Bilkent University

Department of Computer Engineering

Senior Design Project

TechRank

High Level Design Report

Serhat Hakkı Akdağ, Alperen Ustaömer, Mehmet Oğuz Göçmen, İlhami Kayacan Kaya, Pelin Elbin Günay

Supervisor: Çiğdem Gündüz Demir
Jury Members: Mustafa Özdal and Selim Aksoy

High Level Design Report
December 31, 2018

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Design Project course CS491/2.

Table of Contents

High Level Design Report

TechRank

1. Introduction

Online shopping is one of the rapidly growing trends in the last decade. People think that doing shopping with using websites or shopping apps is easier and less time consuming than doing shopping in the stores. Because, they can search the products they want to buy by using search bar without losing time while finding the products in the store. Customers are surfing on e-commerce websites not only for buying the products online but also, even when people are buying things from stores, they are checking out reviews and comments of products online. In any case, checking out reviews is quite common nowadays, as it is hugely beneficial. However, people do not want to read hundreds of comments due to timing constraints. In addition, it is not always easy to understand general consensus on comments because there are inconsistencies among comments. Because of that, seeing the products' rates that are constituted by analyzing all comments is beneficial for the user. By observing rates about the products, customers can easily decide which product is worth buying.

When we consider the common usage of online shopping, we decided to develop a web application in order to help people to decide and find out which technological product is beneficial and necessary for them by analyzing user comments from trusted e-commerce websites and by rating and ranking them into some related categories. Our application will analyze all comments scattered on web, then it will rank the devices on the same category using criteria. TechRank will also decide on overall rating for the device and overall rating for the producer company of the device.

1.1. Purpose of the System

The purpose of this Project is to save people's time to read comments of product in the e-commerce shopping website and provide general view of customer satisfaction. Lots of people that are using e-commerce website faced with this problem. Our project helps people to decide and find out which technological product is beneficial and necessary for them by analyzing user comments from trusted e-commerce websites and by rating and ranking them into some related categories. Our application will analyze all comments scattered on web, then it will rank the devices on the same category using criteria. TechRank will also decide on overall rating for the device and overall rating for the producer company of the device.

1.2. Design Goals

1.2.1. Usability

Usability is one of the most significant design goals of the TechRank since it is a web application that created to make user’s life easier by providing them general ideas about technical products. Also, TechRank also planned a way that target users should use system frequently, and every user should be able to use system without any difficulty or problem. Because of that, system interfaces must be easy to use and system should not cause any problem that reduce the usability of the web application. Therefore, usability is clearly is one of the most important design goals of the TechRank.

1.2.2. Reliability

Reliability is another design goal which always be considered as one of the most important design goals for the TechRank web application. Since TechRank will be daily used web application, system should update all of the information exist inside the service, frequently. In other words, users should be able to reach all of the new reviews that added inside system. By this way TechRank will protect its reviews and ranking truthiness. Therefore, reliability is essential design goal.

1.2.3. Performance

Another significant design goal is high-performance. Since system continuously update database according to updated reviews and users should reach these huge amounts of data frequently, system should work in a very short response time. By this way users will be able to use the TechRank Web Application efficiently with possible minimum latency.

1.2.4. Portability

Portability is another decided design goal of the TechRank. Since users could reach the application from different platforms. Although TechRank will not have any separated mobile application, its web application is designed in a way that perfectly reachable from web services of all type of mobile devices. So, during implementation process portability is one of the commonly considered design goals.

1.2.5. Adaptability

Since TechRank will take reviews and opinions not from just one commonly used shopping web pages, but also it will get data from various other pages. That’s why TechRank System must be able to adapt all of these different webpages. By this way system can successfully reach and take data from different pages and provide better results for users. So, adaptability is another design goal of the project.

1.2.6. Availability

TechRank is a free to use web-based application which will serve lots of different clients at the same time. That’s why it must be available for all of its users for all of the time.

1.3. Definitions, Acronyms, Abbreviations

NLP: Natural Language Processing

HTTP: Hypertext Transfer Protocol

ACID: It is a property set for transaction in relational database. Properties are atomicity, consistency, isolation, durability

HTML: Hypertext Markup Language

CSS: Cascading Style Sheets

JPARepository interface: Java Persistence API repository interface which is found in Spring Framework package

DTO: Data Transfer Object

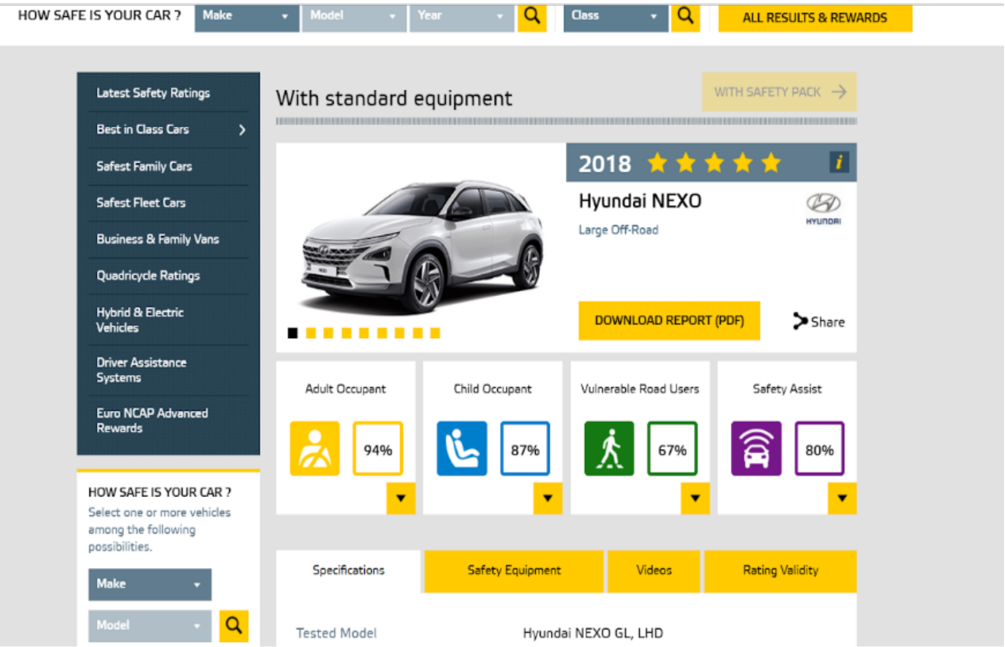
1.4. Overview

Our system collects reviews on electronic products from the trusted e-commerce websites. It analyzes the comments using Natural Language Processing (NLP) and based on the analysis it provides following functionality: Maximum 4 criteria for each product type is pre-selected. For each one of these criteria, TechRank assigns a percentage point by analyzing comments on specific criterion. Ranking of the device amongst all other devices on its category regarding selected criterion is determined. This ranking shows the selected products' place on the market based on only selected criterion. In addition, overall rating for the device is assessed by averaging percentage points for each of the criteria. And by averaging overall ratings of all products of the producer, overall rating of the producer is determined. Using all these rankings, users are able to understand general consensus on products and their specific qualities, and the producers. This way users are not get lost among hundreds or thousands of comments in the web, and are able to understand the rankings of the product they are looking for amongst all of its rivals on the market. In order to provide better experience for users many information on selected device is made available on TechRank. Application shows pictures of the product from different e-commerce website, it provides technical specifications of the device, users are able to see all the comment on web for selected device.

2. Current Software Architecture

2.1. Euroncap (https://www.euroncap.com/en)

Euroncap is an assessment program in which the cars (mostly brand new) are assessed according to the criteria such as pedestrian safety, driver safety, etc. that have been decided previously by the program authorities. The working principle of the assessment consists of taking the cars under several physical tests in which they collect data for the evaluation of the previously defined criteria. After the collection of data, the assessment is concluded by giving percentage evaluations for each criterion of the car that has been assessed. Our working principle has been influenced by the assessment of cars depending on the criteria defined in the assessment process of the program. The data that they gather are based on the physical test results they made on cars. In our case, as we will be considering the gathered data as the user reviews on the particular electronics products. Then, as in parallel with the evaluation of Euroncap, the data will be assessed to specific criterion and the evaluation of these criteria will be made in terms of percentages. In the below screenshot, it can be seen that a car has been assessed according to the following criteria: adult occupant, child occupant, vulnerable road users, safety assist.



With standard equipment

2018 ★★★★★

Hyundai NEXO

Large Off-Road

Adult Occupant	Child Occupant	Vulnerable Road Users	Safety Assist
94%	87%	67%	80%

HOW SAFE IS YOUR CAR ?

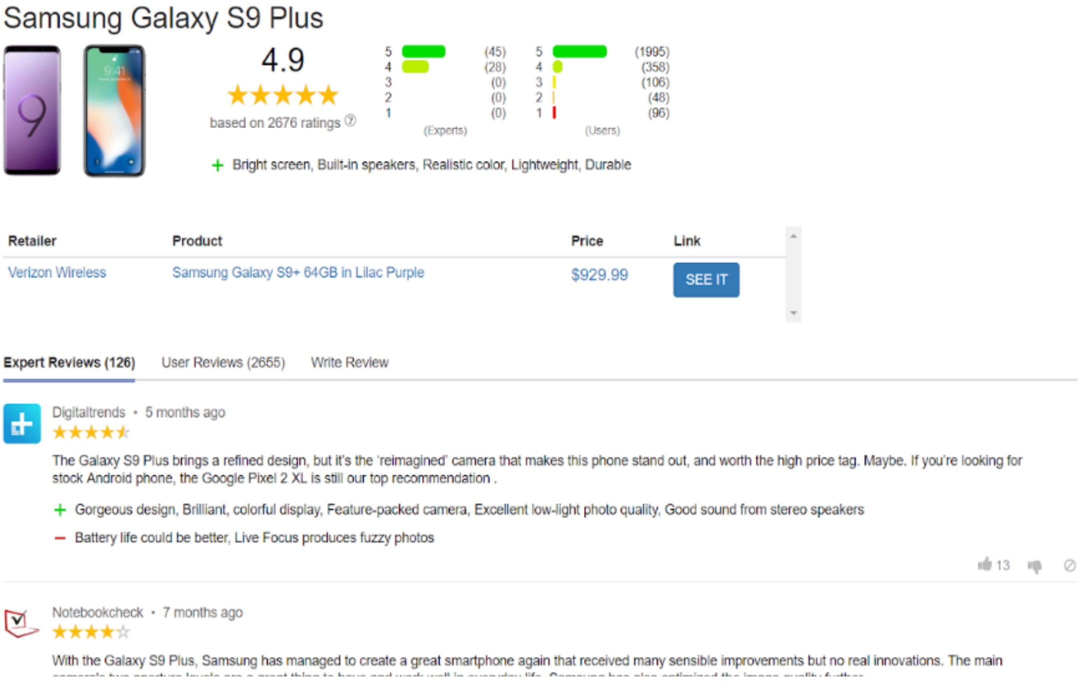
Select one or more vehicles among the following possibilities.

Tested Model Hyundai NEXO GL, LHD

***Figure 1:** Euroncap Website*

2.2. TestFreaks (https://www.testfreaks.com)

TestFreaks is a website which combines thousands of comments from the most-popular e-commerce websites with professional experts' reviews as well. The system that TestFreaks has consists of crawling data and presenting them to the user with the use of some intelligent algorithms so that the users are going to be able to access the best local recommendation results. The website also provides its user an opportunity to see different purchasing options and directing links to those options. Apart from showing the user reviews on a particular electronic device, TestFreaks also shows the test results to its users. Below, there is a screenshot in which a simple electronic device's main page has been shown that includes the reviews, and purchasing option. The pros and cons are directly crawled from the websites that the comments are gathered, thus there needed no processing of the reviews to figure out what the pros and cons are.



Samsung Galaxy S9 Plus

4.9
★★★★★
based on 2676 ratings

(Experts)		(Users)	
5	(45)	5	(1995)
4	(28)	4	(358)
3	(0)	3	(106)
2	(0)	2	(48)
1	(0)	1	(96)

+ Bright screen, Built-in speakers, Realistic color, Lightweight, Durable

Retailer	Product	Price	Link
Verizon Wireless	Samsung Galaxy S9+ 64GB in Lilac Purple	$929.99	SEE IT

Expert Reviews (126) User Reviews (2655) Write Review

Digitaltrends • 5 months ago
★★★★½

The Galaxy S9 Plus brings a refined design, but it's the 'reimagined' camera that makes this phone stand out, and worth the high price tag. Maybe. If you're looking for stock Android phone, the Google Pixel 2 XL is still our top recommendation .

+ Gorgeous design, Brilliant, colorful display, Feature-packed camera, Excellent low-light photo quality, Good sound from stereo speakers

− Battery life could be better, Live Focus produces fuzzy photos

13

Notebookcheck • 7 months ago
★★★★☆

With the Galaxy S9 Plus, Samsung has managed to create a great smartphone again that received many sensible improvements but no real innovations. The main

Figure 2: *TestFreaks Website*

3. Proposed Software Architecture

3.1. Overview

In this part, we will decompose our system in order to analyze components of the high level architecture and show the subsystem structure in great detail. Some UML diagrams are provided below for decomposition. First of all, a general view of subsystems will be presented in a diagram where all the layers are presented. In this diagram (see Section 3.2) all the layers and the connections between them are shown while the details of the packages in the subsystems are not mentioned in Section 3.2. The details of the subsystems with detailed class diagrams found inside the packages are shown in diagrams found in section 4. In the following parts of this section, hardware/software mapping, persistent data management, access control security, global software conditions and boundary conditions are handled in its separate subsections.

3.2. Subsystem Decomposition

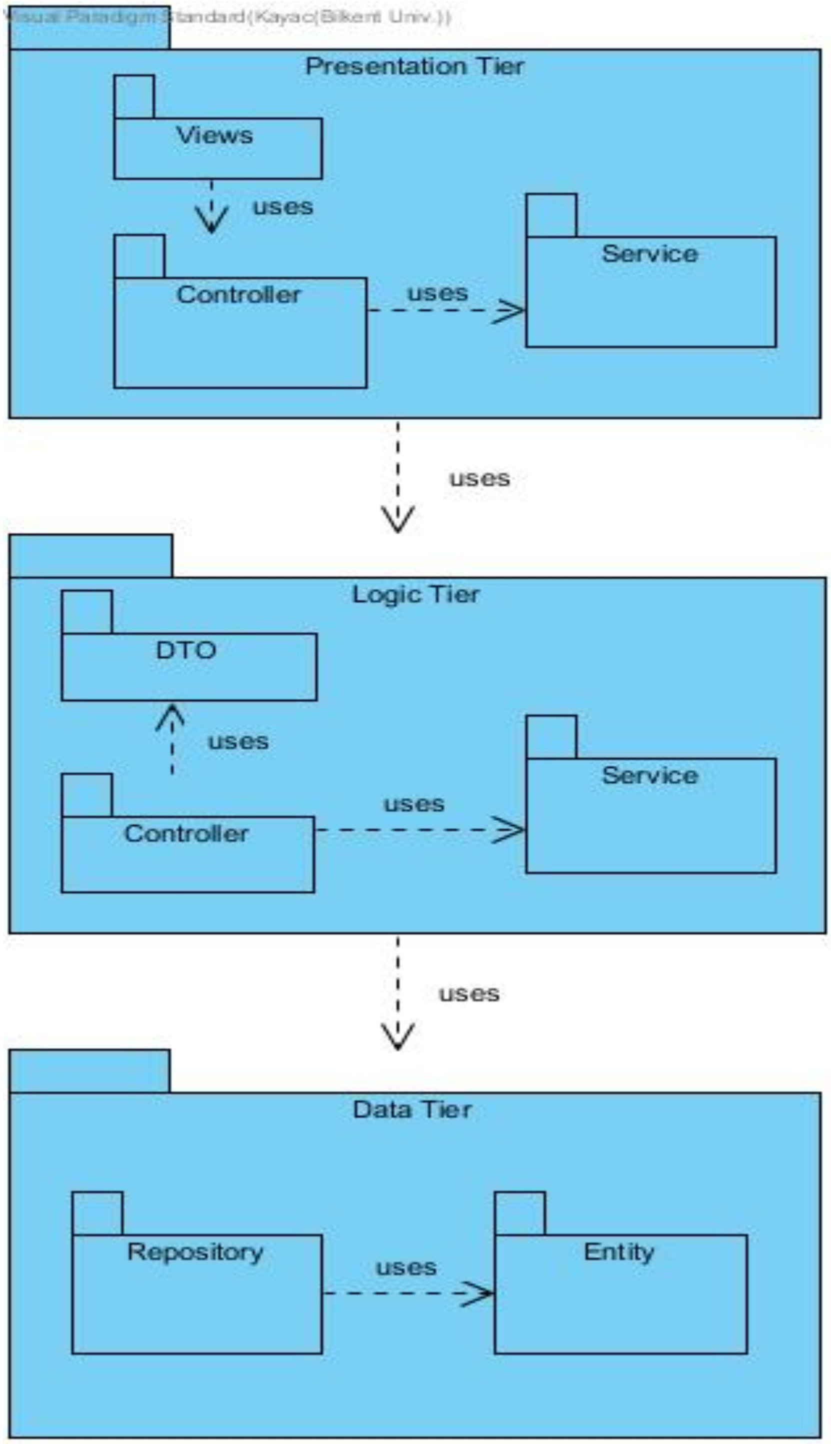


Figure 3: *General Scheme for Subsystem Decomposition*

TechRank application’s software architecture is 3-tier architecture. The working principle of the application is based on client-server system in which when a client side makes a request by means of interacting the interface, the server side responds according to the requests that has been made by the client. The layered structure of 3-tier architecture supports the workflow in the client-server system while encapsulating server side from the client side by limiting the direct interactions between the client (end user) and server(databases). The tiers are presentation, logic and data. The presentation layer is responsible for providing user interface to the end user (client) and listening any action that client is performing on the user interface. This tier is also responsible for making REST calls according to the user actions on the front-end and sending HTTP commands to the related endpoint of the server by means of using request-related controller and service modules that define the request endpoints. Logic tier is responsible for listening for any action on the server by the use of predetermined endpoints. Controllers and services found in this subsystem are responsible for handling the logic behind the REST calls (such as converting RequestDTO’s to Entity classes, navigating requests to related controllers and services in order to communicate with the database). Data tier is responsible for communicating with database and retrieving data required for logic tier to process and return back to the presentation tier. In this process, data tier uses repositories which communicate with databases using the request-related entities.

3.3. Hardware / Software Mapping

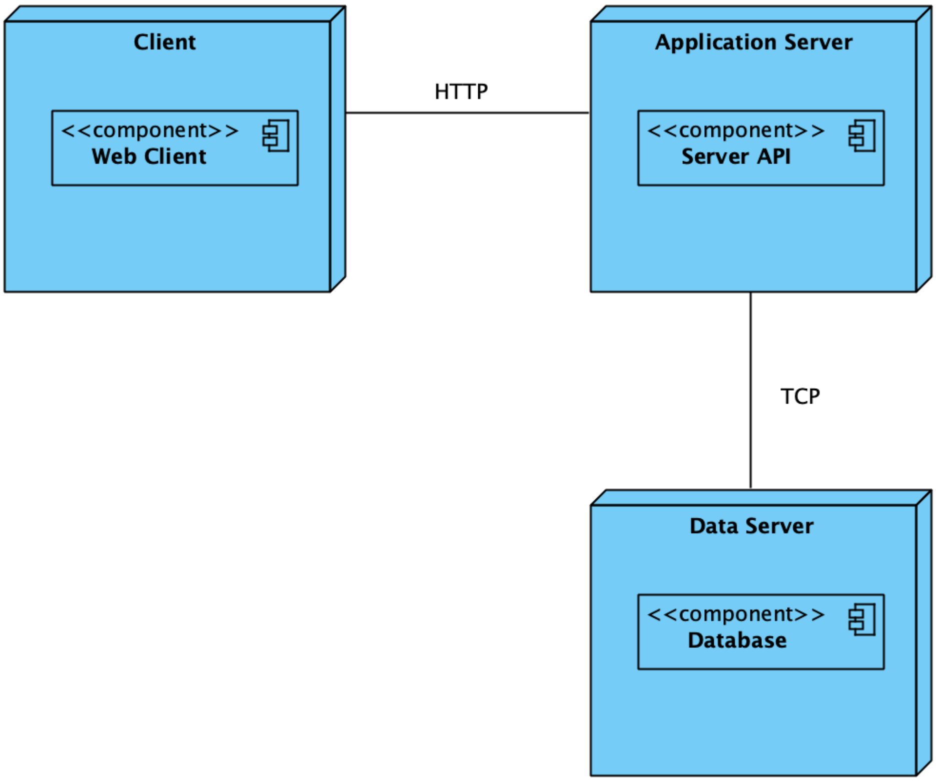


Figure 4: *Presentation Layer*

TechRank would only be available on a web browser; hence, a laptop or a mobile phone can be considered as hardware if user opens up a web browser and enter our URL.

Client will handle the interactions of user with the application; for instance, when user wanted to see a product, relevant HTTP request will be transmitted to application server with appropriate payloads. Moreover, when a result is obtained, frontend will render respective information on web client and user will be able to see the information requested.

Application Server will handle the incoming HTTP requests from client and retrieve the desired data from database. Server will also handle which data will come when a page transition is required; although page transition is handled on front-end side, content within will be retrieved from the server, i.e. server will act like a middleware between front-end end database.

Data Server will hold the relevant data for the application such as product information or comments about a product. In time of a data request, database will be queried by Server and queried data will be returned back to the Server. Moreover, database will also hold the raw data about comments, i.e. crawled and indexed data which is obtained by crawlers/scrapers.

3.4. Persistent Data Management

For our application, we need to store different kind of data in our databases and this data have relations in between each other. This data will be interpreted as different entities in application level since there will be several relations between different entities; such as “review” object being an attribute of “product” object. For this reason and in order to comply with ACID properties, using a relational database as a main storage and structuring the data would be more optimal to handle data transactions.

Since this project heavily relies on data retrieval form different e-commerce websites; initial starting point of this project would be data retrieval by implementing crawlers/scrapers. However, getting so much data from different websites with different data formats is hard to index in a relational database straight-forward. Hence, data retrieved from the websites via crawlers/scrapers will be indexed to a non-relational database initially; after NLP analysis is made on this data, it will be indexed to main storage.

3.5. Access Control and Security

System of TechRank does not require any authentication to use the system. There is no any login system and all end users can access to the system without any authentication. There is one type of user and an administrative user. End users cannot access to database directly and cannot change, add or delete anything from the database. They can just able to search some words from the database. However, administrative user has an authority to change to database of the system. For the security issue, system of TechRank does not take any personal information of the user. Therefore, there is no security problem for the users. Additionally, if there will be any problem about the system, there is an online help system for all users. They can contact with the developers of the system via email.

3.6. Global Software Control

TechRank is client-server application meaning information is requested from client side and provided by server side to the client side of the application. TechRank's software architecture is 3-tier architecture which supports the interactions in client and server systems by means of layering the architecture in terms of presentation, logic and data tiers. With three-tier architecture, data tier holds the information on entities and connects to database. The information requested by the user from the presentation layer are realized by the logic layer and finally entity information on the information requested are provided by entity layer by connecting database. As there are no user-specific action or information, updates on the database are not triggered by users, users only request previously placed data on products, companies and reviews, and the application provides the information to the user. Nonexistence of the user authentication makes system more robust, as the security is not big issue.

3.7. Boundary Conditions

3.7.1. Initialization

The only required initialization for the TechRank is having a proper internet connection since it is relied on updated online services.

3.7.2. Termination

The termination situation of the TechRank is the closing current window of the browser. If this window is not closed it will continue to run at background as long as the web browser stays open.

3.7.3. Failure

Since TechRank is a web application, a sustainable internet connection is mandatory to be able to use application. If there is no internet connection none of the page will be updated and system will not be used by users. This situation will cause a possible failure for the system.

4. Subsystem Services

As we used three-tier architecture to build our application, detailed diagrams and explanations of each tier are given below.

4.1. Presentation Tier

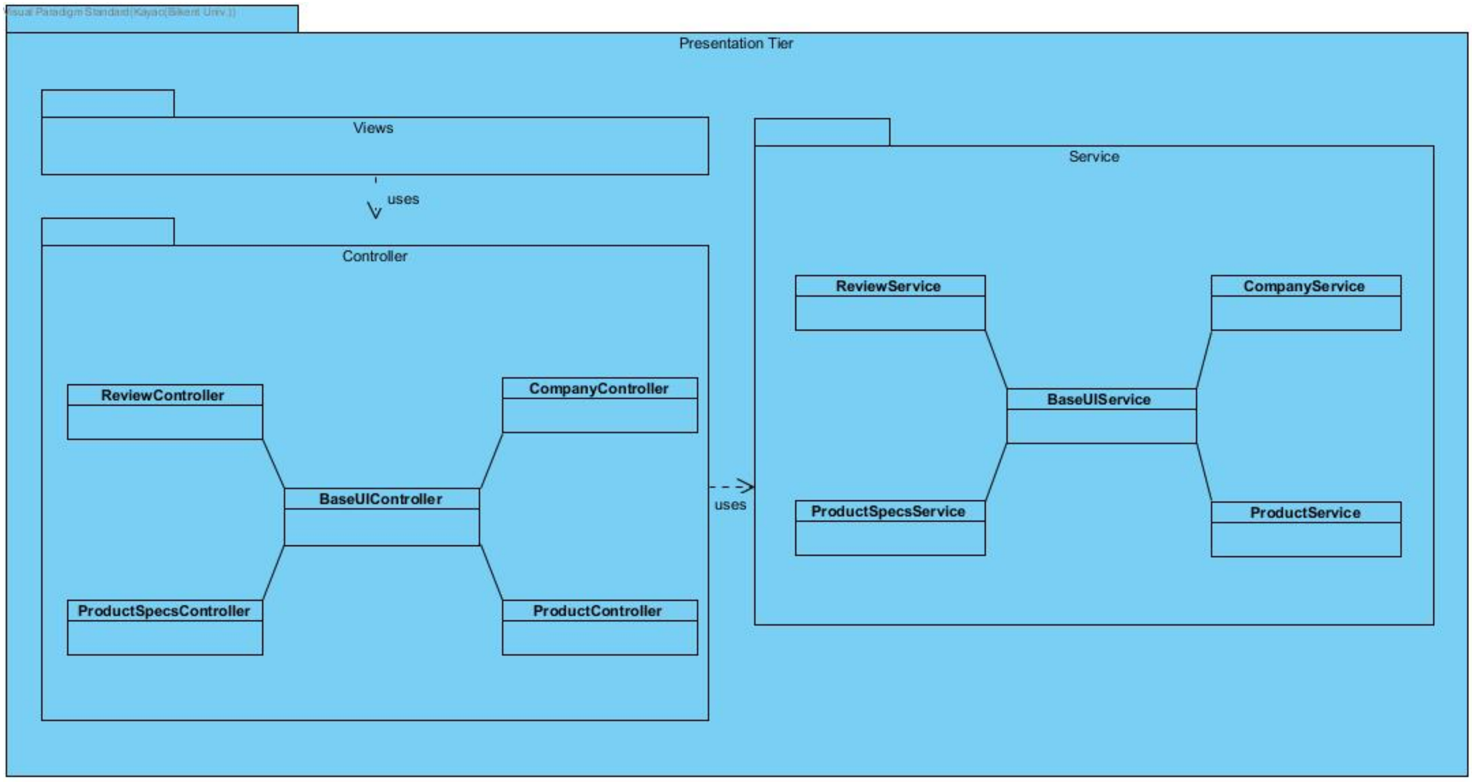


Figure 5: *Presentation Layer*

Presentation tier is responsible for managing interactions between the end-user and the user interface of the application

This tier consists of 3 subsystems: **Views**, **Controller** and **Service** subsystems.

4.1.1. Views

This package inside presentation tier includes HTML and CSS files that will be presented to the client.

4.1.2. Controller

ReviewController: This module is responsible for listening and managing user interactions with the user interface that are related with comments (user reviews).

CompanyController: This module is responsible for listening and managing user interactions with the user interface that are related with companies.

ProductSpecsController: This module is responsible for listening and managing user interactions with the user interface that are related with product's specifications.

ProductController: This module is responsible for listening and managing user interactions with the user interface that are related with product.

4.1.3. Service

ReviewService: This module's functions are used by **ReviewController** in order to make REST calls to the server. It handles the review related endpoints and their communication with the server.

CompanyService: This module's functions are used by **CompanyController** in order to make REST calls to the server. It handles the company related endpoints and their communication with the server.

ProductSpecsService: This module's functions are used by **ProductSpecsController** in order to make REST calls to the servers. It handles the product-specifications related endpoints and their communication with the server.

ProductService: This module's functions are used by **ProductController** in order to make REST calls to the server. It handles the product related endpoints and their communication with the server.

Note: Search bar related endpoints and their RESTful services are handled inside all the controllers and services according to the type of the query passed by the client to the search-bar.

4.2. Logic Tier

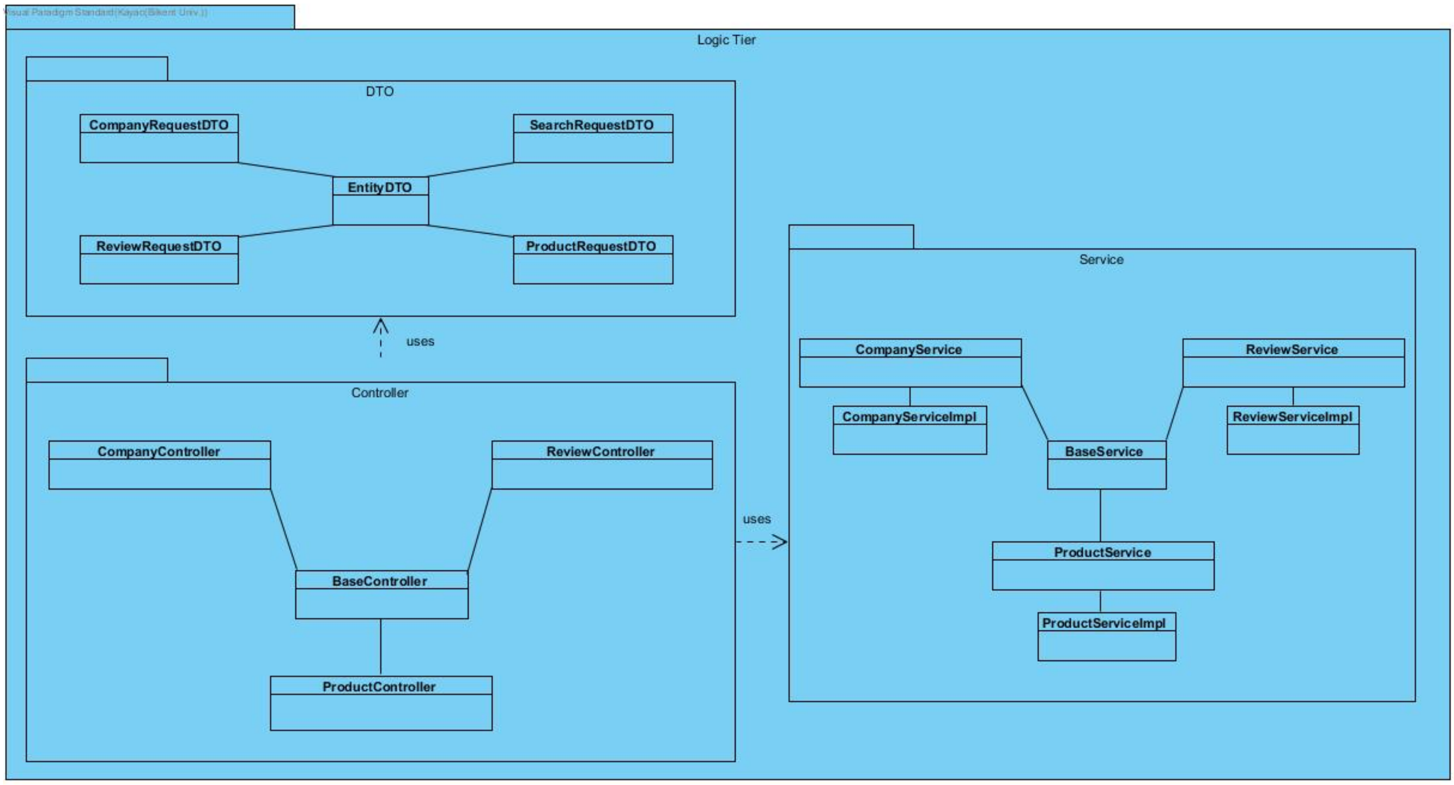


Figure 6: *Logic Tier*

This tier consists of 3 subsystems: DTO, Controller, Service.

4.2.1. DTO

EntityDTO: Base parent class that is responsible for holding information common to all the DTOs. DTO classes are used in the process of communication with the front-end. The server does not respond to the requests directly with the entity classes but with DTO classes to either encapsulate the entity classes or limit the response messages from the server regarding security concerns. DTO classes reshape the entity classes and are then used by controllers while responding the request.

CompanyRequestDTO: This class is responsible for deciding what to return if a company-related request is detected.

ReviewRequestDTO: This class is responsible for deciding what to return if a review-related request is detected.

SearchRequestDTO: This class is responsible for deciding what to return if search request is detected. It in most of the cases encapsulates list of objects to be used and returned back to client.

ProductRequestDTO: This class is responsible for deciding what to return if a product-related request is detected.

4.2.2. Controller

CompanyController: This class responsible for handling the REST request endpoints related to companies originating from the client. It uses **CompanyService** to get data (in Company type) and then convert it to **CompanyDTO** to respond to the request queried by the client.

ReviewController: This class responsible for handling the REST request endpoints related to user reviews (comments) originating from the client. It uses **ReviewService** to get data (in Review type) and then convert it to **ReviewDTO** to respond to the request queried by the client.

ProductController: This class responsible for handling the REST request endpoints related to products originating from the client. It uses **ProductService** to get data (in Product type) and then convert it to **ProductDTO** to respond to the request queried by the client.

4.2.3. Service

CompanyService: This interface is used by **CompanyController** which uses this class' methods for retrieving related information from the database.

CompanyServiceImpl: This class provides the implementation for **CompanyService** interface. This class uses **CompanyRepository** class to ask for the data in the database.

ReviewService: This interface is used by **ReviewController** which uses this class' methods for retrieving related information from the database.

ReviewServiceImpl: This class provides the implementation for **ReviewService** interface. This class uses **ReviewRepository** class to ask for the data in the database.

ProductService: This interface is used by **ProductController** which uses this class' methods for retrieving related information from the database.

ProductServiceImpl: This class provides the implementation for **ProductService** interface. This class uses **ProductRepository** class to ask for the data in the database.

4.3. Data Tier

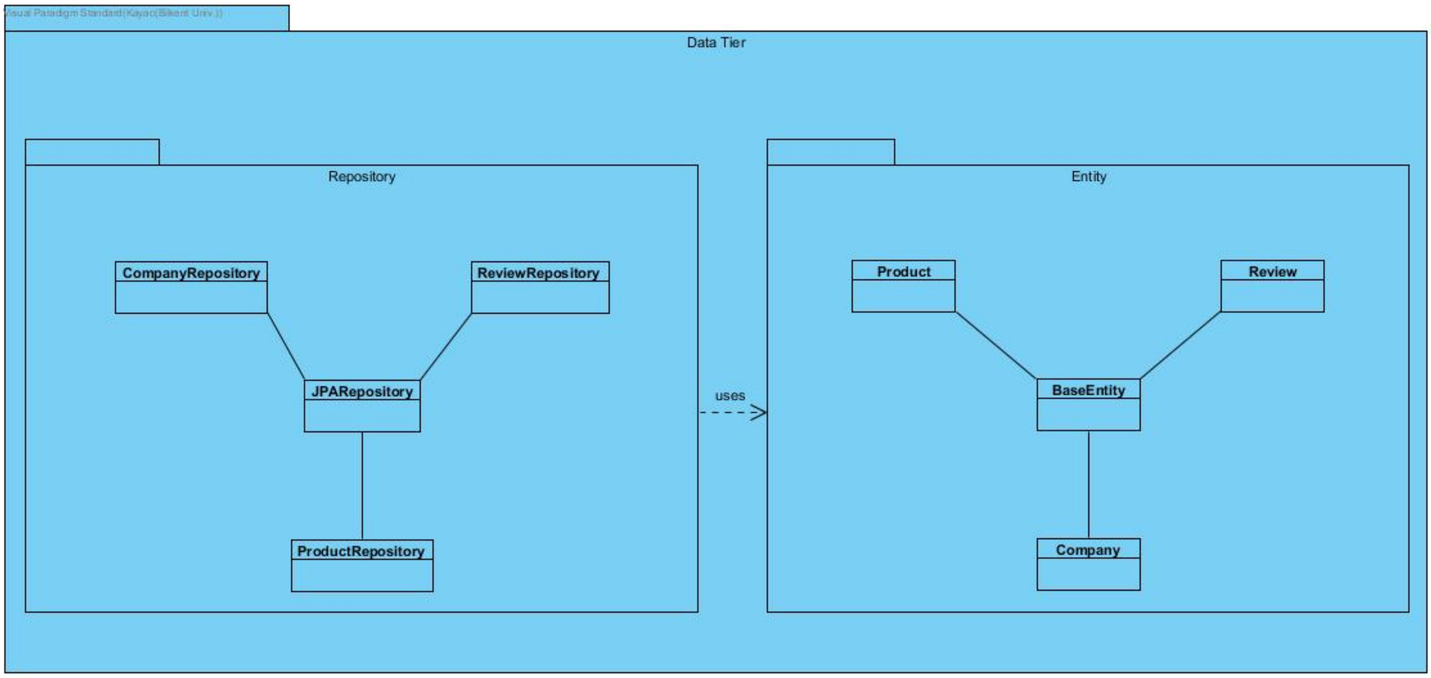


Figure 7: *Data Tier*

This tier consists of 2 subsystems: Repository, Entity.

4.3.1. Repository

CompanyRepository: This class is used by **CompanyService**, it returns back to it after retrieving data related to **Company** entity. This class extends **JPARepository** interface (found in Spring Framework package) which provides wide range of interactive methods to retrieve data from the database.

ReviewRepository: This class is used by **ReviewService**, it returns back to it after retrieving data related to **Review** entity. This class extends **JPARepository** interface (found in Spring Framework package) which provides wide range of interactive methods to retrieve data from the database.

ProductRepository: This class is used by **ProductService**, it returns back to it after retrieving data related to **Product** entity. This class extends **JPARepository** interface (found in Spring Framework package) which provides wide range of interactive methods to retrieve data from the database.

4.3.2. Entity

BaseEntity: Base parent class of entity objects which holds common information that all the entities share. Custom entity classes below which extends this class will be used by related repositories while retrieving information from database.

Product: This class represents the Product entity which is basically a representative of an object retrieved from Product table.

Review: This class represents the Review entity which is basically a representative of an object retrieved from Retrieve table.

Company: This class represents the Company entity which is basically a representative of an object retrieved from Company table.

Note: Repository interfaces, by means of using JPARepository interface, create a vast amount of methods related to the entity object that it is bound to so that complex queries can be handled.

Bibliography

[1]"3-Tier Architecture: A Complete Overview - JReport", *JReport*, 2018. [Online]. Available: https://www.jinfonet.com/resources/bi-defined/3-tier-architecture-complete-overview/. [Accessed: 31- Dec- 2018].

[2]"What is Client Server Architecture? Webopedia Definition", *Webopedia.com*, 2018. [Online]. Available: https://www.webopedia.com/TERM/C/client_server_architecture.html. [Accessed: 31- Dec- 2018].

[3]"What is RESTful API? - Definition from WhatIs.com", *SearchMicroservices*, 2018. [Online]. Available: https://searchmicroservices.techtarget.com/definition/RESTful-API. [Accessed: 31- Dec- 2018].

[4]"IBM Knowledge Center", *Ibm.com*, 2018. [Online]. Available: https://www.ibm.com/support/knowledgecenter/en/SSGMCP_5.4.0/product-overview/acid.html. [Accessed: 31- Dec- 2018].

[5]2018. [Online]. Available: https://www.euroncap.com/en. [Accessed: 31- Dec- 2018].

[6]"Reviews and tests from the best sources, Testfreaks collects it all.", *TestFreaks*, 2018. [Online]. Available: https://www.testfreaks.com. [Accessed: 31- Dec- 2018].