



SOFE 3650U Software Design and Architecture

Art Gallery Website: Project Report

Student	Name
Manreet Kaur	100766207
Haiqa Tikka	100739498
Ammar Salmawy	100756573

System's Requirements

The following is the summary of the relevant requirements collected:

Functional Requirements :

1.) Sales Page :

Sales page is required to display and buy all the art which is for sale. There should be an option to select the required art piece and its quantity.

2.) Code Requirement :

All the coding should be accessible to the website owner and it should be commented properly. Code should be maintainable.

3.) Collection Page :

Collection page is required to display collection of art pieces. The art pieces can be removed as well as updated with time in this page.

4.) Authentication:

There should be a login and logout option so that website can validate the users and users can exit the website safely. This will prevent accessing of each others payment methods.

5.) Feedback :

The system must allow the user to submit feedback through a contact form.

6.) Payment :

The system is required to accept multi payment methods such as credit cards and online banking services, PayPal.

7.) Account Page :

The account page is required to display past purchases and user information.

8.) Bug-free :

The system should be tested for bugs and then repaired prior to launch.

9.) Cross-browser functionality :

The system should perform identically across all popular browsers.

Non-functional Requirements :

1.) User Friendly :

The website should have user friendly features. It should have easy access, simple functions and interactive design.

2.) Background of the website :

There should an art piece image at the background of the website and the theme of the website should be dark.

3.) Time requirements :

Once clicked on an icon, the result should be accessed as fast as possible. For example, when sales page is clicked, sales page should open in less than a minute.

4.) Manageability :

The system shall be easy to monitor and manage by administrators. such as updating available items keeping track of orders and transactions

5.) Security :

The system shall protect the data against malware attacks. i.e. adding errors handlers to prevent database injection

6.) Eye-catching :

The website should be eye-catching to increase attention and generate more sales.

Use Case Model :

The following use case model presents the most relevant use cases that support the art gallery website model. Other use cases are not shown.

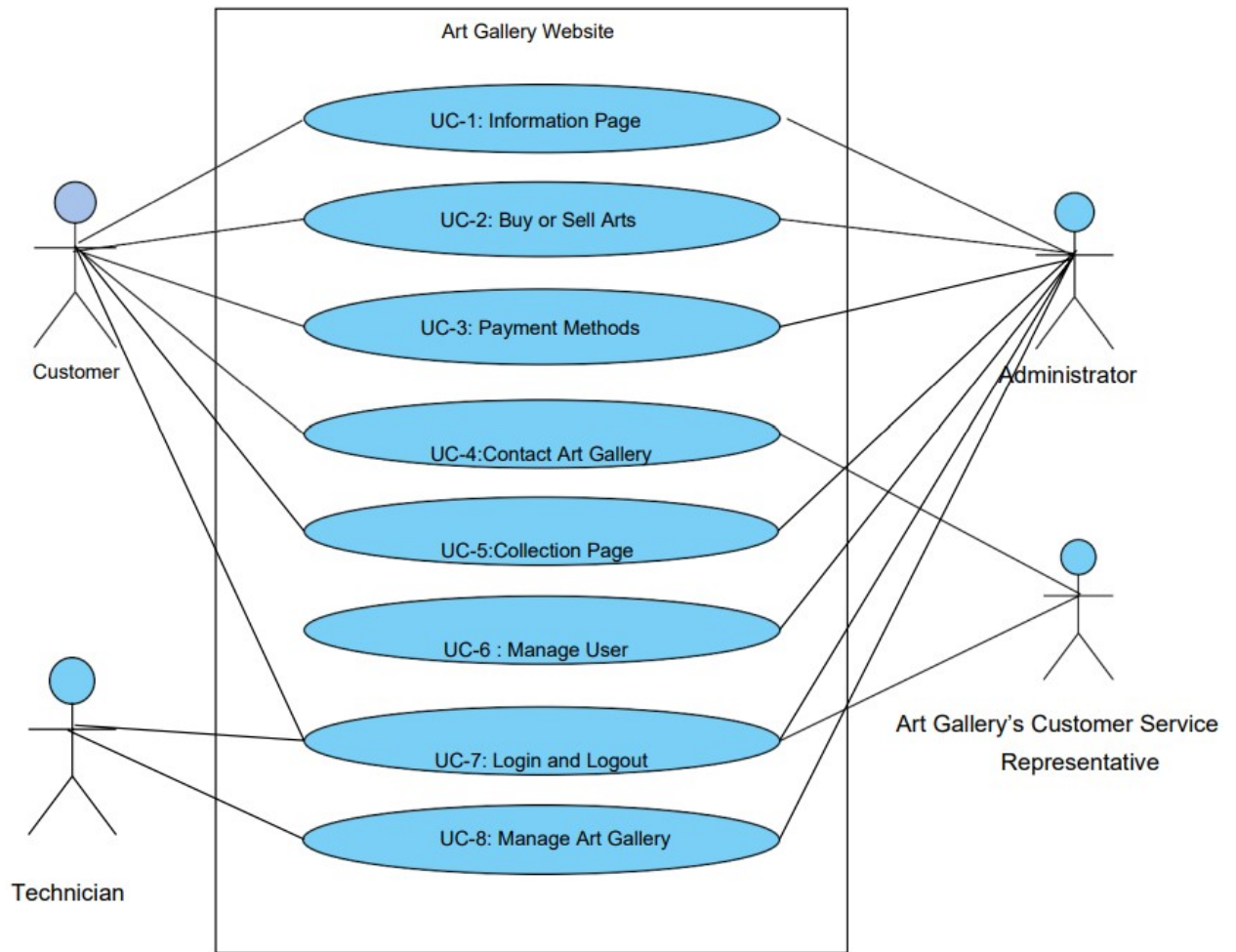


Figure 1:Use Case Model for Art Gallery Website

Each of the use case has been described in the table below :

<u><i>Use Case</i></u>	<u><i>Description</i></u>
UC-1: Information Page	A user can access the information about the art gallery website, history of art gallery website, and the events of the upcoming art gallery and its location Administrator can update this information, upload new information, new events description.
UC-2: Buy or Sell Arts	A user can buy art pieces from the sales page and can select the quantity required to buy. Administrator has the access the to update the art pieces for selling. Administrator can also buy the art pieces from the artists.
UC-3: Make Payment	A user can use the different payment methods to buy the desired art pieces. Administrator can buy art from the users that are selling the art pieces by making payment through this method.
UC-4: Contact Art Gallery	A user can provide feedback, express concerns, make a complaint regarding the services provided at art gallery website through Art Gallery contact page. A customer service representative can see the complaints and concerns of the users through art gallery contact page and can access them accordingly.
UC-5: Collection Page	A user can see the different art pieces listed on the collection page. Collection Page is only meant to display the art pieces. Users can access this page free of cost. Administrator can remove, add or modify the art pieces on this page.
UC-6: Manage Users	The administrator removes or adds the users. Administrator can modify the permissions of the users. User can be removed for a limited period or permanently by the administrator .
UC-7: Login and Logout	A user login into the system through a password /login page will be promoted when the user clicks on sign in. Once the user has been authorized the access to the art gallery, user can navigate the art gallery website according to the roles. Administrator have access to the login and passwords of every user of the art gallery website.
UC-8: Manage Art Gallery	Administrator have access to the art gallery and manages the amount of art pieces displayed, any information or issues related to the art gallery. Technician manages the art gallery by fixing the concerns related to the art gallery functioning. Technician is responsible for the overall maintenance of the art gallery website.

