OntarioTech
UNIVERSITY

SOFE 3650U: Software Design and Architecture
Final Report
Modifying Gym Website

Name	*Student Number*
Arzika Khan	100753164
Shwan Majeed	100749077
Laith Hasan	100738896
Gobikah Balaruban	100742539

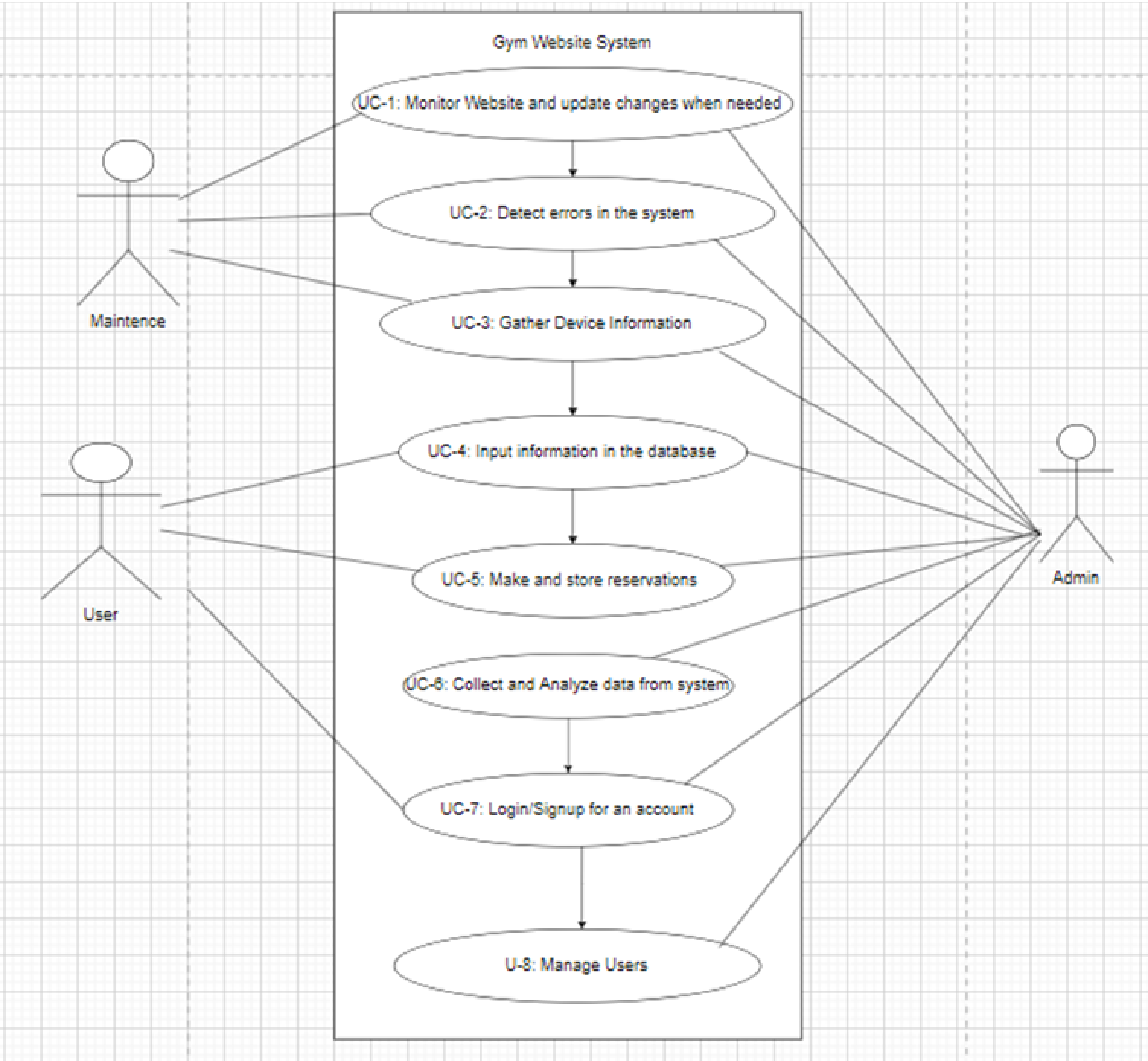
Table of Contents

1. Business Use Case

The growth of gym websites since covid started is unbelievable, and this happened due to the new covid rules and restrictions, so we decided to upgrade and modify our pre-existing gym website to grow our gym business financially and to provide more quality training sessions to our members. Moreover, we also want to ensure our members are able to work out during this pandemic safely and practice physical distancing guidelines. To do so, we created some use cases that will help us reach our goal in a timely manner.

2. System Requirements

2.1 Use Case diagram and description


Gym Website System
UC-1: Monitor Website and update changes when needed
UC-2: Detect errors in the system
UC-3: Gather Device Information
UC-4: Input information in the database
UC-5: Make and store reservations
UC-6: Collect and Analyze data from system
UC-7: Login/Signup for an account
U-8: Manage Users
Maintence
User
Admin

Use Case	**Description**
UC-1:	The website will ensure to be working and up to date with the current information of the Gym.
UC-2:	To ensure the website is properly, the admin and maintenance have to check if there are any errors in the system that affects the functionality. When an error occurs, it needs to be fixed as quickly as possible so the website does not have any down time.
UC-3:	This use case will be used in order to gather device information, such as reservations and user profiles for log in.
UC-4:	The user should be able to add payment information, book time slots, add personal information(i.e name, address, phone number, email) in the system.The admin should be able to add information about the gym such as opening hours, reservation times, general information about the gym and pricing information for products and services offered.
UC-5:	Reservations for gym time slots and fitness classes should be allowed and stored in the website's database so only one user has a time reserved at a time.
