



University Of Bahrain
Collage Of Information Technology
Department Of Computer Science

ITSE302: Software Design & Architecture

Dental Clinic Management System



Project Team Members:

- 1.Ahmed A.hussain - 202004419
- 2.Mohammed Isa - 202002725
- 3.Ahmed Mahmood - 202005802
- 4.Jaffer A.hussain - 202004937



Table of content:

1.	Business case:.....	3
2.	Use Case Diagram:.....	4
3.	Requirement:	5
4.	Use Case:.....	6
5.	Quality Attributes:.....	8
6.	Constrains:.....	9
7.	Concerns:.....	10
8.	Iteration 1: The Design Process.....	11
	Step1: Establishing an Overall System Structure	11
	Step2: Establish Iteration Goal by Selecting Drivers	12
	Step3: Choose One or More Elements of the System to Refine	12
	Step 4: Choose One or More Design Concepts That Satisfy the Selected Drivers.....	13
	Step5: Instantiate Architectural Elements, Allocate Responsibilities, and Define Interfaces	14
	Step6: Sketch Views and Record Design Decisions.....	15
	Step 7: Perform Analysis of Current Design and Review Iteration Goal and Achievement of Design Purpose....	18
9.	Iteration 2:	19
	Step 2: Establish Iteration Goal by Selecting Drivers:	19
	Step 3: Choose One or More Elements of the System to Refine	20
	Step 4: Choose One or More Design Concepts That Satisfy the Selected Drive.....	20
	Step 5: Instantiate Architectural Elements, Allocate Responsibilities, and Define Interfaces	21
	Step 6: Sketch Views and Record Design Decisions.....	22
	UC-4: Search:	26
	UC-7: Appointment Booking	27
	UC-8: Cancel Booking	28
	UC-9: payment:	29
	Step 7: Perform Analysis of Current Design and Review Iteration Goal and Achievement of Design Purpose:...	31
10.	Iteration 3:	32
	Step 2: Establish Iteration Goal by Selecting Drivers	32
	Step 3: Choose One or More Elements of the System to Refine	32
	Step 4: Choose One or More Design Concepts That Satisfy the Selected Drivers.....	33
	Step 5: Instantiate Architectural Elements, Allocate Responsibilities, and Define Interfaces	34
	Step 6: Sketch Views and Record Design Decisions.....	35
	Step7: Perform Analysis of Current Design and Review Iteration Goal and Achievement of Design Purpose: ...	37



1. Business case:

Technology has dominated our current era, as technology has become an essential thing in our daily lives, and all aspects of life have become highly dependent on it. It is communication between us, the users, with the system, and in this case, the person would like to become a dental clinic system that applies in the world to make the process easy.

This dental clinic program will serve patients and even doctors through a system in which dental clinic doctors offer the services they provide, which will enhance this feature by providing sufficient information with pictures and educational videos that will help the patient know the service he needs with full knowledge of the service and also the user can search for services Through the means provided by the clinic, he can choose the appropriate service with the doctor he prefers to perform it to, and book an appointment with the doctor. The available times will be shown to make an appointment reservation with the doctor. In this way, after the end of the service, the patient can submit his reviews on the service. And these reviews will be updated monthly so that other users can benefit from them, and even the doctor can improve his performance through the patient opinions presented in the program. This will enhance the trust between the user and the system because these evaluations will be displayed with full transparency to help the user and the doctor.

From that, there are four main components to the system which are search, appointment booking, cancel booking and payment.

Search: The user will be able to search through the program for the services available in the dental clinic and get all information about the services through the dental clinic.

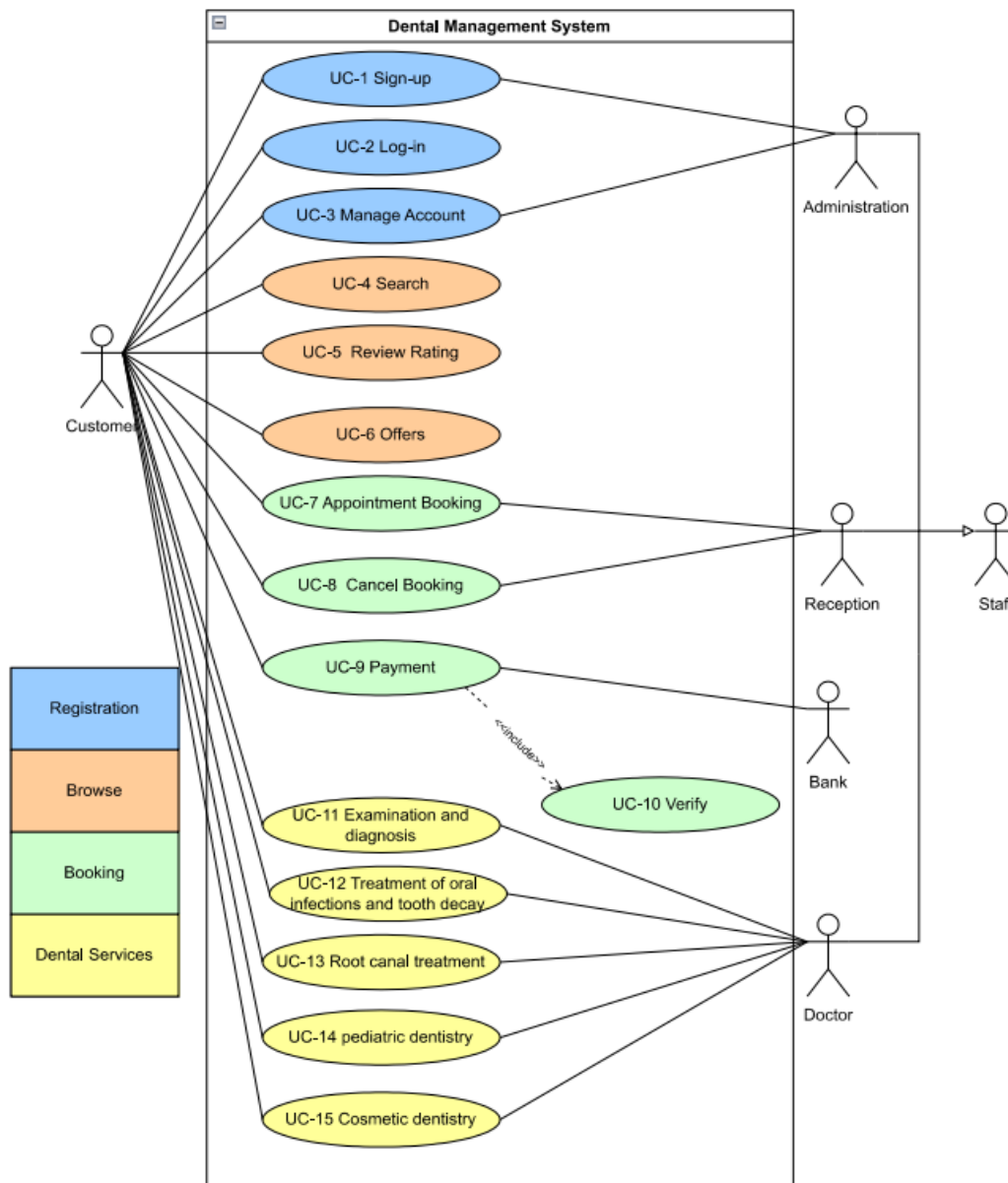
Appointment booking: After searching in the program for the services available in the dental clinic and getting all the information, the user will be able to choose the appropriate services and the doctor he prefers. The times available for reservation will be shown, and the user can choose the most appropriate time to book the appointment.

