## **PROJECT RESEARCH**

**Overview**

Application Area - MonaLisa Beauty is an online booking and salon management software dedicated to hairdressing and beauty industry. [1]

[ASP.Net](http://asp.net/) - MVC 6 application with data base.

Using Entity Framework and code first approach.

Application will be deployed in cloud using AWS (Amazon Web Services), service in use Elastic Beanstalk.

Part of the project are Restful web APIs.

Unit Test is responsible for validation of the code and functionality quality.

**Aims** - Software would allow managers and business owners implement accurate and cost effective scheduling, resource and stock management.

Application would be beneficial in terms of choice appropriate marketing strategy.

Would allow business owners and their customers immediate access at any time from any

destination.

MonaLisa Beauty Platform as a Service product, this cloud computing service allows

to develop, run, and manage application excluding the complexity of building and

maintaining all infrastructure.

**Business Benefits**

- Accurate resources management

- Effective cost management,

- Digital Marketing in place due to accurate APIs analysis

- Highly qualified staff on call

- Data base of hospitals and clinics (clients)

- Automatic Reminders - keeping in touch with customer, help to build positive relationship,

financial, marketing - Appointment history - marketing, financial, inventory.

PAAS - low esb bill, no need for hardware ad software investment, software maintenance,

increased security, no need for code maintenance.

**Design**

**Functionality**

1. Selection of beauticians - external user would be able to book appointment with their favourite beautician. This feature will help build good relationship between business and customer. Customer will feel comfortable with favourite staff, feature will help build trust towards business.
2. Beautician profile specialization and years of experience (Picture plus portfolio)- this functionality will allow customer to select specialist in area that they are looking for particular service. Staff will get a chance to introduce themselves to the customers, tell about their experience and personality.
3. Selection of beauty services (list of services including price) - very important feature for health and beauty salon, wide range of services builds portfolio of the business and differentiate business against their competitors. Service and price offer presented in the form of list will allow customer clear and easy selection of the beauty services.
4. Date picker - this feature allows to select available date for particular beautician and service. Typical MVC date picker.

Date picker - functionality logic

* + - Past dates would be disabled - to prevent booking in the past.
    - Booked dates disabled - to prevent double booking.
    - Out of business dates disabled - to prevent booking when business is closed.

1. Time Picker - this feature is displayed after customer selects available date.
   * + Past time disabled - to prevent booking for time in the past.
     + Booked time disabled - to prevent double booking.
     + Out of business hours time disabled -to prevent booking time when business is closed.

**Functionality for internal users. -APIs analysis**

1. Resources management – staff, business owners and management will be able to predict what kind of stock should be ordered. There will be possibility to forecast what resources will be required and what time of the year. This internal analysis would be excellent feature for Just In Time SCM approach, cost effective and very accurate.
2. Marketing - sales by month - this feature will indicate what’s the seasonal peak, for the business. Sales analysis will help managers and business owners to provide effective marketing strategies, accurate and cost-effective stock management.
3. Marketing – Beautician by sales - this analysis will help management to indicate who is the best employee. How employees KPI compared against their competitors.
4. Marketing – mailing list and phone numbers – personal data of the customers is crucial for reaching out the customers. Kind reminders about the appointments (1 day before appointment), special offers and competitions, all of those interactions with the customers will be very important to develop good relationship with existing customers, keep the up to date with new services and opportunities to get beauty offers in competitive price. Mailing list and phone numbers list will create space for reaching out to the new customers through ‘refer service to a friend, you will get your service for free or in special price’.
5. List of the staff - payroll and resources management. Management will be able to list their employees, by salary, sales, KPIs. Compare their performance and prepare appropriate resources management and apply motivation techniques.
6. List of the clients - business owners and salons managers, knowing customers behavior will be analyze how much money would they spend and when. Marketing can target particular customers at the particular time of the year, with special offer dedicated especially for them.

**Type of users**

1. External users - customers that would like to make an appointment, with particular beautician, select particular service, check pricing and date/time availability.
2. Internal users - staff or management would be able to allocate staff as per booking needs, prediction of seasonal peaks, accurate resources management, marketing purposes - customers contact details.

