

Submission Three: System Design

Reynard Etwaroo(USI#1039877), Tyreck Paul(USI#1024096), Zane Bishop(USI#1041471),
Ricardo Narine(USI#1040581), and Shivesh Mohamed(USI#1039116)

Faculty of Natural Sciences

Department of Computer Science, University of Guyana

CSE2101: Software Engineering I

Alicia Layne, Penelope DeFreitas, Juanelle Marks

December 2, 2022

Table of Contents

General Overview	2
Context Diagram	3
Figure 1. Context Diagram	3
Class Diagram	4
Figure 2. Class Diagram	4
Use Case Diagram	5
Figure 3.1 Use Case Diagram	5
Figure 3.2 Use Case Specification (Login)	6
Figure 3.3 Use Case Specification (View Database)	7
Figure 3.4 Use Case Specification (Perform Queries)	8
Activity Diagram	9
Figure 4. Activity Diagram	9
Sequence Diagram	10
Figure 5. Sequence Diagram	10
System Architecture (MVC)	11
Figure 6. System Architecture	11
Description	12
Scenario	13

General Overview

This document outlines various aspects of the Relational Database Management System (RDBMS) initially proposed by encapsulating and representing them in the form of different diagrams, e.g. context, class, etc. Additionally it also outlines the system architecture that will serve as the basis of the system's design.