UC-6:	With the data given to the website, the administrator will determine the efficiency of the current network by collecting and analyzing the website data. This will ensure network health.
UC-7:	The log in function will help users identify themselves in a safe manner to create reservations.
UC-8:	The administration can visualize the information and make changes to configurations to ensure user safety.

2.2 Quality Attribute Scenarios

ID	Quality Attribute	Scenario	Associated Use Case
QA-1	Performance	The system should be able to update whenever there is a change made by the user as quickly as possible. Whenever someone books an appointment, the gym should be able to see it on their end.	UC-1
QA-2	Modifiability	A new membership option is created for users and is going to be included on the website. This does not get rid of any pre-existing membership options and is only an addition to what is already available.	UC-8,UC-4
QA-3	Availability	Users could use the website services (i.e booking a gym time slot and purchase gym membership) during their convenient time. The system should be available to use 24/7.	UC-2,UC-4,UC-5
QA-4	Security	A user wants to purchase a gym membership and their personal information (billing information and address information) is secure.	All
QA-5	Scalability	A new membership option is added and that causes a spike in users and requests. The system will handle the sudden increase in load.	UC-2
QA-6	Usability	The user should be able to use the system anywhere they are in the world. The website should be accessible with anyone with a disability like using colors so that people who are color blind can read and use this website.	UC-6

2.3 Constraints

ID	**Constraint**
CON-1	A minimum of 100 simultaneous users must be supported on the website.
CON-2	The system must be accessed through a web browser (Chrome, Safari ,Firefox) in different platforms: Windows, OSX, and Linux.
CON-3	An existing relational database server must be used. This server cannot be used for other purposes than hosting the database.
CON-4	Bookings of the last 24 hours must be stored.
CON-5	A user must make a maximum of 2 bookings everyday.
CON-6	Performance data needs to be collected in intervals of no more than 5 minutes, as higher intervals result in time servers discarding data.