**Comparison with similar systems**

**Benchmarking**

There are several online booking applications, few of them are strictly dedicated to hairdressing and beauty industry. [2]

**1. Versum - www.versum.com**- spa and salon management software . Application was introduced in 2010 and is widely used in hairdressing and beauty salons worldwide. At the moments is introduced to the fashion industry as well. Software allows to implement smooth booking management at any time, offers inventory accurate inventory management and effective marketing [2]

**Deployment**

Cloud Hosted - AWS Amazon Web Services

**Devices Supported**

Windows

Mac

Web-based

**The best Features**

Appointment Management

Email Marketing

Booking calendar

Built In CRM

Automatic Reminders

Social Media Marketing

POS

Referral Programs

Inventory Management

Promotional Campaigns

Payroll management

Appointment history

Gift & Rewards System

**Language Support**

English

**Pricing Model**

Monthly payment

**Customer Types**

Small Business

Large Enterprises

Medium Business

**Deployment**

Cloud Hosted

Booth Rental Management - no Client Profiles - yes Employee Management - yes Inventory Management - yes Multi-Location - yes Credit Card System (POS) - no [3]

**2. Salon Iris -** [www.saloniris.com/](http://www.saloniris.com/)

Deployment Installed, Mobile, Cloud

Booth Rental Management - yes

Client Profiles - no

Employee Management - yes

Inventory Management - yes

Multi-Location - yes Credit Card System (POS) - yes **3. Shedul - www.**[shedul.com](http://shedul.com) **—** Deployment Cloud

Booth Rental Management - yes

Client Profiles - yes

Employee Management - yes

Inventory Management - yes

Multi-Location -yes Credit Card System (POS)- yes [4]

**Comparison of features and technical details for 10 booking dedicated softwares:**

1. Deployment a. cloud only 2 b. mobile and cloud 5 c. cloud, mobile, installed 2 d. installed and cloud 1 Booth Rental Management -5

Client Profiles - 5

Employee Management - 8

Inventory Management - seven

Multi-Location 8 Credit Card System (POS) — seven [5]

**Software Architecture**

There are 5 top software architecture methodologies that could be used when developing booking software.

1. **Event-driven architecture**

This architecture builds central unit that is responsible for accepting all of the type of data. unit handle the date and delegates to appropriate modules. Handling data is called ‘event’, then is delegated to code assigned to that type.

This architecture is very different from layered one, event driven application make sure that only right code sees required events.[6]

**Advantages:**

* Dedicated to complex and chaotic environments
* Scale easily
* Easy to extend

**Disadvantages**:

* Individual modules can be tested separately. For this reason testing might be complexed.
* Structure complexity might be problematic for error handling.
* Backup plan is needed in case of module failure.
* Messaging might slow down process.
* Systemwide data development structure might be difficult when events have different needs.
* Modules decoupled and independent - might cause difficulties for consistency of transaction based mechanism.

[6][7]

**Recommended for:**

* Asynchronous system
* Separated data blocks interact with only a few modules
* User interfaces

**2.Microkernel architecture**

Core set of operations that are used in different patterns that depend on the data and the task. For example development tool Eclipse, will open files, annotate them, edit them, and start up background processors. The tool is very popular as is responsible for doing all of the jobs with Java code.When a button is pushed, compiling the code and running it.[6][7]

In this example the basic routines for displaying a file and editing it are part of the microkernel. The Java compiler is an extra part that’s support the basic features in the microkernel.

Plug ins or a plug-in architecture are extra features, layered on top.[6][7]

“Claims processing is necessarily complex, but the actual steps are not. What makes it complex are all of the rules.”[6][7]

**Disadvantages:**

* Microkernel ownership is complexed.
* The plug-ins must include a handshaking code , this is to provide microkernel awareness that the plug-in is installed and ready to work.
* Modification of the microkernel is very complexed almost impossible in late stages of development.

[6][7]

**Advantages:**

* Tools might be used by wide range of people.
* Applications has clear diversification between basic routines and higher order rules
* Applications has a fixed set of core routines, as well as a dynamic set of rules with requirement to frequent update.