Quality Attribute Scenarios :

Quality Attribute	Description	Associate Use Case(s)
QA-1: Performance	The User should not experience delays of more than 5 seconds in retrieving information from the site.	All UC
QA-2: Usability	The User has many methods of paying for the art they desire. The system should reflect this and offer many payment methods (bank transfer, e-transfer, cash by mail etc....)	UC-3
QA-3: Testability	The system may sometimes receive invalid inputs. The system should be able to process these inputs within 10 seconds per invalid input.	All UC
QA-4: Security	User's will store private financial information on the site to buy and sell art. Therefore, the system should encrypt the private information of its users like login and financial info.	All UC
QA-5: Modifiability Usability	The system administrator should be able to modify the art gallery.	UC-1, UC- 2, UC-5, UC-8
QA-6: Testability	The system must be able to complete art transactions	All UC
QA-7: Availability	The system must continue to operate and be <u>available all</u> the time. For UC-4 availability must be predefined	All UC
QA-8: Interoperability	The system must exchange data with external systems such as banking services and online payment providers	UC-3
QA-9: Performance	The users can see new art pieces within thirty minutes from an update by the administrator.	UC-5
QA-10: Security	The management can easily find out the different modes of payments and which users have made the payments	UC-3
QA-11: Security	The administrator can decide which users can be added or removed from the art gallery due to any security concerns which might be temporary or	UC-6

Constraints :

Constraint	Description
CON-1	The system must be accessible and run smoothly by all popular web browsers (<u>mozilla</u> , chrome, edge <u>etc...</u>) and all popular operating systems (Windows, mac <u>os</u> x). Additionally, the system must also support mobile devices.
CON-2	All user data including financial and transactional records must be stored indefinitely.
CON-3	The system must have a mobile-friendly design.
CON-4	The system must work properly when viewed with different resolution monitors. The system must work when the view is stretched or shrunk.
CON-5	All art uploaded to the system must have the authors permission.

Architectural Concerns :

ID	Concern
CRN-1	Establishing an overall system structure.
CRN-2	Allocate work to the members of the development team

The Architecture Design Process

Architecture Driven Design (ADD) Step1: Review Inputs

The inputs are summarized in the following table :

Category	Details											
Design Purpose	Art Gallery is a website that allows users to create accounts to browse and buy art pieces. The system features a login/account system with a payment system for ordering art. The system administrator manages the website by adding/removing art pieces, changing prices, and editing accounts, among other privileges.											
Primary Functional requirements	UC-2: Buy or Sell arts: This directly supports the core of the project through the purchasing of art pieces and the ability to add to the database by the administrator UC-5: Collection Page: This directly supports the core of the project through the ability to modify the art piece collections page UC-8: Manage Art Gallery: This directly supports the core of the project through the ability to modify the art pieces database. It also supports modifiability of the back end with a technician											
Quality attribute scenarios	<table><tr><th>Quality Attribute Scenario</th><th>Importance to the Customer</th><th>Difficulty of Implementation According to the Architect</th></tr><tr><td>QA-1</td><td>low</td><td>low</td></tr><tr><td>QA-2</td><td>high</td><td>medium</td></tr></table>			Quality Attribute Scenario	Importance to the Customer	Difficulty of Implementation According to the Architect	QA-1	low	low	QA-2	high	medium
Quality Attribute Scenario	Importance to the Customer	Difficulty of Implementation According to the Architect										
QA-1	low	low										
QA-2	high	medium										

	QA-3	low	low
	QA-4	high	medium
	QA-5	medium	low
	QA-6	high	low
	QA-7	high	low
	QA-8	high	medium
	QA-9	low	low
	QA-10	low	medium
	QA-11	medium	low
Constraints	All the concerns discussed in initially are included as drivers		
Architectural concerns	All the architectural concerns discussed initially are included as drivers		

Iteration 1: Establishing an Overall System Structure

Iteration 1 outlines the initial results of the design process for the ADD Steps .

Step 2: Establish Iteration Goal by Selecting Drivers

The goal of this iteration is to achieve the architectural concern CNR-1 of establishing an overall system structure. Keeping this in mind, there are several drivers that we as the architects must account for:

1. QA-2: The User has many methods of paying for the art they desire. The system should reflect this and offer many payment methods (bank transfer, e-transfer, cash by mail etc...)
2. QA-4: User's will store private financial information on the site to buy and sell art. Therefore, the system should encrypt the private information of its users like login and financial info.
3. QA-6: The system must be able to complete art transactions
4. QA-7: The system must continue to operate and be available all the time. For UC-4 availability must be predefined
5. QA-8: The system must exchange data with external systems such as banking services and online payment providers
6. CON-1: The system must be accessible and run smoothly by all popular web browsers (mozilla, chrome, edge etc...) and all popular operating systems (Windows, mac os x). Additionally, the system must also support mobile devices.
7. CON-2: All user data including financial and transactional records must be stored indefinitely.
8. CON-3: The system must have a mobile-friendly design.
9. CON-4: The system must work properly when viewed with different resolution monitors. The system must work when the view is stretched or shrunk.
10. CON-5: All art uploaded to the system must have the author's permission.
11. CRN-1: Establishing an overall system structure.

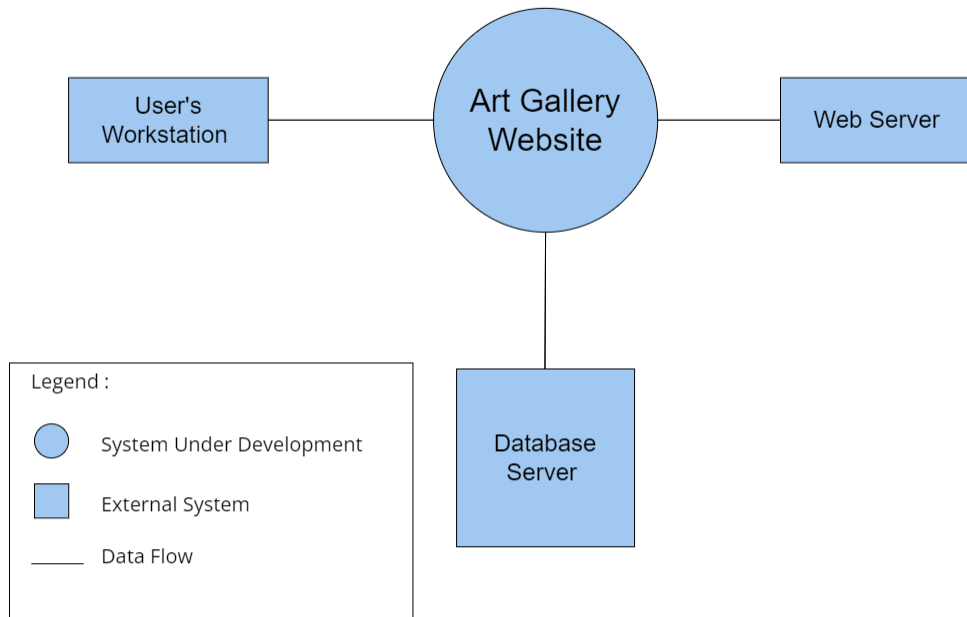


Figure 2: Context Diagram for the Art Gallery Website

Step 3: Choose One or More Elements of the System to Refine

This is a greenfield project for a mature domain, so the entire art gallery system is to be refined.