Cancel booking: The system allows the user, whether a patient or a doctor, to cancel the appointment in any case, but it must be done before the payment process.

Payment: The payment process will be done online. The payment process means confirmation of booking the appointment, and if the payment process is completed, the user will receive a text message or e-mail, or both, confirming that the payment process was successful. After completing the payment process, it will be difficult to cancel the appointment.



2. Use Case Diagram:



3. Requirement:

The Dental Clinic Management System (DCMS) is a system concerned with providing dental clinic services online, such as booking an appointment and selecting services.

System Requirements At the beginning, there will be a registration system in which the user can create an account on the system or log in, and the registration management employee will be able to manage user accounts. Also, all user data will be recorded on a database for preservation, and each user's data will be saved on his account. The system will be in English, but the system will support other languages, enabling the user to choose his preferred language. The user will be able to search and filter through the system, and this will enable the user to search for the services provided by the clinic. The user will be able to select and choose the services provided by the clinic and book an appointment with the doctor he prefers. The system will allow the dashboard management staff to add and modify the clinic services provided to the user or delete the canceled services in the clinic. Notifications will be sent from the system to the users with the details of the appointment, as well as reminding the user about the appointment. The system will provide a safe online payment process that will enable them to pay their bills through bank cards. After the appointment is made and ends, the user will be able through the online reviews system to give his full opinion about his service experience, and this will help other users in choosing, and this will enhance trust between the user and the system.



4. Use Case:

USE CASE	Description
Sign-up	The user will create an account to register in the system by offering required information: first name, last name, CPR, phone number password, date of birth and Location.
Log-in	The user will be able to access their previously created account by entering username, password.
Manage Account	The user can edit their profile information and delete it if they wish to, and they can see their previous Dental Clinic and the payment spent on it.
Search	The user shall be able to search for any clinic in the app with the ability to search for offers.
Review Rating	The user can put on his reviews about a specific Dental Clinic branch and talk about their overall Dental Clinic out of 5.
Offers	The user shall be able to view offers and deals from different clinics.
Appointment Booking	The user shall be able to book an appointment with a certain clinic at a specific time and will be able to select a dentist from that clinic.
Cancel Booking	The user shall have the option to cancel an appointment they have scheduled.
Payment	The user should write his online bank card information to pay for his fees.
Verify	The system should check first if the user card information is correct or not then check if the user Account have enough money to pay for the appointment.
Examination and diagnosis	The user shall be able to meet the dentist with who they have made an appointment for an examination and diagnosis.
Treatment of oral infections and tooth decay	The user shall be able to receive Both tooth decay and mouth infections treatment.



Root canal treatment	The user should receive Root canal treatment depending on the specific case and the dentist's treatment plan.
Pediatric dentistry	The user can get pediatric dentistry like fluoride, cleanings, orthodontics, and sealants.
Cosmetic dentistry	The user has access to cosmetic dentistry treatment like veneers, bonding, and teeth whitening.



5. Quality Attributes:

Quality Attribute	Description	Priorities
QA-1 Modifiability	Users can edit their account information and make any necessary changes, and the Clinic information must be updated daily to reflect the most up-to-date news and services for each clinic.	All Use cases
QA-2 Security	The system will have robust cyber security to prevent unauthorized parties from accessing a user's account, and all transaction payments will be encrypted for security reasons to secure user data.	UC-1 UC-2 UC-9 UC-10
QA-3 Performance	The user may complete tasks and operations in a total of three seconds or less. For instance, if a user searches for a certain clinic service, the system will provide the results in three seconds or less.	UC-4 UC-7 UC-8 UC-9
QA-4 Availability	The system will have backup servers ready to go that will be used in case of server overload or if the server goes down. As a result, the system will be operational around-the-clock during the whole year.	All Use Cases
QA-5 Usability	If there are any errors in the user's input, the system will point them out. If the user is unable to understand how the system operates, there is a user guide in the system that outlines a step-by-step procedure. The user can also cancel any step at any time without having to start over.	All Use Cases
QA-6 Scalability	When the system is deployed, it will be able to support at least 5000 patients, and it will be enhanced to support more.	UC-5



6. Constrains:

ID	Constrains
CON-1	The system only offers services to clients in Bahrain
CON-2	The system must work with all devices.
CON-3	The system will be developed in Java
CON-4	The system must be completed within a year.
CON-5	Access to the system requires an internet connection.



7. Concerns:

ID	Concerns
CRN-1	Creating the base for the whole system
CRN-2	Distributing work among the team's developers individually
CRN-3	BHD currency is accepted in the system.
CRN-4	Both the Java language and its technology are known to the team members
CRN-5	Arabic and English language are used in the system



8. Iteration 1: The Design Process

Step1: Establishing an Overall System Structure

Category	Details		
Purpose	This is a greenfield system from a mature domain. The goal is to create a design that is thorough enough to support the system's building.		
Primary functionality requirement	<p>From phase 1 the primary use cases:</p> <p>UC-4: Search: The User search for a clinic or offers.</p> <p>UC-7: Appointment Booking: The user can choose a dentist and schedule an appointment with a facility.</p> <p>UC-8: Cancel booking: The user wants to cancel a booking they have previously made.</p> <p>UC-9: Payment: the user can pay for the appointment online.</p>		
Quality attributes	QA ID	Importance to customer	Importance to architect
	QA-1: Modifiability	High	High
	QA-2 Security	High	Medium
	QA-3 Performance	High	High
	QA-4 Availability	low	medium
	QA-5 Usability	high	Medium
	QA-6 Scalability	medium	high
concerns	All the constraints are added as drivers		
constrains	All the architectural concerns are selected as drivers		



Step2: Establish Iteration Goal by Selecting Drivers

Establishing the overall system structure and improving the design concepts in accordance with the chosen quality criteria, restrictions, and concerns will be the main goals of Iteration 1. Making the first design choices and building a strong system foundation are the objectives. make sure that crn-1 have been achieved and Architect must consider all drivers of the following system structure:

UC-4: Search

UC-7: Appointment Booking

UC-8: Cancel booking.

UC-9: Payment

QA-2: Security

QA-3: Performance

QA-6: Usability

CON-3 The system will be developed in Java.

CON-2 The system must work with all devices.

CRN-4 Both the Java language and its technology are known to the team members.

CRN-2 Distributing work among the team's developers individually.

Step3: Choose One or More Elements of the System to Refine

Since this is a greenfield project, the entire Dental Clinic Management System has been refined in this instance. In this instance, decomposition is used to refine.



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

Alternative	Reason for discarding Rich client Application
Rich client Application	Our system needs to be regularly updated every few seconds, so we chose against using this reference architecture, which requires installing it on the user's computer. It is therefore not possible to implement it using this reference design.
Mobile application	Because we are not currently concentrating on the creation of such devices, we have chosen not to use this reference architecture created for developing apps specifically for portable devices.
Rich internet application	Although this reference architecture has the benefits of not requiring installation and being simple to update, we chose against using it because of its constrained access to local resources. This restriction was found to be inappropriate because our system depends on optimal performance and resource availability.

Design Decisions	Reason for Decision
Web application	To provide our clients with a solution that is straightforward and simple to use, we chose this reference architecture. Furthermore, we need the system to be updated frequently.
Three-tier deployment	A three-tier deployment is regarded appropriate because the system must be usable on all devices.