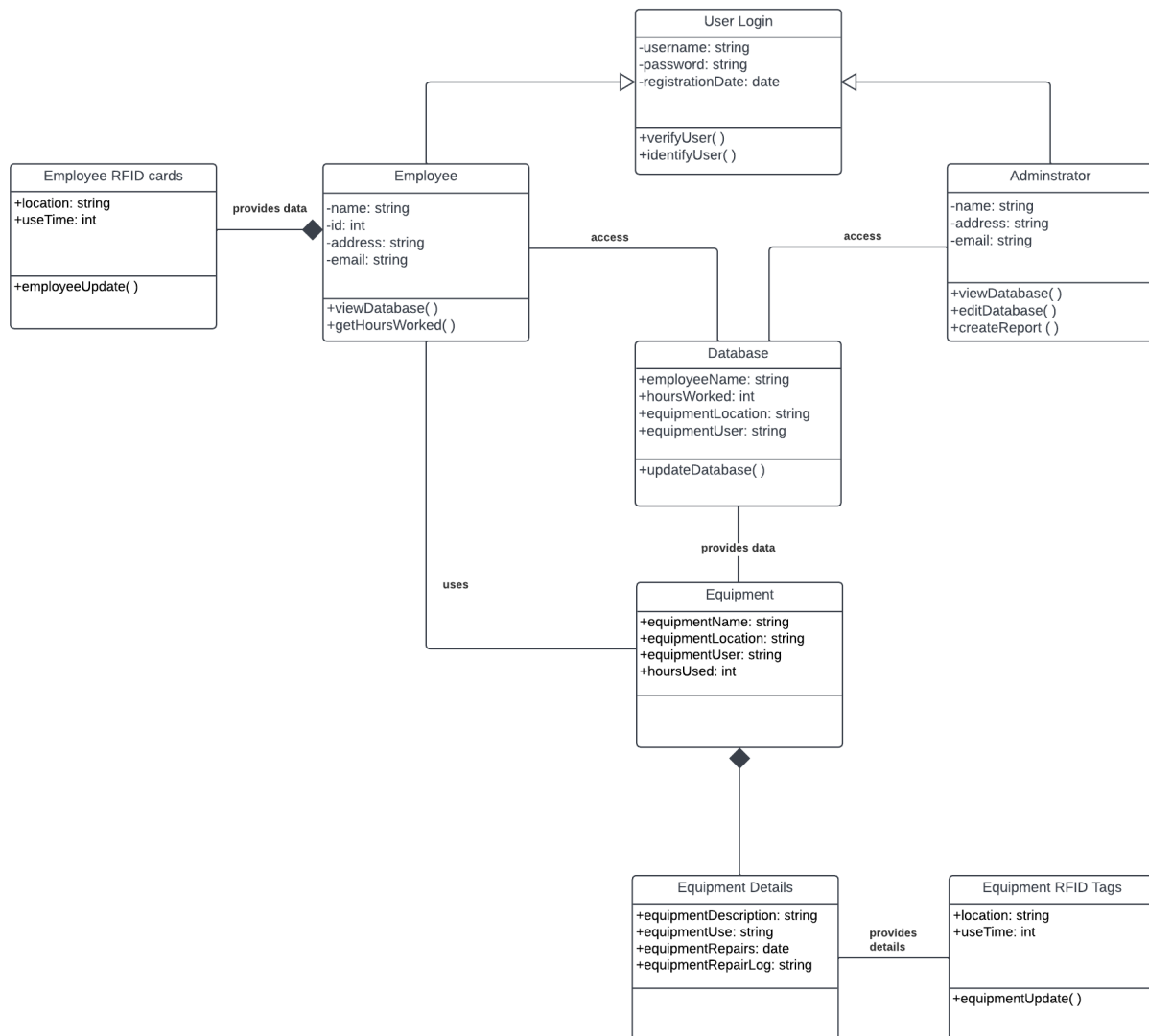
Context Diagram

The following diagram shows how the Relational Database Management System (RDBMS) will interact with various actors that comprise its domain.

Class Diagram

The following class diagram shows the relationships of the various classes within the Relational Database Management System (RDBMS).

Figure 2. *Class Diagram*



[Click to see full file](#)

Use Case Diagram

The following diagram shows a use case scenario between the system and the user with a diagram and a table.

