:

The buyer can add the artwork to their personal gallery. The system updates the personal gallery and sends a confirmation to the buyer.

7. Make Order:

The buyer can place an order for an artwork. The system processes the order, updates the order status, and sends an update to the buyer.

8. Make Payment:

The buyer makes a payment for the order. The system processes the payment, updates the payment status, and sends a confirmation to the buyer.

##### iii. Admin Management:

A screenshot of a computer screen

Description automatically generated

Figure 14: Admin Management Sequence Diagram

This sequence diagram provides a clear overview of the interactions between the admin and the system in the context of an online art gallery.

1. Manage Artists:

* The admin can manage artists on the platform. This process begins with the admin sending a Manage Artists request to the Artist Management system.
* The Artist Management system interacts with the main System to Retrieve Artist Data from the Database.
* The retrieved Artist Data is then displayed to the admin.

2. Manage Buyers:

* Similarly, the admin can manage buyers. This process begins with the admin sending a Manage Buyers request to the Buyer Management system.
* The Buyer Management system interacts with the main System to Retrieve Buyer Data from the Database.
* The retrieved Buyer Data is then displayed to the admin.

3. Update Art Stock:

* The admin can update the stock of artwork. This process begins with the admin sending an Update Art Stock request to the Art Management system.
* The Art Management system interacts with the main System to update the art stock data in the Database.
* A confirmation of the Art Stock Updated is then sent back to the admin.

4. Manage Orders:

* The admin can manage orders on the platform. This process begins with the admin sending a Manage Orders request to the Order Management system.
* The Order Management system interacts with the main System to Retrieve Order Data and Update Order Status in the Database.
* The updated Order Status is then sent back to the admin.

#### c. Entity-Relationship Diagram (ER Diagram):

**A diagram of a company

Description automatically generated**

Figure 15: ER Diagram

##### i. Description:

The ER diagram represents the database structure for an online art gallery. It outlines the relationships between different entities including User, Login, Roles, Permission, Customer, Order, Arts, and Maintenance Bill.

1. User:

This entity represents the users of the online art gallery. It has the following attributes:

* user\_name: The name of the user.
* user\_id: A unique identifier for each user.
* user\_mobile: The mobile number of the user.
* user\_email: The email address of the user.
* user\_address: The physical address of the user.

2. Login:

This entity is linked to the User entity, indicating that each user has login credentials. The attributes include login\_role\_id and login\_username.

3. Roles and Permission:

These entities are related to user access control. Each user has a role (with attributes role\_id, role\_name, role\_desc), and each role has certain permissions (per\_id, per\_role\_id, per\_module, per\_name).

4. Customer:

This entity represents the customers who place orders. It has attributes like cus\_name, cus\_address, cus\_mobile, cus\_email.

5. Order:

This entity represents the orders placed by customers. It includes ford\_type, ord\_num, ord\_desc, ford\_jd.

6. Arts:

This entity represents the art pieces available in the gallery. Each art piece has an arts\_id, arts\_type, and arts\_desc.

7. Maintenance Bill:

This entity, managed by the Permission entity, includes mb\_num, mb\_type, mb\_date, and mb\_desc.

##### ii. Relationships:

The relationships between these entities are represented by diamonds. These relationships help to define how data in one entity is related to data in another entity. They are crucial in understanding how data flows within the system and how different entities interact with each other.

1. User-Login-Roles-Permission (Has):

This indicates that each User entity is associated with a unique Login entity, and each User has one or more Roles, and each Role has one or more Permissions. This means that for each user in the system, there is a unique set of login credentials, and each user has certain roles, each of which has certain permissions. This relationship is crucial for maintaining the integrity and security of user data in the system, and for implementing access control.

* User to Login: One-to-One (1:1) - Each user has a unique set of login credentials, indicating a one-to-one relationship.
* User to Roles: One-to-Many (1:N) - Each user can have multiple roles, but each role is associated with only one user.
* Roles to Permission: One-to-Many (1:N) - Each role can have multiple permissions, but each permission is associated with only one role.

2. University User-Customer-Order-Maintenance Bill (Manage):

This suggests that a University User manages Customer, Order, and Maintenance Bill entities. This means that a university user (perhaps an admin or staff member) has the authority to manage customer profiles, orders, and maintenance bills. This relationship is key to the operation of the system, ensuring that these entities are properly managed and maintained.

* University User to Customer: One-to-Many (1:N) - A university user can manage multiple customers, but each customer is managed by only one university user.
* University User to Order: One-to-Many (1:N) - A university user can manage multiple orders, but each order is managed by only one university user.
* University User to Maintenance Bill: One-to-Many (1:N) - A university user can manage multiple maintenance bills, but each maintenance bill is managed by only one university user.

3. Order-Arts (Has):

This indicates that each Order entity is associated with one or more Arts entities. This means that each order placed in the system includes one or more pieces of art. This relationship is important for tracking which art pieces are included in each order.

#### d. Class Diagram:

**A diagram of a computer

Description automatically generated**

Figure 16: Class Diagram

##### i. Description:

The class diagram serves as a blueprint for your online art gallery project. It helps in understanding the system’s structure, guides the database design, and directs the implementation of the code. It’s a vital tool for communication among stakeholders.

1. Role Class:

This class handles different roles within the system. Each role has an ID, title, and description. Methods include adding a role to the database or deleting it.

2. User Class:

This class represents users with attributes like ID, role ID (linking to Role class for user’s role assignment), name, email, date of birth, and address. Methods are available for adding users to the database or deleting them.

3. Permission Class:

This class manages permissions associated with different roles. It includes permission ID, module (the part of the system where permission is applicable), title, and description. Permissions can be added or deleted.

4. Gallery Class:

This class represents individual galleries in your online art gallery project with attributes like gallery ID, type (perhaps indicating style or medium), title, and description. Galleries can be added or deleted from the database.

5. Payment Class:

This class handles payments made by customers including payment ID, type (such as credit card or PayPal), billing street address, descriptor (like a summary of payment), and date of payment.

6. Customer Class:

This class represents customers with customer ID, name, billing street address, email, and password. Methods are available for adding customers to the database, editing their information, or deleting them.

7. Order Class:

This class manages orders placed by customers including order ID number, type, shipping street address, and description. Orders can be added, edited, or deleted.

8. Arts Class:

This class represents individual pieces of art listed in your online gallery. Each piece has an art ID, title, description, and type. The methods allow for adding, editing, and searching for pieces.

##### ii. Relationships:

The lines connecting different classes represent relationships between them, showing how they interact within this online art gallery system.

1. Role - User:

One-to-many relationship. The Role class is related to the User class. This relationship indicates that each user has a role in the system. The role ID in the User class is likely a foreign key referencing the role ID in the Role class.

2. User - Permission:

One-to-many relationship. The User class is related to the Permission class. This suggests that each user has certain permissions in the system. These permissions are likely determined by the user’s role.

3. Gallery - Arts:

One-to-many relationship. The Gallery class is related to the Arts class. This relationship indicates that each gallery contains multiple pieces of art. The gallery ID in the Arts class is likely a foreign key referencing the gallery ID in the Gallery class.

4. Customer - Order:

One-to-many relationship. The Customer class is related to the Order class. This suggests that each customer can place multiple orders. The customer ID in the Order class is likely a foreign key referencing the customer ID in the Customer class.

5. Order - Arts:

One-to-many relationship. The Order class is related to the Arts class. This relationship indicates that each order contains one or more pieces of art. The art ID in the Order class is likely a foreign key referencing the art ID in the Arts class.

6. Order - Payment:

One-to-one relationship. The Order class is related to the Payment class. This suggests that each order has an associated payment. The payment ID in the Order class is likely a foreign key referencing the payment ID in the Payment class.

#### e. Deployment Diagram:

**A diagram of a application server

Description automatically generated**

Figure 17: Deployment Diagram

This deployment diagram provides a clear representation of how various components interact in a distributed environment, showcasing the integration with external APIs, payment processing, and data storage in a database. we will give a short structure of this diagram:

1. ExternalAPI (External API):

Data Flow: There is a data flow from the External API to the Application Server. This suggests that the Application Server interacts with an external API to either send or receive data.

2. AppServer (Application Server):

* **HTTP Request to External API:** The Application Server sends HTTP requests to the External API, indicating a communication channel for data exchange between the application and the external API.
* **HTTP Request to PaymentGateway (Payment Gateway):** The Application Server communicates with a Payment Gateway using HTTP requests. This is a common pattern for processing online payments in e-commerce applications.
* **Query/Update to DatabaseServer (Database Server):** The Application Server interacts with a Database Server by sending queries and updates. This connection represents the handling of data persistence and retrieval.
* **HTTP Response to Client:** After processing requests, the Application Server sends HTTP responses to the Client. This completes the cycle of client-server communication.

3. Client:

HTTP Request to WebServer (Web Server): The Client sends HTTP requests to the Web Server. This interaction represents the user accessing the web application through a browser.

4. WebServer (Web Server):

* **HTTP Request to AppServer (Application Server):** The Web Server forwards HTTP requests from the Client to the Application Server, suggesting that the Web Server acts as an intermediary between the Client and the Application Server.
* **HTTP Request to AppServer (Application Server):** There is a loopback or internal communication from the Web Server to the Application Server. This can happen when the Web Server needs to request data or services from the Application Server.

5. Analysis:

* **External API Integration:** Our application integrates with an external API, likely for some specific functionality or data exchange.
* **Payment Processing:** The Application Server communicates with a Payment Gateway, indicating support for online payment processing.
* **Database Interaction:** The Application Server communicates with a Database Server for querying and updating data. This reflects the storage and retrieval of information from a database.
* **Client-Server Communication:** The interaction between the Client and the Application Server is facilitated by the Web Server. This is a common architecture for web applications, where the Web Server acts as a gateway to the Application Server.
* **Data Flow:** The arrows indicating data flow help illustrate how information moves between different components, helping to understand the flow of data in our system.

