Software Architecture, Security Architecture and Deployment Architecture

for

Learning Assistance System

**Contents**

[1. OVERVIEW 2](#_Toc32953206)

[1.1 Map User Stories to Quality Attributes 2](#_Toc32953207)

[2 SOFTWARE ARCHITECTURE 4](#_Toc32953208)

[2.1 High Level logical components and their interactions 4](#_Toc32953209)

[2.2 Description of the Modules or System Components of the system 4](#_Toc32953210)

[2.3 Type of connections and integration patterns 6](#_Toc32953211)

[2.4 Architecture Styles 6](#_Toc32953212)

[3 SECURITY ARCHITECTURE 7](#_Toc32953213)

[3.1 Security architecture diagrams 7](#_Toc32953214)

[3.2 Components Description 8](#_Toc32953215)

[4 INFRASTRUCTURE/DEPLOYMENT ARCHITECTURE 8](#_Toc32953216)

[4.1 Deployment Architecture Diagrams: 9](#_Toc32953217)

[4.2 Description of Components 9](#_Toc32953218)

[5 Glossary 12](#_Toc32953219)

[6 References 12](#_Toc32953220)

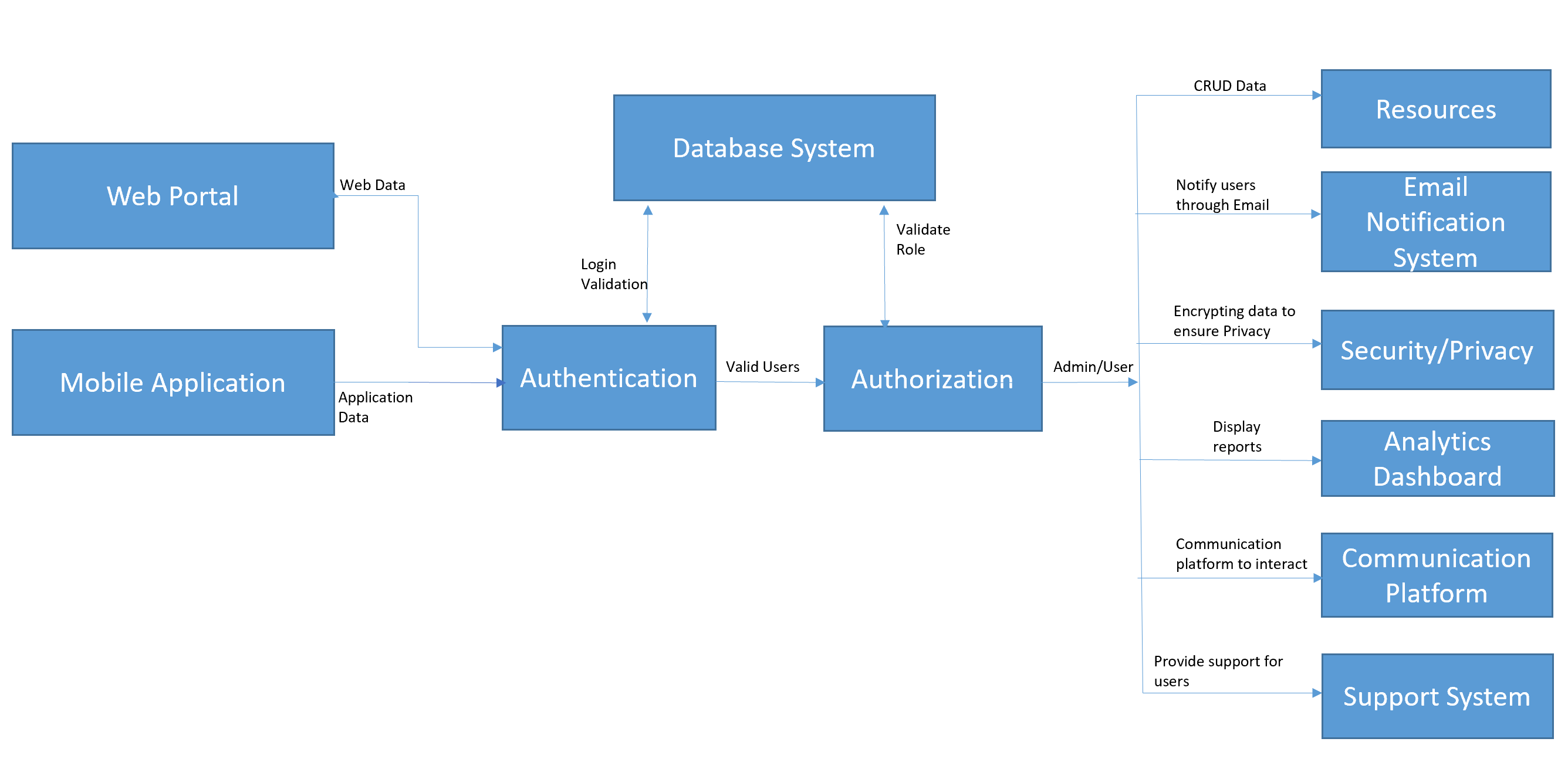
# OVERVIEW

## Map User Stories to Quality Attributes

|  |  |  |
| --- | --- | --- |
| **User Stories** | **Quality attributes mapping** | **Justification for the mapping** |
| Creation of a user-friendly home page for learners | Usability, Supportability,  Flexibility, Portability | Usability- It should be possible to search al the home page contents  Supportability- The system maintains persistent logs of every functional operation in the home page for a registered user  Flexibility - The ability for the solution to adapt to possible or future changes in its requirements  Portability - The system works for all popular browsers (Google Chrome, Internet Explorer, Safari) and mobile operating systems (Android, iOS)  Portability - Users should be able to access the system through a mobile device or desktop |
| Creation of an account with a username/ mobile number/Facebook/Gmail account and password for user | Security, Usability, Portability | Security – The system must be able to validate the user’s role and identity.  Security - All the personal data of the registered user must be encrypted.  Usability – User login has maximum of 3 attempts  Portability - The system works for all popular browsers (Google Chrome, Internet Explorer, Safari) and mobile operating systems (Android, iOS)  Portability - Users should be able to access the system through a mobile device or desktop |