All components should be revised so the overall performance and functionality can be improved. refinement is performed through decomposition.

Step 4: Choose One or More Design Concepts That Satisfy the Selected Drivers

The following table summarizes the selection of design decisions:

Design Decisions and Locations	Rationale
Logically structure the client and server part of the system using the Web Application reference architecture	The Web Applications reference architecture is selected for its overall usability to construct non-rich web applications. This is because our system does not require a rich user interface, it does not need to install anything on the client machine, and it must be accessible over the internet by web browsers (CON-1). This design decision partially affects all of the other drivers, but does not directly impact any other as substantially as the first concern.
Physically structure the application using the three-tier deployment pattern	This system needs to be accessed through a web browser (CON-1) and there should be an existing database that should be used . Hence the three-tier layer deployment is appropriate .

Step 5: Instantiate Architectural Elements, Allocate Responsibilities, and Define Interfaces

The instantiation design decisions made in this iteration are summarized in the following table:

Design Decisions and Location	Rationale
Remove Application Facade for the business layer of the server	The Application facade component is unnecessary to fulfill the requirements of the project overall.
Add Payment System as a external system that communicates with the service agents	The Payment System will be an external system to process payments for the Users. Adding the payment module ensures that many payment methods can be used (UC-3, QA-2), the system can complete art transactions (QA-6), and the system uses an external payment module (QA-8).

Step 6: Sketch Views and Record Design Decision

The diagram below shows the sketch of the module view of the two reference architectures that were selected for the client and server applications. These have now been adapted according to the design decisions we have made.

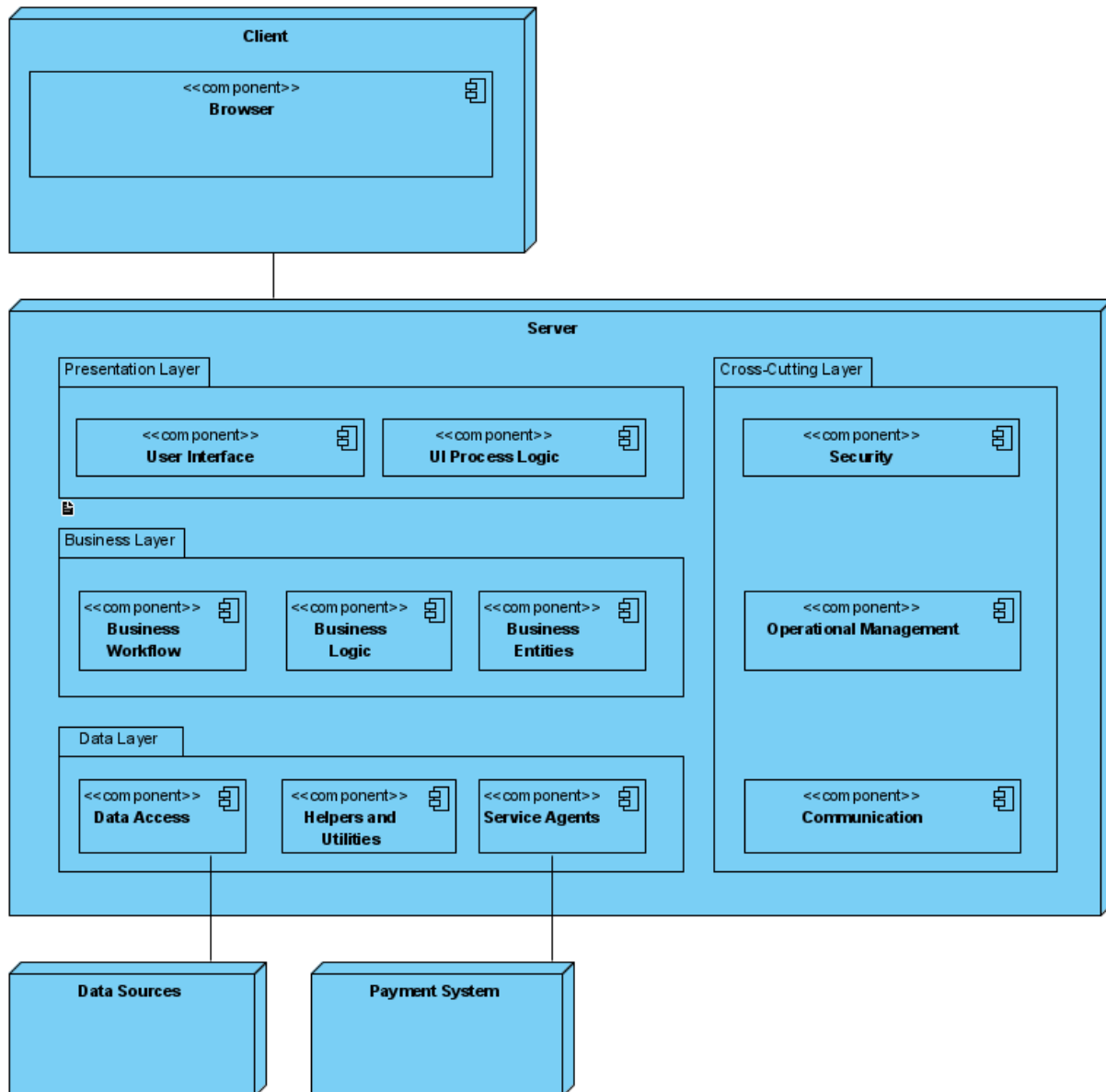


Figure 3: Module View of the website after initial design decisions:

The following table summarize each element of the module view of the diagram along with its responsibilities:

Layer	Component	Responsibility
Client		The presentation layer that communicates with the client service modules
	Browser	Application module that is used by the client to interact with the server applications to provide or display information. Runs on the client machine
Server		This layer exposes the modules and components that users will interact with
Presentation		This layer controls user interaction with the server directly, including use case interactions
	User Interface	These components are responsible for receiving/sending information to the users through inputs like buttons, text fields, etc...
	UI process logic	These components are used to direct the flow of the applications use cases. This can include data validation, providing data from business layer to presentation layer, etc...
Business Logic		Contains modules that perform business logic operations on the server side
	Business Workflow	These components are responsible for managing the processes of the business operations, involving the execution of use cases

	Business Logic	These components retrieve and process the data with business rules.
	Business Entities	These components represent the entities of the business domain
Data		This layer contains modules for data persistence and communication with external systems
	Data access	These components encapsulate persistence mechanisms to provide basic operations like retrieving and storing data
	Helpers and Utilities	These components contain functionality common to other modules in the data layer
	Service Agents	These components are necessary for communicating and transferring data between external services and the system itself
Cross-Cutting		These modules have functionality that are designed to work across multiple layers
	Security	These components include functionality to handle security aspects such as authorization and authentication
	Operational Management	These components handle cross-cutting concerns such as exception management, logging, and instrumentation and validation
	Communication	These components handle communication across the layers and physical tiers of the system

Payment Server		This layer is an external system that will communicate with the system to provide payment methods to the users on the website (UC-3, QA-2, QA-6, QA-8)
----------------	--	--

The following deployment diagram sketches an allocated view that illustrates where the components associated with the modules in Figure 3 will be deployed.