### System Diagram:

#### a. Architecture Diagram:

**A diagram of a software development process

Description automatically generated**

Figure 18: Architecture Diagram

This architecture diagram outlines a typical e-commerce application for our project about the Online Art Gallery. It's structured with distinct modules for user management, artwork management, and order processing. The use of an SQLite Database suggests a lightweight and embedded database solution, suitable for smaller-scale applications such as our project. We will give a short structure of this diagram:

i. Frontend:

* **User Interface:** This is the visual part of our application that users interact with. It includes the different modules for managing user profiles, displaying artwork listings, and facilitating user interactions.
* **User Profile Module:** Handles user account management, allowing users (both artists and buyers) to create, update, and manage their profiles.
* **Art Listing Module:** Displays artwork listings to users, likely with features for searching, filtering, and viewing detailed information about each piece of art.

ii. Backend:

* **User Management Module:** Manages user-related operations, including user authentication, authorisation, and interaction with user profiles. It interacts with the SQLite Database to retrieve and update user data.
* **Artwork Management Module:** Manages the artwork-related operations, such as adding new artworks, updating existing ones, and handling the relationship between artists and their artworks. It also interacts with the SQLite Database to access and manage artwork data.
* **Order Processing Module:** Manages the processing of orders placed by buyers. It likely handles actions such as order creation, order confirmation, and updating inventory. This module interacts with the SQLite Database to store and retrieve order-related information.

iii. Database (SQLite):

Our application uses an SQLite Database to store and retrieve data related to users, artworks, and orders. The User Management Module, Artwork Management Module, and Order Processing Module all interact with this database.

iv. User Roles:

We have two primary user roles: Artists and Art Buyers. Artists interact with the system by uploading their artwork, while Art Buyers place orders for the artwork.

#### b. Technology Diagram:

**A diagram of a computer

Description automatically generated**

Figure 19: Technology Diagram

This technology diagram indicates a technology stack for building our e-commerce Art Gallery website, with considerations for front-end development, back-end logic, payment processing, version control, task management, and database management. we will give a short structure of this diagram:

i. Frontend (HTML, CSS, JS):

* **HTML (HyperText Markup Language):** Used to structure the content and layout of web pages.
* **CSS (Cascading Style Sheets):** Responsible for styling and presentation, enhancing the visual appeal of the user interface.
* **JS (JavaScript):** Add interactivity to the user interface, facilitating dynamic content and client-side functionality.

ii. Backend (Java):

Java: Chosen as the backend programming language. Java is known for its portability, scalability, and robustness so we decided to choose it for building enterprise-level applications.

iii. Payment Gateway (Payment Gateway):

Payment Gateway: Represents the external service or API that handles payment processing. It enables secure online transactions between the buyer and the seller. Common payment gateways include Stripe, PayPal, or other similar services.

iv. Task Management (Notation):

Notation: This tool is used for task management and collaboration. We use this tool to help our team organise, track, and manage member’s work.

v. Version Control (GitHub):

GitHub: A version control platform that facilitates collaborative development and code management. It allows us to work together on the same project, track changes, and manage different versions of the codebase.

vi. Database (SQLite):

SQLite: A lightweight, embedded relational database management system. It is suitable for small to medium-sized applications and is often used in scenarios where a full-scale database server is not necessary, and that is the reason why we chose it.

vii. Analysis:

* **Technology Stack:** The technology stack chosen includes widely used technologies for web development (HTML, CSS, JS) and backend development (Java).
* **Scalability:** Java on the backend is known for its scalability, making it a suitable choice for us to handle a growing number of users and transactions.
* **Payment Processing:** The integration of a Payment Gateway indicates that our e-commerce website supports online transactions securely.
* **Collaboration and Development:** GitHub is a good choice for version control, enabling collaborative development, code sharing, and tracking changes.
* **Task Management:** The inclusion of a task management tool suggests an organised approach to project management, facilitating collaboration and communication within the development team.
* **Database Choice:** SQLite is a lightweight database, suitable for smaller-scale applications. Ensure it meets the data storage and retrieval requirements of our project.

#### c. Data Flow Diagram (DFD):

##### i. Admin:

**A diagram of a computer network

Description automatically generated**

Figure 20: Admin Data Flow Diagram

The purpose of this data flow diagram for the admin panel of an online art gallery is to visually represent and manage the flow of data within the system.

1. Admin:

The admin interacts with the database to verify credentials and roles of access. This ensures that only authorised users can access the system.

2. Check Credential & Check Roles of Access:

These processes interact with the database to verify the user’s identity and permissions. The database stores this information, and it’s retrieved when a user tries to log in.

3. Forgot Password & Send Email to User:

If a user forgets their password, these processes interact with the database to retrieve the user’s email address. An email is then sent to the user with instructions for password recovery.

4. Manage Modules:

This process interacts with the database to manage different modules within the system. The database stores information about these modules, and the admin can add, remove, or modify this information as needed.

5. Database Interaction:

The database is central to the system, storing essential data and interacting with all other elements. It helps manage the system effectively.

##### ii. Artist:

**A diagram of a database

Description automatically generated**

Figure 21: Artist Data Flow Diagram

The purpose of this data flow diagram for the Artist’s functionality in an online art gallery is to visually represent and manage the flow of data within the system.

1. Artist:

The artist interacts with the system to register, log in, and manage their profile and artwork.

2. Registration:

This process interacts with the database to create a new artist account.

3. Login & Change Password:

These processes interact with the database to verify the artist’s identity and update their password if needed.

4. Get Account Details:

This process retrieves the artist’s account details from the database.

5. View Art Gallery: This process retrieves data from the database to display the art gallery to the artist.

6. View Artist Profile:

This process retrieves the artist’s profile details from the database.

7. Add to Personal Gallery:

This process interacts with the database to add artwork to the artist’s gallery.

8. Upload Art:

This process interacts with the database to add new artwork to the system.

9. Database Interaction:

All these processes interact with a central database that stores essential data, such as artist profiles, artwork details, and user credentials. This helps manage the system effectively and ensures data consistency.

##### iii. Buyer:

**A diagram of a diagram

Description automatically generated**

Figure 22: Buyer Data Flow Diagram

The purpose of this data flow diagram for the Buyer’s functionality in an online art gallery is to visually represent and manage the flow of data within the system.

1. Buyer:

The buyer interacts with the system to register, log in, and manage their profile and purchases.

2. Registration:

This process interacts with the database to create a new buyer account.

3. Login & Change Password:

These processes interact with the database to verify the buyer’s identity and update their password if needed.

4. Get Account Details:

This process retrieves the buyer’s account details from the database.

5. View Art Gallery:

This process retrieves data from the database to display the art gallery to the buyer.

6. View Artist Profile & Add To Personal Gallery:

These processes allow the buyer to view the profiles of artists and add artworks to their gallery. They interact with the database to retrieve artist profiles and update the buyer’s gallery.

7. Select Art, Get Bill, Make Order, Make Payment, Make Confirmation:

These processes represent the buyer’s journey from selecting an artwork to receiving confirmation of their purchase. They interact with the database to update the status of the selected artwork, generate a bill, create an order, process the payment, and confirm the purchase.

8. Database Interaction:

All these processes interact with a central database that stores essential data, such as buyer profiles, artwork details, and user credentials. This helps manage the system effectively and ensures data consistency.

## **Key Design Decisions and Justifications:**

### Technology Diagram:

#### a. Design Decision:

The chosen technology stack includes HTML, CSS, JS for the front end, Java for the backend, a Payment Gateway for financial transactions, task management using Notation, version control through GitHub, and SQLite as the database.

#### b. Justifications:

i. Frontend Technologies:

**HTML, CSS, JS:** These are widely adopted technologies for building responsive and interactive user interfaces, ensuring a seamless and engaging user experience.

ii. Backend Language:

**Java:** The use of Java as the backend language provides a robust and scalable foundation. Java's platform independence and mature ecosystem make it suitable for building enterprise-level applications.

iii. Payment Processing:

**Payment Gateway:** Integration with a Payment Gateway ensures secure and reliable financial transactions, a crucial aspect for an e-commerce platform.

iv. Version Control:

**GitHub:** GitHub was chosen for version control, enabling collaborative development, code tracking, and facilitating a structured approach to managing code changes.

v. Task Management:

**Notation:** The use of a task management tool, represented as "Notation," helps in organising and tracking project tasks efficiently, promoting collaboration within our team.

vi. Database Choice:

**SQLite:** The selection of SQLite as the database aligns with the project's needs, providing a lightweight and embedded database solution suitable for smaller-scale applications.

### Deployment Diagram:

#### a. Design Decision:

The Deployment Diagram illustrates the distribution of software components across nodes, including External API, Application Server, Payment Gateway, and Database Server.

#### b. Justifications:

i. External API Integration:

**ExternalAPI Node:** The inclusion of an ExternalAPI node signifies the integration with external services, enabling extended functionality and data exchange.

ii. Payment Processing:

**PaymentGateway Node:** The dedicated node for the Payment Gateway reflects a secure and isolated space for processing financial transactions.

iii. Database Interaction:

**DatabaseServer Node:** The separation of the Database Server node indicates a clear boundary for data storage and retrieval, ensuring efficient management of the application's database.

iv. Client-Server Architecture:

Deployment will illustrate a typical client-server architecture, with clients interacting with the application through the Web Server and the Application Server handling backend processes.

### Architecture Diagram:

#### a. Design Decision:

The Architecture Diagram provides a high-level overview of the system's structure, including Frontend, Backend, Payment Gateway, Task Management, Version Control, and Database components.

#### b. Justifications:

i. Modular Structure:

The modular structure depicted in the Architecture Diagram ensures a clear separation of concerns, promoting maintainability and scalability.

ii. External Service Integration:

The inclusion of ExternalAPI and Payment Gateway components signifies the integration with external services, expanding the system's capabilities beyond the core application.

iii. Version Control and Task Management:

GitHub and Notation Nodes: The presence of nodes for Version Control and Task Management emphasises the importance of organised collaboration and code management.

iv. Database as a Core Component:

Database Node: The Database node is a central component, highlighting the critical role of data storage and management in the system.

v. Client-Server Interaction:

The arrows indicating interactions between the Client, Web Server, and Application Server showcase the flow of information in a client-server architecture, ensuring effective communication.

# **CHAPTER V: DEVELOPMENT**

## **Development Process’s Description:**

### Project Initiation:

Objectives of our platform: connecting artists with buyers, facilitating art transactions, and promoting creativity. Also, we have identified the key stakeholders, including artists, buyers, and administrators. Next, we need to conduct a feasibility study to assess the technical and economic viability of the project.

### Planning:

We will develop a detailed project plan outlining tasks, timelines, and resource requirements.

Then we will identify and prioritise features such as user roles, artwork listings, shopping cart, and payment integration. Finally, plan for scalability and future enhancements.

### Design:

* Create wireframes and prototypes for the user interface (UI) and user experience (UX).
* Design the database schema to store information about artists, artworks, users, and transactions.
* Establish a secure authentication and authorisation system.

### Development/Implementation:

First, we will build the core functionality for our website, starting with user registration and authentication. Then, we implement features for artists to upload artwork, set prices, and manage their portfolios. Next to it, develop the shopping cart and secure checkout process to define credibility for our website. Finally, integrate a payment gateway for secure transactions.