| Creation of an admin account and gaining admin access to do CRUD operation on the system contents. | Modifiability,  Security | Modifiability – The admin has access to add/delete/modify/vary functionality  Security- The admin must have access related to all the users |
| Ability of the user to login successfully after entering valid login credentials and use the learning assistance system of his interest to set and track learning | Security | Security- The user must have access to all the contents |
| Granting of resources (by admin) for the users | Usability, Performance,  Scalability | Usability – The admin makes the system efficient and minimize impact of errors  Performance - The system should be able to support a maximum of 2000 users in its first year  Scalability – The system should grow by 50% per year. |
| Ability to use my rewards, as a registered user to open other courses for my learning and share my accomplishments with other users. | Supportability,  Portability | Supportability - The system should have a built-in mechanism to send emails  Supportability - The system maintains persistent logs of every functional operation in the home page for a registered user  Portability - The system works for all popular browsers (Google Chrome, Internet Explorer, Safari) and mobile operating systems (Android, iOS)  Portability - Users should be able to access the system through a mobile device or desktop |
| Ability of the admin to provide a platform to the users which ensures privacy of users' data. | Security | Security - All the personal data of the registered user must be encrypted |
| Registered users should be able to communicate with other users through a chat box or email. | Supportability | Supportability - The system should have a built-in mechanism to communicate with others through chat box |

# SOFTWARE ARCHITECTURE

## High Level logical components and their interactions



## Description of the Modules or System Components of the system

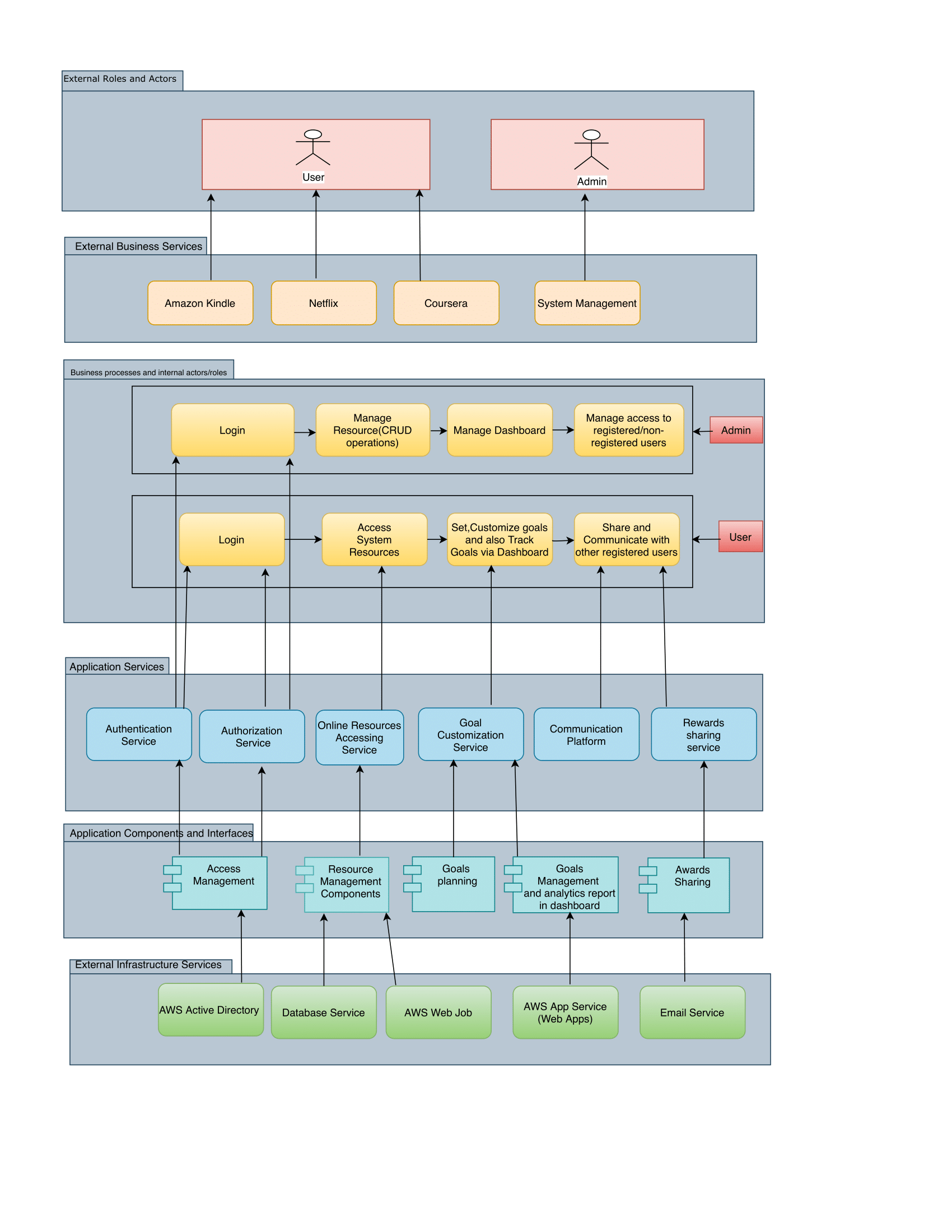
|  |  |  |
| --- | --- | --- |
| **Component** | **Description** | **User Stories** |
| **Web Portal** | This is a web accessible UI for the users to learn depending on their areas of interest and track a record of their learning. Only registered users can track their learning. Non-registered users can also make use of the portal but their learnings will not be tracked. | US1 |
| **Mobile Application** | Mobile App component delivers mobile functionality for all the users. It mimics the web portal with the same UI behavior and all other functionalities. This feature  helps the users to use the learning assistance system more handy in their mobile phones. | US1 |
| **Authentication** | Authentication is needed for the login validation. User can create an account giving username, or mobile number, or by sign in through Facebook or through Gmail. This is required to validate the login credentials. | US2,  US3 |
| **Authorization** | Authorization is done to validate the role. System has an admin who can do CRUD operations on system contents and has the right to give access to the users depending upon the users who are registered or not. Admin is the one who give access to the users to use the resources. Only registered users can track their learning. | US3,  US4 |
| **Database System** | Database systems are required to store the users data for login and also to validate the role(admin/registered/non-registered user)It is also required to store the resources. | US2,  US3,  US5 |
| **Security/Privacy system** | Admin should take care of protecting users data to ensure privacy | US2,  US3, |
| **Analytics Dashboard** | Dashboard is available to all registered users to track their learning. Registered users can track the courses they have completed as per their plan, on-going learnings, time spent per session, and recommended courses in their dashboard | US4 |
| **Communication Platform** | Users can use communication platform to discuss with other registered users. Users can also share their learning with others users and can also share their achievements in the communication platform | US7 |
| **Resources** | Resources include online resources such as eBooks, articles, documentaries, google links, blogs, amazon’s kindle, Netflix etc., live streaming and other pre-recorded courses. Admin is the one who give access to the users to use the resources. | US5 |
| **Email Notification System** | An email Notification system to notify the users regarding their progress or other activities, new courses added or time left to finish the courses etc. | US5 |
| **Support system** | Support system to help the users who face difficulties while using the system. | US4 |

## Type of connections and integration patterns

Below provide details of the connectors between certain components.

|  |  |
| --- | --- |
| **Connector Name** | **Description** |
| Login Validation | The user data entered during login is validated by checking in the database |
| Validate Role | Authorization is done by validating the role of the user. User can be admin/registered /un-registered user. |
| CRUD data | Admin can Create, Read, Update and Delete the resources. Users can use these resources. |
| Encrypting data to ensure privacy | Encryption has to done in order to ensure the user’s data privacy else the data will be misused |
| Display reports | Registered users can track their learning using the reports displayed in the dashboard |
| Valid users | Valid users are identified during authentication after the user provides the data to login the system. |

## Architecture Styles



|  |  |
| --- | --- |
| **Component** | **Description** |
| **Client Layer** | End users to access application through this layer. Various browsers or thin desktop clients are examples of clients. |
| **Presentation Layer** | The presentation layer deals with the presentation logic and the page rendering. It contains all the components needed to allow interactions with an end-user. |
| **Business Layer** | This layer encodes the real-world business rules that determine how data can be created, displayed, stored, and changed. Various business processes and its interaction, governed by various rules are managed at this layer. |
| **Data Layer** | Rules related to data access and management are implemented at this layer. It abstracts the logic required to access the underlying data stores. |

# SECURITY ARCHITECTURE

Security architecture is a unified description of the various components and mechanisms that describe how the potential risks and threats to the system are addressed. It also specifies when and where to apply security controls.

The key attributes of the description of security architecture are as follows:

* Components – these could include firewalls, network devices
* Mechanisms – protocols, monitoring techniques
* Standards followed – Encryption standards, Legal standards
* Controls – Access control, User groups, audit logs

## Security architecture diagrams

A screenshot of a map

Description automatically generated

## Components Description

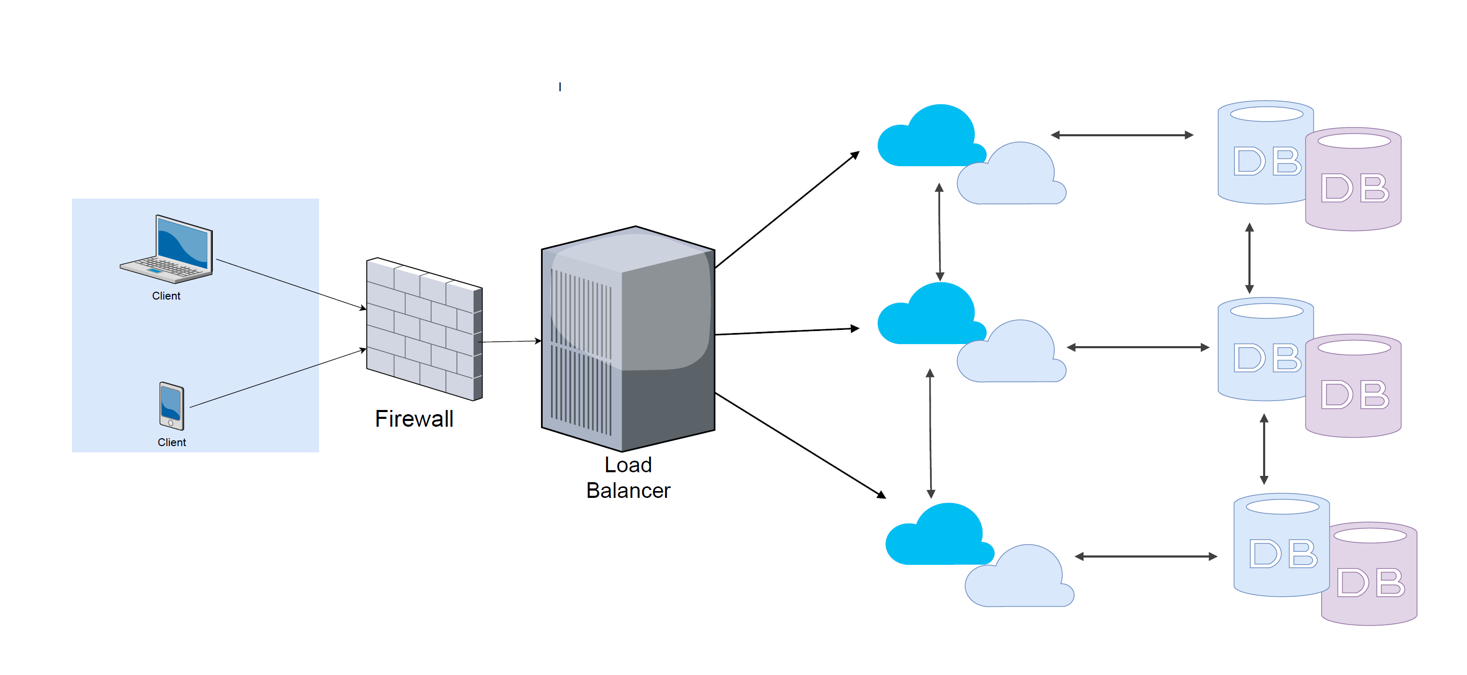
|  |  |
| --- | --- |
| **Component** | **Description** |
| **Authenticator** | This component is responsible for authenticating the service request received from the agents or applications outside the boundary |
| **Session Manager** | This component is responsible for providing real-time monitoring and control of all privileged account sessions to prevent and to detect malicious activity. |
| **Cache Manager** | This component is responsible for the primary mechanism of retrieving a [Cache](https://docs.jboss.org/infinispan/4.0/apidocs/org/infinispan/Cache.html) instance, and is often used as a starting point to using the [Cache](https://docs.jboss.org/infinispan/4.0/apidocs/org/infinispan/Cache.html). |
| **Cryptography** | This component is responsible for constructing and analyzing the [protocols](https://en.wikipedia.org/wiki/Communications_protocol) that prevent third parties or the public from reading private messages |
| **Authorizer** | This component is responsible for defining the access policy |
| **Session DAO** | This component is responsible for enabling [session](https://shiro.apache.org/static/1.2.5/apidocs/org/apache/shiro/session/Session.html) access to an EIS (Enterprise Information System) |
| **Pluggable Realms** | This component is responsible for providing application-specific security entities such as accounts, roles, and permissions to perform authentication and authorization operations. |

# INFRASTRUCTURE/DEPLOYMENT ARCHITECTURE

Deployment architecture covers:

1. Compute hardware
2. Storage hardware
3. Network hardware
4. Physical deployment diagram
5. Geographical distribution diagram
6. Trend/SLA/Growth
7. Communication details (type of cable, network, speeds…)