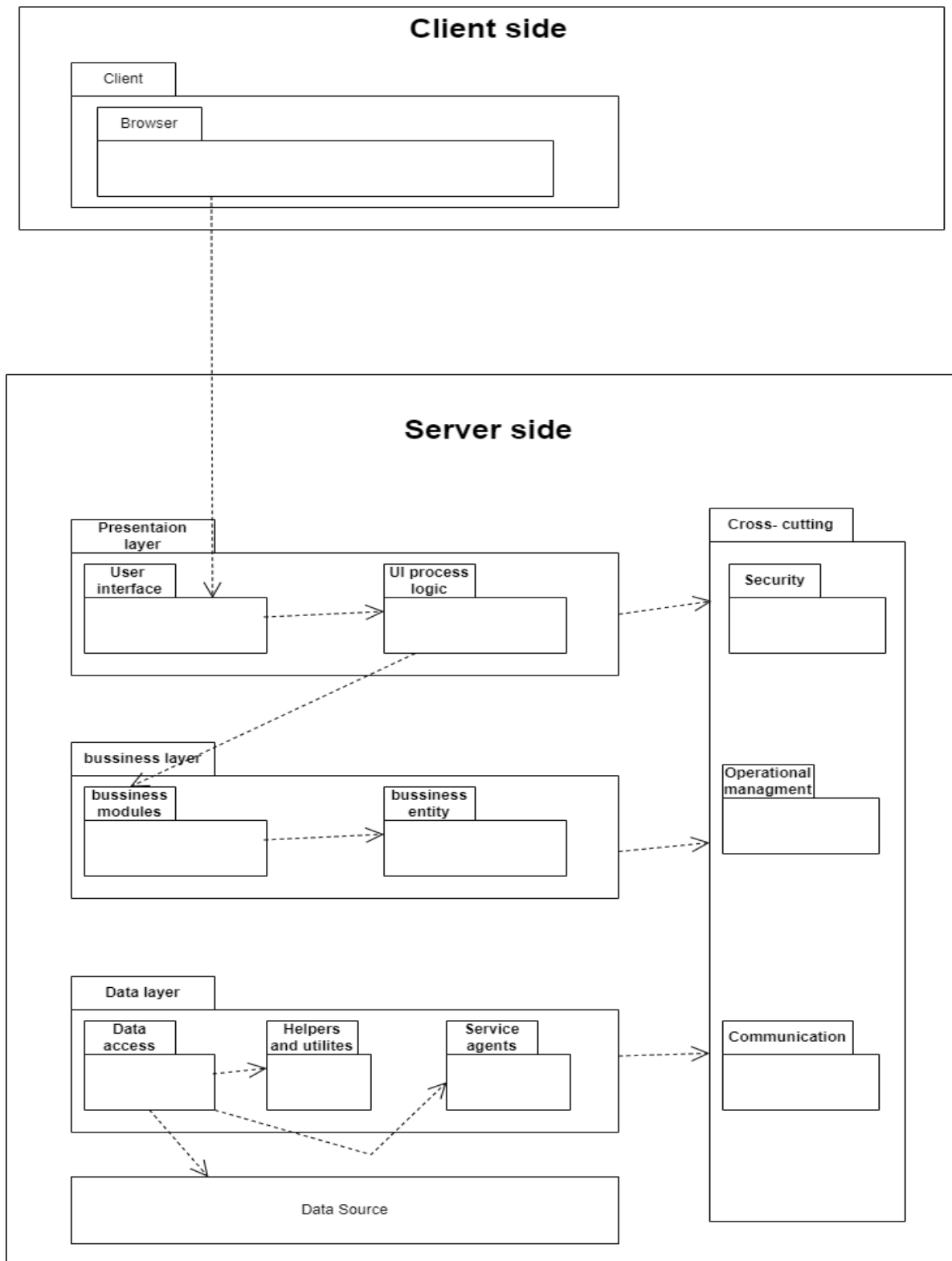
Step5: Instantiate Architectural Elements, Allocate Responsibilities, and Define Interfaces

The instantiated of the design decisions that considered and made are summarized.

Design Decision and Location	Rationale
Develop a specific business layer module to manage clinic bookings in addition to the data source.	Develop a specific business layer module to manage clinic bookings in addition to the data source. Through access to the business layer, the client side can manage the booking clinic data, and adjustments are reflected in the data source.
combining the host server and web application into a single entity	This increases system efficiency and makes it easier to obtain data from the web layer.



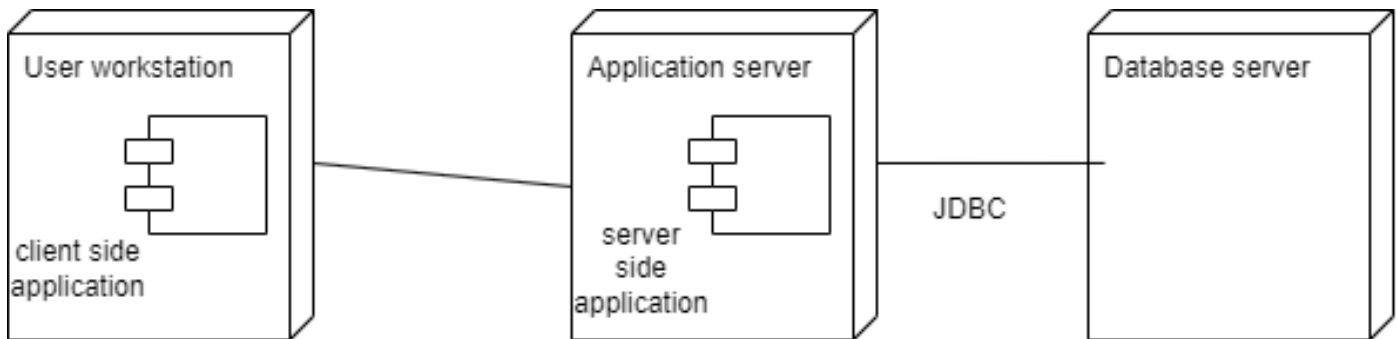
Step6: Sketch Views and Record Design Decisions



Element	Responsibility
Presentation layer	This specific layer is made up of elements in charge of controlling use case flow and user interaction.
Business layer	This layer is made up of modules that conduct business logic tasks using server-side processing.
Data layer	This layer includes modules for data handling and connection with the hotel management system.
User interface	These elements are responsible for interacting with users and displaying information to them. They include user interface components like text fields and buttons.
UI process logic	These parts oversee controlling how the operations of the application proceed. Data validation, managing interactions between various business logic components, and supplying data from the business layer to the user interface are just a few of the duties they do.
Business module	The execution of business processes is conducted by this module.
Business entity	These elements embody the business domain's entities and their related business logic.
Data access	These elements provide commonly used operations for getting and storing information and function as a storage mechanism..
Helpers and utilities	These elements include features that are common to all the data layer's modules but are not exclusive to any one module.
Service agents	The communication protocols used to deliver data to external services are abstracted by these components.
Cross-cutting	These elements include the communication protocols used to send data to external services.
Security	These modules have functionalities including input/output operations, logging, and security that cut across numerous layers.
Operation management	This part is responsible for the system's security, encompassing operations like authorization, authentication, exception management, logging, instrumentation, and validation. It also has cross-cutting functionality.
Communication	These components include cross-cutting functionality that controls communication channels between multiple physical tiers and levels.



The deployment diagram.



Element	Responsibility
User workstation	the server in charge of hosting the application's client side.
Application server	the server in charge of hosting the application's server-side logic.
Database server	The system in charge of maintaining the database

Relationship	Description
Application server and Database server	The JDBC protocol will be utilized for database communication.