### Testing:

* Conduct rigorous testing of each module, including user authentication, artwork uploads, and payment processing.
* Perform usability testing to ensure a smooth and intuitive user experience.
* Identify and resolve any bugs or issues during our development.

### User Feedback and Iteration:

* Collect feedback from potential users, artists, and buyers through an online survey feedback box.
* Iterate on the design and functionality based on feedback to enhance the user experience.
* Implement improvements and refinements.

### Deployment/Release:

* Deploy the e-commerce website on a hosting platform.
* Monitor the launch for any issues and address them promptly.
* Notify stakeholders and promote the platform to attract artists and buyers.

### Post-Launch Support and Maintenance:

* Provide ongoing support to address any post-launch issues.
* Monitor website performance and address scalability concerns.
* Regularly update the platform with new features and improvements due to customer feedback.

### Legal Compliance:

* Ensure compliance with data protection and e-commerce regulations.
* Clearly communicate terms of service, privacy policy, and other legal aspects to users.

## **Programming Languages, Frameworks and Tools:**

### IDE :

* VS Code and XAMPP
* Reference: https://code.visualstudio.com
* Reference: https://www.apachefriends.org/download.html

### Language:

* **Frontend:** HTML ( HyperText Markup Language ), JS ( JavaScript ), ejs (Embedded JavaScript templates), and CSS ( Cascading Style Sheets).
* **Backend:** Java

### Support:

* **Framework:** In our project, we are using the framework Vue.js ( https://vuejs.org ) to define for frontend
* **Library:** In other to control our database, we have used SQLite ( https://www.sqlite.org/about.html ) and Xampp ( https://www.apachefriends.org/index.html ) to define the Backend more easily

## **Challenges:**

Software development projects often encounter various challenges, and of course, we have a lot of challenges during our project too. Here are some challenges that our team have encountered during the development of our Art Gallery website:

### Integration Issues:

* **Challenge:** Difficult to seamlessly integrate different components into the system.
* **Solution:** Thoroughly test integrations, meet multiple times with code leaders, and ensure compatibility before deployment

### Scalability Concerns:

* **Challenge:** Anticipate increased users and ensure the system can scale to accommodate increased traffic.
* **Solution:** Regularly evaluate performance, conduct testing, and optimise the code base for scalability, and conduct surveys to optimise data performance

### Security Hole:

* **Challenge:** Identify and address potential security vulnerabilities
* **Solution:** Implement secure encryption methods, conduct regular security testing, and stay up to date with security best practices

### Cooperation Issues:

* **Challenge:** Difficult coordination between team members, especially because this is the first time working together and often work far apart, difficult communication between members
* **Solution:** We promoted communication, effectively used collaboration tools, and established clear guidelines for teamwork

### Version Control Conflict:

* **Challenge:** Conflicts occur when merging code changes from different branches.
* **Solution:** Centralise ramifications and classifications clearly, while communicating code changes to members effectively and resolving conflicts promptly.

### User Experience (UX) Challenges:

* **Challenge:** Ensure a seamless and intuitive user experience for both artists and buyers.
* **Solution:** Conduct user testing, gather feedback, and iterate on the design based on user input.

### Performance Bottlenecks:

* **Challenge:** Identify and resolve performance bottlenecks that can impact website responsiveness.
* **Solution:** Conduct performance testing, optimise database queries, and address resource-intensive processes.

### Data Management Issues:

* **Challenge:** Process large amounts of artwork images and related data efficiently.
* **Solution:** Implement effective data storage strategies, leverage content delivery networks, and optimise image loading.

### Legal and Compliance Issues:

* **Challenge:** Navigate legal considerations related to data protection, intellectual property, and e-commerce regulations.
* **Solution:** Consult legal experts, stay informed about relevant regulations, and take necessary compliance measures.

In addition, during the process of working together, we encountered many different small and large problems, but we sat together and tried to cooperate to solve the challenges we encountered. Currently, this project still has many errors, but we are still trying every day to improve our product in the most positive direction to bring the best experience to our users.

## **Code Management and Version Control:**

Code management and version control played a key role in our project. The main tools used for this purpose are GitHub for collaboration and version control, Google Drive for code backup and image storage, and Zalo for team communication. This report provides an overview of how these tools are used and their impact on our project development.

### GitHub: Version Control and Collaboration

* **Version Control with Git:** GitHub serves as our primary version control system, using the Git distributed version control system. This allows us to track changes made to the codebase, maintain a comprehensive history, and facilitate collaboration between team members. Git's branching and merging capabilities allow for the concurrent development of features, ensuring a smooth and organised development process.
* **Collaborative Workflow:** GitHub's collaboration features, including pull requests, issues, and project boards, have been instrumental in streamlining our collaborative workflow. Pull requests provide a structured process for code review, allowing our team to review proposed changes before merging them into the main codebase. Issues and project boards help in tracking tasks, bugs, and project milestones.
* **Contributor Management:** GitHub also provides robust tools for contributor management. Each of our team members has their own GitHub account, and access to repositories is controlled through permissions. This ensures a secure and controlled environment, allowing us to manage contributions effectively and maintain code quality.

### Google Drive: Backup and Image Storage

* **Code Backup:** Google Drive is used as an additional layer of backup for our codebase. Regular backups of important project files, configuration settings, and documentation are stored on Google Drive. This redundancy enhances data security and provides a fallback option in case of unforeseen issues with the primary GitHub repository.
* **Image Storage:** Given the visual nature of our Art Gallery project, Google Drive is used to store images related to artworks and other visual assets. This allows for efficient sharing and collaboration on image files among team members.

### Zalo: Team Communication

* **Group Chat on Zalo:** Zalo serves as our primary communication platform, providing an instant messaging solution for team members. The group chat on Zalo facilitates quick communication, updates on project progress, and coordination among team members. It serves as a centralised communication hub for both technical and non-technical discussions.
* **Real-time Collaboration:** Zalo's real-time communication features enable quick decision-making and issue resolution. The ability to share updates, discuss code-related matters, and seek immediate feedback enhances the overall efficiency of our development process.

### Conclusion:

The combination of GitHub for version control and collaboration, Google Drive for backup and image storage, and Zalo for team communication created a cohesive and productive ecosystem for our project. This integrated approach ensures code integrity, provides backup solutions, and promotes effective communication between team members. Going forward, we will continue to leverage these tools to maintain a streamlined development process and achieve project success.

# **CHAPTER VI: TESTING AND QUALITY ASSURANCE**

## **Testing Methodologies:**

Applying the Agile model to our online gallery project brings many benefits. Agile helps us enhance flexibility and interaction in the product development process.

Firstly, we use the Scrum method to manage the project. We divide the project into short sprints, usually ranging from 1 to 4 weeks. Each sprint focuses on developing a small part of the product, allowing us to quickly receive feedback from customers and make necessary changes in the development direction.

Secondly, we regularly hold daily meetings to update progress and evaluate completed tasks. This allows us to identify issues early on and find ways to resolve them quickly.

Thirdly, we use a Kanban board to track tasks and the progress of each team member. This helps us know who is doing what and keeps everyone synchronised.

Finally, we evaluate the completion of each sprint and hold a meeting after each sprint to assess and improve the development process. This helps us learn from mistakes and apply that knowledge to the next sprints.

## **Results and Bug Tracking:**

### Test Case 1:

|  |  |
| --- | --- |
| **Test Case #:** 1 | **Test Case Name:** Verify Art Display. |
| **System:** Art Gallery System. | **Subsystem:** Artwork Display. |
| **Designed by:** Mr. Nguyen Ba Phuc. | **Design Date:** 27/10/2023 |
| **Executed by:** Mr. Truong Tan Phat. | **Execution Date:** 25/11/2023 |
| **Short Description:** This test case verifies that the artwork is displayed correctly in the art gallery. | |
| **System Testing:** Functional Testing. | |

Table 21: Test Case 1 Information

|  |
| --- |
| **Pre-conditions:**   * The Art Gallery System is properly installed and configured. * The database containing the artwork information is populated with valid and accurate data. * The Art Gallery System is accessible and functional. |

Table 22: Test Case 1 Pre-conditions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comments** |
| 1 | Launch the Art Gallery System. | The art gallery system opens successfully without any errors or delays. | Pass | The art gallery system launched seamlessly, and all features and functionalities are working as expected. The system interface is user-friendly and intuitive, making it easy to navigate through different galleries, view artwork, and explore various exhibitions. |
| 2 | Navigate to the artwork display section. | The user is directed to the designated artwork display section of the art gallery system. | Pass | Upon navigating to the artwork display section, the system successfully directs the user to the designated area where artwork is showcased. The layout and organisation of the artwork display section are visually appealing and easy to navigate. |
| 3 | Click on the "Artwork Display" tab or button. | The artwork display interface loads and becomes visible on the screen. | Fail | Unfortunately, upon clicking on the "Artwork Display" tab or button, the expected system response did not occur. Instead, there was a delay in loading the artwork display interface, and when it finally appeared, some elements were missing or not functioning properly. |
| 4 | Wait for the artwork display interface to load. | The artwork display interface loads within a reasonable time frame, without any noticeable delays. | Pass | The artwork display interface loaded smoothly and within a reasonable time frame, without any noticeable delays. The system efficiently processed the request and presented the artwork display interface with all its elements fully functional. |
| 5 | Check that the relevant information, such as title, artist, and medium, is accurately displayed for each artwork. | The artwork information, including title, artist, and medium, is correctly associated with each respective artwork and is displayed alongside the artwork thumbnail or image. | Pass | The system successfully displays the relevant information, such as the title, artist, and medium, for each artwork. |
| 6 | Ensure that the artwork display is responsive and adjusts appropriately to different screen sizes or orientations. | The artwork display interface adapts seamlessly to different screen sizes or orientations, maintaining the proper layout and readability. | Pass | The artwork display interface demonstrates excellent responsiveness, seamlessly adjusting to different screen sizes or orientations. Whether viewed on a desktop, tablet, or mobile device, the system ensures that the layout and readability of the artwork display remain intact. |

Table 23: Test Case 1 Description

|  |
| --- |
| **Post-conditions:**   * The artwork display interface remains accessible and functional. * The displayed artwork thumbnails or images, along with their associated information, are accurate and consistent with the database. * The artwork display interface maintains its responsiveness and adaptability to different screen sizes or orientations. |