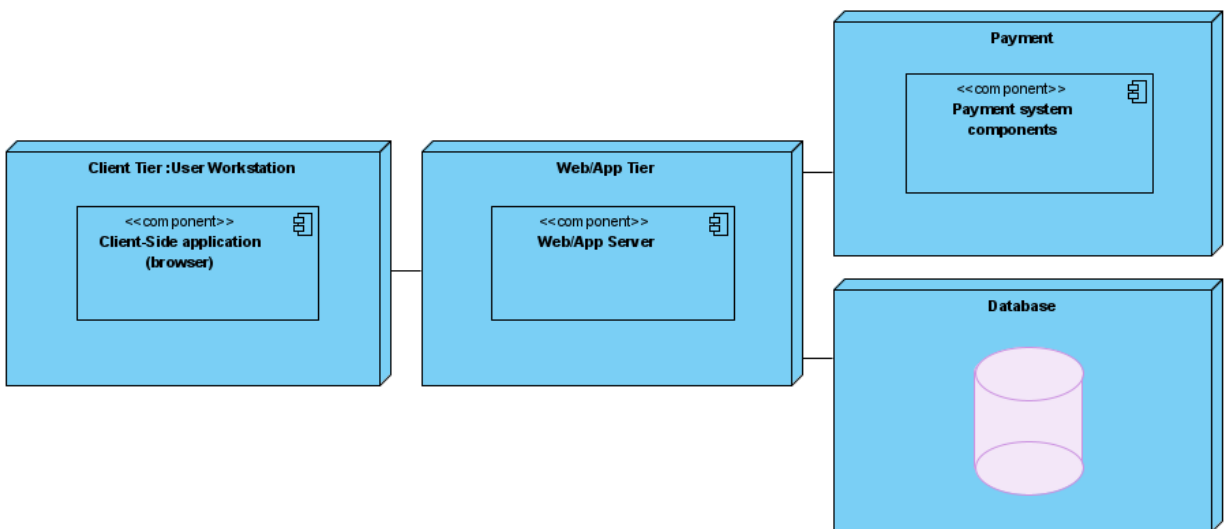


Figure 4:Deployment Diagram of the system

The following table summarizes the responsibilities of the elements :

Element	Responsibility
User Workstation	Users PC, hosts client side application, in this case the user would only need a browser to connect to the server
Web/App Server	Hosts the server-side logic and web pages of the application
Database Server	The server that hosts the database
Payment system	The external system used to make payments on the website

Step 7: Perform Analysis of Current Design and Review Iteration Goal and Achievement of Design Purpose

Not Addressed	Partially Addressed	Fully Addressed	Rationale
	UC-2		Introduction of an external payment system supports the functionality of this use-case
	UC-5		Selected reference architecture establishes required modules that will support this functionality
	UC-8		Selected reference architecture establishes required modules that will support this functionality
	QA-2		Introduction of an external payment system supports the functionality of this quality attribute
	QA-4		Selected reference architecture establishes required modules that will support this functionality
	QA-6		Selected reference architecture establishes required modules that will support this functionality
	QA-7		Selected reference architecture establishes required modules that will support this functionality
		QA-8	Introduction of an external payment system

	CON-1		Selected reference architecture establishes required modules that will support this functionality
	CON-2		Selected reference architecture establishes required modules that will support this functionality
CON-3			No relevant decisions were made
	CON-4		Selected reference architecture establishes required modules that will support this functionality
CON-5			No relevant decisions were made
		CRN-1	Selection of reference architecture and deployment pattern
CRN-2			No relevant decisions were made

Iteration 2: Identifying Structures to Support Primary Functionality

The goal of this iteration is to reason the units of implementation. This will affect the team's formation , interfaces and means by which the development task may be implemented .

Step 2: Establish Iteration Goal by Selecting Drivers

The goal of this iteration is to address the general architectural concern of identifying structures to support primary functionality. Identifying these elements is useful for understanding how functionality is supported and for addressing CRN-2:Allocate work to the members of the development team. Keeping this in mind, the following are the drivers that we as the architects must account for besides CRN-2:

- UC-2: Buy or Sell arts: This directly supports the core of the project through the purchasing of art pieces and the ability to add to the database by the administrator
- UC-5: Collection Page: This directly supports the core of the project through the ability to modify the art piece collections page
- UC-8: Manage Art Gallery: This directly supports the core of the project through the ability to modify the art pieces database. It also supports modifiability of the back end with a technician

Step 3: Choose One or More Elements of the System to Refine

The elements that are to be refined are those that are directly associated with the different layers that were previously defined by the reference architectures. Specifically, the main functional requirements must be refined in reference to the architecture designed in iteration 1.

Step 4: Choose One or More Design Concepts That Satisfy the Selected Drivers
Selected design concepts

The following table summarizes the selection of design decisions in this iteration:

Design Decisions and Location	Rationale and Assumptions
Create a Domain Model for the application	A Domain Model is useful for the system to identify major entities and their relationships within the domain. The Domain Model always exists within a system, however, the earlier it is designed the easier it becomes to understand. So, an initial Domain Model must be created early in the design process.
Identify Domain Objects that map to the functional requirements	After the initial Domain Model, each Domain Object must then be identified and encapsulated in its own building block.
Decompose Domain Objects into generalized and specialized Components	Once the Domain Objects have been identified and encapsulated, they must be specialized into modules and components that are specific to the layer they are located in.

Step 5: Instantiate Architectural Elements, Allocate Responsibilities, and Define Interfaces

The instantiation design decisions made in this iteration are summarized in the following table:

Design Decisions and Location	Rationale
Create only an initial domain model	An initial domain model is created to accelerate this design phase . In this domain model , the entities of the primary use cases are identified and modeled .
Map the system use cases to domain objects	System's use cases are analysed which results in the identification of domain objects. Domain objects of all the use cases are identified to address CRN-2.
Decompose the domain objects across the layers to identify layer-specific modules with an explicit interface	This method of working confirms that the module that maintains all of the features are recognized. The use case will be handled by the architect. This helps the remaining team members to identify the module, which allows the work to be equally distributed.

Step 6: Sketch Views and Record Design Decisions

Several diagrams are created as a result of design decisions made in step 5 are as follows:

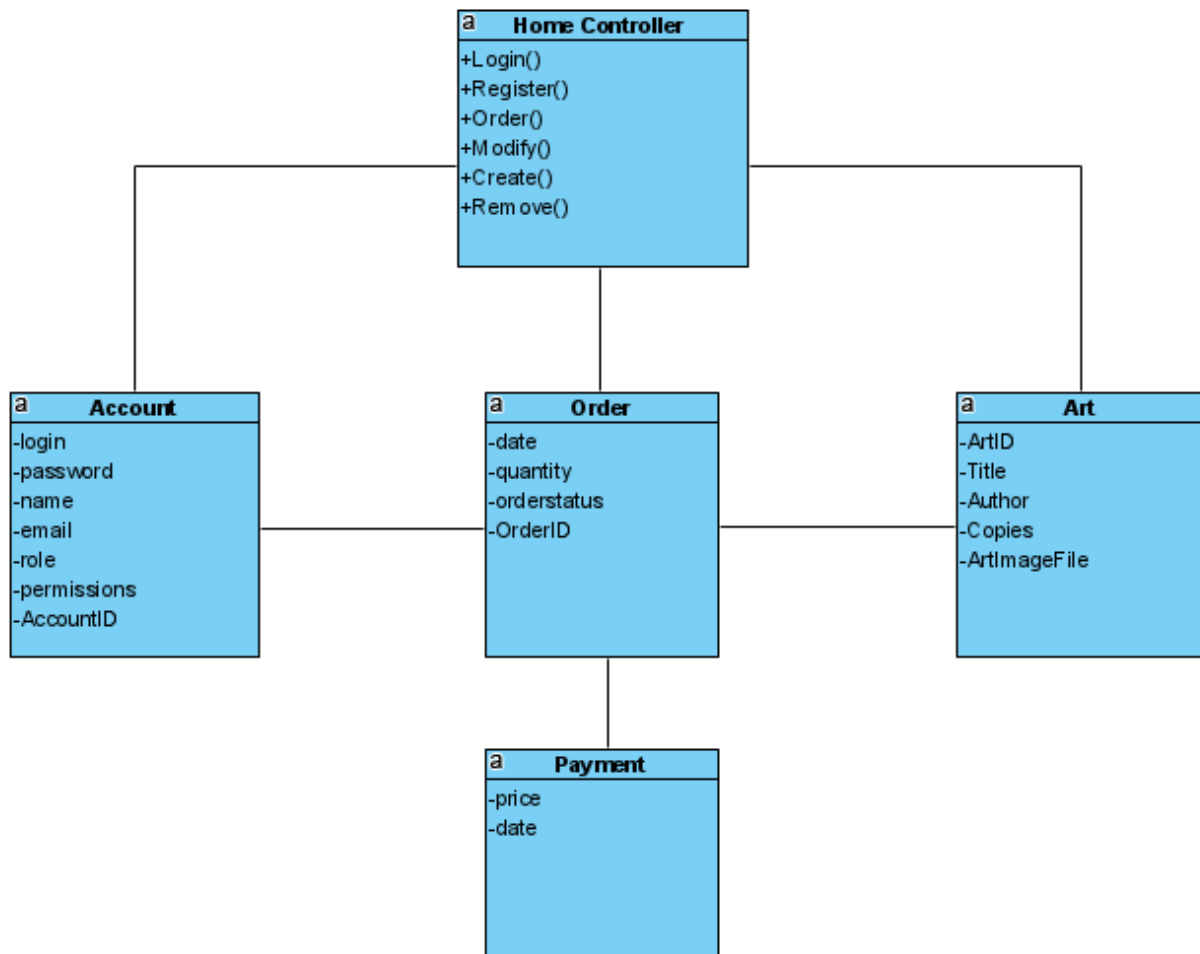


Figure 5: Initial Domain model for the Art Gallery Website

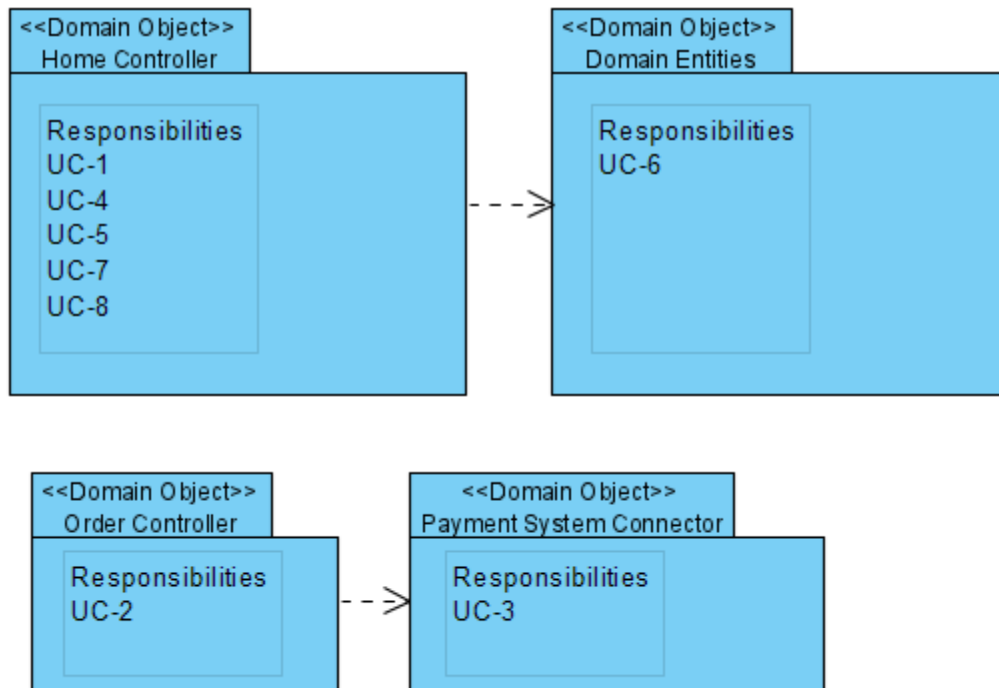


Figure 6:Domain Objects associated with the Use Case Model.