Step 7: Perform Analysis of Current Design and Review Iteration

Goal and Achievement of Design Purpose

Not Addressed	Partially Addressed	Completely Addressed	Design Decision made during the iteration
	UC-4		Selected reference architecture establishes the modules that supports this functionality
	UC-7		Selected reference architecture establishes the modules that supports this functionality
	UC-8		Selected reference architecture establishes the modules that supports this functionality
	QA-9		Selected reference architecture establishes the modules that supports this functionality
QA-3			No relevant decision made
		QA-2	Selected reference architecture provides high security
QA-6			No relevant decision made.
		CRN-3	While the advantage of developers' proficiency in Java is acknowledged, it is also important for them to possess a broader range of expertise in other areas.
CRN-2			No relevant decisions made
	CON-3		The adoption of the Java programming language has been carefully considered for this system's development and is supported by the reference architecture.
CON-2			No relevant decision made.



9. Iteration 2:

The second round of design for the Dental Clinic Management System, known as Iteration 2, involves identifying the structures necessary to support the main functions of the system. During this stage, we move away from broad descriptions of functionality used in the first iteration and focus on making more specific decisions that will guide implementation and team formation. This progression from general to is deliberate and part of the ADD method, as we cannot design everything at once and must make disciplined decisions to ensure a systematic design that addresses the most significant risks first before moving to more detailed aspects. In the first iteration, our objective was to establish an overall system structure. With that goal met, our new objective for Iteration 2 is to consider the implementation units, which will impact team composition, interfaces, and how development tasks can be distributed, outsourced, and executed in sprints.

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 not only for understanding how functionality is supported, but also for addressing CRN-2 that is, the allocation of work to members of the development team. In this second iteration, besides CRN-2, the architect considers the systems.

Primary use cases:

UC-4: Search

UC-7: Appointment Booking

UC-8: Cancel Booking

UC-9: payment



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

The element that will be redefined in this iteration will be the modules found in different layers defined by the previous reference architecture from iteration one the support of functionality in this system needs the collaboration of components associated with modules that are found throughout different layers.

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

Design Decisions and Location	Rationale and Assumptions
Create a Domain Model for the Application	It is crucial to create an initial domain model for the system before moving on to functional decomposition and identifying the primary entities of the domain and their relationships. Neglecting to create a proper domain model can result in an ad hoc architecture that is challenging to comprehend and maintain. While a domain model may eventually emerge, it may not be optimal, leading to a suboptimal system design. Therefore, creating a domain model is necessary to improve the system's design and ensure its long-term maintainability.
Identify Domain Objects that map to functional requirements	The application's unique functional components should be contained in a self-contained building block known as a domain object. While it is possible to decompose the layers directly into modules without considering a domain object, this approach carries a high risk of neglecting essential requirements. Therefore, it is crucial to encapsulate the application's functional elements in a domain object to ensure that all requirements are considered and met.
Decompose Domain Objects into general and specialized Components	Domain objects encompass complete sets of functionalities, but these functionalities are supported by more finely grained elements located within the layers. In this pattern, these elements are referred to as modules, and they are connected to the specialization of the layers in which they are located (such as UI-Modules). There are no viable alternatives to decomposing the layers into modules to support the application's functionality.



Use Spring framework	Spring is a highly popular web application framework known for its fast and secure development of websites. It is also scalable and versatile, making it a top choice for many developers. As a full-stack framework, Spring includes all the necessary components, and it does not require any additional extras. Additionally, it is highly compatible with other frameworks, making it easier to achieve QA-3 in the development process. Other alternatives were considered such as ruby framework but due to the team's structure it was discarded for the CRN-4.
-----------------------------	--

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

Design Decisions and Location	Rationale
Create only an initial domain model	To speed up the design process, it is necessary to identify and model the entities that play a role in the primary use cases. However, at this stage, only an initial domain model is created.
Map the system use cases to domain objects	An initial identification of domain objects can be made by analyzing the system's use cases. To address CRN-5, domain objects are identified for all the use cases in UML model
Decompose the domain objects across the layers to identify layer specific modules with an explicit interface	The approach guarantees the identification of modules that support various functionalities. The architect will conduct the tasks for the primary use cases, enabling other team members to identify the remaining modules and allocate them among the team. This realization leads to the creation of a new architectural concern, namely CRN-8: evaluating the primary modules. It is essential to evaluate these modules since the modules that implement the user interface functionality are challenging to evaluate independently. Therefore, the focus of this concern is limited to the primary modules only.



Connect components. associated with modules using Spring	Using this framework can increase productivity when checking system speed and controlling execution, which can aid in defining the critical aspects of modules in terms of speed, productivity, and execution. This, in turn, can assist in fulfilling CRN-8 (a concern related to evaluating the primary modules).
Associate frameworks with a module in the data layer	The modules in the data layer encapsulate the ORM mapping. The Spring framework comes with an ORM (Object-Relational Mapping) built-in, and the previously chosen framework relates to these modules.

Step 6: Sketch Views and Record Design Decisions

Figure (3.1) shows an initial domain model for the system.

Figure (3.2) shows the domain objects that are instantiated for the use case model in UML model.



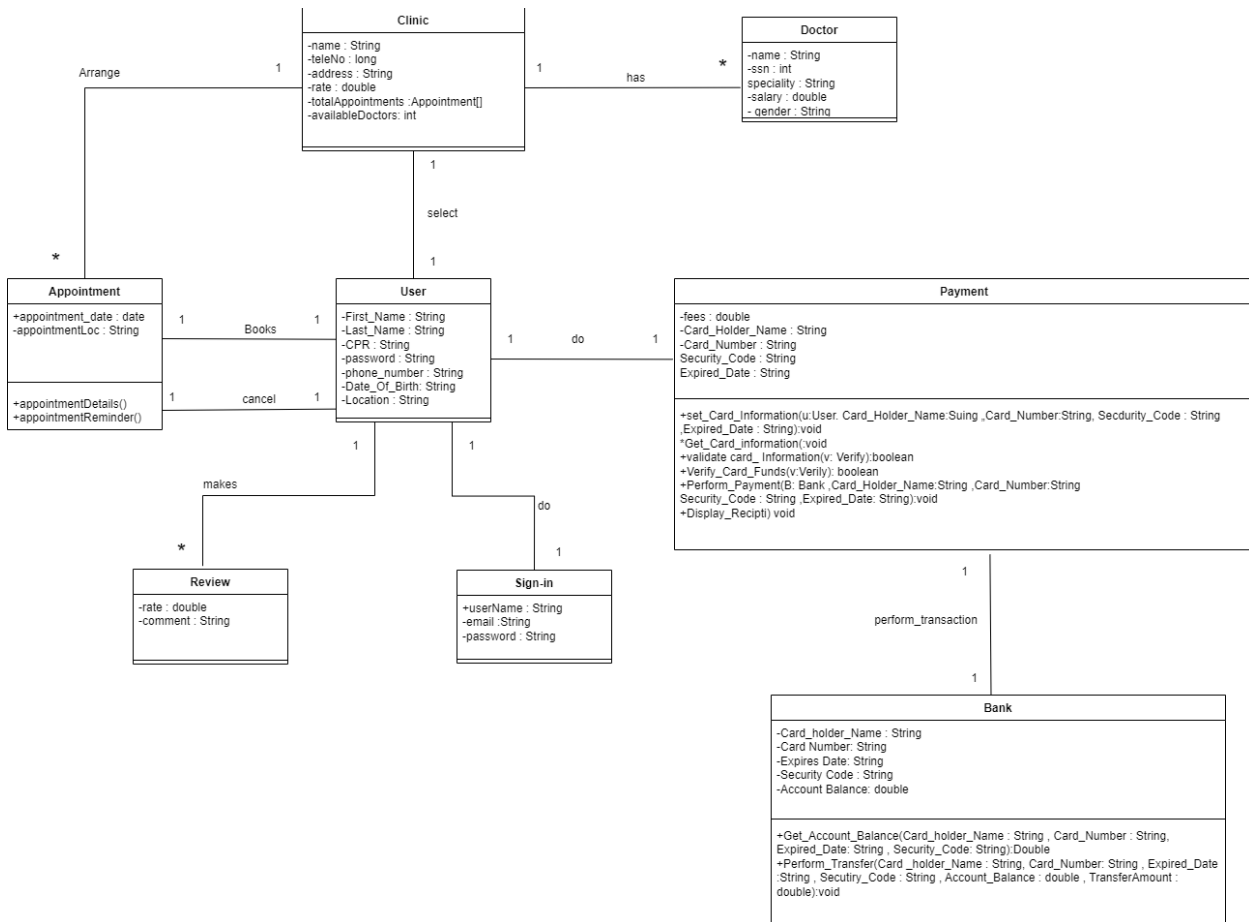


Figure 3.1

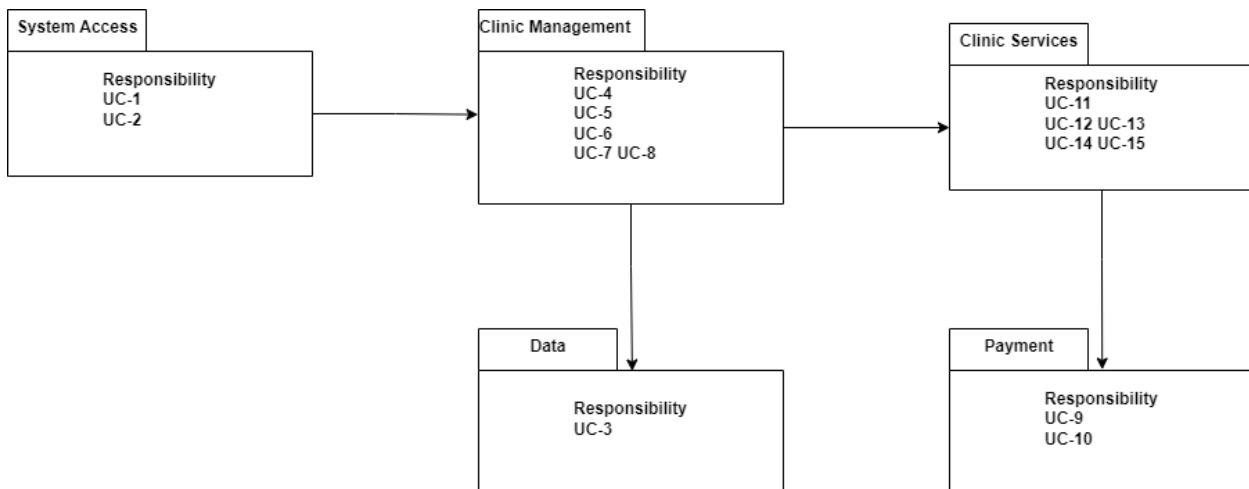
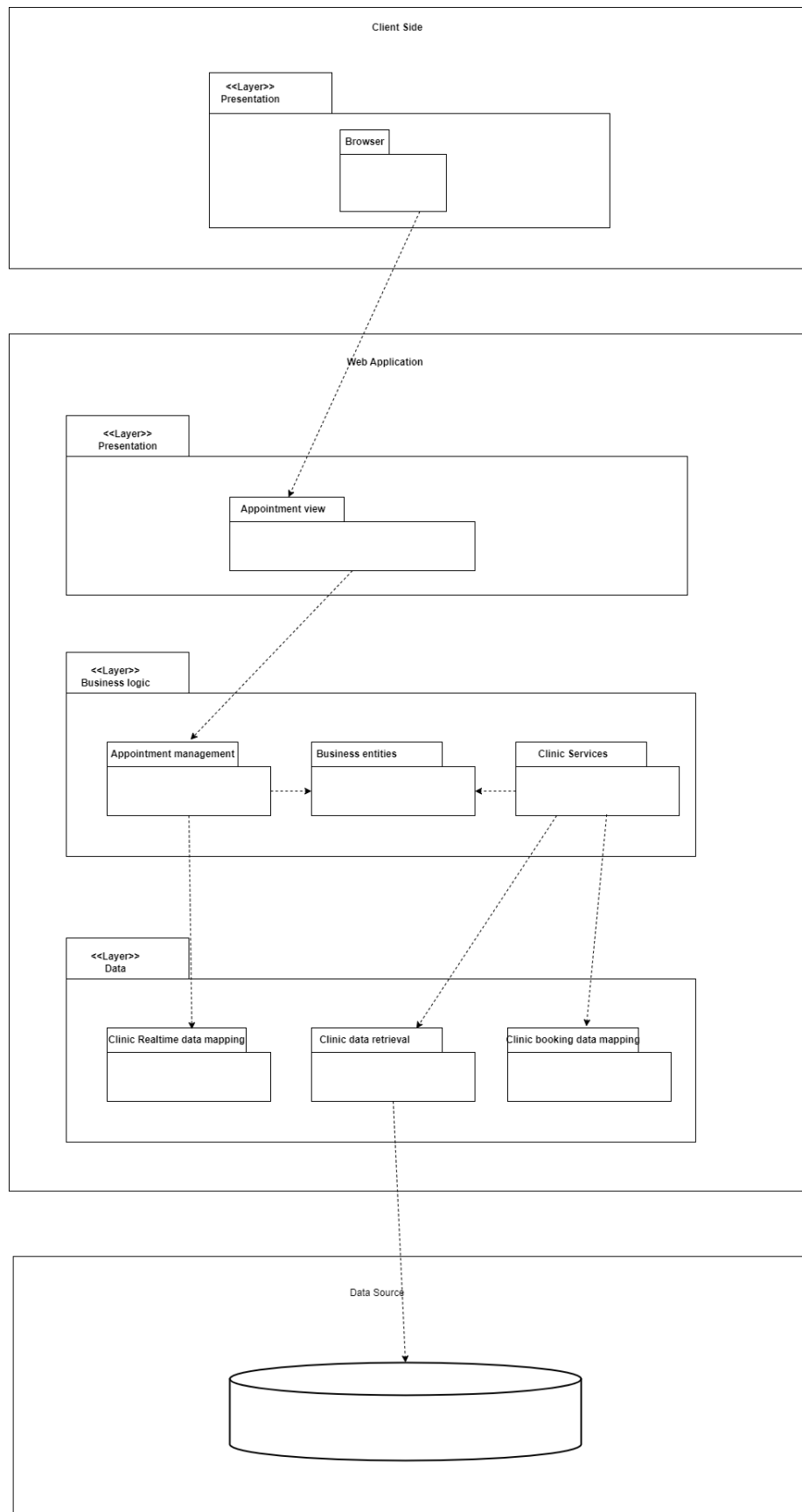


Figure 3.2



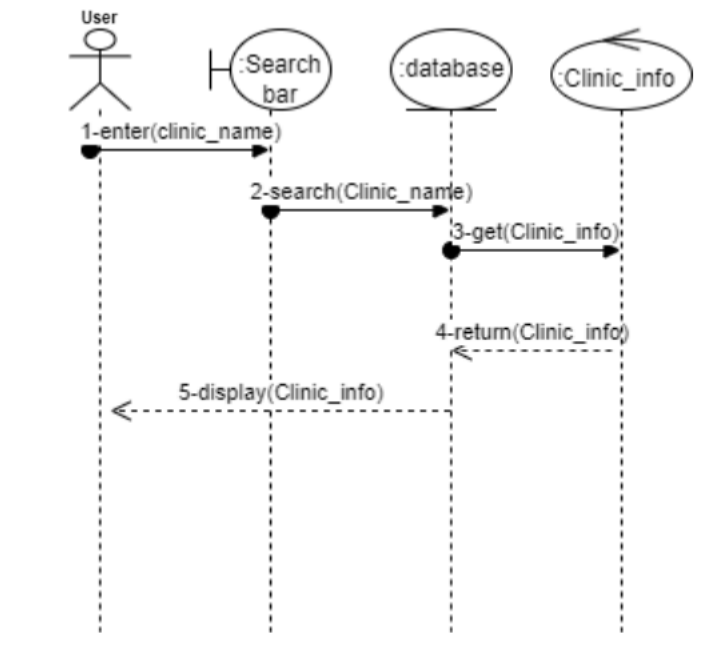


Element	Responsibility
Appointment view	It is responsible for showing different types of clinics to the User and gives the choices to Book clinics and see their details and information
Appointment management	The employee changes the reservation status of a clinic room and chosen doctor and make them unavailable to stop user is from booking again in the same room and same doctor and date
Business entities	Contains the entities for the business domain model.
Clinic service	Responsible for the Clinic services and how to make it easier to the user to contact with the Clinic such as booking and cancel booking and pay from online and choose preferable doctor.
Clinic real time data mapping	The Clinic must give out accurate information to the employee to give the real time data about the Clinic rooms and his doctor and when he is available to book again
Clinic data retrieval	The Clinic shares information of what the user did in the Clinic such as the satisfaction of the services like how smooth booking or cancelling appointment of how satisfied they are with their doctors.
Clinic booking data mapping	Responsible for checking the reservation and the chosen doctor of booking and sending receipts to the user with all the information they needed to not cause confusion.
Data source	Responsible for saving all reservation and information about the system.



The following sequence diagrams for UC-4, UC-7, UC-8, and UC-9 were created in the previous step of the method to define interface:

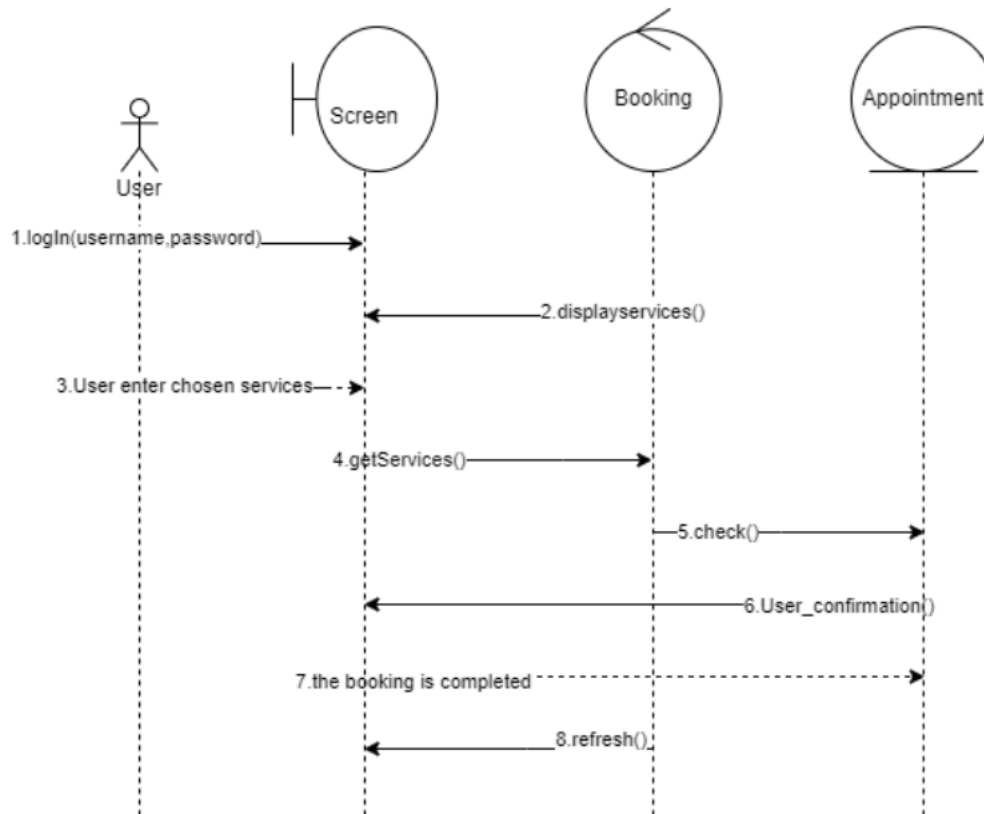
UC-4: Search:



	Method Name	Description
User	Enter()	The user enters clinic name to search about it.
Search bar	Search()	This method contains the name of clinic entered by user
database	getInfo()	This method will get information from clinic info to display it for user
	Display()	This method will display the information about the clinic that get it from clinic info
Clinic_info	Return()	This method returns the information to database that mentioned from get method



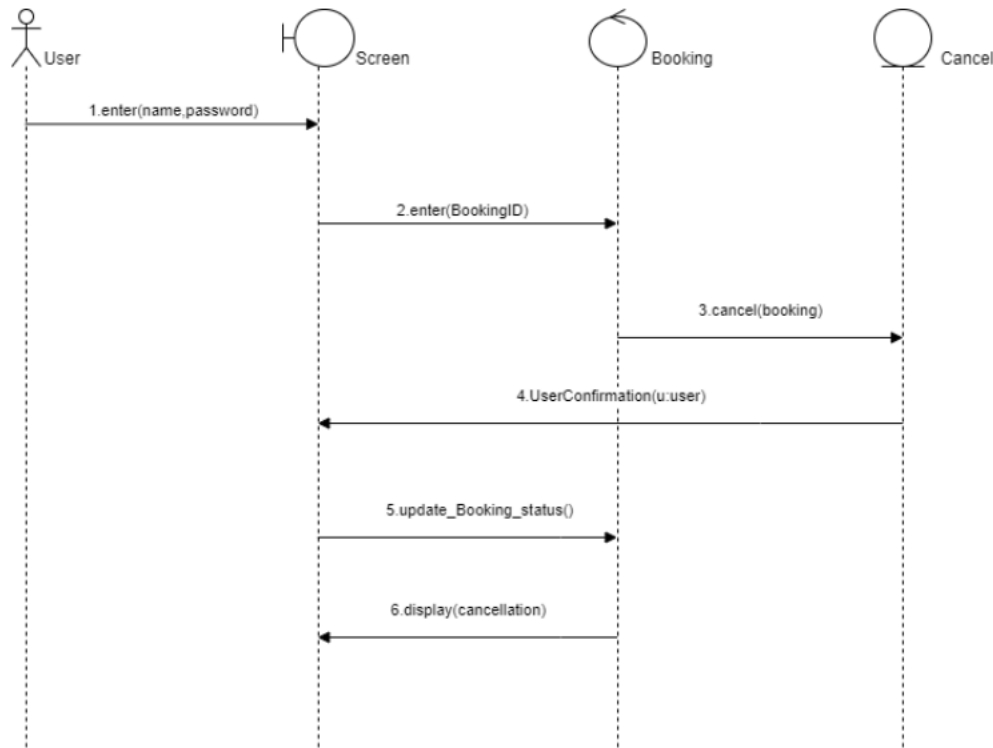
UC-7: Appointment Booking



	Method Name	Description
User	login()	The login method is important to complete booking. To take more information about users.
	displayAmount()	shows the user the total bill amount
booking	displayServices()	This method will give the user the services provided from clinic
	Check()	This method will check the database and give the user time available for booking
Screen	GetServices()	This method will ensure that the service is available or not
	Refresh()	This method will refresh the screen to give the user signal of appointment is register successfully
Appointment	User_confirmation()	This method will ask the user is sure about the information entered and accept the date



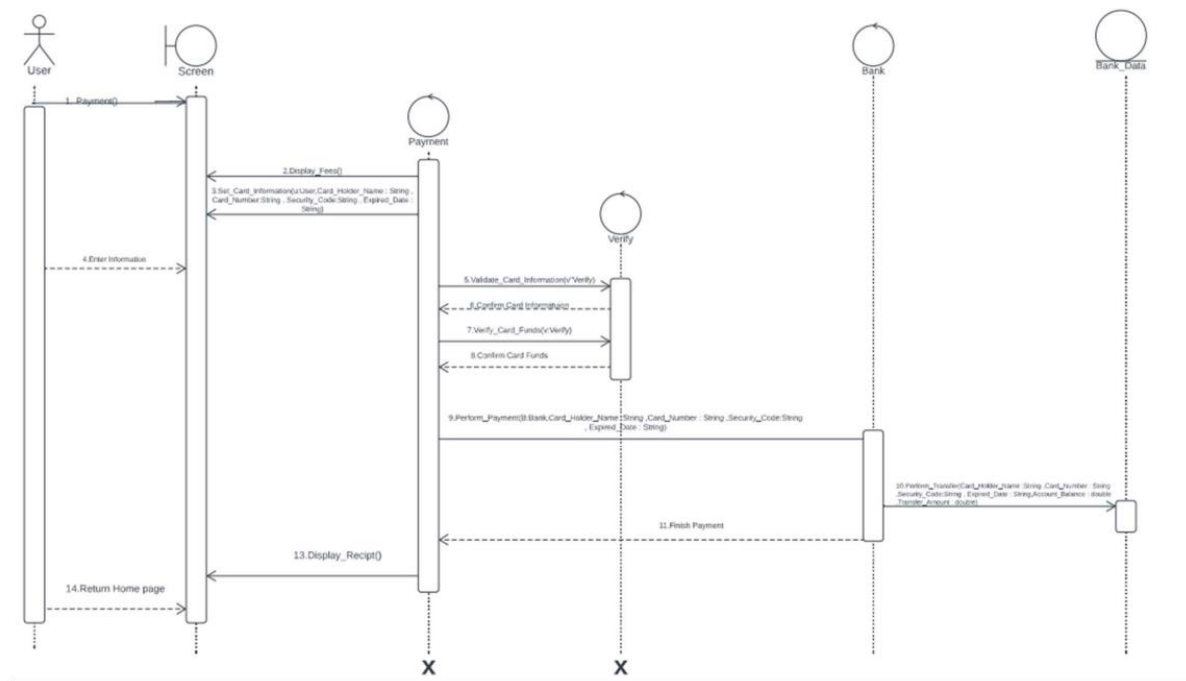
UC-8: Cancel Booking



	Method Name	Description
User	Enter()	This method is same as login it is important to ensure that this user account and want to cancel appointment
Screen	Enter_AppintmentID()	This method will save more time for the system to collect the information about user appointment.
	Update_database()	After the user confirmation the system will communicate with database to cancel the appointment.
Booking	Cancel()	This method will call the cancel object.
	display()	After the appointment cancel the booking will communicate with the screen to display cancellation information.
Cancel	User_confirmation()	This method is responsible to take user confirmation after calling from booking.



UC-9: payment:



User	Method Name	Description
	Enter_information ()	This method will give the system info about appointment so the system will calculate the amount.
Screen	Payment()	This method will display payment page
	Homepage_return()	This method will take the user to home page after completing the payment.
Payment	Display_fees()	After display, the payment page will prompt screen to display fees.
	Verify()	This method will verify the card entered by user.
	Display_receipt()	After payment complete, the system will prompt the screen to display receipt



Bank	Verifycard()	This method will get the card information from payment and check if exist or not.
	Transection()	After verifying the card and check the amount in this account this method will trans the fees to clinic bank account.
Bank_data	CheckAccount()	This method is called after the bank verify the card, and after verifying the bank will check the amount in this account.



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

Not addressed	Partially addressed	Completely addressed	Iteration Design decisions
		UC-1	Modules have been defined across layers and diagrams to support this use case.
		UC-7	Modules have been defined across layers and diagrams to support this use case.
		UC-9	Modules have been defined across layers and diagrams to support this use case.
	QA-1		No relevant decision made.
QA-2			No relevant decision made
QA-6			No relevant decision made.
	QA-7		The elements that support the associated use cases (UC-1) and (UC-11) has been identified
	CON-4		No relevant decision made.



10. Iteration 3:

In this third iteration of the process, we will address a Quality Attribute Scenario Driver (QA-4) by showcasing the outcomes of the activities performed in each step of ADD. Based on the foundational structural choices made in the first two iterations, we can begin to consider how to satisfy some of the critical quality attributes. This iteration will concentrate on a single quality attribute scenario.

Step 2: Establish Iteration Goal by Selecting Drivers

The aim of this iteration phase is to improve the quality of the system in terms of QA-4 by tackling the driver of the quality attribute scenario. This scenario demands that the system functions continuously, even if there is a failure. To achieve this, the system must be designed to switch to backup servers smoothly, guarantee that maintenance and repair are uncomplicated, and allow the faulty server to restart without losing data. By prioritizing these drivers, the objective is to enhance the system's agility in responding to evolving requirements and ensure its continued operation, even in unpredictable situations.

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

In this availability scenario the elements that will be refined are the physical nodes they were identified in the first iteration:

- Application server
- Database server



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

The design concepts used in this iteration are the following:

Design Decisions and Location	Rationale and Assumptions
Active redundancy replicates critical components for high availability and reliability in software architecture. It improves performance by distributing the workload and simplifying maintenance and upgrades. Prioritizing active redundancy enhances modifiability, allowing the system to adapt to changing requirements while reducing costs and downtime.	This approach reduces downtime and maintenance costs while improving system performance and modifiability. And include seamless replication, effective workload distribution, and no compromise on other quality attributes.
The Dental Clinic Information Inspector	The Dental Clinic Information Inspector is a system designed to improve patient data reliability and accuracy. It assumes a well-designed system with consistent data, accurate error detection, and timely notification. The inspector's role is to facilitate continuous monitoring, alert staff to issues, and improve the quality of patient care by ensuring accurate and up-to-date information.



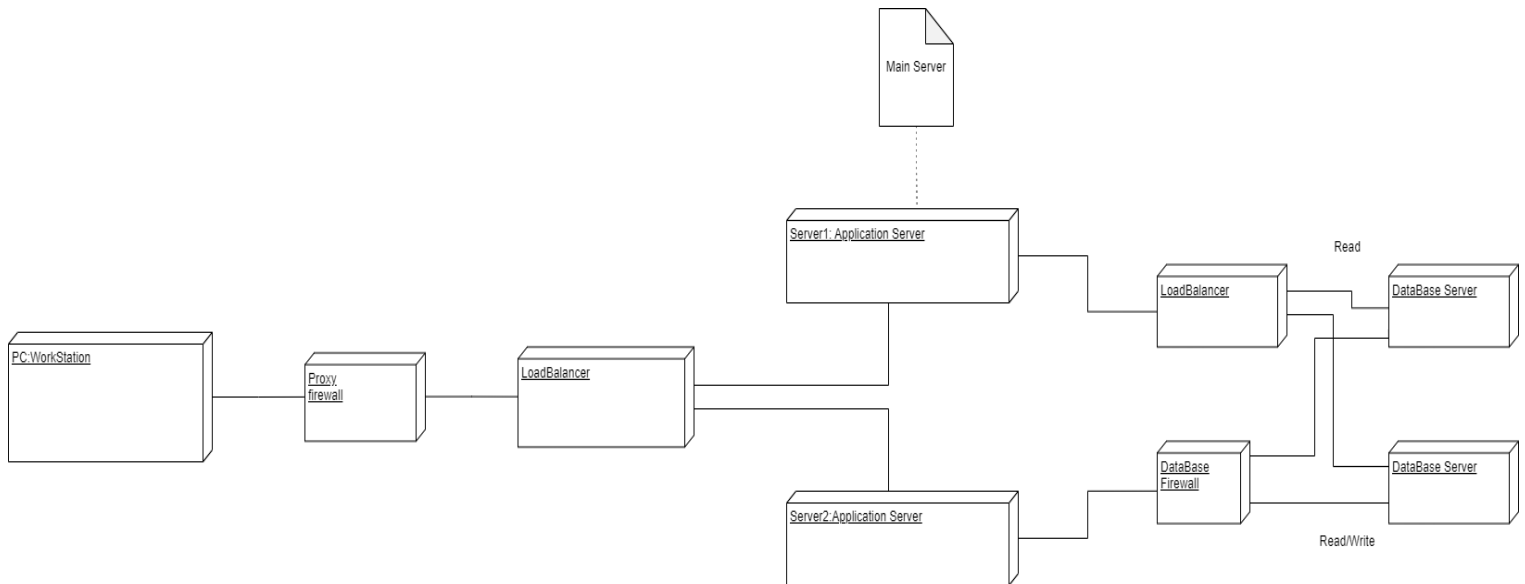
Step 5: Instantiate Architectural Elements, Allocate Responsibilities, and Define Interfaces

The instantiation design decisions are summarized in the following table:

Design Decisions and Location	Rationale
The application server uses active redundancy and load balancing to improve system reliability and performance.	Active redundancy and load balancing improve the application server's reliability, performance, and scalability by distributing incoming requests across multiple instances of the server.
Utilize technical support to implement load balancing and redundancy.	The system can avoid developing a unique solution that might be less developed and more difficult to maintain by employing current technological solutions for load balancing and redundancy.
The workstation and the server application should each have a firewall proxy between them.	Adding a firewall proxy to the workstation and server applications adds an extra layer of protection to manage and monitor incoming traffic from both ends, preventing unauthorized access and malicious activities, and improving the security of data transmission and communication.

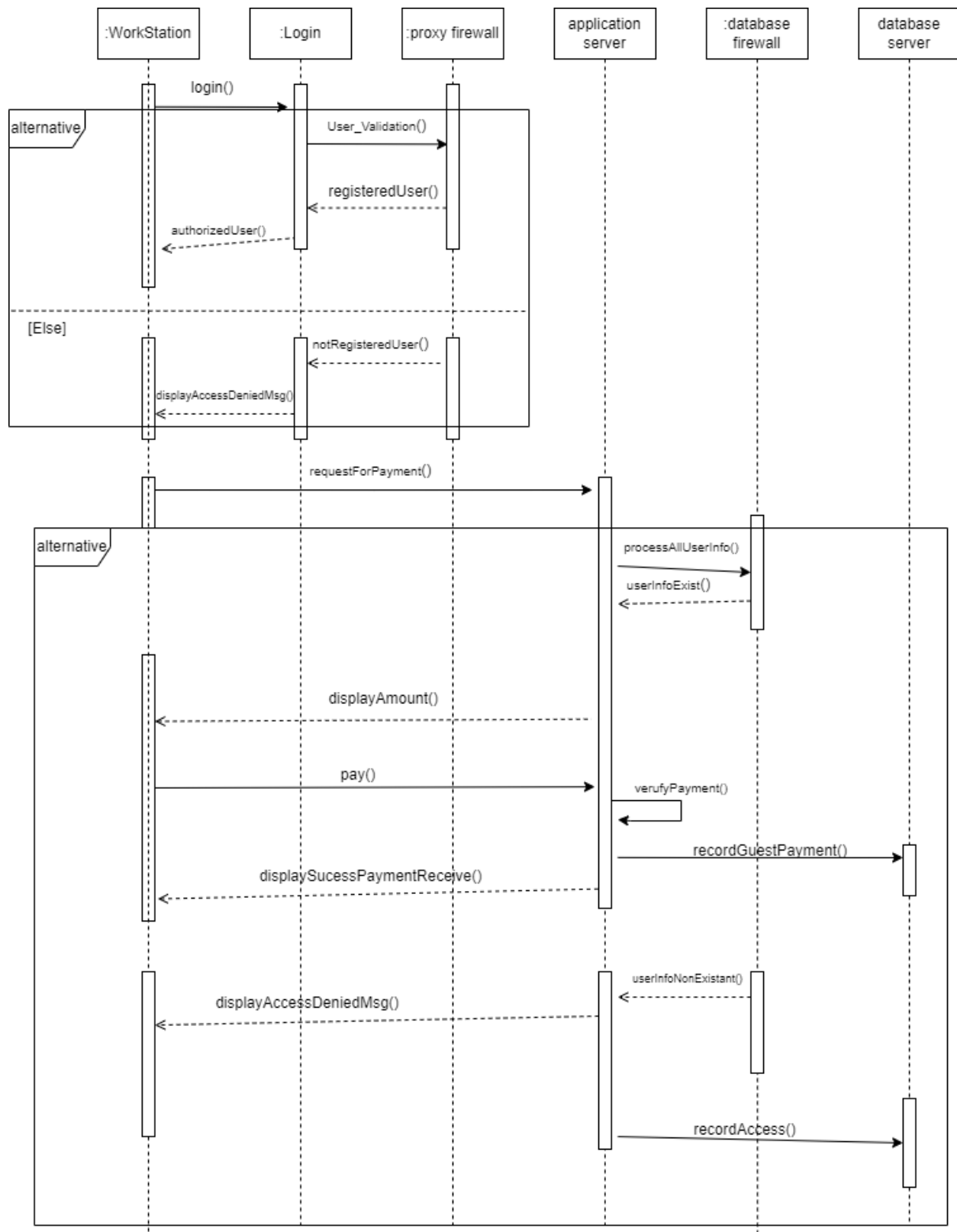


Step 6: Sketch Views and Record Design Decisions



Element	Responsibility
Load balancer	To minimize the overloading of a single application server and lower the danger of system crashes, a load balancer splits user traffic among two or more application servers. Users are still able to use the system while the failed server is recovering if one or more servers encounter excessive demand or failure. This is done by the load balancer, which diverts traffic to the remaining server(s).
Proxy firewall	The application server's security will be improved by this component by filtering and vetting all service requests and messages.
Database firewall	This component is used to guard against unauthorized access by preventing non-authorized individuals and networks from logging in and monitoring the database for potential assaults.





Step7: Perform Analysis of Current Design and Review Iteration Goal and Achievement of Design Purpose:

Not addressed	Partially addressed	Completely addressed	Iteration Design decisions
		QA-4	Redundant application servers have decreased the risk of system failure.
	QA-2		The system will operate more effectively if a load balancer is included.
	QA-6		The system's scalability will be increased by using a load balancer and replicating the application server.
		QA-2	The system's security will be improved with the addition of a firewall.
		CON-5	A responsive web application design enables accessibility and usability on many devices and screen sizes, offering a smooth and practical user experience.