Table 24: Test Case 1 Post-conditions

### Test Case 2:

|  |  |
| --- | --- |
| **Test Case #:** 2 | **Test Case Name:** Product Searching. |
| **System:** Art Gallery System. | **Subsystem:** Product Search. |
| **Designed by:** Mr. Do Minh Quan. | **Design Date:** 26/10/2023. |
| **Executed by:** Mr. Nguyen Can. | **Execution Date:** 15/11/2023. |
| **Short Description:** Verify that the search functionality accurately retrieves relevant products based on user input. | |
| **System Testing:** Functional testing. | |

Table 25: Test Case 2 Information

|  |
| --- |
| * **Pre-conditions:** The user is on the registration page of the system. * The user registration form is visible and accessible. * The user has a stable internet connection. |

Table 26: Test Case 2 Pre-conditions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comments** |
| 1 | Enter a specific product name in the search bar. | The search results page should display products that match the entered product name. | Fail | Unfortunately, upon entering a specific product name in the search bar, the system did not provide the expected response. The search results page did not display products that matched the entered product name accurately. |
| 2 | Click on the "Add category" button or press Enter. | The displayed products should be relevant to the search query. | Pass | Upon clicking on the "Add category" button or pressing Enter, the system successfully provided relevant products that matched the search query. The displayed products were appropriate and aligned with the entered search criteria, allowing users to find the desired products effectively. |
| 3 | Review the search results page. | The search results should be displayed in a clear and organised manner, making it easy for the user to browse and select products of interest. | Pass | The search results page is presented in a clear and organised manner, successfully meeting the expected system response. The system effectively displays the search results, allowing users to browse through the products with ease and select the ones that are of interest to them. |

Table 27: Test Case 2 Description

|  |
| --- |
| **Post-conditions:**   * The user registration form should capture all required user information accurately, including fields such as name, email address, password, and any other necessary details. * The user registration form should enforce proper validation rules for each field, ensuring that the entered data meets the specified criteria (e.g., valid email format, minimum password length). |

Table 28: Test Case 2 Post-conditions

### Test Case 3:

|  |  |
| --- | --- |
| **Test Case #:** 3 | **Test Case Name:** Add Item to Cart. |
| **System:** Art Gallery System. | **Subsystem:** Shopping Cart. |
| **Designed by:** Mr. Tran Tuan Nghiep. | **Design Date:** 26/10/2023. |
| **Executed by:** Mr. Nguyen Minh Thong. | **Execution Date:** 20/11/2023. |
| **Short Description:** Verify that the user can successfully add items to the shopping cart. | |
| **System Testing:** Functional testing. | |

Table 29: Test Case 3 Information

|  |
| --- |
| **Pre-conditions:**   * The user is on the product details page of the E-Commerce website. * The product is available for purchase. * The user has a registered account and is logged in. |

Table 30: Test Case 3 Pre-conditions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comments** |
| 1 | Navigate to the product details page of the desired item. | The system should load the product details page, displaying all relevant information about the product, such as its name, description, price, and any available options or variants. | Pass | Upon navigating to the product details page of the desired item, the system successfully loaded the page and displayed all relevant information about the product. The expected system response was met, providing users with comprehensive details about the product, including its name, description, price, and any available options or variants. |
| 2 | Select the desired quantity of the item. | The system should update the quantity selection field to reflect the user's chosen quantity. The system may also validate the entered quantity to ensure it falls within the allowed range or meets any other specified criteria. | Pass | Upon selecting the desired quantity of the item, the system successfully updated the quantity selection field to reflect the user's chosen quantity. The expected system response was met, allowing users to specify the desired quantity accurately. |
| 3 | Click on the "Add to Cart" button. | The system should add the selected item to the shopping cart. It should update the shopping cart icon or indicator to reflect the addition of the item. Additionally, the system may display a confirmation message indicating that the item has been successfully added to the cart. | Pass | Upon clicking on the "Add to Cart" button, the system successfully added the selected item to the shopping cart. The expected system response was met, and the necessary updates were made to the shopping cart icon or indicator to reflect the addition of the item. |
| 4 | Review the shopping cart page. | The system should navigate the user to the shopping cart page. The shopping cart page should display the added item, along with its details such as name, quantity, price, and any additional relevant information. | Fail | Upon reviewing the shopping cart page, it was observed that the system did not meet the expected response. The system failed to navigate the user to the shopping cart page and did not display the added item or its details. |

Table 31: Test Case 3 Description

|  |
| --- |
| **Post-conditions:**   * The item should be successfully added to the shopping cart. * The shopping cart page should display the added item, including its name, quantity, price, and any additional relevant information. * The shopping cart page should provide options to update the quantity or remove the item from the cart. |

Table 32: Test Case 3 Post-conditions

### Test Case 4:

|  |  |
| --- | --- |
| **Test Case #:** 4 | **Test Case Name:** Artwork Deletion |
| **System:** Art Gallery System. | **Subsystem:** Artwork Deletion. |
| **Designed by:** Mr. Ta Vi Khang. | **Design Date:** 25/10/2023. |
| **Executed by:** Mr. Nguyen Ba Phuc. | **Execution Date:** 24/11/2023. |
| **Short Description:** Test the functionality to delete an image from the gallery. | |
| **System Testing:** Functional testing. | |

Table 33: Test Case 4 Information

|  |
| --- |
| **Pre-conditions:**   * The Art Gallery System is installed and running. * The user has appropriate access rights and permissions to delete images from the gallery. * The gallery contains at least one image that can be deleted. |

Table 34: Test Case 4 Pre-conditions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comments** |
| 1 | Open the Art Gallery System and navigate to the section or page where images can be managed or deleted. | The gallery management section is displayed, showing a list of images available for deletion. | Pass | The user successfully accessed the gallery management section. |
| 2 | Identify the specific image in the gallery that needs to be deleted based on its attributes or reference. | The selected image is highlighted or visually distinguishable to indicate that it has been selected for deletion. | Pass | The user was able to locate the desired image for deletion. |
| 3 | Click on the delete button or perform the necessary action to initiate the deletion of the selected image. | A confirmation message or dialog box prompts the user to confirm the deletion action. | Pass | The system successfully prompts the user for confirmation before proceeding with the deletion. |
| 4 | Click on the "Confirm" or "Delete" button in the confirmation message or dialog box to proceed with the deletion. | The selected image is permanently removed from the gallery, and a success message is displayed indicating the successful deletion. | Pass | The system successfully deletes the selected image from the gallery and provides a confirmation message. |
| 5 | Check if the deleted image is no longer present in the gallery or the list of available images. | The deleted image is no longer visible in the gallery or the list of images. | Pass | The system successfully removes the deleted image from the gallery, confirming its deletion. |

Table 35: Test Case 4 Description

|  |
| --- |
| **Post-conditions:**   * The deleted image is permanently removed from the gallery and no longer visible in the system. * The gallery management section or page reflects the updated list of images, with the deleted image no longer present. * Any associated metadata or information related to the deleted image (e.g., title, description, tags) is also removed from the system. |

Table 36: Test Case 4 Post-conditions

### Test Case 5:

|  |  |
| --- | --- |
| **Test Case #:** 5 | **Test Case Name:** Self-cleaning. |
| **System:** Art Gallery System. | **Subsystem:** Configuration Testing. |
| **Designed by:** Mr. Truong Tan Phat. | **Design Date:** 26/10/2023. |
| **Executed by:** Ms. Huynh Ngoc Anh Thu. | **Execution Date:** 26/11/2023. |
| **Short Description:** This test case ensures that the test environment is returned to its pre-test state and does not render the environment unusable. It specifically focuses on configuration testing to verify that the system remains functional and stable after changes are made. | |
| **System Testing:** Installation testing. | |

Table 37: Test Case 5 Information

|  |
| --- |
| **Pre-conditions:**   * The Art Gallery system has been successfully installed and configured on the target environment. * The system's configuration settings are accessible and modifiable. * A baseline or reference configuration state has been established for comparison. |

Table 38: Test Case 5 Pre-conditions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comments** |
| 1 | Open the Art Gallery system's administration panel and navigate to the configuration settings. | The configuration settings interface is displayed. | Pass | Ensure that the configuration settings interface is accessible and properly displayed. |
| 2 | Modify the "Number of displayed artworks" setting from 10 to 5. | The configuration setting is successfully updated. | Pass | Confirm that the system allows for modifying the configuration settings. |
| 3 | Click on the "Save" button to apply the configuration changes. | The system indicates that the configuration changes have been saved. | Pass | Ensure that the system confirms the successful saving of configuration changes. |
| 4 | Ensure that the system restarts or refreshes to reflect the new configuration. | The system restarts or refreshes without any errors. | Pass | Verify that the system can restart or refresh after applying the configuration changes. |
| 5 | Check that the updated configuration is reflected in the system's interface, displaying only 5 artworks per page. | The system successfully displays the updated number of artworks per page. | Pass | Confirm that the system can revert the configuration changes successfully. |
| 6 | Repeat steps 2-6 with various configuration changes. | The system consistently applies the configuration changes and can recover to its original state. | Pass | Validate that the system can handle multiple configuration changes and recover reliably. |

Table 39: Test Case 5 Description

|  |
| --- |
| **Post-conditions:**   * The Art Gallery system remains functional and stable after applying and undoing the configuration changes. * The system's configuration settings are restored to their original state. * The system continues to display the expected number of artworks per page according to the original configuration. |

Table 40: Test Case 5 Post-conditions

## **Quality Assurance Measures:**

* **Test Effort:** Measure the effort put into designing, reviewing, and executing test cases. This metric helps set baselines and compare the final test results.
* **Test Effectiveness:** Evaluate the success of test cases in detecting bugs. This metric measures the ability of a test case to detect bugs and is represented as a percentage.
* **Defect Resolution Percentage:** Assess the efficiency of the development team in analysing and fixing bugs reported by the QA team. This metric calculates the percentage of defects fixed out of the total defects reported.
* **Defect Age:** Measure the average time taken by developers to fix a defect, from its creation to resolution.
* **Number of Test Cases:** Track the total number of test cases, including the number of test cases passed, failed, and blocked.
* **Test Case Review Success:** Evaluate the success of test case review and analysis. This measure assesses the criteria and metrics used to enhance the test case review process.
* **Process Metrics:** Assess the characteristics and performance of the testing process, such as enhancement and conservation of the Software Development Life Cycle (SDLC) procedures.

# **CHAPTER VII: DEPLOYMENT AND IMPLEMENTATION**

## **Deployment Process Details:**

The Online Art Gallery project was deployed through many steps to ensure its implementation was good and smooth.

