Department of Computer Science & Engineering
The University of Texas at Arlington

Architectural Design Specification
CSE 4316: Senior Design I
Summer 2020

eMarket

sell out Old, bring in New

Clouders
eMarket - A fully Cloud Based Web application

Mohan Karki
Roshan Kandel
Nabin Panthi
Sunny Raj Bhandari
Noel Tamang

Revision History

Revision	Date	Author(s)	Description
0.1	08.03.2020	MK	document creation
0.2	08.07.2020	MK, RK, NP, SB, NT	complete draft
0.3	08.08.2020	MK, RK, NP, SB, NT	release candidate 1
1.0	08.10.2020	MK, RK, NP, SB, NT	official release

Contents

List of Figures

List of Tables

1 INTRODUCTION

EMarket web application is designed especially focusing to help the local business and people to sell their products.It is an classified-ad websites where the local people or business can post their ad of the products that the buyers can search and contact them to pay and get the item. The key features of the application are listed below:
1) Registration for the new account, Login and logout.
2) Post an Ad for certain amount of time and can be sponsored to be displayed at the top of the page.
3) User can seen the seller review and contact them through provided information in the application.
4) User can search for the products through name, zip code.

This web application is not designed focusing for the particular customer rather it will be designed for the many people and local business i.e. overall class of customers. Thus, our product when released will be available for the general people.

2 System Overview

This section should describe the overall structure of eMarket software system. An architectural "layer" is the top-level logical view, or an abstraction, of eMarket design. EMarket consists of the three different layer as Presentation Layer, Application Layer and Repository Layer as shown in the figure below. This architecture layer give a easy and simple approach showing how the information is passed, received and stored. The top-level of system layer allows user to request and display the response received from application layer whereas application layer will performs the business logic using the data received from the repository layer doing the CURD operations.

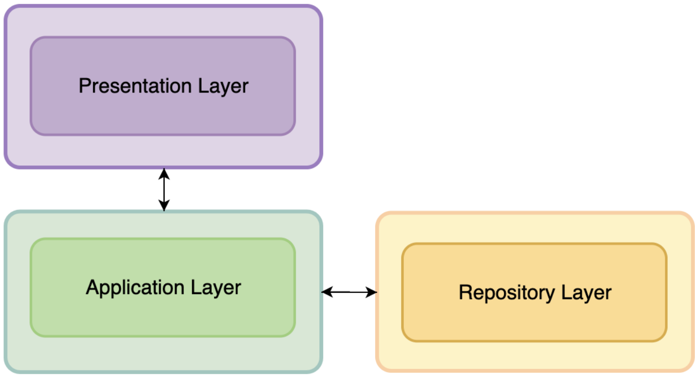

Figure 1: A simple architectural layer diagram

2.1 Presentation Layer

This is the first layer of the EMarket System. It is the front-end part consisting of the user interface. It is responsible for presenting the content and user interaction. In this layer, the user will make the request and system will present the response that is redirected from the system. This layer will be using Thymeleaf, CSS, bootstrap components for building this layer. Thymeleaf is the java HTML5 template engine whereas CSS is cascading style sheet to beautify the view and bootstrap is used for modern user interface. In this layer, there will be different components displayed like registration form, posting ad form, buttons to display the contact information and when the user will fill the details or making any other request, then this layer will communication with only the application layer.

2.2 Application Layer

After the presentation layer make the request then this layer play the important role performing mostly the service interface and business logic. In order to carry out the business logic, this layer will be using the Java programming Language and Spring framework. Through using spring framework, it will be using spring security library for authentication and authorization features. Since, this layer will act as the middle layer after receiving the request from the authenticated users, the business logic will handle and process by communicating with the Repository layer in order to store and retrieve data, and finally passing it to the presentation layer for displaying the user. Therefore, this layer is responsible handling the request from presentation layer, communicate with repository layer and process it for providing the application services in a consistent way.

2.3 Repository Layer

In this layer, Amazon Relational Database Services is being used in order to store the authenticated users information that is being received from the presentation layer and processed using the logic from application layer. Our system will be implementing the database management system, MYSQL through Amazon RDS. This layer will get request from the application layer and return back the stored information to be displayed in the presentation layer after being processed by the application layer. In this layer, there will be repository of users, products, role stored in the database and their attribute value will be received from the presentation layer that the user will provide.