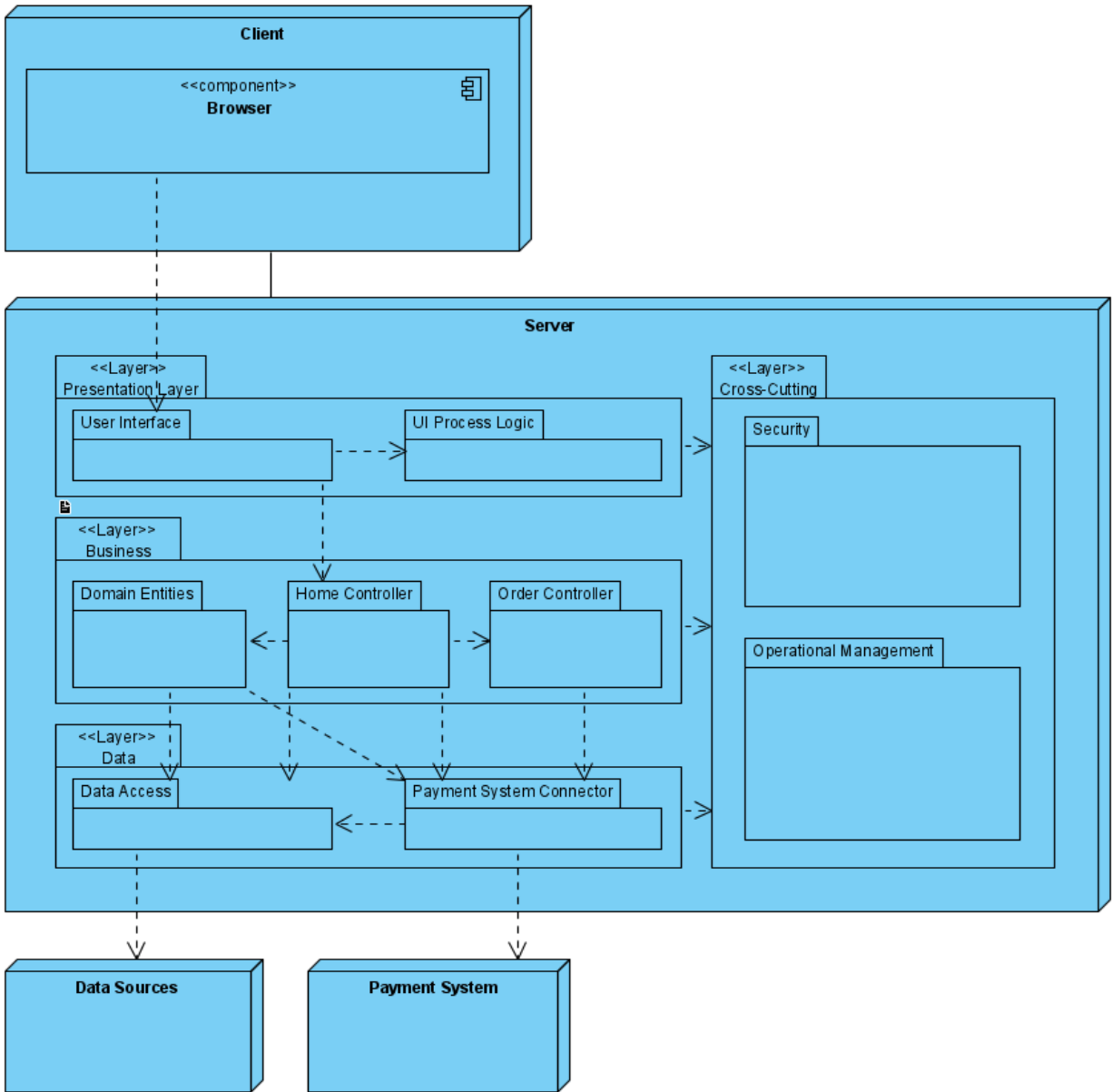


Figure 7:Module view with modules that support primary use cases

The following summarizes the responsibilities for the elements identified in Figure 7:

Element	Responsibility
Browser	Application module that is used by the client to interact with the server applications to provide or display information. Runs on the client machine

User Interface	These components are responsible for receiving/sending information to the users through inputs like buttons, text fields, etc...
UI Process Logic	These components are used to direct the flow of the applications use cases. This can include data validation, providing data from business layer to presentation layer, etc...
Domain Entities	Contains the entities from the domain model. These include account processing, registration, etc...
Home Controller	Contains business logic pertaining to most use cases of the system. This includes logging in, registering, modifying the art gallery, etc... (UC-1, UC-4, UC-5, UC-7, UC-8)
Order Controller	This controller processes business logic pertaining to the ordering of artwork (UC-2)
Data Access	This module encapsulates persistence mechanisms to provide basic operations like retrieving and storing data
Payment System Connector	This connector is responsible for communication between the order controller and the external payment system, in other words a service agent
Security	These components include functionality to handle security aspects such as authorization and authentication
Operational Management	These components handle cross-cutting concerns such as exception management, logging, and instrumentation and validation

Sequence Diagrams for primary use cases:

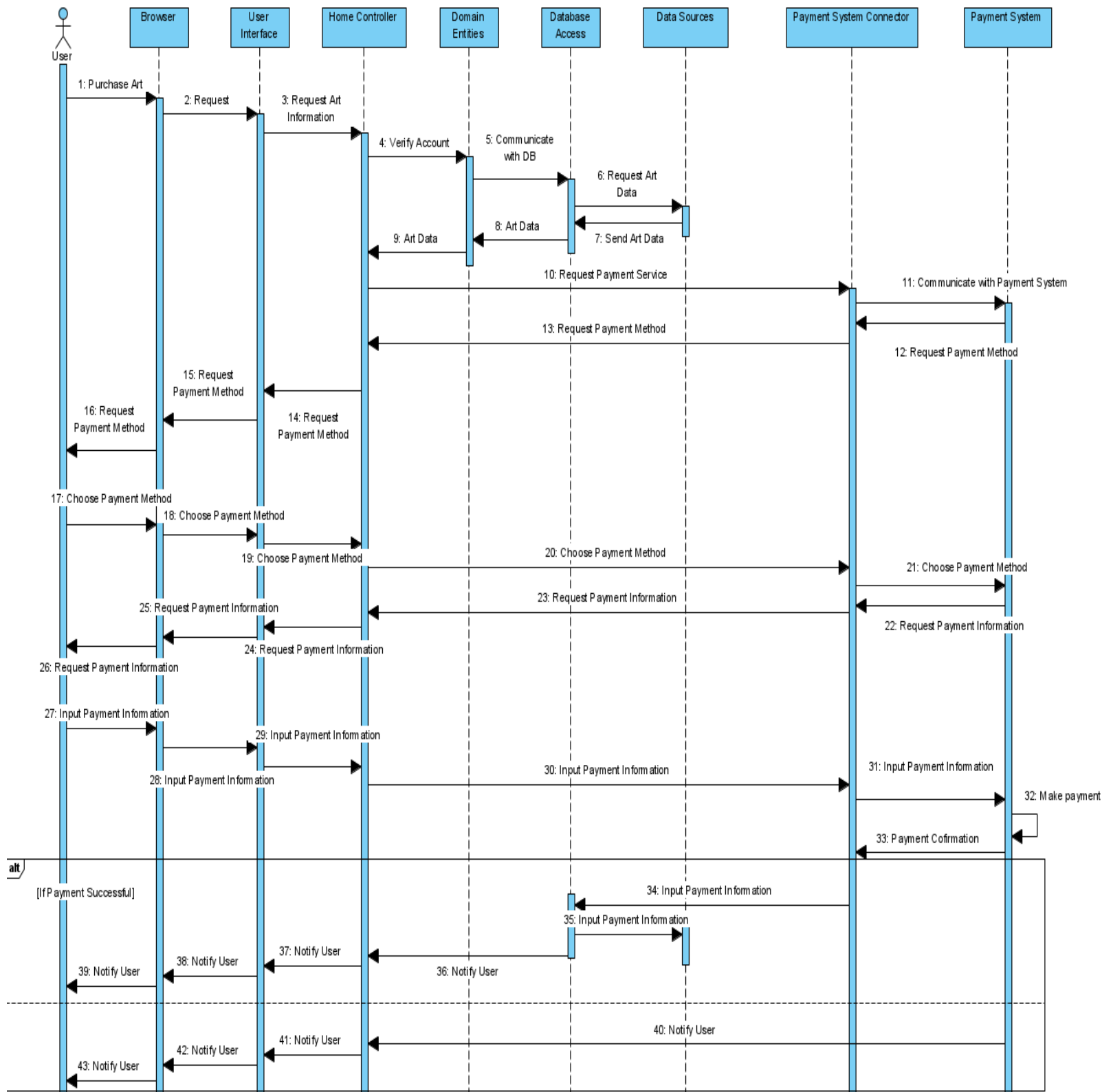


Figure 8: Sequence Diagram for UC-2: Buy or Sell Art

The following table describe the interactive elements of the Figure 8:

Method	Description
Element: Browser	
Request	Browser is requesting the information pertaining to the art requested by the user
Choose Payment Method	Payment System asks user what payment method they would like to use, then user selects from given list and browser is passing along parameters
Input Payment Information	User inputs information for given payment method and browser is passing along parameters
Element: User Interface	
Request Art information	User interface is requesting the information pertaining to the art requested by the user to display to send to the browser
Choose Payment Method	User Interface is passing parameters of payment system chosen to payment system connector
Input Payment Information	User Interface is passing parameters of payment information to payment system connector
Element: Home Controller	
Verify Account (Request Art Information)	This method verifies that the user account is valid with the account management system in Domain Entities and requesting art information through that
Element: Domain Entities	
Communicate with DB	This method requests art information that

	the user requested to purchase
Element: Database Access	
Request Art Data	This method requests art information that the user requested to purchase
Input Payment Information	Access Database to input payment information (invoice etc)
Element: Payment System Connector	
Communicate with Payment System	Connect to the external Payment system Application
Choose Payment Method	Transfer user inputted information to the Payment System Application
Input Payment Method	Transfer user inputted information to the Payment System Application
Input Payment Information	Transfer information from payment (invoice etc) to the database
Element: Payment System	
Make Payment	Payment System Application uses user inputted information to make a payment