2.4 Architectectural Concerns

ID	**Concern**
CRN-1	Creating the website
CRN-2	Leveraging the team’s knowledge on Databases and HTML
CRN-3	Creating a timeline for members for the team
CRN-4	Distributing tasks among team members

3. Design Process

3.1 ADD Step 1: Review Inputs

Category	Details
Design purpose	The purpose of our design was essentially to create a gym reservation website for users, where users can create and modify reservations for a gym.
Primary functional requirements	The primary use cases that were selected from 2.1 are: UC-1: The website status must be functioning at all times UC-2: Related to technical issues as detecting errors fixes any technicality UC-4: Relates to our core driver(QA-3) as the user should be able to input information all time.

Quality Attribute Scenarios:

Scenario ID	Importance to the Customer	Difficulty of Implementation According to the Architect
QA-1	High	High
QA-2	Medium	Medium
QA-3	High	Low
QA-4	High	Medium
QA-5	Medium	High
QA-6	High	Medium

Selected as drivers: QA-1, QA-2, QA-3
Constraints: All of the constraints discussed in the project process are included as drivers.
Architectural concerns: All of the architectural concerns discussed in the project proposal are included as drivers.

3.2 Iteration 1: Establishing an Overall System Structure

Iteration 1 was used in order to establish the overall system structure for our project. To start, we reviewed our inputs and identified which requirements will be considered as drivers. After that, we identified our selected drivers in order to address the general architectural design.

ADD Step 2:Establish Iteration Goal by Selecting Drivers

As this was the first iteration, the team's established goal for this iteration was to achieve the architectural concern 1, which was to create a website.

From this iteration goal, we then selected drivers to help influence the structure of the website. The selected drivers were:

- QA-1:Performance, meaning the system should be able to update whenever there is a change by the user, such as a reservation.
- QA-2:Modifiability, meaning an option for users to make an account should be available.
- QA-3:Availability, where the website's services should always be running 24/7 in order to satisfy users.
- CON-1: supporting users, where we need to make sure a minimum of 100 users can be supported simultaneously on the website
- CON-2: system accessibility, meaning the system must be supported on multiple platforms in order to satisfy different types of users.

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

For the iteration goal of creating a website, the elements of the system that need to be refined are:

- Supporting quality attributes
- Supporting project concerns
- Entire website

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

Design Decisions and Location	**Rationale**
Logically structure the client part of the system using the RCA (Rich Client Application) reference architecture	This reference architecture was chosen due to its capability in supporting our system. This architecture helps create a higher server capacity (CON-1). Also provides more flexibility for personal computers (CON-2).
Logically structure the server part of the system using the Service Application reference architecture	This architecture doesn't help provide a user interface, but exposes serverices that are consumed by other applications.
Physically structure the application using the three-tier deployment pattern	This client-server architecture stores information, handles logic, and has a GUI which is communicated throughout the system; this helps achieve QA-1, as well as QA-3.

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

The instantiation design decisions considered and made are summarized in the following table:

Design Decisions and Location	**Rationale**
Create a module dedicated to changes websites data due to user's inputs in Service Application	This will help further facilitate the achievement of QA-1 and QA-3.
Create a module for users to create accounts in Service Application	This will help further facilitate QA-2.
Remove local data sources in the Rich Client Application	Since the network connection is reliable, there is no need to store data locally.

The results of these instantiation decisions are recorded in the next step. In iteration 1, it is typically too early to precisely define functionality and interfaces. In the next iteration, which is dedicated to defining functionality in more detail, interfaces will begin to be defined.