[6][7]

**3. Microservices architecture**

The microservice architecture is dedicated to help develop software that is not monolithic, heavy and inflexible. The concept is that for each feature you develop one little program instead of developing one big program.

“If you go onto your iPad and look at Netflix’s UI, every single thing on that interface comes from a separate service,”

[6][7]

Approach similar to event-driven and microkernel approaches, however used when tasks can be easily separated.

Example of this architecture is Netflix , as an representation of the compilation of little websites. It allows for easy up and down service development as per demand.

[6][7]

**Disadvantages**:

* Independent services on large scale otherwise it might cause cloud imbalance.
* Sometimes applications don't have tasks that can be split into the single units.
* Case when tasks are spread between different microservices might affect the performance
* Too many microservices can confuse users as parts of the web page appear much later than others.

[6][7]

**Advantages**:

* Small components of webservices
* Boundaries of data centres are well defined.
* Recommend for rapidly developed web application
* Recommended for development teams spread out around the world.

[6][7]

**4. Space-based architecture**

The space-based architecture is designed to handle large load for data bases. It’s achieved splitting processing and storage between servers.

Another words it’s cloud based architecture.

“It’s all in-memory objects,” [6]. “The space-based architecture supports things that have unpredictable spikes by eliminating the database.”[6]

**Disadvantages**:

* Transactional support facing problems with RAM databases.
* Challenging generation of enough load for testing the system , however individual nodes can be tested independently.
* Catching data for speed without corrupting multiple copies is difficult.

[6][7]

**Advantages**:

* High-load data like click streams and user logs
* Not bank transactions
* Social networks

[6][7]

**5 . Proposed system architecture MVC6 - Layered Architecure (n-tier)** it’s the most common used software architecture it’s built around data base, as many businesses storing data in the tables.

MVC - Model - Controller - View , model is built on the way that data enters the top layer - View - and works their way until it will reach bottom layer and data base.

Three interconnected parts are responsible for different task, MVC pattern allows for code reuse and parallel development.

**Model** is the main component of the MVC framework. It defines application's dynamic data structure and directly manages the data, overall logic and rules of the application.

**View** - visual representation of the application.

**Controller** responsible for accepting and converting input, then translating into the commands and passing for the model or view.

[6][7]

MVC architecture become extremely popular for web application development, JavaScript, Python, Ruby, PHP, Java, and C# most common used programming languages.

Layered architecture separating concerns. Each layer can focus on their role.

**Advantages:**

Maintainable

Testable

Flexible to assign separate "roles"

Flexible for separated layers update

**Recommended for:**

New applications with short turnaround

Enterprise or business

Inexperienced developers that dont understand other architectures

Applications with maintainable and testable standards

[6][7] [8]