Figure 3.1 Use Case Diagram

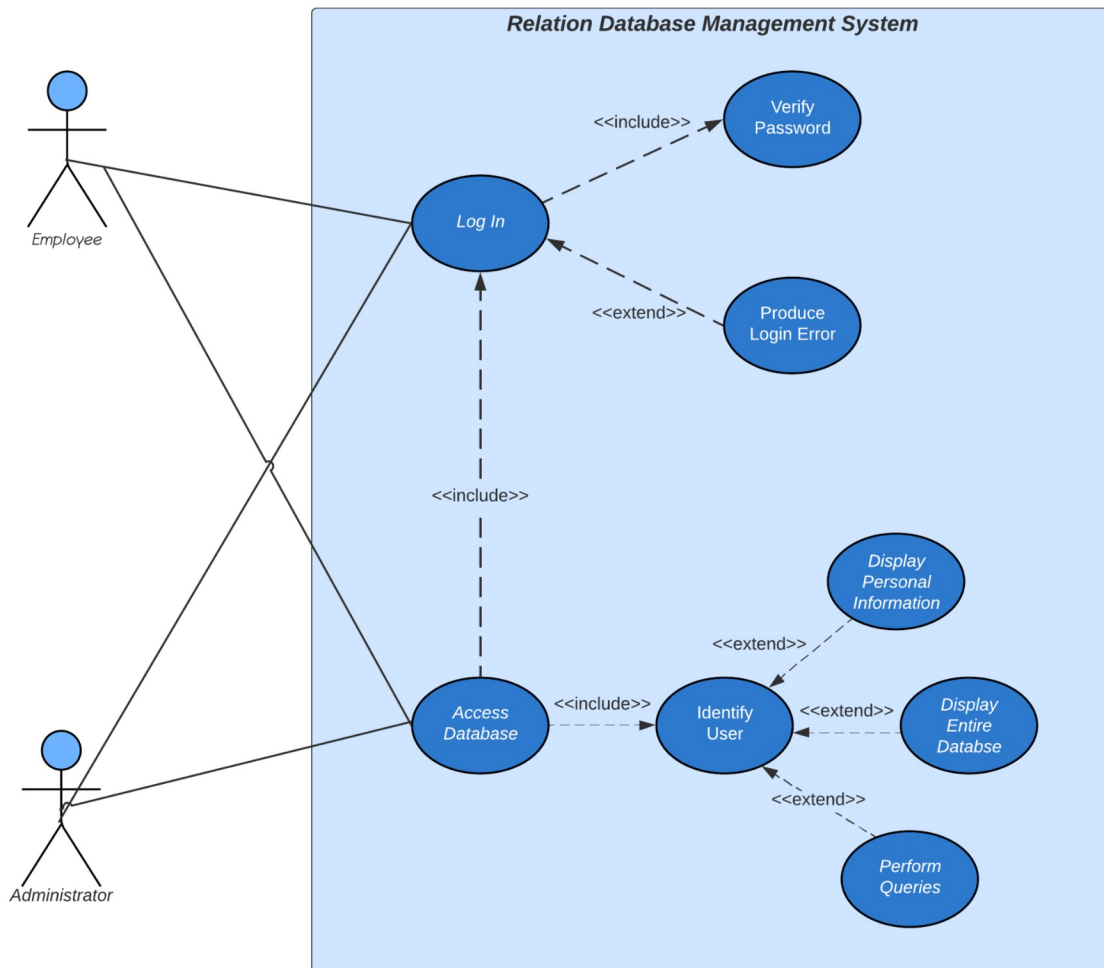


Figure 3.2 *Use Case Specification (Login)*

Use case:	Log In	
Actors:	Employees, Admin	
Cross Reference:	Fig 1.1	
Description	The user accesses the login screen on their device and enters their credentials. The system then verifies the credentials and identifies the user.	
Steps Specification		
General	Actor/System communication	System Response
1. Begins when the user accesses the login screen.		
	2. The user enters their credentials into the specified fields.	
		3. The system verifies the credentials and identifies the user as an Admin or Employee.

Figure 3.3 *Use Case Specification (View Database)*

Use case:	View Database