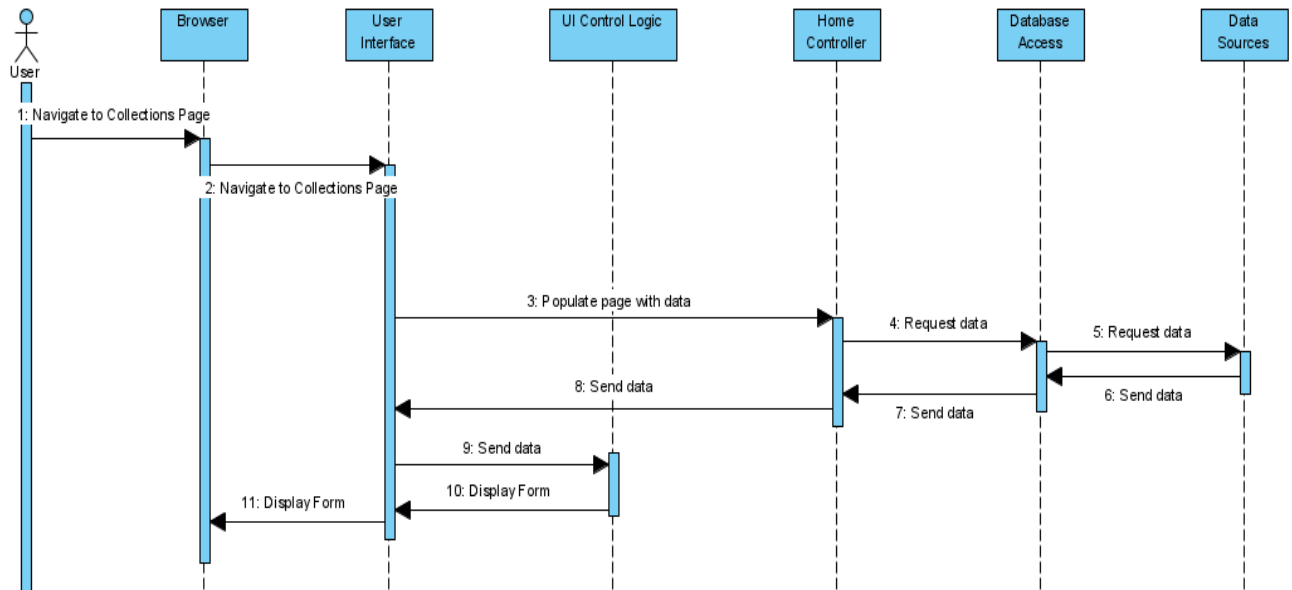


Figure 9: Sequence Diagram for UC-5: Collection Page

The following table describe the interactive elements of the Figure 9:

Method	Description
Element: Browser	
Navigate to Collections Page	This method calls the User Interface to provide a form for the User Interface
Element: User Interface	
Populate page with data	Request information from database to display to the user viewing the collections page
Send data	Send database information to UI Control Logic to receive updated element information
Element: Home Controller	
Request Data	Request information from database

Element: Database Access	
Request Data	Request access to read database information

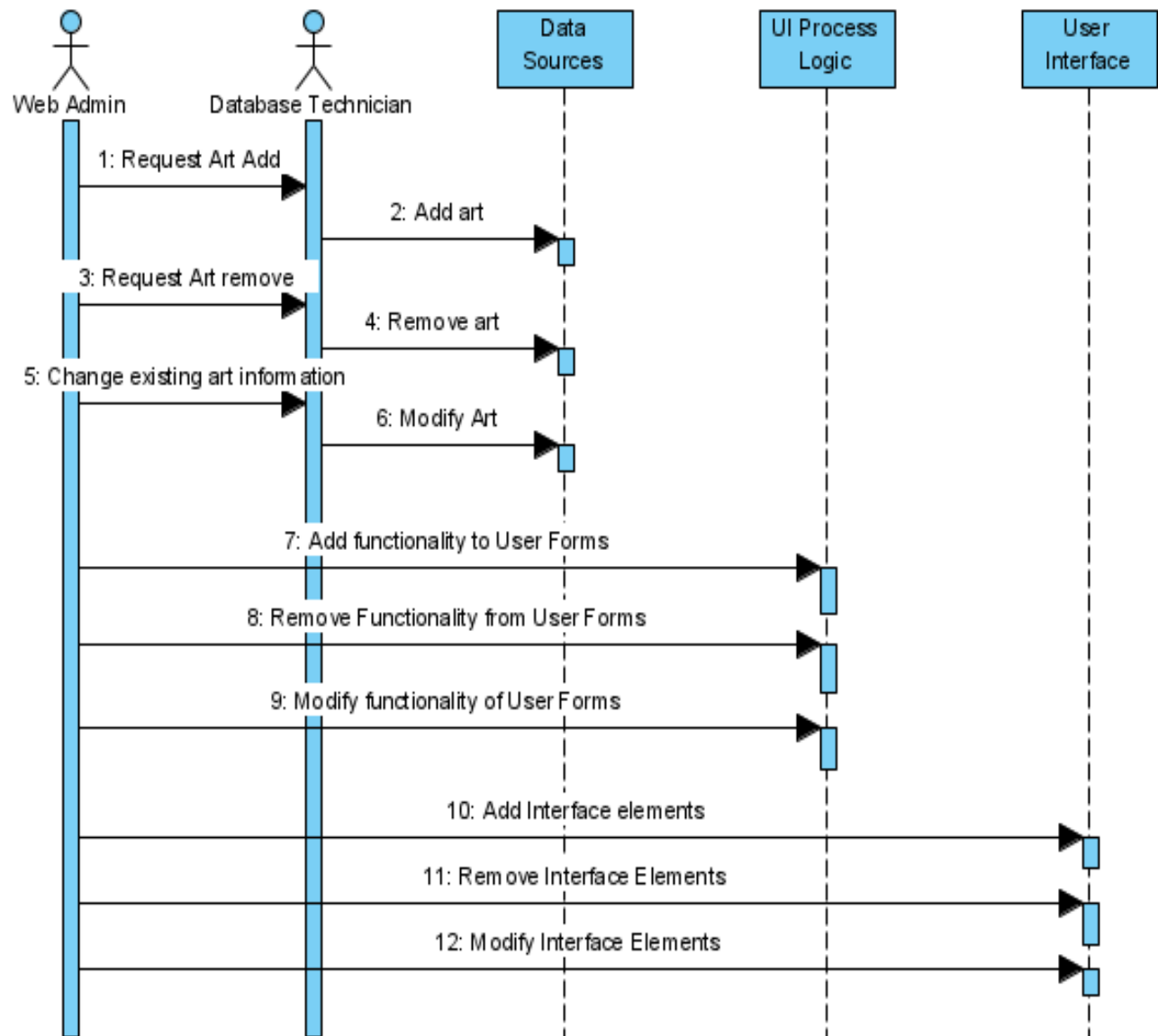


Figure 10: Sequence Diagram for UC-8: Manage Art Gallery

The following table describe the interactive elements of the Figure 10:

Method	Definition
User: Web Admin	
Request Art Add	Web Admin makes decision about which art to add and then notifies Database Technician
Request Art Remove	Web Admin makes decision about which art to remove and then notifies Database Technician
Change existing art information	Web Admin makes decision about which art to modify and then notifies Database Technician
Add functionality to User Forms	Web Admin makes decision about which functionalities should be added to the system and directly accesses the UI Process Logic to add them
Remove functionality to User Forms	Web Admin makes decision about which functionalities should be removed to the system and directly accesses the UI Process Logic to remove them
Modify functionality to User Forms	Web Admin makes decision about which functionalities should be modified to the system and directly accesses the UI Process Logic to modify them
Add Interface Elements	Web Admin makes decision about which interface elements should be added to the system and directly accesses the User Interface to add them
Remove Interface Elements	Web Admin makes decision about which interface elements should be removed to the system and directly accesses the User Interface to remove them

Modify Interface Elements	Web Admin makes decision about which interface elements should be modified to the system and directly accesses the User Interface to modify them
User: Database Technician	
Add art	Database Technician receives information from Web Admin to update database by adding art information
Remove Art	Database Technician receives information from Web Admin to update database by removing art information
Modify Art	Database Technician receives information from Web Admin to update database by modifying existing art information

Step 7: Perform Analysis of Current Design and Review Iteration Goal and Achievement of Design Purpose

The design decisions in this iteration provided an initial understanding of how functionality is supported in the system. Modules associated with the functionality of the system were identified and defined.

Not Addressed	Partially Addressed	Fully Addressed	Rationale
	UC-2		The payment system to pay for art has been implemented, but a system that pays the users selling art on the website must still be implemented to fulfill UC-2
		UC-5	The collection page is fully implemented as a modifiable page that can be displayed to the user if they wish to browse art on the art gallery

		UC-8	The art gallery is fully modifiable through direct access to the database and the user interface forms are also modifiable directly
	QA-2		The elements that support the associated use case (UC-2) have been identified
	QA-4		The elements that support the associated use cases (UC-2, UC-8) have been identified
	QA-6		The elements that support the associated use case (UC-2) have been identified
QA-7			No relevant decisions made.
	CON-1		No relevant decisions made.
	CON-2		Modules responsible for collecting data have been identified.
	CON-3		No relevant decisions made.
	CON-4		No relevant decisions made.
	CON-5		Modules responsible for authentication and data modification have been identified.
		CRN-2	Modules associated with all of the use cases have been identified and work was distributed among the team members.

Iteration 3: Addressing Quality Attribute Scenario Driver (QA-4)

In this iteration , the design phase, the outcomes of the actions completed in each of the ADD phases are presented. We now begin to think about the fulfilment of some of the more essential quality attributes, based on the fact that users may store private financial information on the site. This iteration concentrates on one of the quality attributes possibilities that is QA-4.

Step 2: Establish Iteration Goal by Selecting Drivers

This iteration , we focus on the QA-4 quality attribute scenario: User's will store private financial information on the site to buy and sell art . Therefore the system should encrypt the private information of it's users like login and financial information.

Step 3: Choose One or More Elements of the System to Refine

For this iteration we will be choosing Database Server and Web Server elements to refine . These are the two elements within the deployment plan that can be controlled in order to fulfill the security attribute.

Step 4: Choose One or More Design Concepts That Satisfy the Selected Drivers

The following table summarizes the selection of design decisions in this iteration:

Design Decisions and Location	Rationale and Assumptions
Implementing authenticate actors technique by having a login system and password	The website may be able to authenticate users by ensuring that user or computer is who they claim to be . Users/computers are required to have their passwords and a username to enter the access all the features if website

Using Encrypt Data technique by securing credentials in configuration files and using cryptographic algorithms .	The website must exchange the financial information with the payment system securely. Also users' financial information should be encrypted .
Implementing change default settings technique .	The website may force the users to change the default password settings by popping up a password change web page .

Step 5: Instantiate Architectural Elements, Allocate Responsibilities, and Define Interfaces

The instantiation design decisions made in this iteration are summarized in the following table:

Design Decisions and Location	Rationale and Assumptions
Authenticate actors by having them enter their unique username and password on the login page before accessing the website.	The website using login technique will help to authenticate users and will ensure that the user's information is saved and being accessed by the right user.
Establishing a cryptographic algorithm to ensure secure exchange of the information between the website and external payment system . Configuring files for the database .	A cryptographic algorithm can be used in the website to help exchange of financial or other confidential information securely between the system's.
Establish the change default settings by having a webpage which will pop-up if the user is login in for the first time.	Changing default settings should be mandatory to access the website if the user is login in the website for the first time .

Step 6: Sketch Views and Record Design Decisions

In this step , the deployment diagram that was made in iteration 1 is refined.

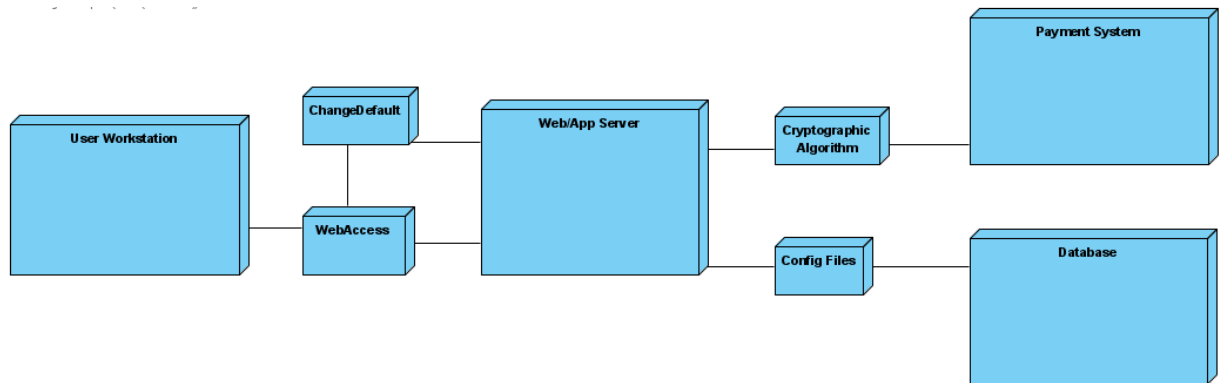


Figure 11:Refined Deployment Diagram

The following table summarizes the new elements that were not introduced in deployment diagram in iteration 1 :

Element	Responsibilities
WebAccess	This takes user to the login webpage where the user will login using his username and password
ChangeDefault	This will take those users who are using the website for the website and will let users change the default settings of the website along with the password and username.
Cryptographic Algorithm	This will ensure that the exchange of data between the webserver and payment system is secured and there is no misuse of the data.
Config Files	This helps in encrypting the database files by configuring them , hence securing them from hackers.

Step 7: Perform Analysis of Current Design and Review Iteration Goal and Achievement of Design Purpose

The focus of this iteration was on QA-4. The following table shows how the design decisions that were taken during this iteration affected the drivers.

Not Addressed	Partially Addressed	Fully Addressed	Rationale
	UC-2		No relevant decisions made.
	QA-2		No relevant decisions made.
	QA-4		By introducing authenticated actors and changing default settings techniques we can monitor any abnormal activities on the website and reduce the probability of any malicious use by keeping users aware, and have control, of the security settings. Also using encryption for exchanging data between the payment system and for the users' information in the database will help protect users' information from data breaches.
	QA-6		No relevant decisions made.
QA-7			No relevant decisions made.
CON-1			No relevant decisions made.
	CON-2		No relevant decisions made.
CON-3			No relevant decisions made.
CON-4			No relevant decisions made.
		CON-5	Modules associated with all of the use cases have been identified