**Overview of proposed system architecture**

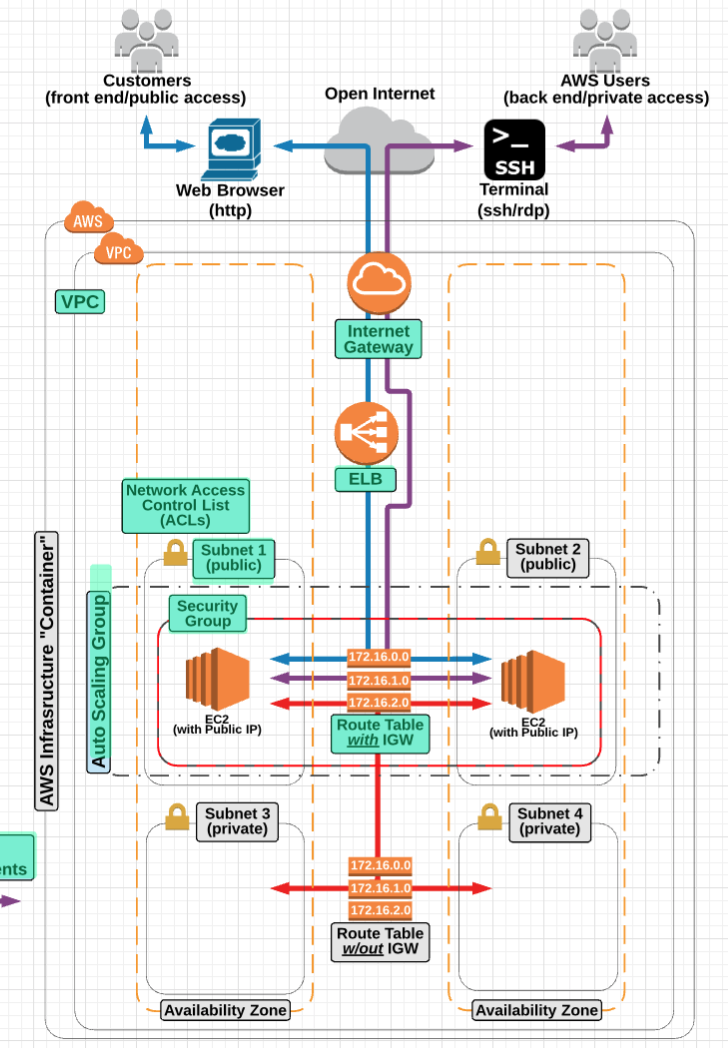
**Framework**

[ASP.Net](http://asp.net/)  - MVC 6 application with data base.

Using Entity Framework and code first approach.

**Cloud Architecture**

Application will be deployed in cloud using AWS (Amazon Web Services), service in use Elastic Beanstalk. Other AWS services in use: EC2 instance, Internet Gateway, Security Groups, Route Table, NACLs, RDS, ELB, Autoscaling, Route 53.



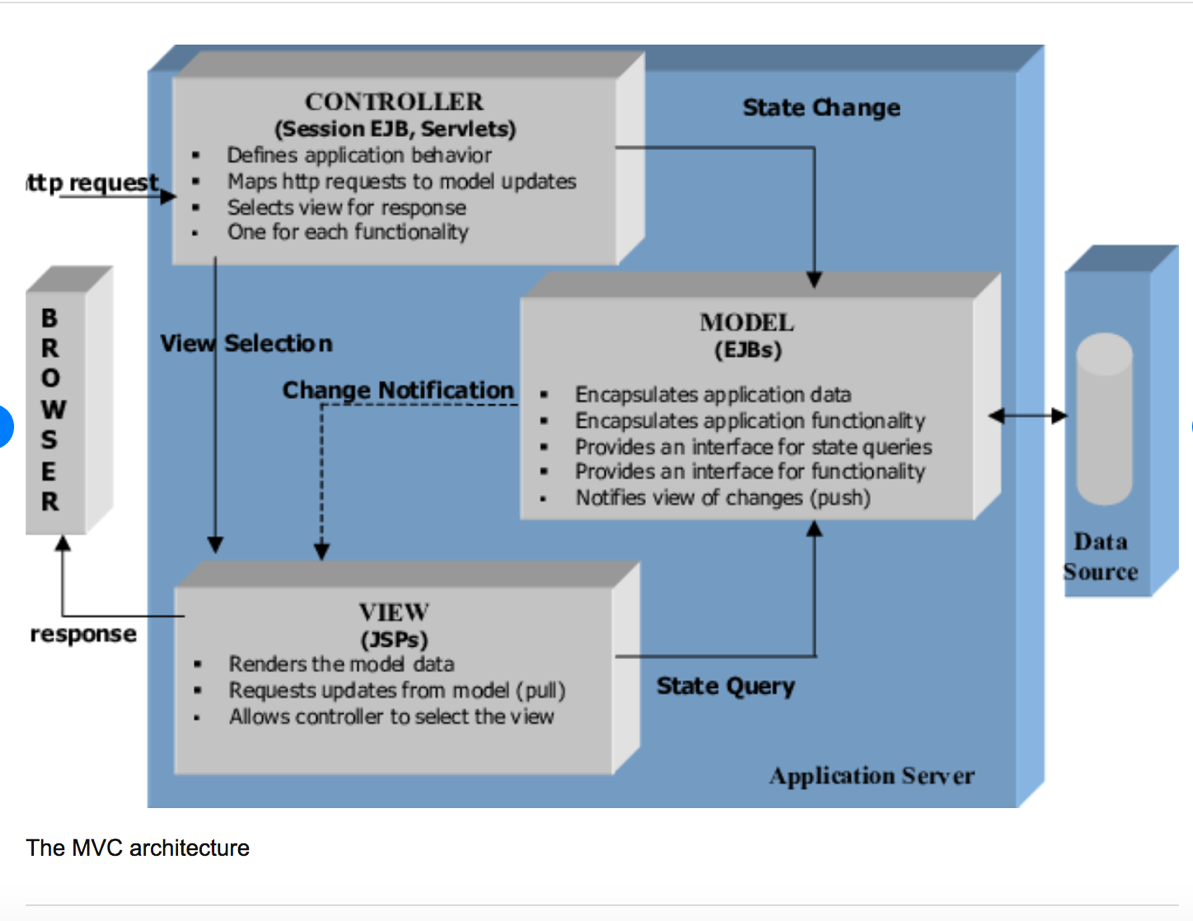
Sample of the Highly Available AWS Cloud Architecture and Fault Tolerant VPC [10]