------------------	---------------

Actors:	Employees, Admin	
Cross Reference:	Fig 1.1	
Description	After the system has identified the user, it displays personal information if the user is an employee or displays the entire database if the user is an admin.	
Steps Specification		
General	Actor/System communication	System Response
1. After identifying the user, the system displays information pertinent to their position.		
		2. The system provides the personal records of an individual if they are identified as an employee.
		3. The system provides all records if they are identified as an administrator.

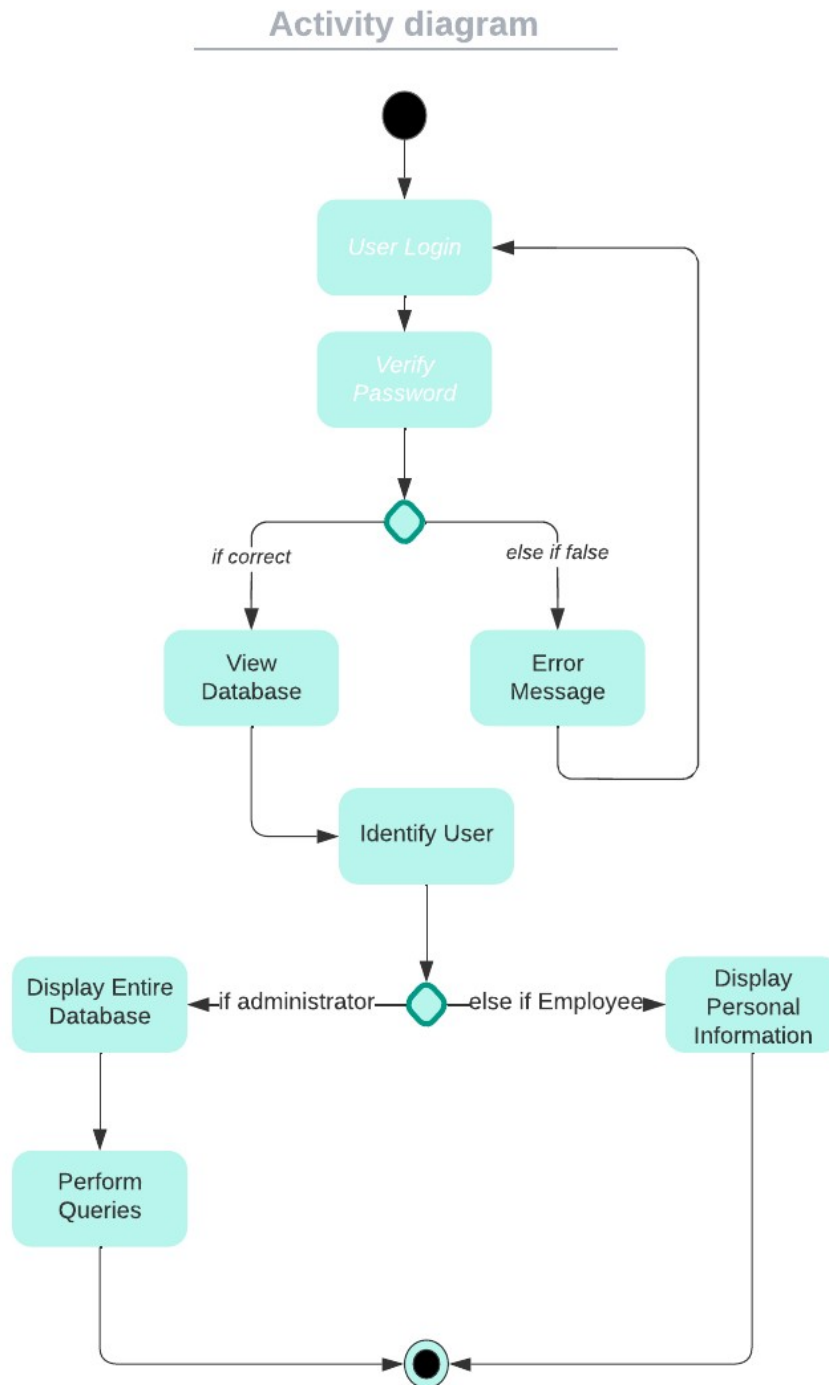
Figure 3.4 *Use Case Specification (Perform Queries)*

Use case:	Perform Queries	
Actors:	Admin	
Cross Reference:	Fig 1.1	
Description	Upon verifying that the user is an admin, the system then provides the user with various tools to aid in creating queries.	
Steps Specification		
General	Actor/System communication	System Response
1. After identifying the user, the system displays all records within the database.		
	2. The admin is presented with an interface that allows them to perform various queries depending on their needs.	
		3. The system iterates the database and executes the query to create an output.

Activity Diagram

This diagram shows the activity diagram explaining the sequence from the login to the database interface.

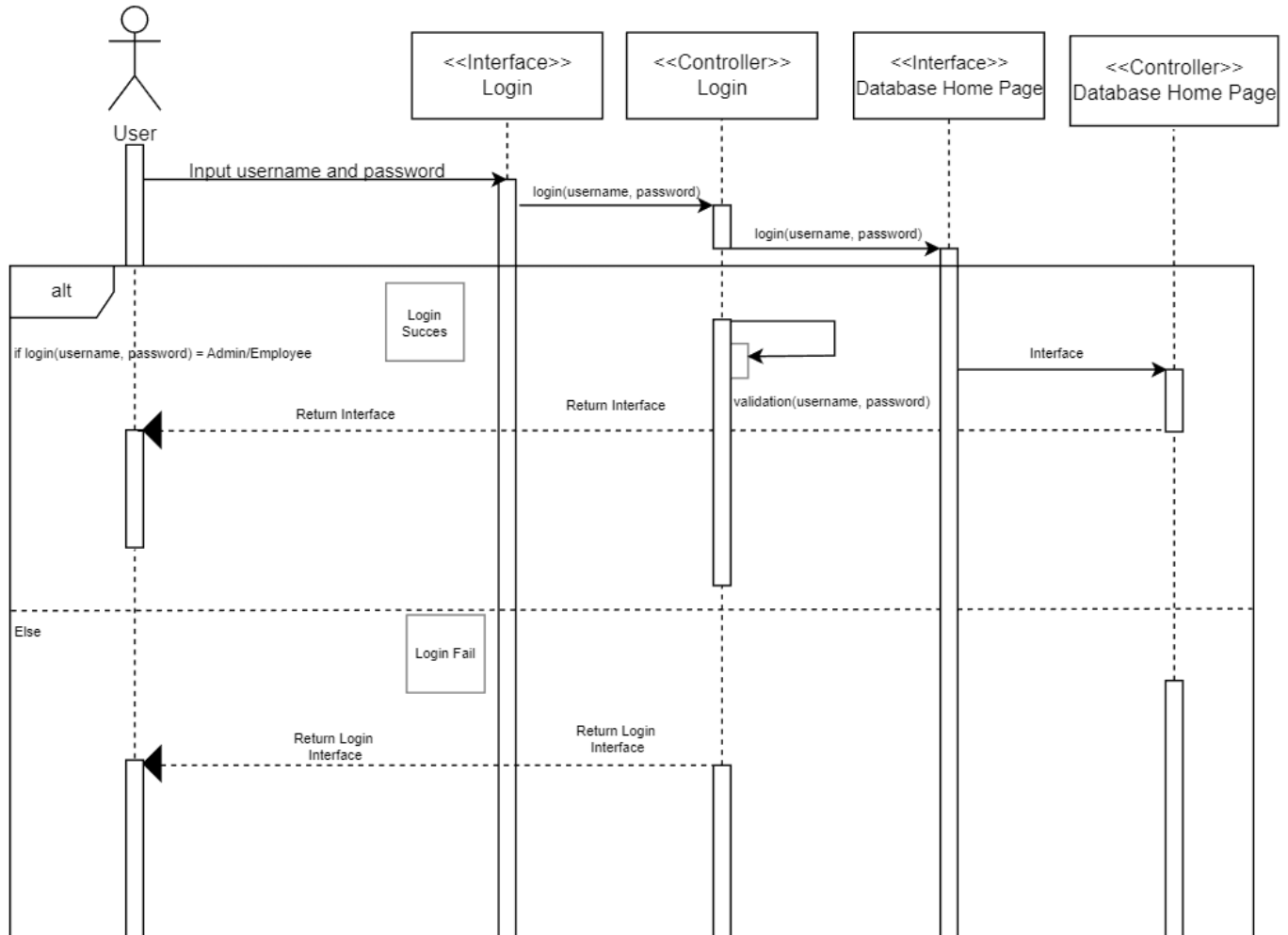
Figure 4. *Activity Diagram*



Sequence Diagram

The following sequence diagram shows the interrelation of the various classes within the Relational Database Management System (RDBMS).

Figure 5. *Sequence Diagram*

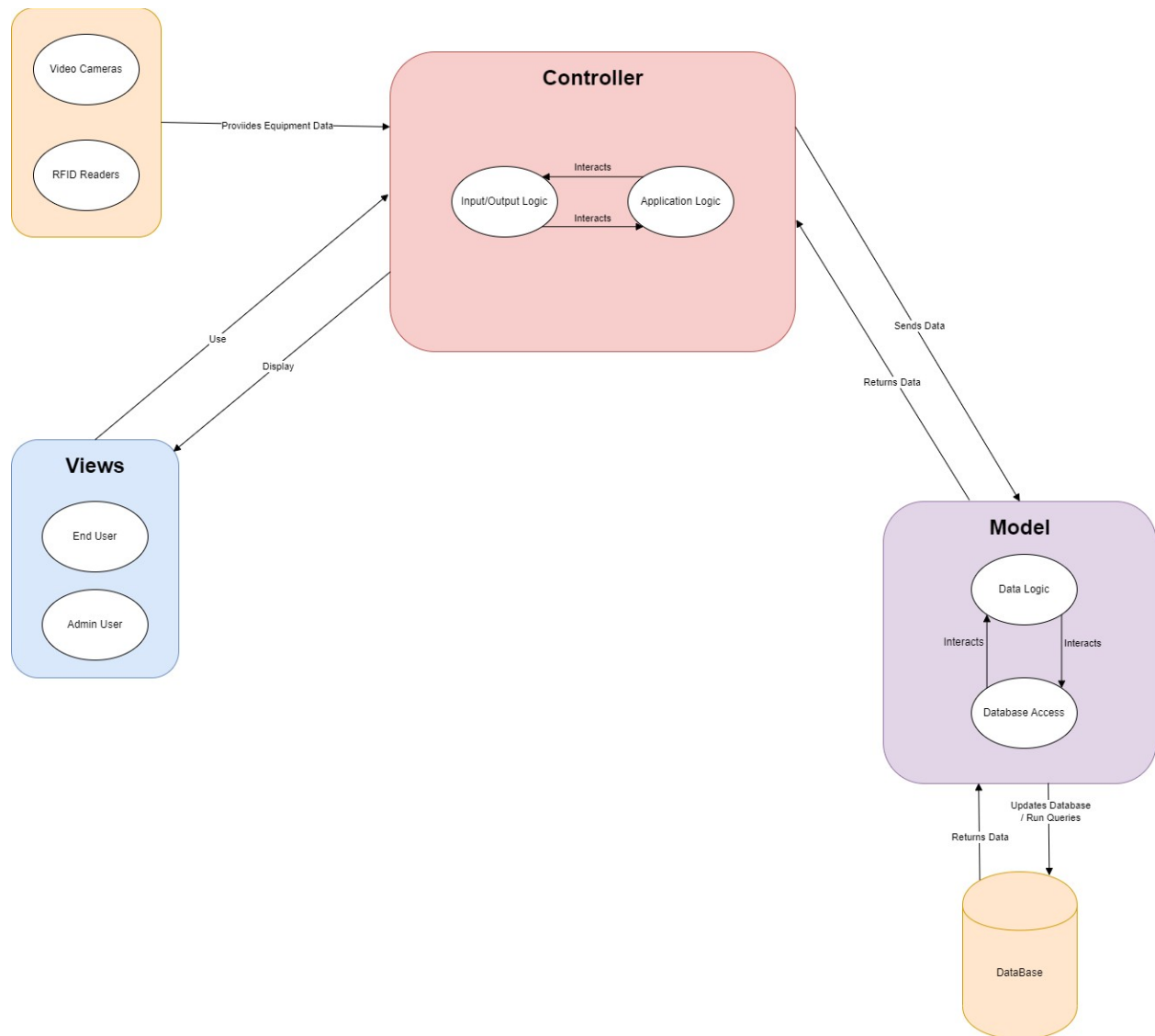


[Click to see full file](#)

System Architecture (MVC)

Figure 6. *System Architecture*

[Click here to see full file](#)



Description

View:

They are two primary views of the system: a regular end user (employee) view and an administrator (managers) view. These two components can inherit from one generalized class and append additional functionalities to create their respective interfaces with their associated privileges. End users, for example, are intended to only be allowed to use the system to book equipment usage time. However, administrators are allowed to view equipment usage logs, taken from the database.

Involved in the system are also input devices such as the cameras and rfid readers. These do not present views to the users but do serve a function of feeding data to the controller which then sends the data to the model to be stored in the database.

Controller:

The controller for this system essentially will comprise of the input/output logic and the underlying application logic. A controller in this system will not only respond to user input from the view interfaces but also manage input data coming from the rfid readers and cameras. The two logic modules: input/output logic and equipment monitoring logic will be coupled to provide whatever necessary functions are necessary depending on the desired input.

Model:

The model consists of data logic and database access logic. The data logic component will enforce the format in which it will be accessed while the database access logic will essentially allow you to interact with the database to get or update the desired data.

Scenario

Scenario: Administrator User would like to see a usage log for a specified time.

The view shown would be the Admin interface. On that interface, the user will select the desired functionality and criteria. In this case, the interface could display two date time slots to input the range of logs the Admin would like to see.

In the controller, the input logic will register the input, then the application logic will form a request to be sent to the model.

The model will then process the request data into a desirable format, which the database access logic will then take and convert to a query for the database. The results will be returned and sent to the controller which will process the data through the application logic and then the input/output logic will convert the data into a desirable format to be sent back to the view.