ADD Step 6:Sketch Views and Record Design Decisions

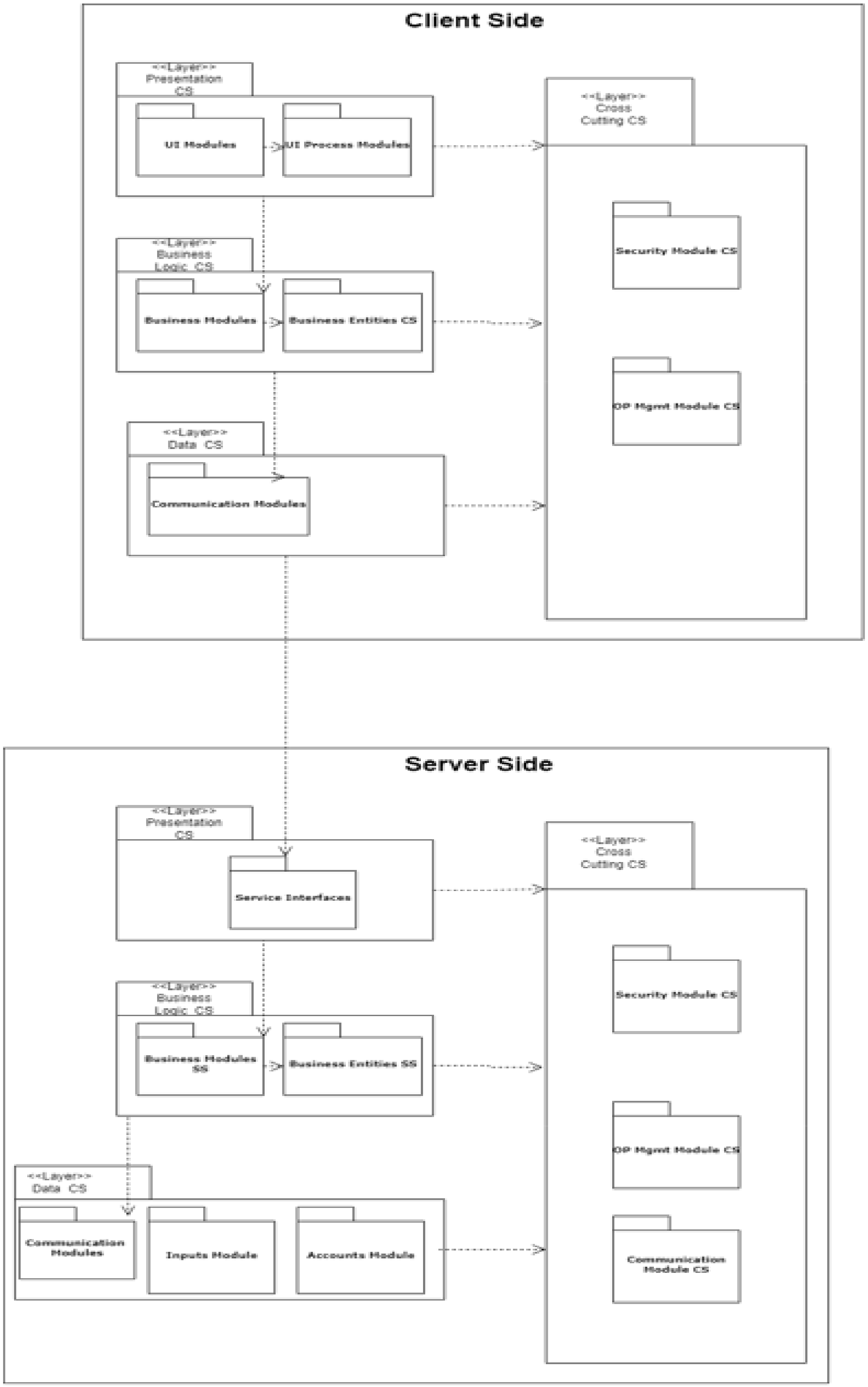


Figure 1.1: Module view of two reference architectures

Information table for Figure 1.1:

Element	**Responsibility**
Presentation Client Side CS	This layer contains modules that control user interaction and use case control flow.
Business Logic CS	This layer contains modules that perform business logic operations that can be executed locally on the client side
Data CS	This layer contains modules that are responsible for communication with the server.
Cross-cutting CS	This "layer" includes modules with functionality that goes across different layers, such as security, logging, and 1/0.
UI process modules	These modules are responsible for control flow of all the system use cases (including navigation between screens)
Business modules cs	These modules either implement business operations that can be performed locally or expose business functionality from the server side.
Business entities CS	These entities make up the domain model. They may be less detailed than those on the server side.
Communication modules CS	These modules consume the services provided by the application running on the server side.
Services server side SS	This layer contains modules that expose services that are consumed by the clients.
Business Logic SS	This layer contains modules that perform business logic operations that require processing on the server side.
Data SS	This layer contains modules that are responsible for data persistence
Cross-Cutting SS	These modules have functionality that goes across different layers, such as security, logging, and 1/0.
Service interfaces SS	These modules expose services that are consumed by the clients.
Inputs Module SS	This module stores and updates inputs given by the user into the website.
Accounts Module SS	This module helps create accounts for users.