3 Subsystem Definitions & Data Flow

Presentation layer will be displayed once the users browse the domain address of the eMarket in their web browsers. After this the user will make the request like Post an Ad, Login through the help of the thymeleaf, CSS and bootstrap and communicate with the application layer where the system will validate, authenticate and authorized tasks based on their request. Then this application layer will communicate with the repository layer with the storage logic and query and responds back to presentation layer sending response to users using thymeleaf, CSS and bootstrap.

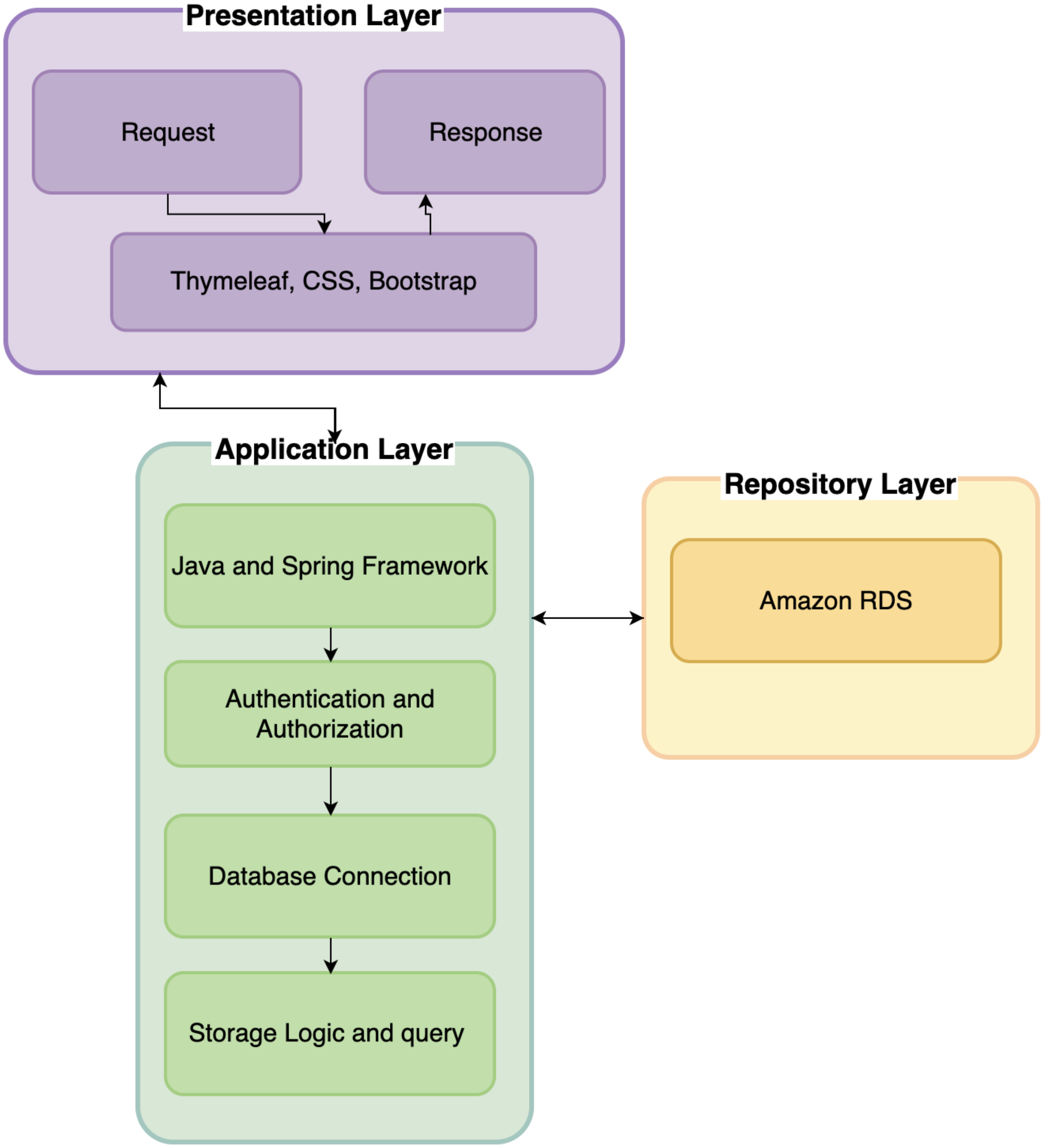

Figure 2: A simple data flow diagram

4 Presentation Layer Subsystems

This layer will plays the vital role in displaying the content to users whenever they will browse the eMarket domain address. It will be nice looking modern design that will be used to get the request from the users and response back with the useful output. This layer will presenting the different functionalities and features of the application such as login, posting an Ad through the help of the backend services available in the system.

4.1 Request

In this subsystem, when the users will open the domain address of the eMarket, they can look at the content available and presented by the system like the list of products, login and registration form. The users can make alot a request like viewing the full description of the products, in order to post an Ad, register to use the full services.

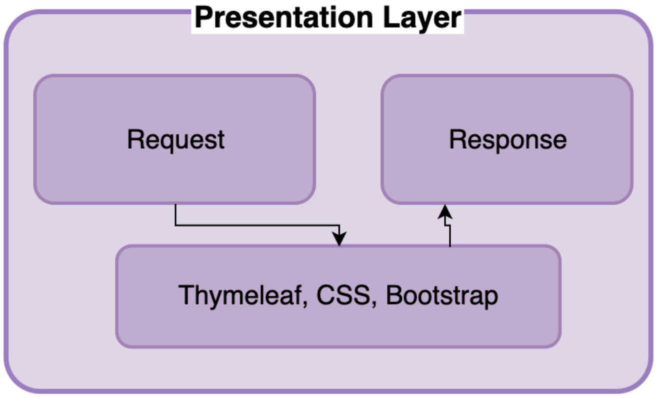

Figure 3: Presentation Layer subsystem description diagram

4.1.1 Assumptions

User will provide the correct details and information as instructed by the system which will be processed and passed to application layer.

4.1.2 Responsibilities

This sub-system is responsible to get the users data and instruction in order to pass to the Application layer through the help of keyboard, mouse or touch pad.

4.1.3 Subsystem Interfaces

Each of the inputs and outputs for the Request subsystem are defined below:

Table 2: Subsystem interfaces for Request

ID	Description	Inputs	Outputs
#1	User want to use the full features of the application	clicks in the Registration Forms button	Displays the Registration Form
#2	Users want to see their dashboard.	Clicks the Login button for it.	Displays the Login Form
#3	Users want to post an Ad of their product.	Clicks the Ad button for it.	Displays the Ad Form
#4	Users want to see the contact details of the sellers.	Clicks the seller info button.	Displays the seller information of respective seller.

4.2 Thymeleaf, CSS, Bootstrap

This is the very important sub system layer of the presentation layer which presenting in viewing and designing the user interface of the system. Thymeleaf is the Java Html5 template engine for displaying the data and information that is received from the server to users by adding beautify through the help of CSS and Bootstrap.

4.2.1 Assumptions

We can assume that the thymeleaf will provide convenience in presenting the data and using some logic like if-else in the HTML page whereas CSS will beautify the way of presenting it through the help of bootstrap.

4.2.2 Responsibilities

Thymeleaf will more responsible for creating a web page and helps to display the items as needed by the users request like forms.

4.2.3 Subsystem Interfaces

Each of the inputs and outputs for this subsystem are defined below:

Table 3: Subsystem interfaces for Thymeleaf, CSS, Bootstrap

ID	Description	Inputs	Outputs
#1	The system will display nice looking registration form	Request for the Registration Form	Displays the Registration Form
#2	Users want to see their dashboard.	Request to login	Displays the Login Form
#3	Users want to post an Ad of their product.	Request for Post an Ad	Displays the Ad Form
#4	Users want to see the contact details of the sellers.	Request seller info.	Displays the seller information of respective seller.

4.3 Response

The subsystem is mostly important to display the results from the users based on their request through the help of the thymeleaf.

4.3.1 Assumptions

We can assume that the desired response will be provided by the system to the users for every time they make a request and it will be quick.

4.3.2 Responsibilities

It will responsible to show the desired output that the users want based on the request they made.