**Data Base**

A database containing 3 Tables: Beautician, Services, Clients

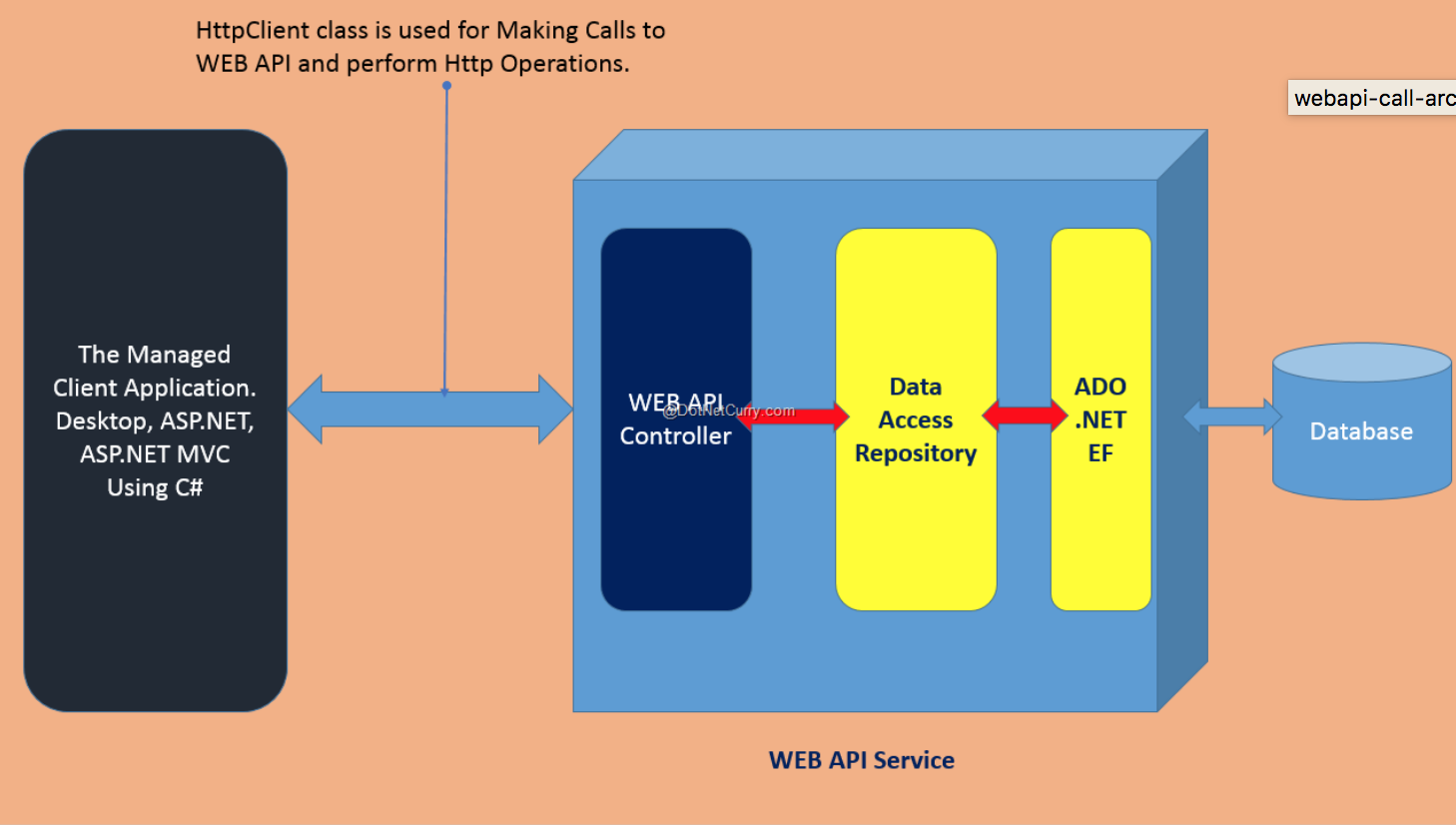
(RDB AWS)