* Step 1: Setting up the component, we use a cloud-based platform (such as AWS) that provides hardware and software resources needed for deployment of the project including web server, database storage, networking, security, content management system, and any additional tools.
* Step 2: Testing the project on the local environment to ensure its functionality and performance work well.
* Step 3: Reviewing and configuring the project’s data and function to suit the client’s requirements.
* Step 4: Use version control systems such as Git to manage the source as well as manage the work of developers and upload the art’s products to the database.
* Step 5: Verifying its availability, functionality on the web, payment security, and compatibility across different devices and browsers,
* Step 6: Launch the Online Art Gallery on the web.

## **Rollout Strategy and Timeline:**

The rollout of the Online Art Gallery project was carefully scheduled to optimise the time and work in case of disruption and accidents. The following timeline is:

### *Preparation*:

* Duration: 1 week.
* Our team members will prepare many meetings for verification of the website setup, payment security testing, and other issues relevant to customers’ experiences and make necessary adjustments.

### *Soft Lau*nch:

* Duration: 1 week.
* Launch the software to small groups of audiences including sponsors, artists, gallery staff, clients, or small volunteer groups for further testing and gathering feedback, improving the quality of the product before the official launch.

### *Public Laun*ch:

* Duration: 1 day.
* After the soft launch, the Online Art Gallery project was officially launched to the public. At this time, we hired a marketing team to promote our website through various channels and to attract as many customers as possible.

## **User Training and Onboarding:**

* **Interactive Guides:** For every user who is on the first time enter the web, on-screen tips appear to assist users in navigating the basic features and functionalities of the website. This allowed them to explore the platform easily.
* **User Manual:** The Online Art Galley also included a user manual that explained in detail the features and some basic functions of the art online gallery. This includes how to register, log in, create the profile, search, and purchase the artwork for customers, as well as how to upload the artwork update the artwork, and set the price for the artist. In addition, full documentation contains some regular FAQs and troubleshooting, and guides are provided in PDF file or video format.
* **Support Channels:** We have many channels to help customers get acquainted with the onboarding process. It includes our community forum, an available 24/7 chatbot, and an email system to help users solve their problems as soon as possible.

## **Post-deployment Support Plan:**

After deployment, the art online gallery has the potential to grow further. These issues regularly require maintenance and updating of the content and interface of the website. That is the reason why the post-deployment support plan is crucial, and it ensures the project’s quality and reliability. It includes the following aspects:

* **Use Feedback:** Establish a service team that responds to the user’s queries and complaints through many of the support channels. In addition, the collected feedback will be used to improve the art online gallery interface and function for the next update.
* **Maintenance Regularly:** The software also needed to be updated regularly to avoid bugs, security issues, and cyber-attacks. This updated version includes fixing bugs, adding new features, changing the interface, optimising code, enhancing payment security, etc. Based on the user’s data, report, and feedback, we can improve our software to be more reliable and stable.
* **Community Forums:** Creating online forums such as a Facebook page, Reddit, or Discord helps to develop a community-driven support system. Users might exchange information, ask questions, and solve problems together.

# **CHAPTER VIII: USER DOCUMENTATION**

## **User’s Manuals or Guides:**

### Getting Started:

#### a. User Registration:

To unlock the full potential of the Art Online Gallery, start by registering an account:

* Visit the registration page.
* Fill in the required details.
* Click "Register."
* Verify your account through the email sent to your registered email address.

#### b. Account Verification:

Upon registration, check your email for a verification link. Click the link to activate your account and gain full access to the platform.

#### c. Login and Logout:

Log in to your account using your credentials. Remember to log out after each session to ensure the security of your account.

#### d. Forgot Password:

If you forget your password:

* Click on "Forgot Password."
* Enter your email address.
* Follow the instructions sent to your email to reset your password.

### User Profiles:

#### a. Artist Profile Management:

i. Uploading Artwork:

* Showcase your artistic creations:
* Navigate to your artist dashboard.
* Click "Upload Artwork."
* Add detailed descriptions, set prices, and upload high-quality images.

ii. Setting Prices:

Specify the cost of your artwork during the upload process. Prices should reflect the value you place on your creations.

iii. Managing Art Descriptions

Edit and enhance your art descriptions regularly to provide potential buyers with meaningful insights into your work.

#### b. Buyer Profile Management

i. Updating Account Details:

Keep your buyer profile information up to date:

* Access your profile.
* Click "Edit Profile."
* Update your details.
* Save changes.

ii. Changing Password

Enhance your account security by changing your password regularly:

* Go to "Account Settings."
* Click "Change Password."
* Follow the prompts to update your password.

## **System Requirements for End-users:**

The system requirements for end-users to use the art online gallery project are as follows:

* All user devices can connect to the Internet such as laptops, computers, smartphones, and tablets.
* A web browser that supports HTML5, CSS3, and JavaScript, such as Microsoft Edge, Google Chrome, or Mozilla Firefox.
* A stable and fast internet connection that can handle high-quality images and videos.
* A screen resolution of at least 1024 x 768 pixels for optimal viewing experience.

## **Troubleshooting Tips and FAQs:**

### Troubleshooting Tips:

|  |  |  |
| --- | --- | --- |
| **Type of Issue** | **Issue Description** | **Troubleshooting Tip** |
| Forgot Password | Unable to reset the password. | Ensure you are entering the correct email address. Check your spam folder for the password reset email. |
| The password reset link expired. | Request a new password reset link and use it promptly. |
| Account Lockout | Account locked due to multiple failed login attempts. | Wait for a specified duration or use the "Forgot Password" option to reset your password. |
| Artwork Upload and Display Issues | Artwork images appear pixelated or distorted. | Upload high-resolution images and ensure they meet the platform's image size requirements. |
| Uploaded artwork is not visible in search results. | Check your artwork settings to ensure they are set to be publicly visible. |
| Transaction and Payment Issues | Payment transaction unsuccessful. | Verify that your payment details are correct. If the issue persists, try using an alternative payment method. |
| Order Confirmation Email | Did not receive an order confirmation email. | Check your spam or junk folder. Ensure your email settings allow messages from the Online Art Gallery platform. |

Table 41: Issues and Troubleshooting Tips

### Frequently Asked Questions (FAQs):

Q: What is the Art Online Gallery?

A: The Art Online Gallery is a platform connecting artists and art enthusiasts. Artists can showcase and sell their work, while buyers can explore and purchase a diverse range of artwork.

Q: How do I Register?

A: Visit the registration page, fill in the required details, and follow the account verification process outlined in the user manual.

Q: How Can I Showcase My Artwork?

A: Log in, go to your artist dashboard, and click "Upload Artwork." Follow the prompts to add descriptions, set prices, and upload images.

Q: Can I Edit My Artwork Descriptions?

A: Yes, you can edit your artwork descriptions at any time from your artist dashboard.

Q: How Can I Make a Purchase?

A: Add the desired artwork to your shopping cart, proceed to checkout, and follow the payment process.

Q: Is My Payment Information Secure?

A: Yes, we prioritise the security of your payment information. We use secure payment gateways and encryption protocols.

# **CHAPTER IX: PROJECT EVALUATION:**

## **Evaluation Criteria:**

The success of standard software can be defined and measured by several evaluation criteria, which are dependent on techniques, stakeholders, and project goals. Value, time, cost, scope, and quality are five common and important evaluation criteria for almost all software.

### *Scope*:

The first and crucial criterion for any software. The project’s scope and objectives help the developers to accomplish the goals on time. These objectives must be SMART-specified, measurable, attainable, relevant, and time-bound.

### *Quality*:

Refers to how well the software completes functional and non-functional criteria, as well as customer expectations and standards.

### *Time*:

Refers to the length of the software development process, as well as adaptation to the schedule and deadlines.

### *Cost*:

The number of resources such as software resources, hardware resources, and additional resources that are consumed in the software development process. Managing the budget is an essential criterion to prevent the run of budget or income loss.

Finally, value refers to the benefits and outcomes provided by software to customers, users, and organisations, as well as the return on investment and impact on corporate goals. The more value that software brings the more successful the project is.

## **Goals-checking Process:**

Depending on time limitations, our team just did the basic elements of the art online gallery. Up till now, the goals we achieved include:

* **Establish a Fully Functional Art Gallery Platform:** Develop and deploy a fully functional online art gallery platform accessible to both artists and art buyers.
* **User Registration and Profile Management:** Enable users (artists and buyers) to register accounts, create profiles, and manage their personal information efficiently.
* **Artwork Upload and Management:** Implement a seamless artwork upload feature, allowing artists to showcase their creations with detailed descriptions, pricing, and categories.
* **Efficient Transaction Mechanism:** Establish a secure and efficient transaction mechanism, enabling buyers to easily purchase artworks and artists to manage sales.
* **User-Friendly Interface:** Design and implement a user-friendly interface for both artists and buyers, ensuring a positive and engaging user experience.
* **Artwork Search and Discovery:** Implement a robust search and discovery feature, allowing users to find artworks based on various criteria such as artist name, category, or keywords.
* **Artist Profile Management:** Provide artists with a user-friendly dashboard to manage their profiles, upload new artwork, set prices, and update information.
* **Quality Assurance and Bug Resolution:** Conduct thorough testing to identify and resolve any bugs or issues, ensuring a stable and reliable platform.
* **Implement User Feedback Mechanism:** Integrate a feedback mechanism to collect user input, allowing for continuous improvement based on user suggestions and concerns.
* **Documentation and Reporting:** Maintain comprehensive project documentation, including progress reports, user manuals, and troubleshooting guides.
* **Marketing and Awareness Initiatives:** Initiate marketing efforts to create awareness about the online art gallery, potentially leveraging social media campaigns, SEO strategies, and collaborations with influencers.
* **Collaborate with Artists and Collectors:** Actively collaborate with artists to encourage participation in the platform, expanding the art collection and attracting art enthusiasts and collectors.
* **Security and Privacy Measures:** Implement robust security measures to ensure the privacy and protection of user information, particularly in transactions.
* **User Training and Support:** Develop user manuals and support systems to assist both artists and buyers in navigating and using the platform effectively.
* **Achieve Project Milestones:** Successfully achieve key project milestones as outlined in the initial project plan, demonstrating progress and adherence to the established timeline.

## **Users’ Feedback and Satisfaction:**

### Introduction:

The user’s feedback and satisfaction with the project Online Art Gallery was collected using the survey “Customer Survey based on Actual Situation”. This survey was conducted to gather valuable insights from our users, helping us understand their experiences and expectations. The feedback provided by our esteemed users serves as a crucial guide for the continuous improvement and enhancement of our platform.

The survey, titled "Customer Survey based on Actual Situation," aimed to evaluate various aspects of user interaction, including the login interface, overall user experience, and the specific experiences of artists using our platform. We appreciate the time and effort invested by each participant in sharing their thoughts, as their feedback is instrumental in shaping the future of our platform.

### Overall Satisfaction:

One of the key metrics assessed in this survey is the overall satisfaction of our users. Participants were asked to rate their satisfaction on a scale of 1 (very bad) to 10 (very good). The following section provides an overview of the overall satisfaction levels, offering a glimpse into how our users perceive their interactions with the Online Art Gallery website.  
Biểu đồ câu trả lời của biểu mẫu. Tên câu hỏi: Overall assessment of our website&apos;s user interface
. Số lượng câu trả lời: 3 câu trả lời.

Figure 23: Overall Assessment of the website's user interface

### Login Interface Experience:

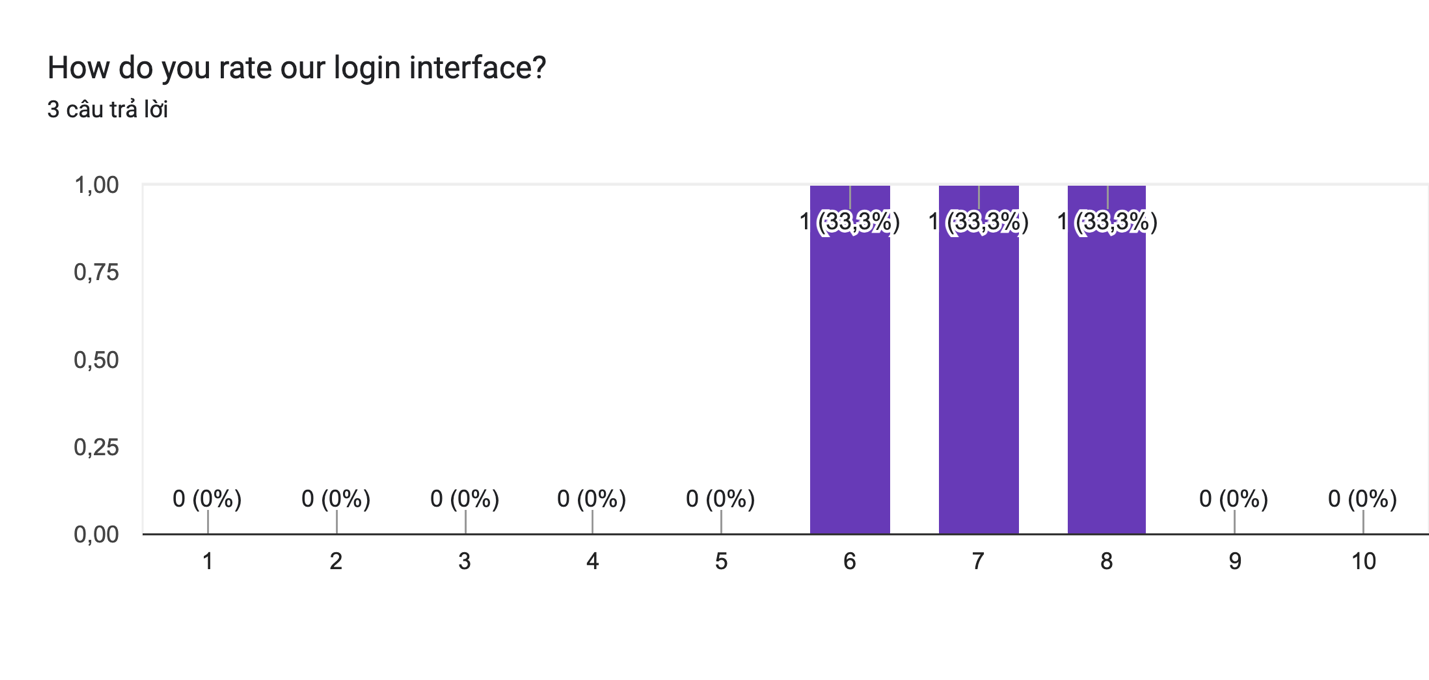
The evaluation of user satisfaction with the Login Interface yielded a moderate rating on average. Users, when assessing their experience with the platform's login interface, expressed a nuanced perspective, indicating a balanced level of satisfaction. This moderate satisfaction level suggests that while users found certain aspects of the login interface to be satisfactory, there may be opportunities for enhancement or refinement. Further exploration of specific user feedback and preferences will be valuable in pinpointing areas for potential improvement, contributing to the continuous optimization of the login interface to better align with user expectations and preferences.

Figure 24: Login Interface Experience

### User’s Website Experience:

The feedback received from users and their overall satisfaction with the system were consistently positive. Users provided commendable remarks regarding their experience with the platform's login interface, indicating a generally high level of satisfaction. The positive sentiments expressed by users underscore the effective design and functionality of the login interface, contributing to an overall positive perception of the system. It is evident that users found the login interface to be user-friendly and satisfactory, enhancing their overall interaction with the platform.Biểu đồ câu trả lời của biểu mẫu. Tên câu hỏi: How is your experience using our website as a user?
. Số lượng câu trả lời: 3 câu trả lời.

Figure 25: User's Website Experience

### Artist’s Website Experience:

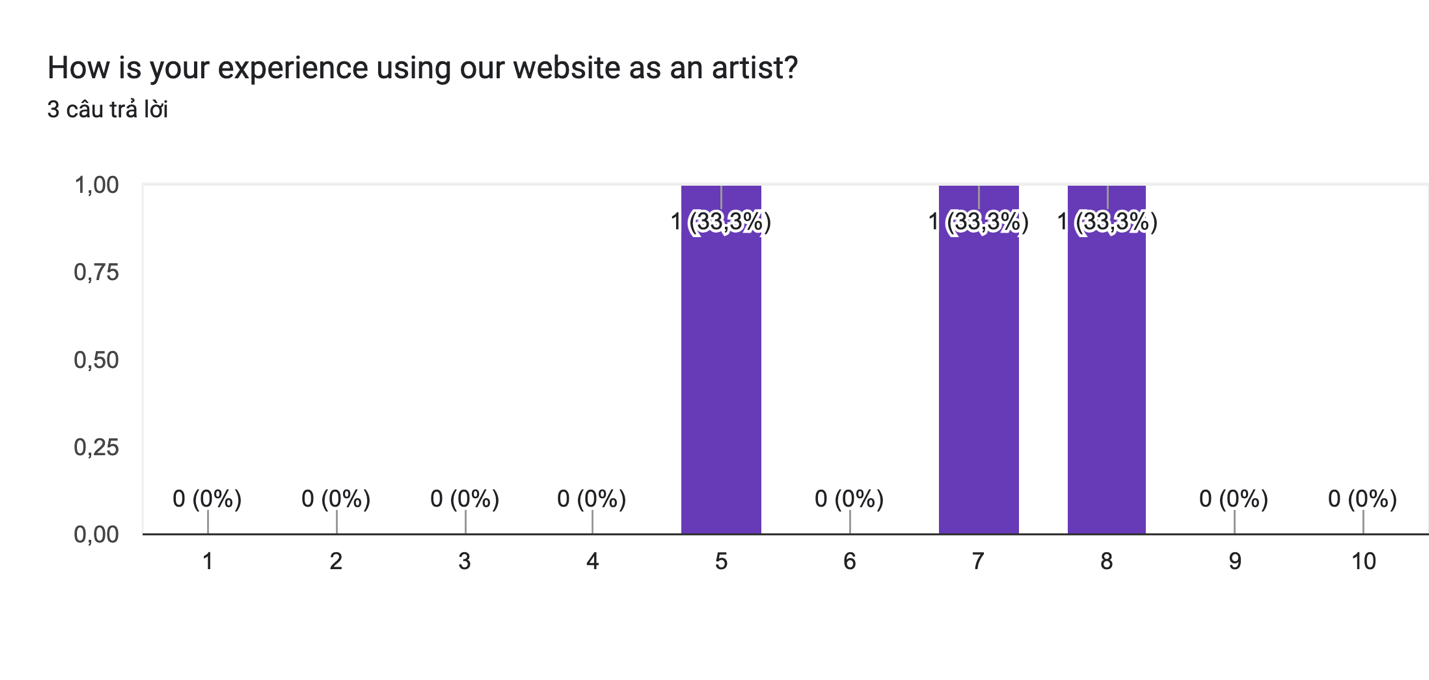
Artists, upon evaluating their experience on our website, provided a rating that falls within the spectrum of average to good satisfaction. This assessment implies a moderate level of contentment among artists using the platform, signifying that the overall user experience meets or slightly exceeds their expectations. The nuanced nature of this rating suggests a mix of positive and constructive aspects in artists' interactions with the website. Exploring specific feedback and preferences from the artist community will be instrumental in identifying areas of improvement and refining the user experience. This nuanced evaluation underscores the importance of an ongoing commitment to enhancing the platform to align more closely with the diverse needs and expectations of the artist user base.

Figure 26: Artist's Website Experience

### Overall Assessment:

The satisfaction level, as indicated by users, falls within the range of 7 to 8 on a scale of 10. This implies a relatively positive sentiment among users regarding their experience with the platform. The specific positioning within this range suggests a favourable but not exceptional level of satisfaction. Analyzing feedback within this range will be crucial in identifying both strengths and areas that may benefit from improvement. This nuanced satisfaction score underscores the importance of addressing user feedback and making targeted enhancements to elevate overall satisfaction levels on the platform.Biểu đồ câu trả lời của biểu mẫu. Tên câu hỏi: Overall assessment of our website&apos;s user interface
. Số lượng câu trả lời: 3 câu trả lời.

Figure 27: Overall Website's User Interface Assessment

### General Assessment from Users:

“Positive aspects:

Visually appealing interface: High-quality images of artwork, clean layout, and intuitive navigation made browsing enjoyable.

Curated collections: The website presented interesting curated collections that helped me discover new artists and styles.

Artist information: Detailed biographies and artist statements provided valuable context for understanding the artwork.

Easy search and filtering: I could easily find specific types of art or artists I was interested in using search filters and categories.

Negative aspects:

Limited artist representation: The website seemed to focus on certain demographics or artistic styles, limiting the overall diversity of artists presented.

Pricing transparency: Some artwork descriptions lacked clear pricing information, making it difficult to make informed buying decisions.

Technical issues: I encountered a few minor bugs, such as slow loading times or broken links, which disrupted the browsing experience.”

“the web is quite good but i think the web should have more picture”

“User friendly interface, i did not need much time to learn how to use it”

# **CHAPTER X: LESSONS LEARNED:**

## **Reflection:**

### *Outcome*:

* **Interface:** we are successful in designing a simple and appealing online gallery platform for users. We aim to provide an easy-to-use and effective website to attract more customers.
* **Transactions:** The payment security is almost finished. We intend to apply more security techniques in the future to ensure the safest website for purchasing artwork.
* **Maintenance and Update:** the feedback function and other support channels through social media are established to gather feedback and improve regularly.

### *Time*:

The art online gallery process is strictly sticking to the plan and schedule. Due to the midterm, our project was delayed 2 weeks, but we adjusted it to be suitable for the current phase of the project.

### *Budget*:

The cost to design the art online gallery is optimised.

### *Value*:

We gained some good feedback from audiences about the simplicity and effectiveness of the art online gallery.

## **Further Improvements:**

* Applying advanced technology to solve the problem of lagging when transactions or many customers access the web at a certain time.
* Implementation of the AI chatbot to maximise the support customer’s service.
* Improved the interface to look more fashionable.
* Expand our database to store more customers' information and artwork.

## **Insights:**

After finishing the process of this project, we have learned these skills and experiences:

* Teamwork: Collaboration between each team member through GitHub or meetings.
* The full process of development and software product includes giving ideas, planning, implementation, testing, deployment, and maintenance. Applying agile development methodologies to ensure each sprint is done correctly and fulfils the client’s requirements, as well as scheduling the project and finishing it on time.
* Basic knowledge about coding web using CSS, HTML, Python, JavaScript, Database, …
* Presentation (arguments, present ideas, evaluation, etc)
* Problem-solving skills.
* Design PowerPoint slides professionally.
* Make a professional speech for presentation.

## **Recommendations:**

* **Enhanced User Engagement Metrics:** Implement advanced analytics to gather more comprehensive user engagement metrics. This can include detailed insights into user behaviour, popular artworks, and time spent on the platform. Enhanced analytics will provide valuable data for future improvements.
* **Interactive Art Discovery Features:** Introduce interactive features to enhance art discovery. Incorporate augmented reality (AR) or virtual reality (VR) capabilities to allow users to visualise artworks in their own spaces before purchase. This can revolutionise the online art-buying experience.
* **Real-time Collaboration with Artists:** Facilitate real-time collaboration between artists and buyers. Introduce a messaging or chat feature that allows potential buyers to communicate directly with artists, seek insights into the creative process, or request customisations.
* **AI-Powered Art Recommendations:** Implement artificial intelligence (AI) algorithms for personalised art recommendations. Analyse user preferences, purchase history, and interactions to provide tailored suggestions, making the art discovery process more intuitive and enjoyable.
* **Blockchain for Art Authentication:** Explore blockchain technology for art authentication. Implementing blockchain can enhance the transparency and authenticity of artworks, assuring buyers of the originality and provenance of the pieces they purchase.
* **Expanded Marketing Efforts:** Expand marketing efforts to reach a broader audience. Utilise social media campaigns, influencer collaborations, and partnerships with art-related events to increase platform visibility and attract a diverse range of users.
* **Mobile Application Development:** Develop a dedicated mobile application for both Android and iOS platforms. A mobile app will enhance accessibility, allowing users to explore and interact with the Art Online Gallery on their smartphones and tablets.
* **Gamification for User Engagement:** Introduce gamification elements to boost user engagement. Consider incorporating challenges, achievements, or reward systems to incentivise users to explore more artworks, interact with artists, and make purchases.
* **Expanded Payment Options:** Diversify payment options to cater to a global audience. Integrate additional payment gateways and consider supporting cryptocurrencies, providing users with a variety of secure and convenient payment methods.
* **Community Forums and Events:** Create community forums or virtual events within the platform. These spaces can encourage discussions, art critiques, and virtual exhibitions, fostering a sense of community among artists and art enthusiasts.

# **CHAPTER XI: CONCLUSION:**

## **Summary of the Outcomes:**

In the final report, we also did almost the fundamental part of the Online Art Gallery project. The report outlines the objectives and goals, implementation progress, testing procedures, quality assurance measures, deployment process, results, challenges, and documentation. Building the “Online Art Gallery" project has created opportunities for team members to learn the entire process of developing software. We are not only learning about the technician for building a basic website such as coding skills and fixing bugs, frontend, backend, and databases but also knowing how to make decisions, schedule, plan, cooperation, and argumentation to obtain the result of the project, although it is not the perfect version one.

## **Reiteration of Key Achievements:**

The project's main achievements are summarised in the following way.

In the first place, we have developed a comprehensive user registration and verification system to ensure users' security and privacy. This feature facilitated the creation of a trusted environment for artists to display their works, as well as for art lovers to explore and take an interest in art.

Second, this project has been able to introduce advanced search and category functionality that makes it easy for users to find a large collection of works. This functionality makes it possible for art lovers to discover new artists and works of art based on their preferences, thereby enhancing the overall user experience.

In addition, a smooth transaction between artists and buyers has been made possible by integrating a secure payment gateway. This feature has also helped artists to sell their works more effectively, giving art enthusiasts a chance to support their favourite artists by buying their creations.

Finally, to improve the overall user experience and accessibility, the project has shown a simplified and user-friendly interface. By using a variety of support channels, clients and users will be acquainted with the Online Art Gallery web.

## **Closing Remarks:**

The project-building process brought the team new experiences, albeit with a few shortcomings including registration and log-in, browsing and category, secure payment processing, and scalability has resulted in an online platform that fulfils the needs and expectations of both artists and art enthusiasts. Because of the time limit, there are still some parts that have not been completed and the team will work on them in the future. The team is also grateful for the teachers' attention and for pointing out the shortcomings of the art online project.

# **CHAPTER XII: APPENDICES**

## **GitHub:**

Click on the following link to view the code fully via GitHub: https://github.com/panadolextra91/Software-Engineering.git

## **Glossary of Terms:**

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Agile Methodology | A flexible and iterative project management approach that adapts to changing requirements and promotes continuous improvement throughout the development process. |
| Application Programming Interface (API) | A set of rules and protocols that allows different software applications to communicate with each other. It defines the methods and data formats for requests and responses, enabling seamless integration between diverse systems. |
| Architecture Diagram | An architecture diagram represents the high-level structure and organisation of a system's components. It may include modules, layers, components, and their relationships, offering a conceptual view of the system's architecture. |
| Backend | The backend is the server-side of a software application, responsible for processing requests, managing data, and performing operations that are not directly visible to the user. It often involves server logic, databases, and application integration. |
| Buffer Time | Additional time incorporated into the project timeline to account for unforeseen delays or challenges, providing flexibility to adjust schedules without affecting critical milestones. |
| Cascading Style Sheets (CSS) | A style sheet language used to describe the presentation and formatting of HTML documents. It enables the separation of content and design, allowing developers to control the layout, colors, and styles of web pages. |
| Class Diagram | A class diagram in UML depicts the static structure of a system by illustrating classes, their attributes, methods, and relationships. It provides a blueprint for the classes in a system and their associations. |
| CRUD Operation | CRUD stands for Create, Read, Update, and Delete. It refers to the basic operations that can be performed on data in a database or storage system. These operations are fundamental for interacting with and managing data. |
| Data Flow Diagram (DFD) | A data flow diagram represents the flow of data within a system. It illustrates how data moves between processes, data stores, and external entities, providing insights into the system's data processing and storage. |
| Deployment Diagram | A deployment diagram in UML visualises the physical deployment of software components on hardware nodes. It shows how software artifacts are distributed across different nodes, facilitating understanding of system architecture and deployment configurations. |
| Embedded JavaScript (EJS) | A templating language used in the project's frontend for generating dynamic content on the login page. |
| Encryption Protocols | Techniques used to implement robust encryption for safeguarding user data during transmission and storage. |
| Entity-Relationship Diagram (ER Diagram) | An ER diagram is a visual representation of the data model that defines entities, their attributes, and the relationships between entities in a database. It is commonly used in database design to depict the structure of a database system. |
| Framework | A pre-built structure or set of tools that provides a foundation for developing software applications. It includes reusable code, libraries, and conventions to help developers build applications efficiently and consistently. |
| Frequently Asked Questions (FAQs) | A compilation of common questions and their respective answers related to a product, service, or topic. They serve as a resource to address user queries and provide quick solutions to common issues. |
| Frontend | The frontend refers to the user interface and user experience components of a software application or website that users interact with directly. It encompasses design, layout, and functionality visible to the end user. |
| GitHub | A web-based platform for version control using Git. It provides tools for collaboration, code review, and project management. Developers use GitHub to host and share code repositories, track changes, and work on projects collaboratively. |
| Google Drive | A cloud-based file storage and synchronisation service provided by Google. It allows users to store files in the cloud, synchronise them across devices, and collaborate with others through shared documents, spreadsheets, and presentations. |
| Hypertext Markup Language (HTML) | A standard markup language used to create the structure and content of web pages. It utilises tags to define elements such as headings, paragraphs, links, images, and other components of a webpage. |
| Import Statements | Statements in Java (e.g., import java.io.File;) used to bring in necessary classes and functionalities for file handling, database connectivity, SQL operations, and date/time handling. |
| Integrated Development Environment (IDE) | A software application that provides comprehensive tools and features to facilitate the development of software. It typically includes code editor, debugger, compiler, and other utilities, streamlining the development process. |
| Java | Java is a high-level, versatile, and object-oriented programming language. Known for its platform independence, Java is widely used for building scalable, robust, and enterprise-level applications. |
| Java Database Connectivity (JDBC) | A Java-based API that enables Java applications to interact with relational databases, such as MySQL. In this context, JDBC is used for database connectivity in the project. |
| JavaScript (JS) | A scripting language that enables the creation of dynamic and interactive content on web pages. It is commonly used for client-side scripting to enhance user experience by handling events, modifying content, and interacting with the browser. |
| JavaScript Object Notation (JSON) | A lightweight data interchange format that is easy for humans to read and write, and easy for machines to parse and generate. It is often used to transmit data between a server and a web application as an alternative to XML. JSON is based on a subset of the JavaScript programming language. |
| Library | A collection of pre-written code or functions that developers can use in their software projects. Unlike a framework, a library does not dictate the overall structure of an application but provides specific functionalities. |
| Middleware | Middleware is software that acts as an intermediary layer between different applications or components in a distributed computing environment. It facilitates communication and data management between disparate systems, often handling tasks such as data translation, security, and transaction management. |
| Mitigation Strategies | The proactive plans and actions devised to address and minimise the impact of identified risks, ensuring effective risk management and project resilience. |
| Objectives | Specific, measurable, and time-bound goals that an individual, team, or organisation aims to achieve. They provide a clear direction and purpose, guiding efforts toward desired outcomes. |
| Online Art Gallery/The project | An innovative digital platform designed to streamline the art industry by connecting artists and art buyers, providing a central hub for art display, discovery, and transactions. |
| Quality Assurance (QA) | Systematic monitoring and evaluation to ensure standards are met. |
| Quality Assurance Measures | Rigorous testing procedures, including code cross-checks, stakeholder discussions, and flexible resource allocation, implemented to ensure optimal program performance and user satisfaction. |
| RESTful Interface | RESTful, which stands for Representational State Transfer, is an architectural style for designing networked applications. A RESTful interface is an application programming interface (API) that adheres to the principles of REST. It uses standard HTTP methods (GET, POST, PUT, DELETE) to perform operations on resources, providing a scalable and stateless communication approach. |
| Risk Assessment | The identification and evaluation of potential challenges or uncertainties that may impact the project, including technical, resource-related, timeline, security, user experience, and market competition risks. |
| Role-based Authentication | A system where access and functionalities are determined based on the user's role (e.g., Admin, Customer, Artist). |
| Rollout Strategy | A plan for the gradual and controlled implementation of a new product, feature, or system. It involves phased releases, testing, and monitoring to ensure a smooth transition and minimise potential risks or issues. |
| Scope | The boundaries and extent of a project. It defines what is included and excluded from the project, outlining the goals, deliverables, tasks, and constraints. |
| Sequence Diagram | A sequence diagram in UML represents the interactions between different components or objects in a specific order over time. It shows the flow of messages and actions between objects, helping to visualise the dynamic behavior of a system. |
| SQLite | A lightweight, embedded relational database management system. It is self-contained, serverless, and widely used for applications where simplicity, minimal configuration, and ease of use are essential. |
| Stakeholder | An individual, group, or entity that has an interest, involvement, or influence in a project or organisation. Stakeholders may include customers, employees, investors, partners, and others who can affect or be affected by the project. |
| System Diagram | A system diagram provides an overview of the components and interactions within a system. It may include hardware, software, data, and communication elements, giving a holistic perspective on the system's structure. |
| Technical Resilience | The ability of the project to overcome technical challenges, including system integrations and compatibility issues, ensuring a robust and reliable online gallery. |
| Technology Diagram | A technology diagram illustrates the technology stack and infrastructure components used in a system. It showcases the hardware, software, and network elements that contribute to the system's functionality. |
| Transaction mechanism | The process implemented on the platform to facilitate secure and efficient financial interactions between artists and art buyers. |
| Unified Modeling Language Diagram (UML Diagram) | Visual representations of system structures and behaviors used in software engineering. The Unified Modeling Language provides a standardised way to depict various aspects of a system, facilitating communication among stakeholders. |
| Unique Selling Proposition (USP) | The distinctive features and advantages that set the Art Online Gallery apart from competitors, contributing to its appeal and value proposition in the market. |
| Use Case Diagram | A use case diagram in UML illustrates the interactions between actors (users or external systems) and a system, showcasing various use cases or functionalities. It provides a high-level view of how users interact with the system. |
| Use Case (UC) | A written description of how users will perform tasks on your website. |
| User Experience (UX) | The overall experience a user has when interacting with a product or system. It encompasses usability, accessibility, aesthetics, and the overall satisfaction users derive from their interactions. |
| User Interface (UI) | The visual and interactive elements of the online gallery that users interact with, including the website layout, design, and features. |
| User’s Manual/User Guide/Instruction Manual | A document that provides information and instructions on how to use a product or system. It typically includes details on features, setup, troubleshooting, and maintenance. |
| Visual Studio Code (VS Code) | A source-code editor developed by Microsoft for Windows, Linux, and macOS, used as the development environment for coding the project. |
| XAMPP | A free, open-source web server solution that simplifies the setup of a local development environment. It includes Apache, MariaDB, PHP, and Perl, offering an easy-to-install platform for testing and developing web applications on personal computers. |
| Zalo | A popular messaging and social media app in Vietnam. It provides features such as text and voice messaging, video calls, social networking, and more. |

Table 42: Glossary of Terms

## **Survey Questionnaire:**

To view the entire survey questionnaire, please click on the following link: https://docs.google.com/forms/d/e/1FAIpQLSdzv9TZ6cSJaM62fS30cp52RW6rOopgkj65oBsyvJbuGSK4GQ/viewform

**Survey Name: Customer Survey Based on Actual Situation**

**Section 1: Compulsory Information**

Email:

Name:

ID:

**Section 2: User’s Satisfaction**

On a scale from 1 (very bad) to 10 (very good), please rate the following:

How do you rate our login interface?

How is your experience using our website as a user?

How is your experience using our website as an artist?

Overall assessment of our website's user interface?

**Section 3: General Assessment:**

Please give us a general assessment of your experience, including any error problems you encountered, and positive and negative aspects of our website.

## **Stakeholder Communication Plan (Draft):**

* **Stakeholder Identification:** List of identified stakeholders, categorised by their level of influence and interest in the project.
* **Communication Objectives:** Clear objectives for each stakeholder group, specifying the desired outcomes of communication.
* **Communication Channels:** A summary of communication channels selected for each stakeholder group, including meetings, emails, and collaboration tools.
* **Frequency of Communication:** A schedule indicating how often communication will occur with each stakeholder group, ensuring timely updates without information overload.
* **Key Messages:** Core messages to be conveyed to stakeholders, aligned with project goals, and addressing their specific concerns.
* **Feedback Mechanism:** Description of mechanisms for stakeholders to provide feedback, ensuring a two-way communication flow.
* **Escalation Procedures:** Procedures for addressing issues or concerns that require escalation to higher levels of management or decision-makers.
* **Monitoring and Evaluation:** Criteria for assessing the effectiveness of communication strategies, allowing for adjustments based on stakeholder responses.

## **Goals and Objectives Review:**

In this section, a comprehensive review of the project's goals and objectives is conducted to assess progress and ensure alignment with the project's overarching vision. The revisit emphasises the continued relevance of each project goal to the overall mission, reinforcing its significance in achieving project success.

Each specific objective associated with the established goals undergoes a detailed assessment. This evaluation focuses on completion status and the contribution of each objective to the overarching goals. It provides a nuanced understanding of how individual objectives collectively contribute to the project's success.

Alignment with the defined project scope is a key aspect of this review, ensuring that goals and objectives remain within the project's original parameters. This prevents deviations that could potentially impact the project's overall success and ensures that efforts are consistently directed towards the intended outcomes.

Measurable outcomes linked to each objective are scrutinised to determine the extent of their achievement. This analysis identifies areas of success and highlights any objectives that may require further attention or refinement to meet the desired standards.

The review also includes an examination of the timeline associated with each goal and objective. This ensures that the project adheres to its established schedule, allowing for timely completion and delivery of outcomes.

Stakeholder satisfaction is considered by incorporating feedback regarding the goals and objectives. This feedback loop ensures that the project remains responsive to stakeholder expectations, fostering a collaborative and inclusive approach to project management.

Identification and assessment of risks that may have affected the attainment of goals and objectives are integral to this review. Corresponding mitigation strategies are reviewed and, if necessary, adjusted to enhance the project's resilience to unforeseen challenges.

Lessons learned during the pursuit of goals and objectives are reflected upon. These insights contribute to a collective understanding of successful strategies and areas for improvement, providing valuable guidance for future projects and supporting a culture of continuous improvement.

This review section serves as a pivotal checkpoint in the project management process. It offers nuanced insights into the project's trajectory, identifies successes, and highlights areas for adjustment, ensuring a comprehensive and informed approach to project completion.

# **CHAPTER XIII: REFERENCES**

1. EJS Documentation:

EJS. (n.d.). Documentation. Retrieved November 16, 2023, from <https://ejs.co/#docs>

1. YouTube Channel - Nịt Coder:

Nịt Coder. (n.d.). YouTube. Retrieved December 05, 2023, from <https://www.youtube.com/@than369/featured>

1. YouTube Video - Git and GitHub for Beginners - Crash Course:

freeCodeCamp.org. (2020 May 28). Git and GitHub for Beginners - Crash Course [Video]. YouTube. <https://www.youtube.com/watch?v=RGOj5yH7evk>

1. Governors State University - OPUS Open Portal to University Scholarship:

Vasquez, F. E. (Year). Title of the Document. OPUS Open Portal to University Scholarship. Retrieved November 01, 2023, from <https://opus.govst.edu/cgi/viewcontent.cgi?article=1507&context=capstones>

1. Online Gallery - Art Example:

Artist Last Name, First Initial. (n.d.). Title of the Artwork. Online Gallery. <https://onlinegallery.art/en/>

1. Mozilla Developer Network:

Mozilla. (n.d.). Resources for Developers, by Developers. Mozilla Developer Network. Retrieved November 12, 2023, from <https://developer.mozilla.org/fr/>

1. XAMPP - About:

Apache Friends. (n.d.). About XAMPP. Apache Friends. Retrieved October 30, 2023, from <https://www.apachefriends.org/about.html>

1. Heroku - Main Server:

Heroku. (n.d.). Our Main Server. Heroku. Retrieved December 06, 2023, from <https://www.heroku.com>

# **ACKNOWLEDGEMENTS**

We extend our sincere gratitude to all those who have contributed to the completion of this report. This project would not have been possible without the support, guidance, and collaboration of various individuals and organisations.

We express our deepest appreciation to Mrs. Nguyen Thi Thuy Loan and Mr. Nguyen Quang Phu, whose expertise and encouragement have been invaluable throughout the entire duration of this project. Their guidance has been instrumental in shaping the direction of the research and ensuring its academic rigour.

We are thankful to the International University (IU), HCMC, VNU, for providing the necessary resources and environment conducive to research and learning. The facilities and infrastructure have played a crucial role in the successful execution of this project.

We would also like to acknowledge the contributions of our colleagues and peers who have provided insights, feedback, and support at various stages of the project. Their diverse perspectives have enriched the research and added depth to its findings.

Our appreciation extends to our friends and family for their unwavering support, understanding, and encouragement throughout this academic journey. Their belief in our abilities has been a constant source of motivation.

Lastly, we are grateful to all the participants and stakeholders involved in this project, including Ms. Nguyen Quynh Mai Thanh and the development teams, for their cooperation and willingness to share their experiences and expertise. Their involvement has been instrumental in shaping the practical aspects of this research.

This report stands as a collective effort, and we are deeply appreciative of the collaborative spirit that has fueled its completion. Each contribution, no matter how small, has played a vital role in bringing this project to fruition. Thank you to everyone who has been a part of this journey.