4.3.3 Subsystem Interfaces

Each of the inputs and outputs for this subsystem are defined below:

Table 4: Subsystem interfaces for Response

ID	Description	Inputs	Outputs
#1	User want to use the full features of the application and wants to register an account.	Users fill all correct details of the registration form	Registration form complete
#2	Users want to see their dashboard.	enters the valid login credentials.	Authenticated and authorized to see the dashboard.
#3	Users want to post an Ad of their product.	Clicks the Ad button for it and fill the details	Ad successfully posted for listing message sent.

5 Application Layer Subsystems

This application layer section breaks down layer abstraction to another level of detail. In this layer, the application will be using the java programming language as the base programming language and spring framework with its spring security features for authentication and authorization of the users. After this, the application will connection to the database and store the user information and retrieve from the database using the storage logic and query.

5.1 Java and Spring Framework

In this layer of the sub-system, the application will be using the Java programming language and the Spring framework for the designing and building this application. Spring MVC which is Model View Controller is so effective in building the website. Here, the model are build which is processed by the controlled for mapping and the view are used for displaying it.

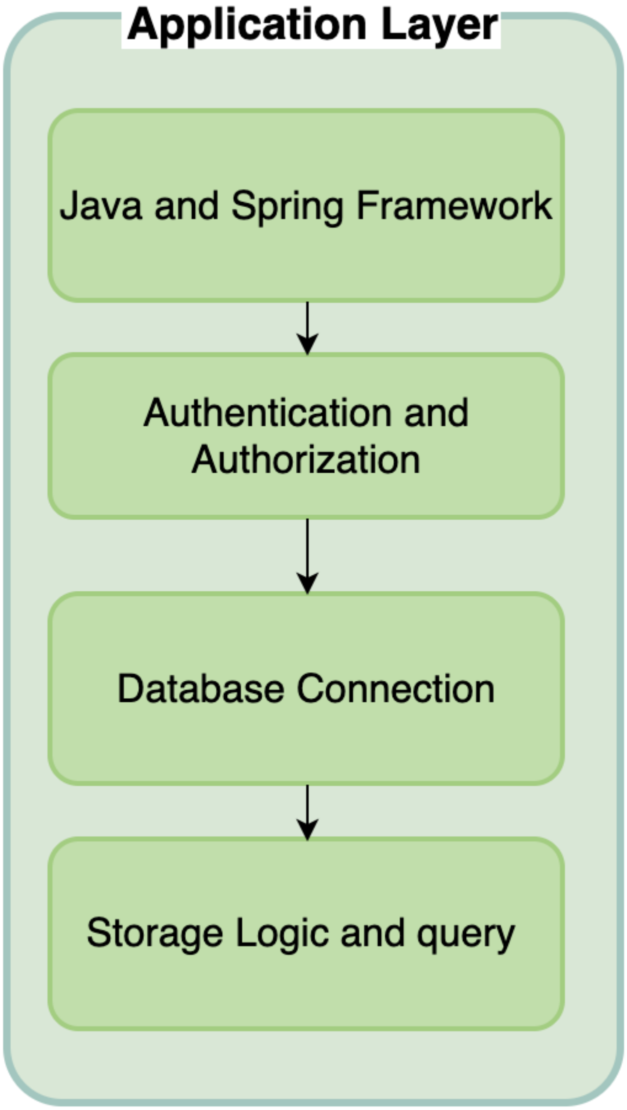

Figure 4: Application Layer subsystem description diagram

5.1.1 Assumptions

We assume that java and spring framework will so efficient in creating, designing and building this web application.

5.1.2 Responsibilities

It will be responsible for creating the bean, controller layer and other services interface.

5.1.3 Subsystem Interfaces

Table 5: Subsystem interfaces for Java and Spring Framework

ID	Description	Inputs	Outputs
#1	Debug the certain function	Java function used for printing.	Prints in the console

5.2 Authentication and Authorization

In this sub system, spring security play the vital role for the authentication and authorization for the users role based on the credentials they provided for login.

5.2.1 Assumptions

Users will provide the valid credentials for the registration and login.

5.2.2 Responsibilities

It will responsible for validating the user credentials and authenticated and authorized for using the services provided by the application.

5.2.3 Subsystem Interfaces

Table 6: Subsystem interfaces for Authentication and Authorization

ID	Description	Inputs	Outputs
#1	Users will be authenticated and authorized for using the application.	Correct login credentials entered	Welcome message is displayed in the screen.
#2	Users will not be authenticated and authorized for using the application.	Incorrect login credentials entered	Error message is displayed in the screen.

5.3 Database Connection

In this sub system, it will setup and make the connection with the Repository layer i.e. database by defining the parameter needed for it in application properties of the application. In this layer, the database URL, username, password and MYSQL java connector is used. Similarly, for the JPA, the dependency is added in the pom.xml file and the hibernate is also defined in application properties.

5.3.1 Assumptions

We assume that database connection will be quick and stable while running the application.

5.3.2 Responsibilities

This layer is responsible for the efficient communication with the database and its services.

5.3.3 Subsystem Interfaces for Database Connection

Table 7: Subsystem interfaces for Database Connection

ID	Description	Inputs	Outputs
#1	Make connection with the database	enters credentials of the database	Connection successful message.

5.4 Storage Logic and query

This is the important sub system of the application layer where the logic for storing and retrieving data from Repository layer takes place. In this layer, our application will be using JPA(Java Persistence API), hibernate for making the easy and quick communication with the Repository layer. Here the logic includes creating the table for database, retrieving user information by using the query such as findbyID(), save() function available through JPA.

5.4.1 Assumptions

We assume that this logic designed here will be quick in order to save and retrieve info to present for the users in the presentation layer.

5.4.2 Responsibilities

This layer is responsible for storing and doing query efficiently without any application malfunction through available database logic defined here.

5.4.3 Subsystem Interfaces

Table 8: Subsystem interfaces for Storage Logic and query

ID	Description	Inputs	Outputs
#1	Users want to see the details of the seller.	Request for details	returns the information of the seller after retreiving from database
#2	Users want to update their information for contact through dashboard.	Users enter the update information	Information is stored in the database and user is notified for it

6 Repository Layer Subsystems

This is the repository layer of the subsystem where all user are stored present in the web. Similarly, it will also used to fetch the data for displaying in the presentation layer. In this section the user information such as name, email, phone number, address are stored as well as the product information such as product name, price, category, image will be stored in their respective tables or entity.

6.1 Amazon RDS

Amazon Relation Database services is the relational database service provided by the Amazon web services. It is since running in the cloud and helps to simply the setup, operation, and scaling of a relational database for applications. Under this services, we will be using the Database management system, MY SQL which will responsible for storing information based on the logic applied in application layer. There will be different Entity and their relation will be defined in this database workplace for more efficient and quick storage and retrieval of the information as per the user request.

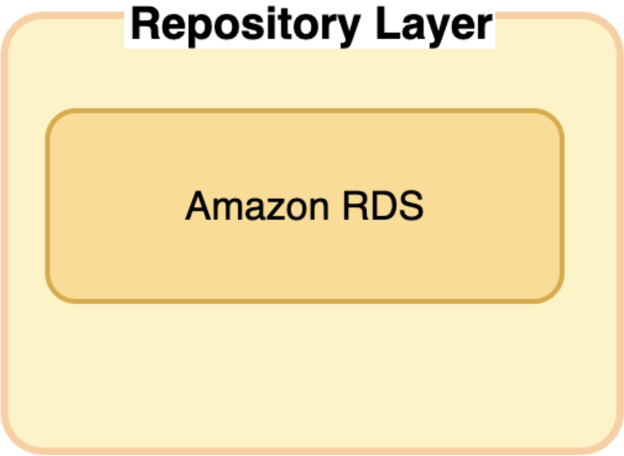

Figure 5: Repository Layer subsystem description diagram

6.1.1 Assumptions

We assume that this service will be online every time and each seconds for storing the information and retrieving using the database login available in the application layer.

6.1.2 Responsibilities

This system will be responsible for storing the information provided by the user in the presentation layer and using the logic the available in the application layer. Similarly, it will be also responsible for making the data available for retrieval at any time of day.

6.1.3 Subsystem Interfaces

Table 9: Subsystem interfaces for Amazon RDS

ID	Description	Inputs	Outputs
#1	User enter the data in the presentation layer such as registration, posting an Ad and it is stored in the database.	Gets the information from application layer.	Store the information
#2	Users want to view the products for buying in the presentation layer.	Query for data retrieval	Return the information based on request.

References