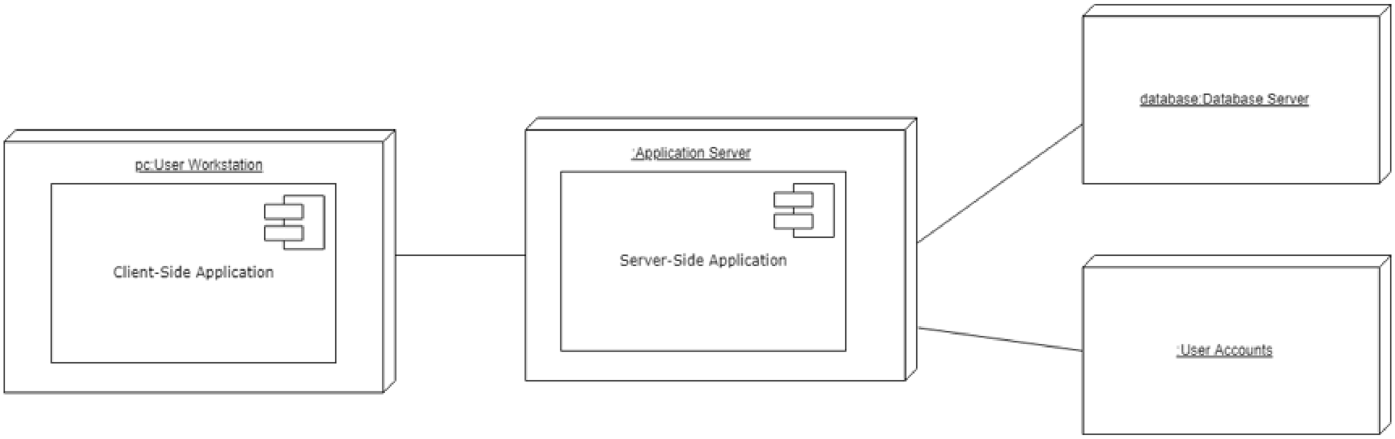


Figure 1.2: Initial Deployment Diagram

Information table for Figure 1.2:

Element	**Responsibility**
User workstation	The user's PC, which hosts the client side logic of the application
Application server	The server that hosts server side logic of the application and also serves web pages
Database server	The server that hosts the legacy relational database
User Accounts server	This server hosts all of the user’s account data

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

The following table summarizes the design progress using the Kanban board technique:

Not Addressed	**Partially Addressed**	**Completely Addressed**	**Design Decisions made During the Iteration**
	UC-1		Selected reference architecture establishes the modules that will support this functionality.
	UC-2		Selected reference architecture establishes the modules that will support this functionality.
	UC-4		Selected reference architecture establishes the modules that will support this functionality.
		QA-1	The systems performance (ability to update) was addressed in the Service application
	QA-2		The systems modifiability was addressed in the Service application
		QA-3	The systems availability was addressed in the Service application
QA-4			No relevant decisions made

QA-5			No relevant decisions made
QA-6			No relevant decisions made
	CON-1		This constraint was discussed in the RCA
	CON-2		This constraint was discussed in the RCA
		CON-3	The database server was addressed within the deployment diagram
		CRN-1	The whole iteration was provided in order to satisfy this concern
CRN-2			No relevant decisions made
CRN-3			No relevant decisions made, discussed outside iteration 1
CRN-4			No relevant decisions made, discussed outside iteration 1

3.3 Iteration 2: Identifying Structures to Support Primary Functionality

Iteration 2 will be focused on use cases that are important in identifying structures to support primary functionality. Identifying these particular elements will be useful in how the primary functionality is supported and the constraints of the project.

ADD Step 2: Establish Iteration Goal by Selecting Drivers
Iteration 2 will focus on the following use cases that support primary functionality:

- UC-1: Monitor website and update changes when needed
- UC-2: Detect errors in the system
- UC-4: Input information into the database

ADD Step 3: Choose One or More Elements of the System to Refine
For this iteration the element of the system we are choosing to refine is:

- Server
- Database
- Other aspects of the website in relation to the use cases

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

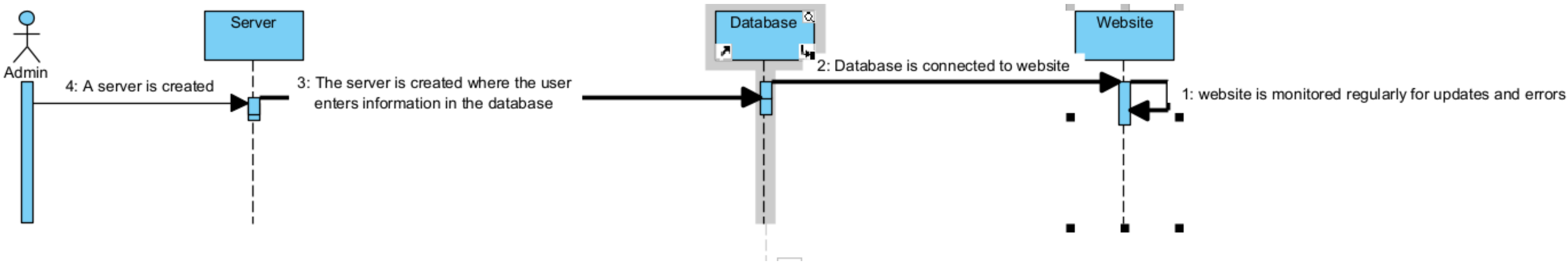
Design Decisions and Location	**Rationale and Assumptions**
Create a server	This server must be created so a least 100 users can enter the website with minimal amount of errors and downtime
Create a Website	Another design decision is creating a user-friendly website that can maintain and represent our system.
Create a Database	The database must be created in order to get and hold user and system data to help the main functionality in the system. The database will be within the website.

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

Design Decisions and Location	**Rationale**
Using Apache or Wamp to create our server that can hold at least 100 users	These software are used to create servers on many websites found on the web. These websites have many users at a time also having very few errors and downtime. This is associated with UC-2
Using web programming such as HTML,CSS,Javascript, Bootstrap,Node J.S,PHP etc. to create a user-friendly	Using web programming languages will help make the website more accessible where it allows to create a more visually appealing website and the user have access to the website and help connect to the database. This is associated with UC-1
Using a SQL data management system to create our database.	A data management system will store data in the system when new information is being entered. This will also be easier to retrieve information when being asked by a user or admin. This is associated with UC-4.

ADD Step 6: Sketch Views and Record Design Decisions

Sequence Diagram:


Admin
Server
Database
Website
4: A server is created
3: The server is created where the user enters information in the database
2: Database is connected to website
1: website is monitored regularly for updates and errors

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

Not Addressed	**Partially Addressed**	**Completely Addressed**	**Design decisions made during the iteration**
		UC-1	The website will be updated regularly used various softwares and monitored regularly

		UC-2	The optimal server will be used to address this use case
	UC-3		Partially addressed as many use cases discussed is related to this and not directly
		UC-4	The solution for this was addressed for this iteration
	UC-5		Concepts related were addressed but not directly
UC-6			This was not addressed
	UC-7		Related to other use cases address but not directly

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

Iteration 3 will address the quality attribute scenario driver chosen. Now that the fundamental structural decisions are made, more rationale can be put into fulfilling one of the important quality attributes. This iteration will focus on availability (QA-3).

Step 2: Establish Iteration Goal By Selecting Drivers
This iteration will focus on the following scenario: The system experiences an unknown event that causes a failure. The management system resumes activity in less than 5 minutes.

Step 3: Continue One or More Elements of the System to Refine
For the scenario described above, the elements that are refined are:

- Website Server
- Database Server

Step 4: Choose One or More Design Concepts That Satisfy the Selected Drivers
The design concepts that satisfy availability are defined below including the rationale and assumptions for each.

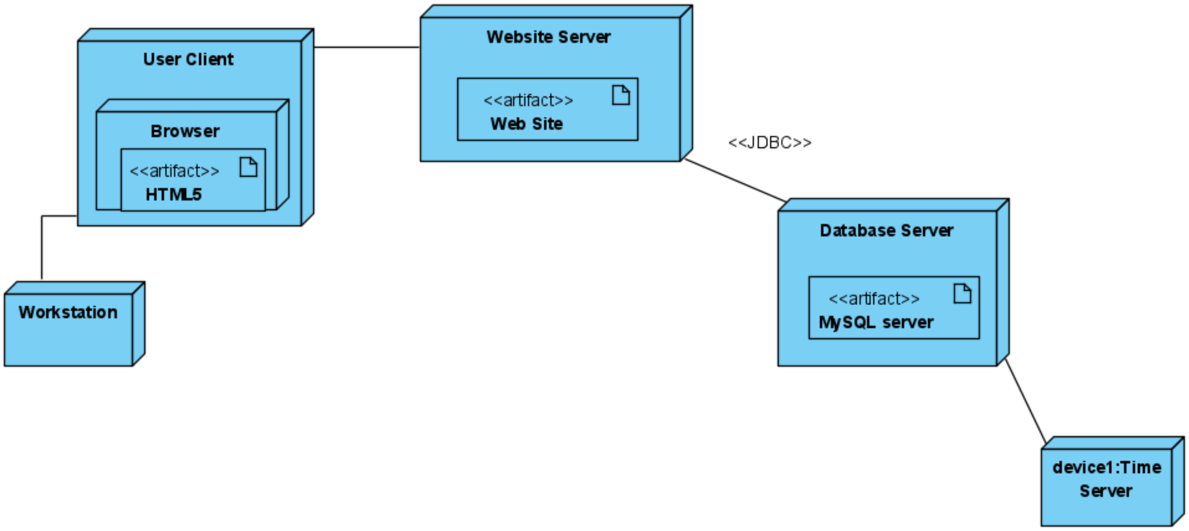
Design Decisions and Location	**Rationale and Assumptions**
Implementing predictive modelling to predict future outcomes to prevent those situations from happening.	Predictive modelling is based on statistics of previous data and is used to foresee any failures that may occur.
Implementing exception prevention to prevent the website from crashing in the first place.	It is good to implement this strategy for events that are exceptional. Exception prevention is a good practice to handle unplanned events from occurring that may cause the site to crash.

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

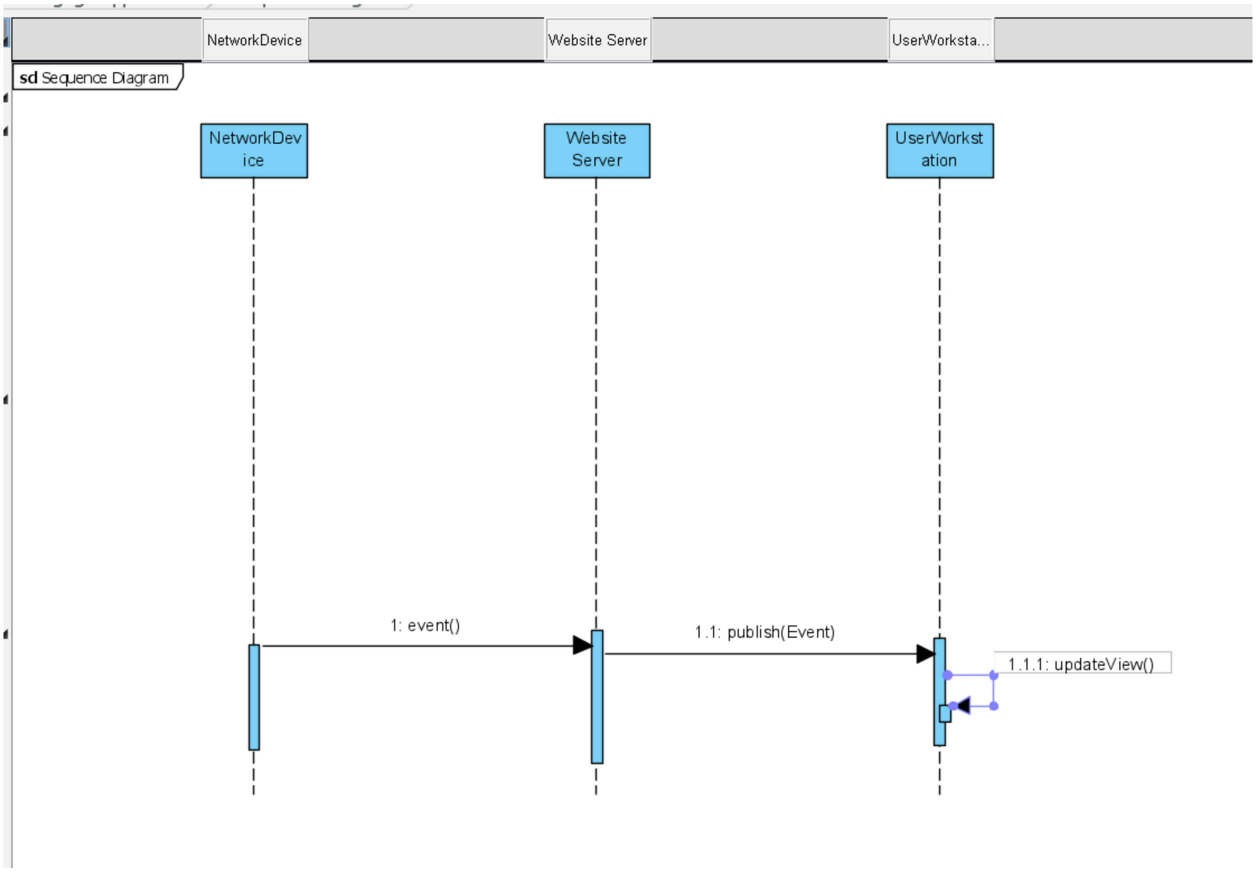
Design Designs and Location	**Rationale**
Maintaining audit trail to keep track of user and system activity.	This can help trace any activity that could lead to system failure, such as an attacker.
Handling exceptions and retrying.	The exceptions can be dealt with by reporting and handling it. The exceptions can be corrected.

Step 6: Sketch Views and Record Design Decisions

Refined deployment diagram (Key:UML)



Sequence Diagram (Key:UML)



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

Not Addressed	**Partially Addressed**	**Completely Addressed**	**Design Decisions Made During the Iteration**
	QA-1		Performance of the website is addressed, as it should run smoothly when a user makes processes
	QA-2		It is modified to handle any faults properly.
		QA-3	This iteration mainly focuses on this

			quality attribute and how the website tries to maximize its availability.
	QA-4		Security is important in case there are any faults that may cause system failure.
QA-5			Not addressed in this iteration
	QA-6		Usability is important in this iteration as it is addressing the way users access our site too.
		CON-1	Being able to handle many users without the website crashing is important.
	CON-2		The browser should work fine without any errors.
	CON-3		This database has to be ready
	CON-4		Bookings should be stored without any faults.
CON-5			Not addressed
	CON-6		Performance can affect the availability