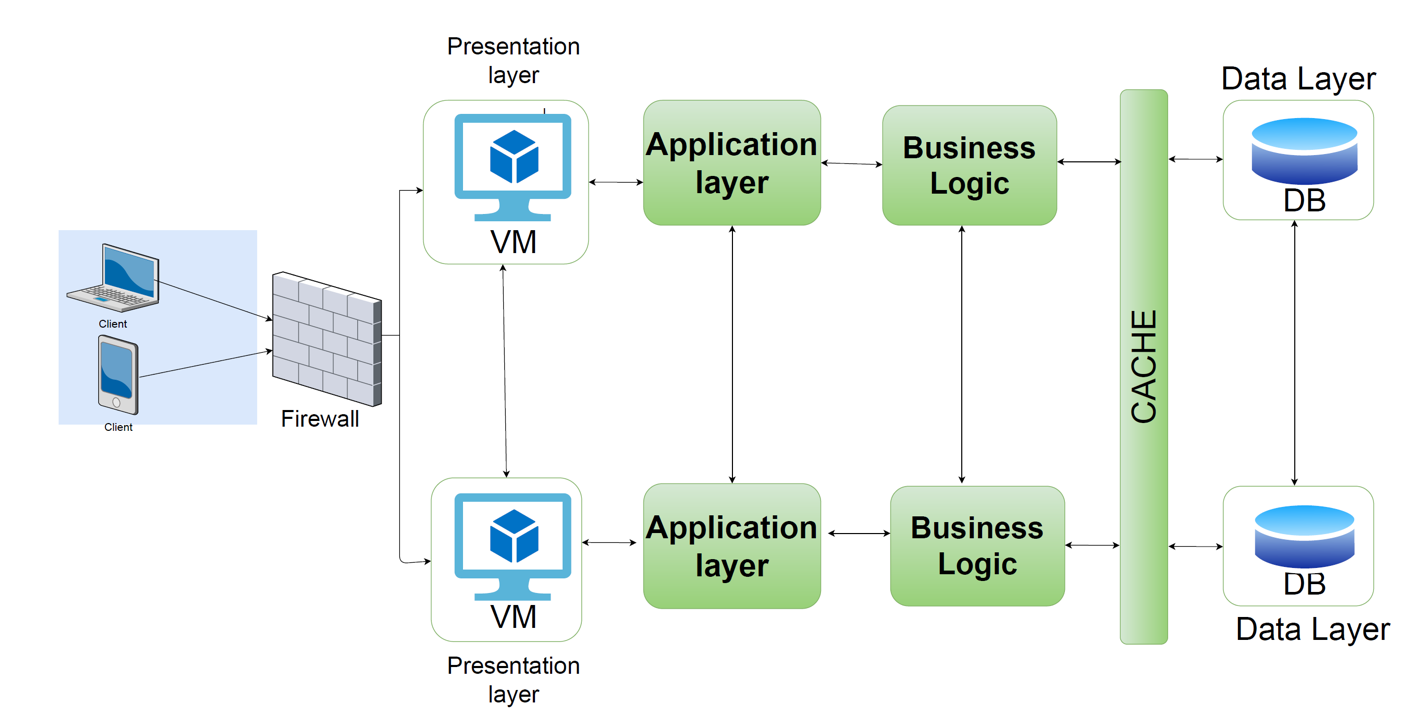
## Deployment Architecture Diagrams:



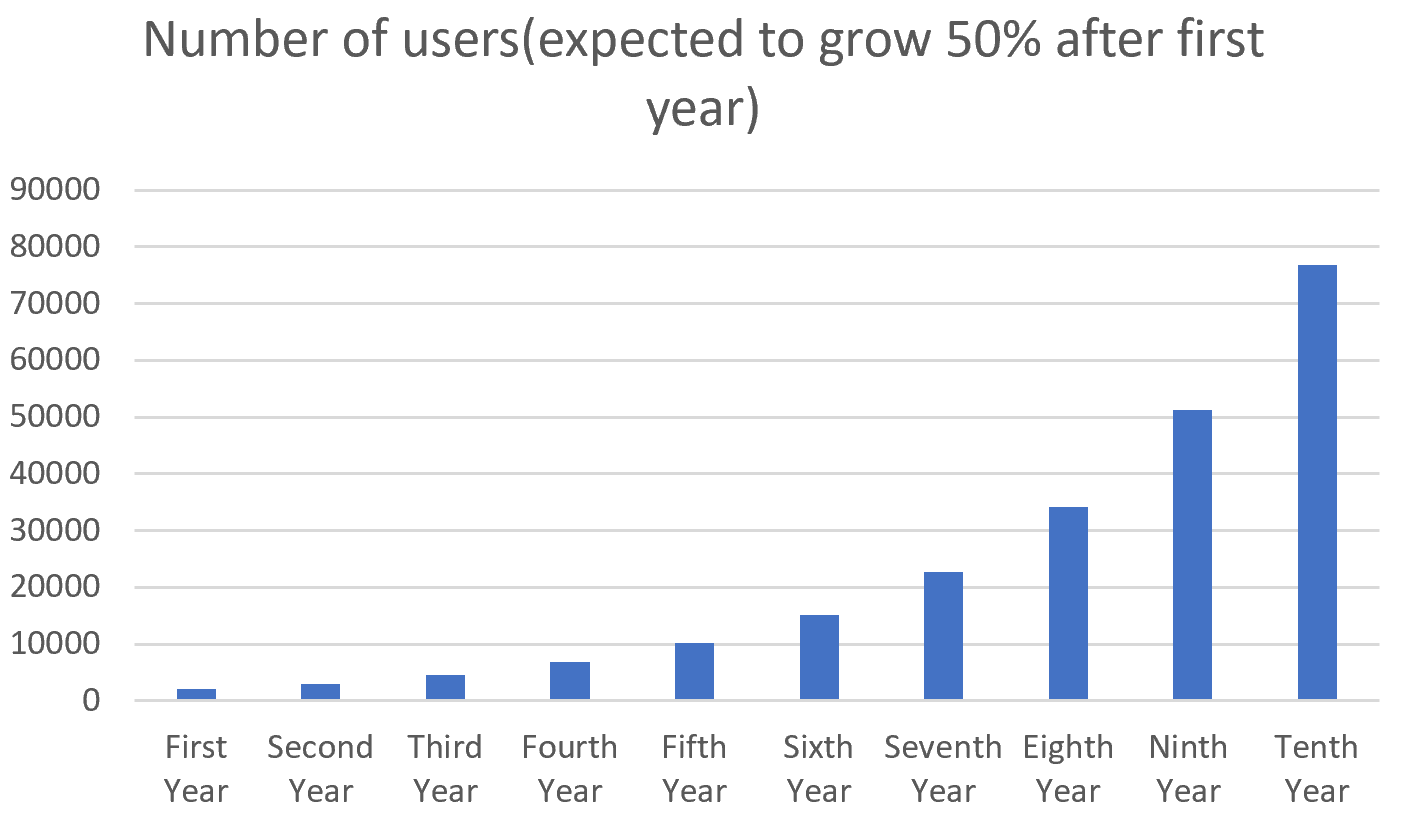
## Description of Components

|  |  |
| --- | --- |
| **Component** | **Description** |
| Database | A cloud database service used to build and accessed through a cloud platform. |
| Load Balancer | A Load balancer is used to distributes network or application traffic across several servers and to increase capacity (concurrent users) and reliability of applications. |
| Visualization Platform (VMs) | Running on a clustered Virtual Windows machine on cloud |
| Client (Browsers) | Supporting Google chrome, Internet explorer, Safari, Firefox, Opera browsers |
| Mobile (Application) | Supporting Mobile App in Android, IOS, Windows |
| Firewall | A Firewall is designed to prevent unauthorized access to or from a private network |

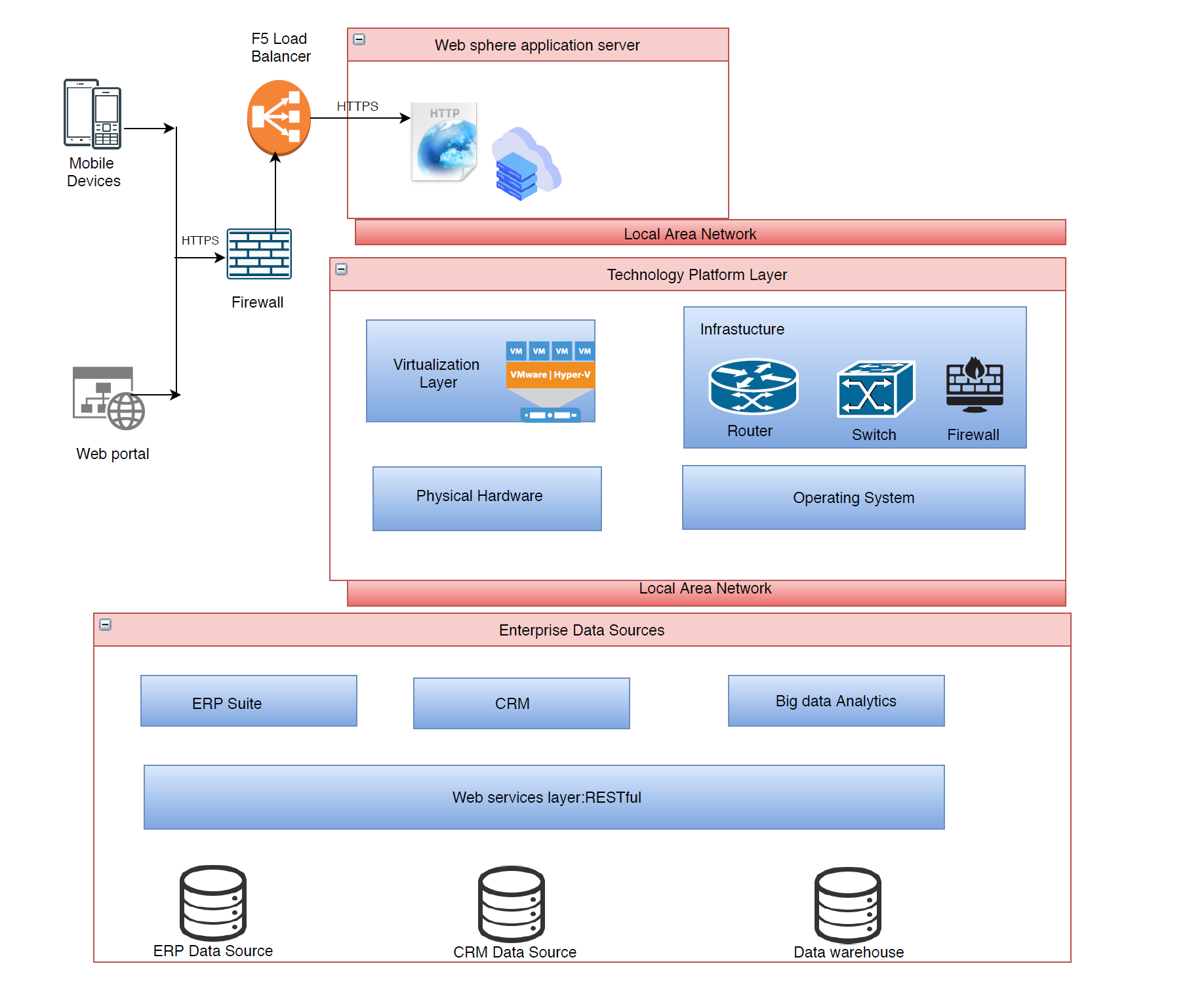
**Geographical Distribution Diagram**



**Graphical Representation of Users Growth**



**Network Computing Hardware Diagram**



# Glossary

|  |  |
| --- | --- |
| Term | Description |
| CRUD | Non-Functional Requirement(s) |
| ERP | Enterprise Resource Planning |
| HTTPS | Hyper Text Transfer Protocol Secure |
| VM | Virtual Machine |
| CRM | Customer Relationship Management |
| DB | Database |
| MVC | Model View Controller |
| JDBC | Java Database Connectivity |
| SLA | Service-level Agreement |

# References

* <https://towardsdatascience.com/10-common-software-architectural-patterns-in-a-nutshell-a0b47a1e9013>
* <https://www.tutorialspoint.com/software_architecture_design/introduction.htm>
* <https://www.techopedia.com/definition/72/security-architecture>
* <https://docs.microfocus.com/SM/9.61/Codeless/Content/install/sm_install/deployment_architecture.htm>
* <https://docs.microsoft.com/en-us/dotnet/architecture/serverless/architecture-deployment-approaches>