**Proposed system architecture MVC6 - Layered Architecture**

****

The MVC architecture [10]

**WEB API Service Architecture [11]**



**Software or hardware requirements**

**Operation system**

Windows 7 64-bit

Windows 8.0

Windows 10

**Software:**

visual Studio

Google Chrome or Mozilla

**Hardware** - no special requirements required for Application using Amazon Web Services

**Bibliography**

[1] -Meet the #1 Software for Salons and Spas, Schedul, 2019, Accessed: 30. 05.2019 19:10

<http://www.shedul.com/?utm_source=capterra&utm_medium=paid&utm_campaign=salon>

[2] -Salona Software, Salon Software Advice, 2019, Accessed: 05.06.2019 19:30 <https://www.softwareadvice.com/ie/salon/>

[3] -Finances online, review for business, May 2019, Accessed: 05.06.2019 20:10 Accessed:<https://reviews.financesonline.com/p/versum/>

[4] - <http://www.shedul.com/?utm_source=capterra&utm_medium=paid&utm_campaign=salon>

[5] -Salon Software, accessed: 05.06.2019 20:15

<https://www.capterra.com/sem-compare/salon-software?gclid=CjwKCAjw8qjnBRA-EiwAaNvhwL5ooed26BpeR1Hba05_TzCrlgpV2Yrcz_buQnKFwrsctPhnqORvOBoC3pMQAvD_BwE>

[6] - Software Architecture Patterns, Mark Richards, February 2015, ISBN: 9781491971437

[7] The top 5 software architecture patterns: How to make the right choice, [Peter Wayner](https://techbeacon.com/contributors/peter-wayner), accessed: 09.06.2019 20:33

-<https://techbeacon.com/app-dev-testing/top-5-software-architecture-patterns-how-make-right-choice>

[8] Fitting Security Into Your Software Lifecycle: Automation and Integration, Michelle Duffy, Sonatype; Dennis Hurst, Saltworks; Jimmy Rabon and Brent Jenkins, Micro Focus, April 2019, accessed: 09.06.2019 21:30

https://www.microfocus.com/en-us/marketing/esp-webinars?commid=356244&utm\_source=techbeacon&utm\_medium=techbeacon&utm\_campaign=00134846

[9] Orion Papers, February 2018, accessed:17.06.2019 22:02

<https://interactive.linuxacademy.com/diagrams/TheOrionPapers.html>

[10] [Journal of Web Engineering (JWE)](https://www.researchgate.net/journal/1540-9589_Journal_of_Web_Engineering_JWE) 1(1):37-60 · January 2002 with 1,737 Reads

[11] Consuming a Web API Asynchronously in ASP.NET MVC or WPF, Mahesh Sabnis, 10/1/2015, accessed: 17.06.2019 23:02

<https://www.dotnetcurry.com/aspnet/1192/aspnet-web-api-async-calls-mvc-wpf> , accessed: