

Spring 5-2020

VZWAM Web-Based Lookup

Ruben Claudio
Dakota State University

Follow this and additional works at: <https://scholar.dsu.edu/theses>



Part of the [Databases and Information Systems Commons](#), [Information Security Commons](#), and the [Other Computer Sciences Commons](#)

Recommended Citation

Claudio, Ruben, "VZWAM Web-Based Lookup" (2020). *Masters Theses & Doctoral Dissertations*. 356.
<https://scholar.dsu.edu/theses/356>

This Thesis is brought to you for free and open access by Beadle Scholar. It has been accepted for inclusion in Masters Theses & Doctoral Dissertations by an authorized administrator of Beadle Scholar. For more information, please contact repository@dsu.edu.

VZWAM WEB-BASED LOOKUP

A graduate project submitted to Dakota State University in partial fulfillment of the
requirements for the degree of

Master of Science

In

Information Systems

May 2020

By

Ruben Claudio

Project Committee:

Dr. Ronghua Shan

Dr. Christopher Olson

Dr. Stephen Krebsbach



**MSIS Project
Plan Approval Form
(Form #2)**

Important: Your project plan must be reviewed and approved by your project supervisor before you can register for the implementation course. (See Project Guidelines on grad office website - current students link for detailed description of requirements)

Student Name: Ruben Claudio

Expected Graduation Date: Fall 2019

Committee:

Faculty Project Supervisor: Dr. Shan

Committee member: Dr. Olson

Committee member: Dr. Krebsbach

Master's Project Title: VZW Rep Look Up Tool

Description of Project: 100-word summary of your formal plan. You may attach additional pages to this form. The signed approval form with additional pages should be attached as first page of your formal plan. Be sure to include:

1. Introduction (very brief overview of what you proposing to do and why you are doing it.)
2. Statement of problem or question you have identified and brief summary of current situation (literature search)
3. Goals, objectives, purpose (what you plan to achieve - desired outcome of this project)
4. Scope of Work, Plan of Action, Activities (how you plan to achieve the objectives, the specific activities you will undertake)
5. Work Breakdown Structure (WBS) and Gantt Chart.
6. Deliverables (what you will actually have once you have completed your project, e.g., a database, a website, a program, etc.)

DO NOT WRITE THE DESCRIPTION HERE. ATTACH WORD-PROCESSED DOCUMENT


Students must bring the original form to the Office of Graduate Studies and Research.

You should retain a copy for your files.

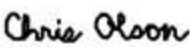
Approvals/Signatures:

Student: Ruben Claudio

Date: July 26, 2019

Faculty supervisor: 

Date: 7/26/2019

Committee member: 

Date: July 26, 2019

Committee 

member: 

Date:

ACKNOWLEDGMENT

I would like to thank Dr. Shan for his continued support, feedback, and guidance through the INFS 788 courses and this report. Thank you, Dr. Olson, for working with me setting up the webserver and feedback for the web app --thanks for your patience! Thank you also, Dr. Krebsbach, for accepting my invitation to be part of this committee.

And thanks to my family for continued support through my schooling and while I worked on this project including the report.

ABSTRACT

This web-based lookup will allow V employees to find territory sales rep much faster. It will simplify the process and eliminate manual processes.

At the moment, a combination of multiple manual processes is needed to find territory sales reps. The company's CRM does not allow to find rep sales quickly. When an in-house sales representative is talking to a prospect, this sales rep has to go through a few series of steps to find an outside or territory sales rep --which is usually needed to schedule in-person meetings, that results in delays while doing transactions with the prospects. Besides, because of the convoluted process, not everyone knows these steps and sometimes this process ends up in the hands of more experienced reps or supervisors which means the use of unnecessary additional resources. This will be a web-based app connected to the organization's database. Employees will have the ability to search for outside reps assigned to specific territories based on their zip codes. Thus, with just a couple of clicks, any employees will find territory sales reps which means that employees will utilize their time more efficiently and avoid escalating the situation.

Due to data security and privacy, A dummy database will be used. The only cost will involve time. The estimate is approximately 12 hours of coding using HTML, CSS, PHP, SQL database, and phpMyAdmin.

DECLARATION

I hereby certify that this project constitutes my product, that where the language of others is set forth, quotation marks so indicate, and that appropriate credit is given where I have used the language, ideas, expressions, or writings of another.

I declare that the project describes original work that has not previously been presented for the award of any other degree of any institution.

Signed,

Ruben Claudio

TABLE OF CONTENTS

PROJECT APPROVAL FORM	3
ACKNOWLEDGMENT	4
ABSTRACT	5
DECLARATION	6
TABLE OF CONTENTS	7
LIST OF TABLES	8
LIST OF FIGURES	9
INTRODUCTION	10
Background of the Problem	10
Statement of the problem	13
Objectives of the project	13
LITERATURE REVIEW	14
SYSTEM DESIGN	27
IMPLEMENTATION	37
CONCLUSIONS	45
REFERENCES	46
APPENDIX A: USERS' MANUAL	49
APPENDIX B: SYSTEM TECHNICAL DOCUMENTATION	51
APPENDIX C: WEB-APP EMAIL HYPERLINK AND AUTO-COMPLETE	56
APPENDIX D: DATABASE SCHEMA	57

LIST OF TABLES

Table 1. Software process modeling constructs <Sample table>	3
--	---

LIST OF FIGURES

Figure 1. Asset decision sub-space <sample figure>	3
--	---

CHAPTER 1

INTRODUCTION

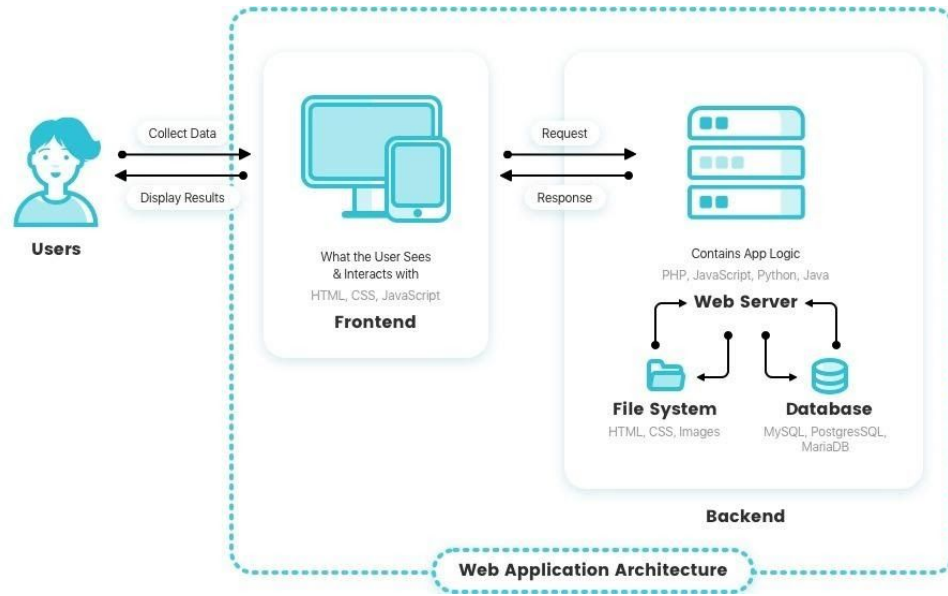
Background of the Problem

While working at the V Company, it was noticed the difficulties that inside sales representatives face when they try to find an outside sales representative. After talking to some more employees including inside sales reps and they all agreed how non-friendly users the current process is. Currently, inside sales reps have to go their CRM type a zip code, and the type of marketing lead this prospect came from (Kraenzel et al., 2017). This populates hundreds of records including records that are outside the scope either by the marketing type or zip code, then they have to check record after record to find the exact outside sales representative. All this usually happens while the inside sales rep is talking to a prospect over the phone (Weaver et al., 2018).

Almost three decades have passed since the introduction of online services. At the same time, several related concepts (such as service center architecture, cloud computing, and the Internet of Things) are integrated into the development area of the website (Ianevski et al., 2017). However, the concept of internet services and the definition of related basic concepts have not been fully updated. The generic term "web application" and the concept behind its definition do not encompass the different types of applications currently available that use web services. This article provides an overview of general network applications and then

extends the scope of the discussion from today's perspective by proposing the general term "web applications" to cover a broader range of systems than "network applications".

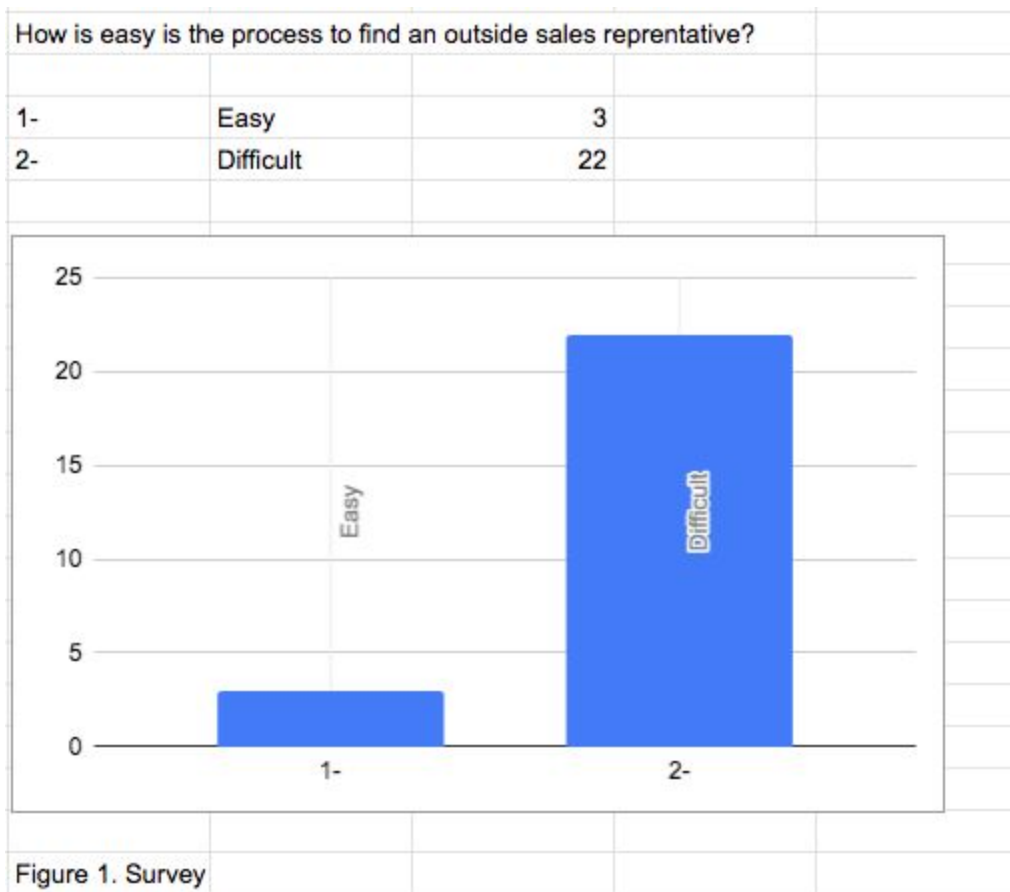
The concept of computer networks can be translated from the early 1960s. Since then, various types of services (such as file sharing and email) have been introduced into network systems that enable a new type of calculation (Rowe et al., 2018). The system is possible. Compared to stand-alone computer applications that run entirely on one computer, these distributed systems consist of different components that run on different computers, and these components communicate over a network in a client-server architecture. In this case, the concept of online services was introduced in the early 1990s and became popular so that online services could outsource the functions of other types of online services. Nowadays, web services are mainly used to create distributed customer-oriented systems (often referred to as "web applications") and the field is evolving rapidly (Bello et al., 2017). However, during the current investigation, it was found that the information regarding the domain is disordered and that the relationship between the different concepts of the domain is not easy to understand. Besides, it was noticed that the concept of web application concepts has evolved, but the definition has not been updated properly. It was noted that the concept under the generic term "web application" and its definition does not include the different types of applications currently available that use web services. The definition of concepts is important because they provide a precise consensus on the topic of interest that will help in the correct application of the concept.



It was proposed the generic term "web application", which encompasses a larger system, (Aslam et al., 2016) and the generic term "web application", the definition of which is likely to include systems. Next, it was proposed a definition of "web applications", highlighting the abstract architectural formalities of these web systems. Besides, it was discussed the development of web applications into a development medium and according to the proposed definition. A literary review was performed of basic network concepts (such as protocols and architectures) to gain a deeper understanding of the modern applications of these concepts. At the same time, it was carried out a series of experiments to obtain empirical evidence and to understand the application of these concepts to the development process. The experience is based on a prototype (Lam et al., 2016). The second part of this article contains general information about the rest of the article, as well as an overview and development of online applications (Huang et al., 2017). The third part extends the discussion of the proposal and the definition of the concept of web applications from a modern perspective. This section also explains the types and restrictions of online applications. The fourth part deals with

development methods and techniques (TT) of the different parts of the web application and corresponds to the proposed definition. Part V concludes the discussion and describes future work.

Below is the results of a survey I did with some of the sales reps



Statement of the problem

Currently, a lot of resources are used to be able to make a simple search. From the in-house sales reps wasting a lot of time to junior reps having to escalate the situation to management or senior reps because they just can't find the exact outside sales rep.

Objectives of the project

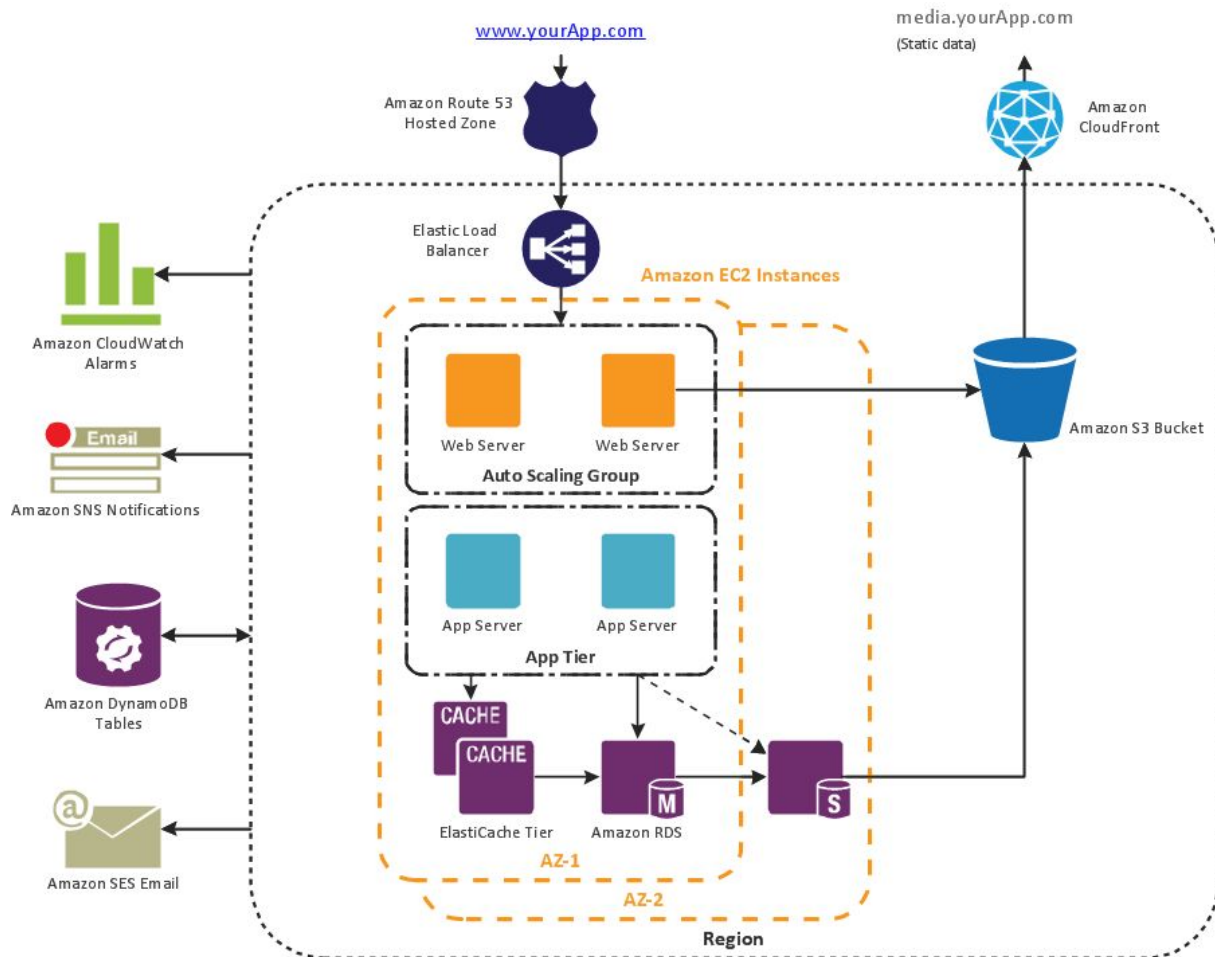
This web-based lookup app allows anybody to simply enter a zipcode or city and quickly find the name of the outside sales representative.

CHAPTER 2

LITERATURE REVIEW

Usually for small organizations, this is not a priority as they want their representatives to figure out the process. Besides, these small businesses usually have only a few sales representatives, so there is no need to use the equipment, and sometimes these names are learned only from the company's experience (Keshavan et al., 2018).

For large organizations, however, this can be a very tedious process, as they can have hundreds of external sales representatives, as my employer has encountered. Before I joined the company, my employer had some sort of program, but the software crashed and they didn't put the resources in the right place to solve the problem. That's why at large companies, this problem is usually solved by programming or purchasing.



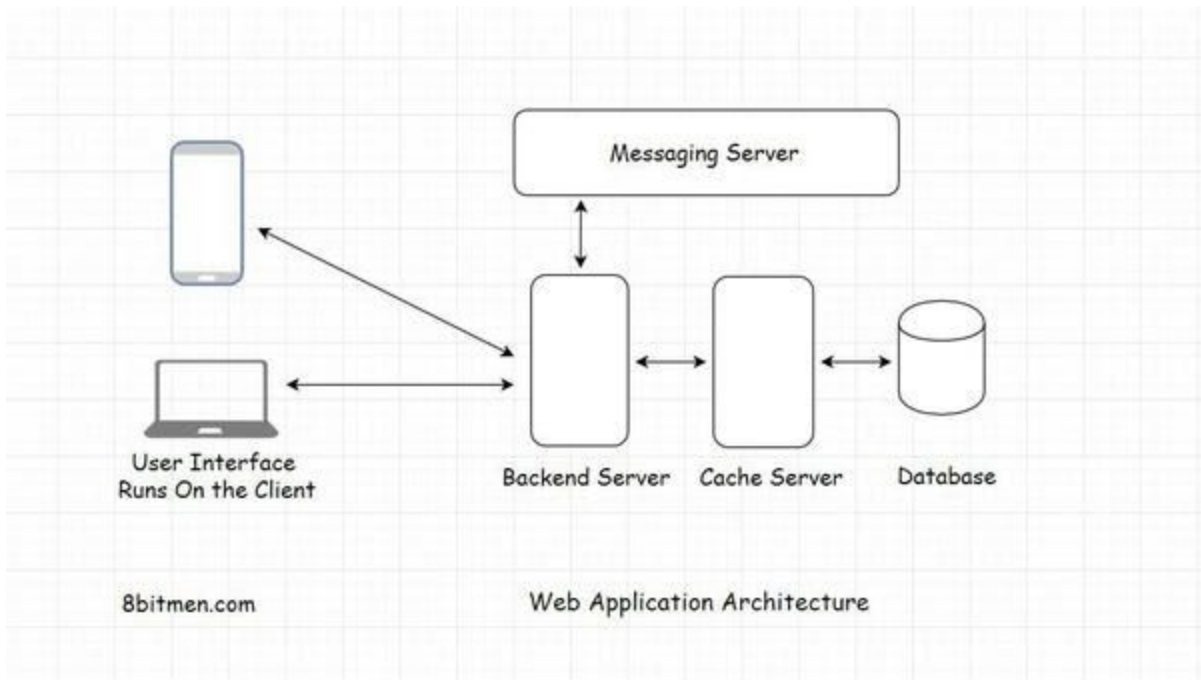
Since my current manager does not have a suitable solution, they have to find these names manually in CRM. Internal sales usually represent a long time. My employer needs these kinds of solutions because the company has more than 300 in-house sales representatives and more than 10,000 external sales representatives, so internal sales representatives save a lot of application time and make them more productive (Calvo et al., 2016).

This type of software is usually not available out of the box. I did a quick search on Google and nothing like it appeared. A search brings results to software that helps vendors in other ways. For example, Skynamo can help external sales representatives manage the day more efficiently by providing everything needed in mobile applications.

TRIKZO This section gives an overview of online applications and distinguishes them from desktop applications. He then discusses the development of web applications and emphasizes the concept that web applications have surpassed the term “web applications” (Maxbauer et al., 2016). A. Box and Network Applications Box applications are programs that run independently on computer devices (such as desktops, laptops, and even mobile devices). All components of these applications are located on the device, and these components are used in a single address in system memory. The components of the web application are shared. This is why the system works at multiple address points. PCMag has defined an online application in such a way that a web application downloads all or part website every time it runs.

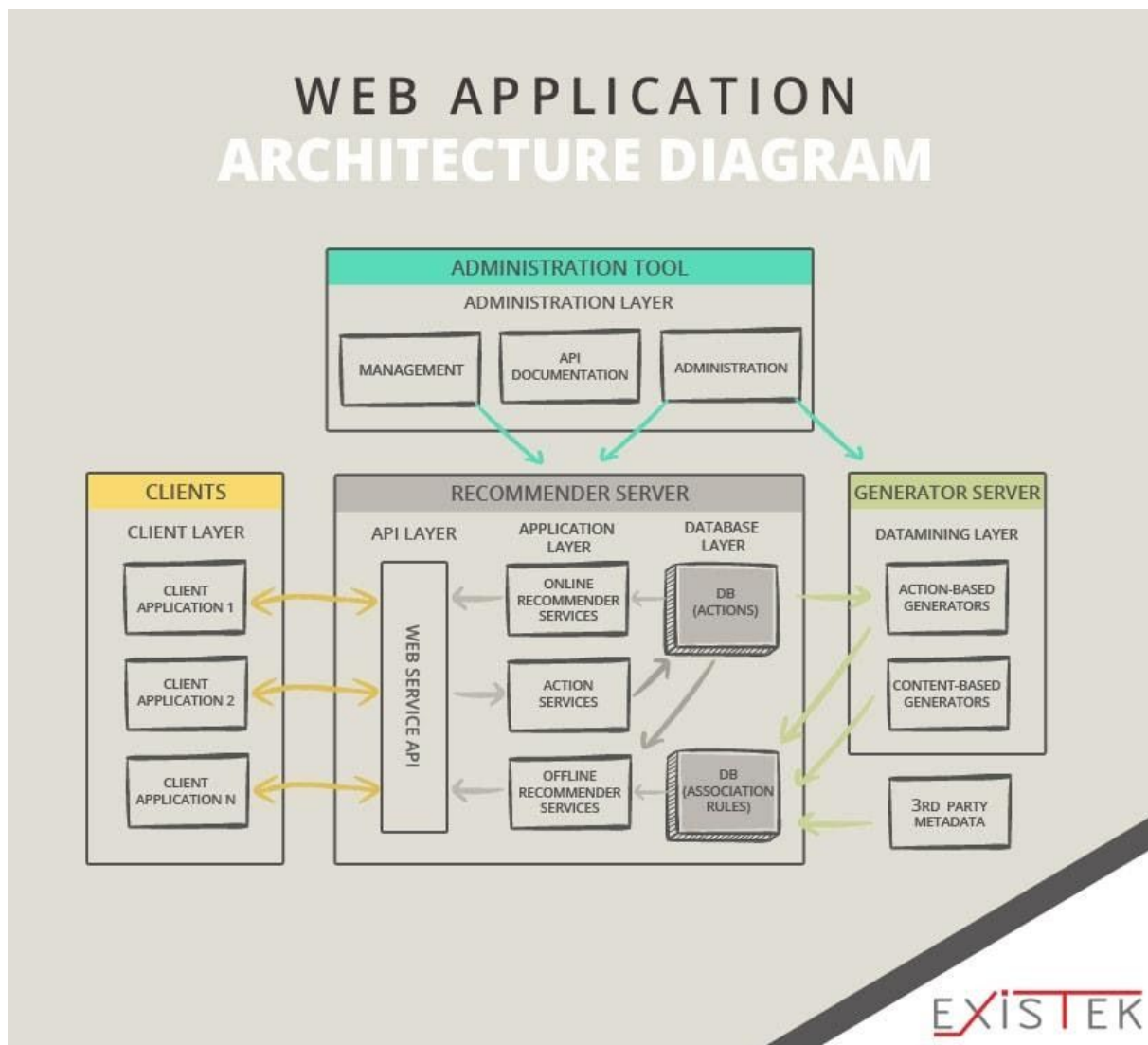
It is believed that “rich client applications” or “rich online applications” have their domains higher than some online applications (Gupta and Gupta, 2017). Therefore with this definition, it can be interpreted that the scope of web applications is limited to web-based applications.

B. Network Development Evolution. Earlier systems that used web services were called “websites” With continuous development, today’s network applications provide a platform that can deliver other service functions through network protocols obtainable from data networks (such as email and file transfer).

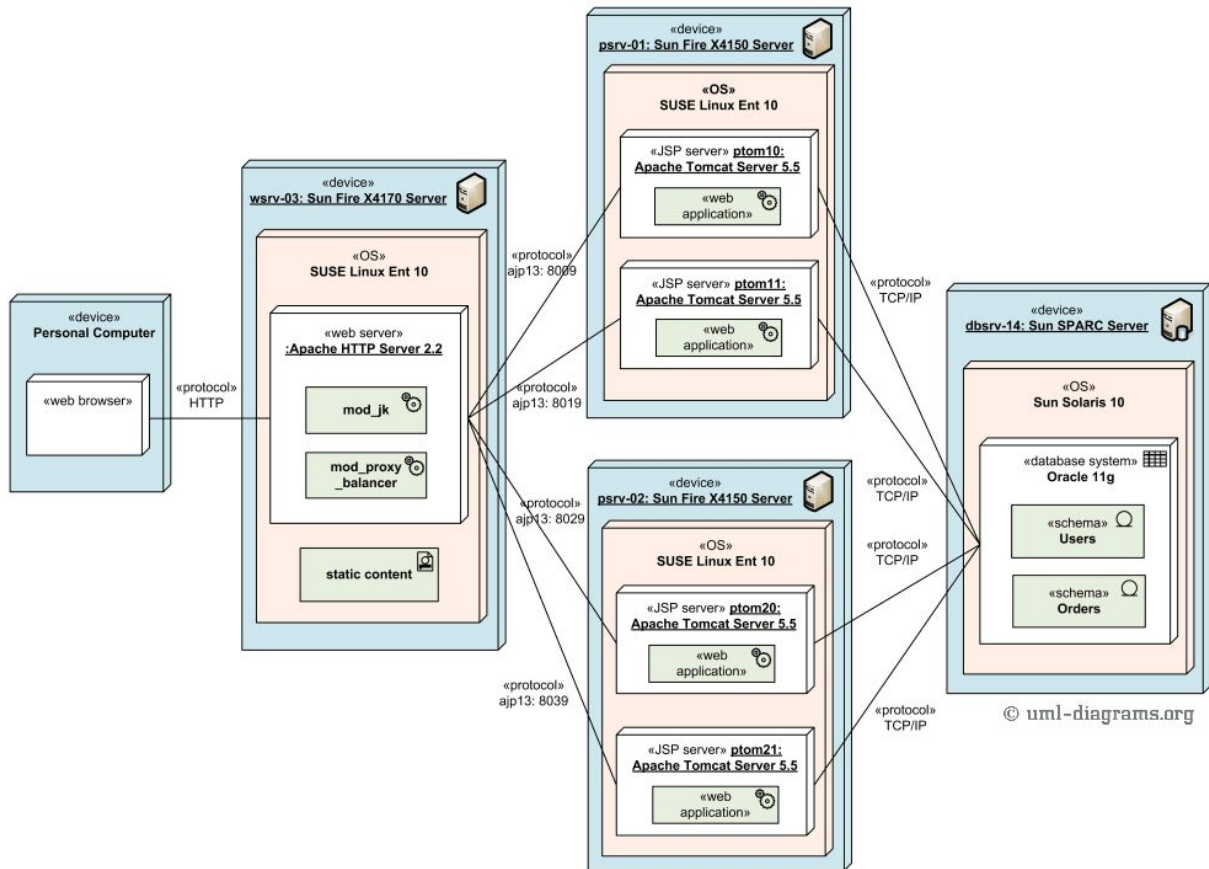


With all of these features, online and offline application services have become advanced, powerful, and very sophisticated units. Online applications can provide services to multiple users at the same time, which is less annoying than desktop applications. One of the main advantages of a web application is its use. He continued that the use of web applications is generally a matter of customizing server-side components on the network. The customer does not need a special program or configuration (Li et al., 2019). "In this case, web applications are easier to manage, maintain, and configure compared to a large number of concurrent users, and the workload is reduced. The web applications do not require any special software or settings that require a platform and local standalone access offer certain hardware dependencies, but modern TT reduces these barriers, has reached a minimum, and has grown rapidly. New systems such as mobile applications, system service centers, and the Internet of Things have become popular and use Internet services for communication (Rudik et al., 2017).

Note that the definition of "web application" does not cover this type of application and can be seen as a much more complex system than traditional conventional online applications. However, they all have similar architectural features in the region. This installation starts with creating the complexity in the web application. These dependencies on the location of the facts and technical factors in development, complex structures, different sizes, and extended features and functions together make systems that use Internet services very advanced and complex (Kaplinger et al., 2019).

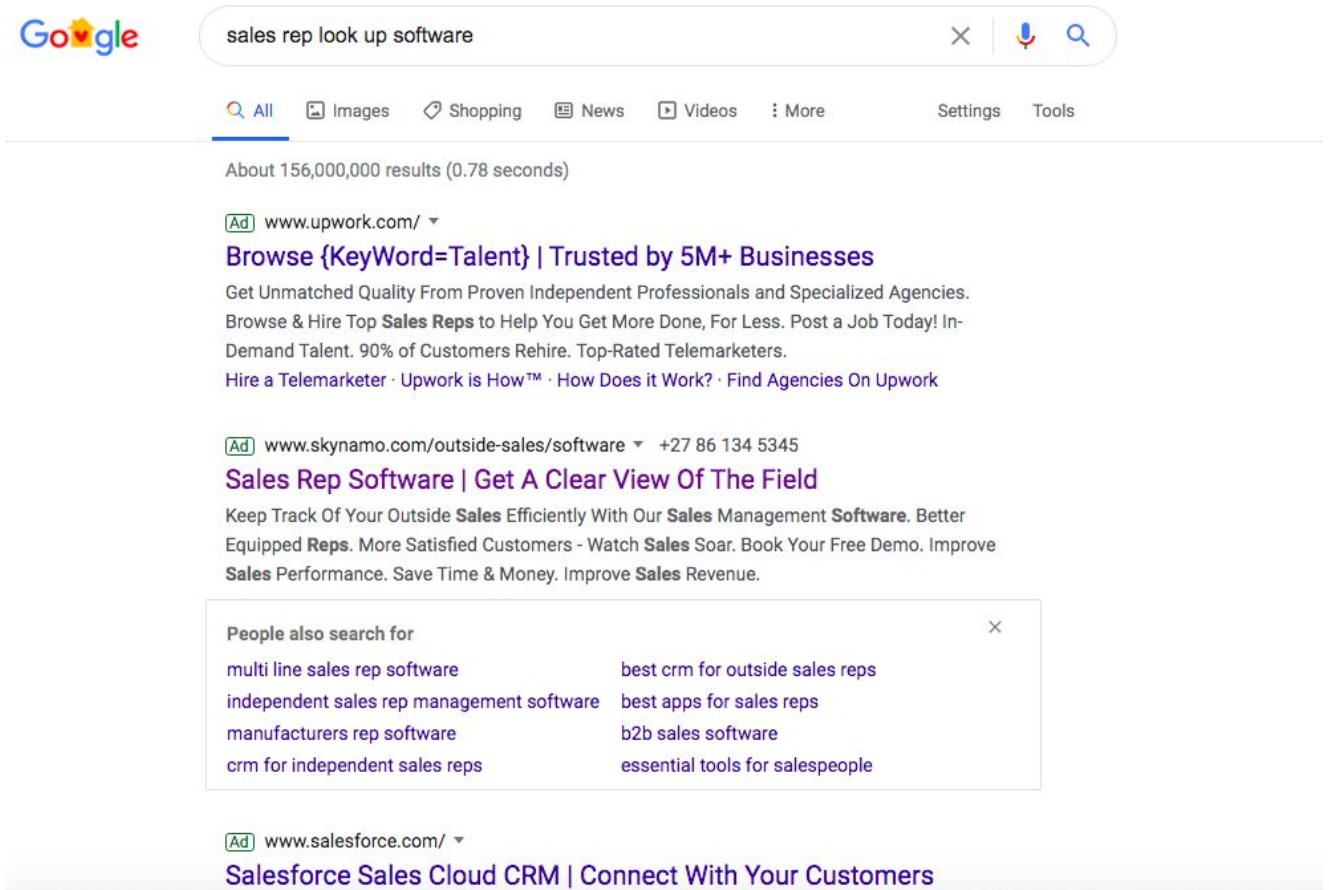


As mentioned in Chapter II.A, the term "web application" encompasses basic concepts of browser-based customer service applications. As mentioned in Part B, the use of Internet services now goes beyond web applications and general-purpose browsers. It is, therefore, believed that the term "network application" or its definition does not apply to all of these systems in general. However, as described in Section 2, Part B below, the architecture of these systems is always based on a client-server style using an application response model based on HTTP (Ofoegbu et al., 2019). Therefore, they can be summarized in one area. It is recommended the term "web applications" for all applications that are based on a client-server model and use web services for communication. This term can be viewed as a general term that encompasses a broader system than the term web application inside and outside the browser and thus solves the doubling of ordinary web applications (Kraenzel et al., 2017). During the literature search, it was decided on the definitions for the term "web application". Network applications are defined as follows: Network applications are all programs that can be accessed via a network connection via HTTP, not programs in the device memory (Weaver et al., 2018).



Before defining web-based applications, it is better to understand their form. As mentioned in Section II.A, a desktop application is an application that works independently as an application embedded in a laptop/desktop computer, mobile device, or even any other device. You can use Web page technology to communicate, process data, store, etc to add remote components to particular applications. Systems that use web services can be considered "web applications." Note that the browser itself is a desktop application that uses web services. Based on this discussion, it is recommended that you define web-based applications as follows (Rowe et al., 2018). This definition also defines the scope of network application functionality. They use web services based on a client-server architecture, application response model, standard HTTP, and other related technologies (Bello et al., 2017). To clarify

this definition, it is recommended that the base model for web-based applications be considered as "standalone web-based applications." This concept is expressed by the following formula.



Skynamo promotes that they help eliminate paperwork from outside sales reps:



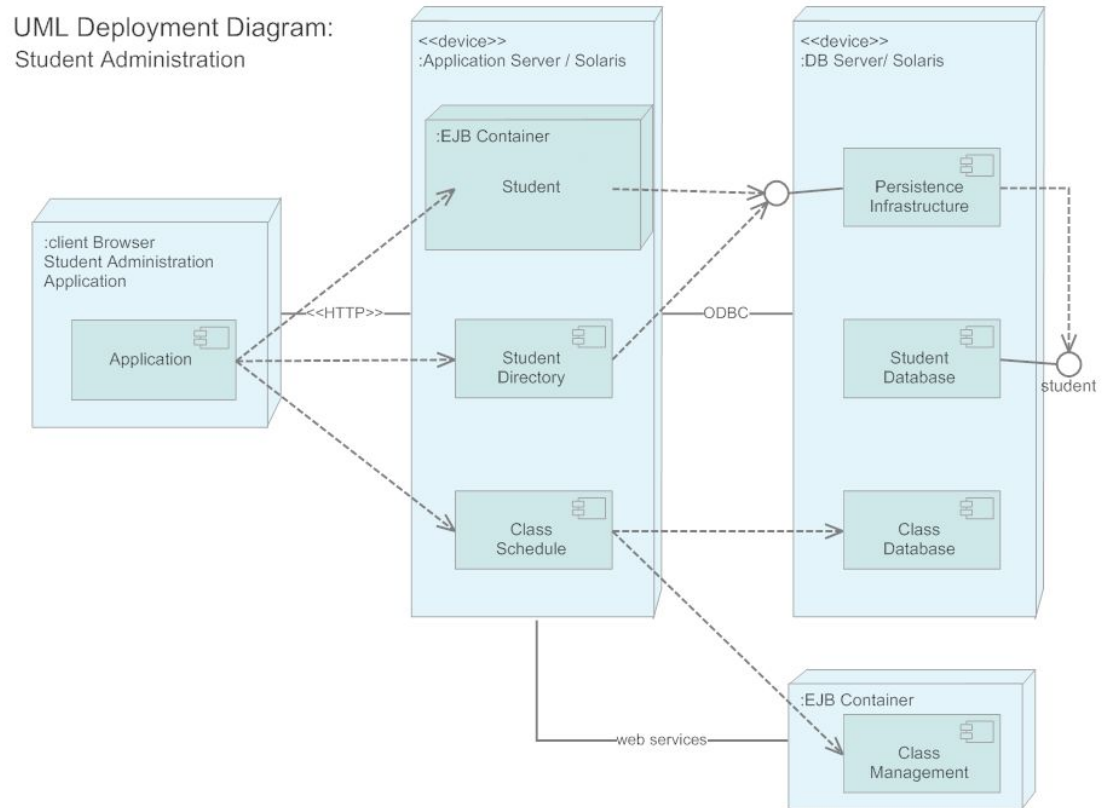
Then with the Skynamo app:



Skynamo also claims they reduce the time for outside sales reps to log mileage, review customer history, and check scheduled appointments from 10 minutes with a manual process to 1 minute using their mobile app (Aslam et al., 2016).

Again, this is not a competitor as the capabilities are very different, this is why I will not go into much depth here, but when trying to find similar software online, Skynamo came up on the list. Also, the more research I did to find similar apps and learn how others are doing, the more I notice that there is no software built out of the box with the capabilities my software offers. The other software which is now not working was built in house by a group of programmers in the company.

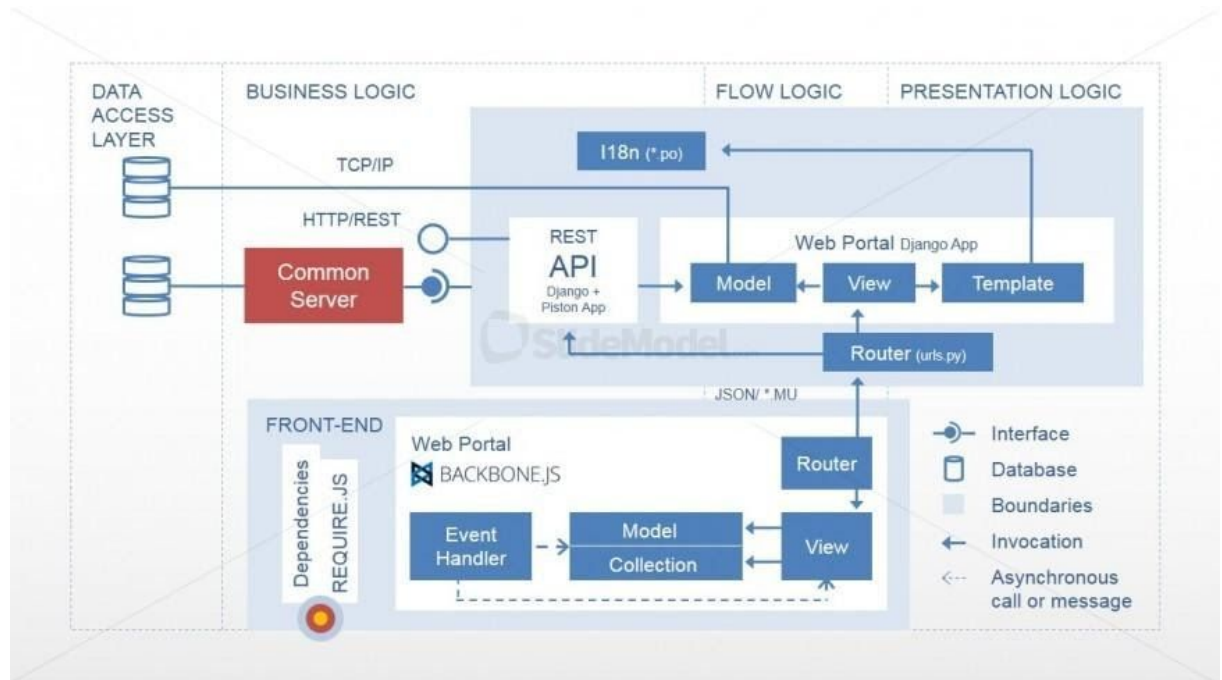
This section discusses the types of web-based applications to better understand the scope and definition of the proposed term “web-based applications”. Taivalaari and Mikkonen proposed similar ideas and introduced the concept of taxonomy to online software. In taxonomy, they used this dimension in three categories (Huang et al., 2017): "related to applications, operating environments, and overall architecture." The focus of their conversations is on the cloud service, in addition to the regular browser. Outside the app, they have very little discussion (Lam et al., 2016).



Keshavan et al., (2018) presented the framework in a similar context. He attempted to migrate any desktop program to the making to the source code. So there is no need to understand web technology." Its approach is not to improve knowledge of a network, but to take a traditional network project to a new level. However, the applicability of this method is questionable (Calvo et al., 2016). It is better to distribute web-based applications according to the type of customer item. The client item can be a browser or a browser. That's why the

division of web-based applications into two categories: browser-based and non-browser applications. Browser-based applications are usually available on the web. It can give good examples like Google and Facebook apps. The client component of a browser-based application is implemented in a browser (Maxbauer et al., 2016).

The browser loads these client components when the user starts using the program and continues to run until the user closes the browser. As a browser behavior, desktop applications can be used as client components to access a web server and access network services. Compared to browser client components, this has the advantage that native desktop applications can include rich and complex graphical user interfaces and can be easily developed compared to browser web applications. This is not a widely used technology for desktop computers/laptops and has all the disadvantages associated with managing and maintaining desktop applications compared to browser web applications. However, it is useful and practical in some situations. This concept is often used in mobile applications. Besides, these non-browser applications may be components of some embedded applications on some devices, and such systems are part of a modern concept called the Internet of Things.



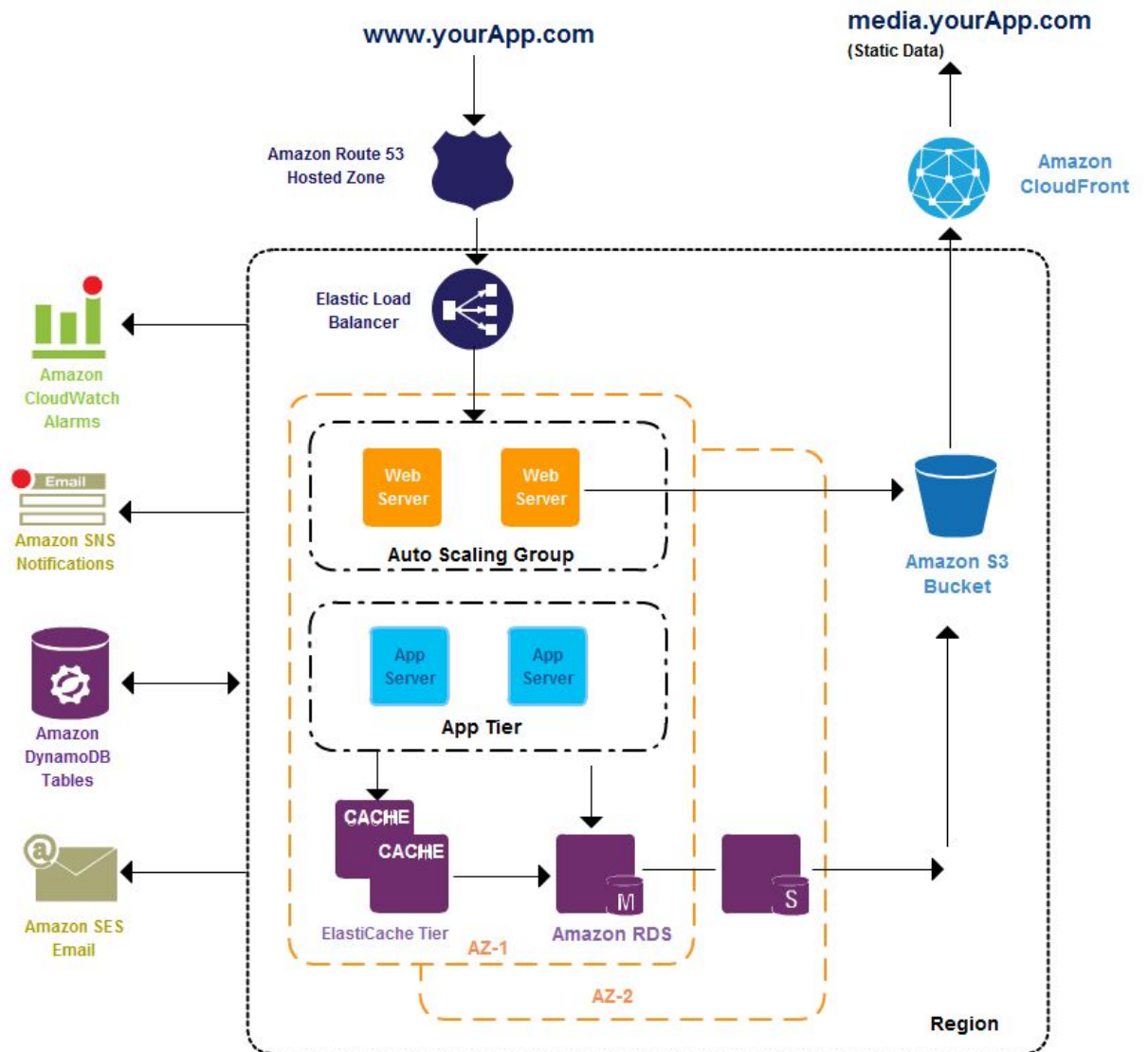
Limitations of web applications although web applications have many advantages over desktop applications, there are some limitations to web-based web applications. Some of their limitations also apply to all types of online applications. Gupta and Gupta, (2017) defined that these limitations are discussed in detail and analyzed in the context of five main web-based applications. 1. Poor graphical user interface: This affects web-based, HTML-restricted applications, and therefore lacks a rich user interface compared to desktop applications. Due to the poor user interface, in most cases, the implementation of a single function is spread across multiple pages, so users must test a traditional page sequence pattern to browse through serial pages to accomplish a single task (Prokhorenko et al., 2016). 2. Slow response speed: Because browser applications use a standby mode, page refresh limits their interaction.

CHAPTER 3

SYSTEM DESIGN

Of course, using a company's CRM to solve this problem has produced a domino effect, as aggravating the situation and wasting resources also cause other problems. Other than the use of technology, there is no other way to solve this problem, and the company has not invested resources to solve technical problems. In the case of a senior representative or sales manager, the current process takes an average of 3 to 12 minutes (Li et al., 2019). From entering a zip code to displaying information, this web page takes 5 to 15 seconds to display the required information on the screen.

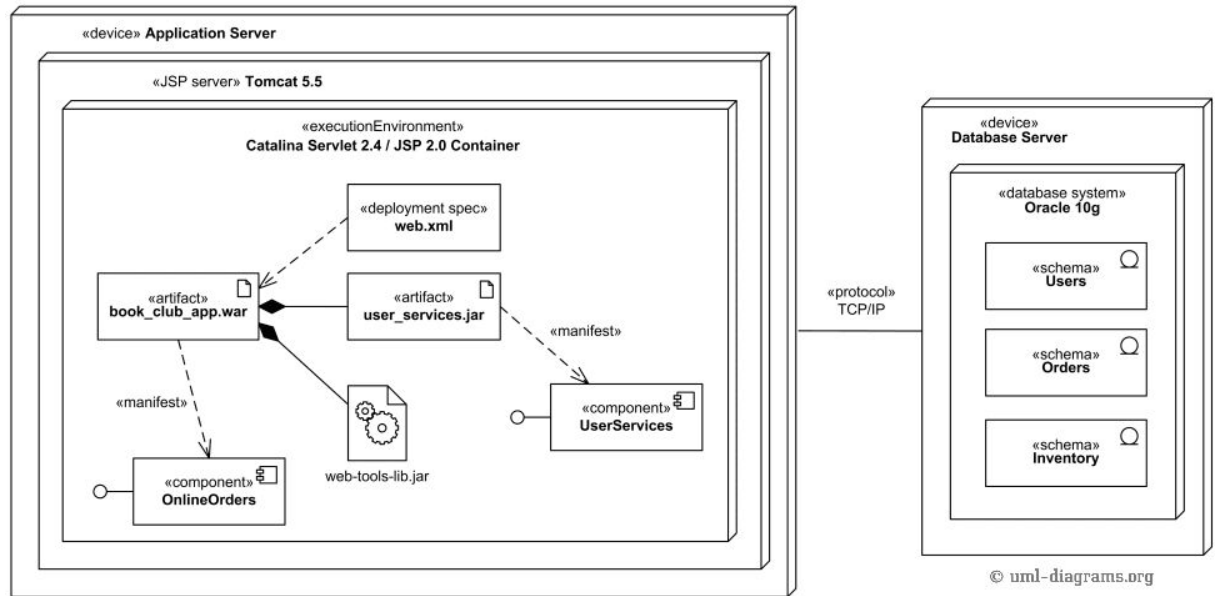
3-Tier Auto-scalable Web Application Architecture



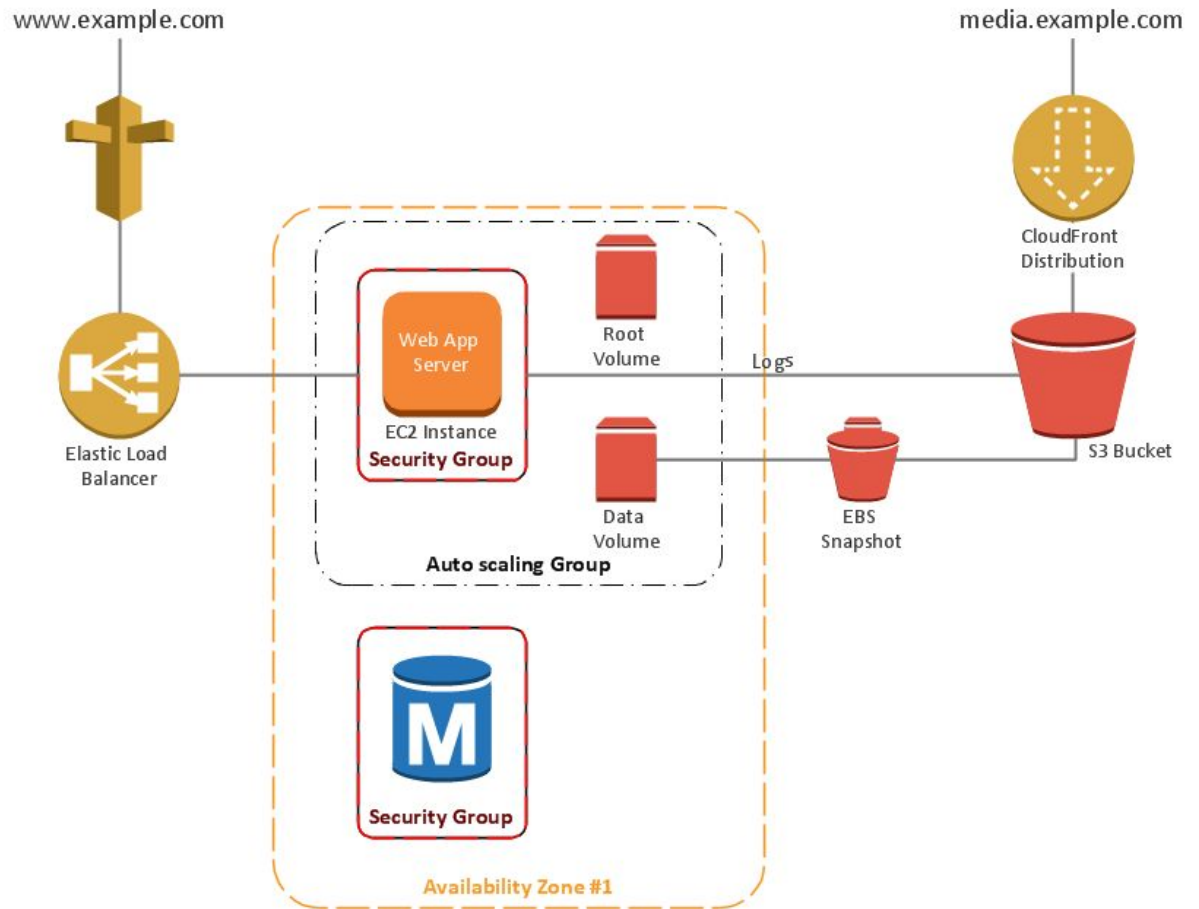
That is, techniques and techniques for developing web applications are essential to understand the full potential of system functions and to solve the complexity of development, as well as to fully understand the development of TT and its use. Many people have discussed the CT network from various perspectives. Our attempt to classify TT is based on Fielding's concept of architectural elements (components, connectors, and data), which focuses on the

architectural form of network applications and considers it consistent with the proposed definition. The TT of the website application client part of the client component TT is handled in the following categories: 1) browser, 2) non-browser. 1) Web-based customer web TT is the main resource for a web-based website or application website. This resource is loaded into the client browser to provide an interface that enables users to interact with the system via a graphical user interface. Typically, in web applications, clients and server components are developed on the same web page. If the business logic is completely separate from the website, some components of the website must be developed to be used with the business logic (Rudik et al., 2017). These components developed on the web page are implemented on the server before sending the web page to the browser and the server-side is developed via TT, as described in Section IV.B.

While the browser itself is a platform-related desktop application, like any other desktop program, the platform given by the browser application to the client component can also help the web application be independent of the platform. In this installation, the browser acts as the intermediate software for web-based applications. Carefully avoid detailed discussions about browsers, as the real focus is on the application components that work in the browser. Standard web-based web for the development of client-side components are: development of HTML content, including the GUI, CSS content formation the desired look, and address and address the behavior of JS content (Kaplinger et al., 2019).



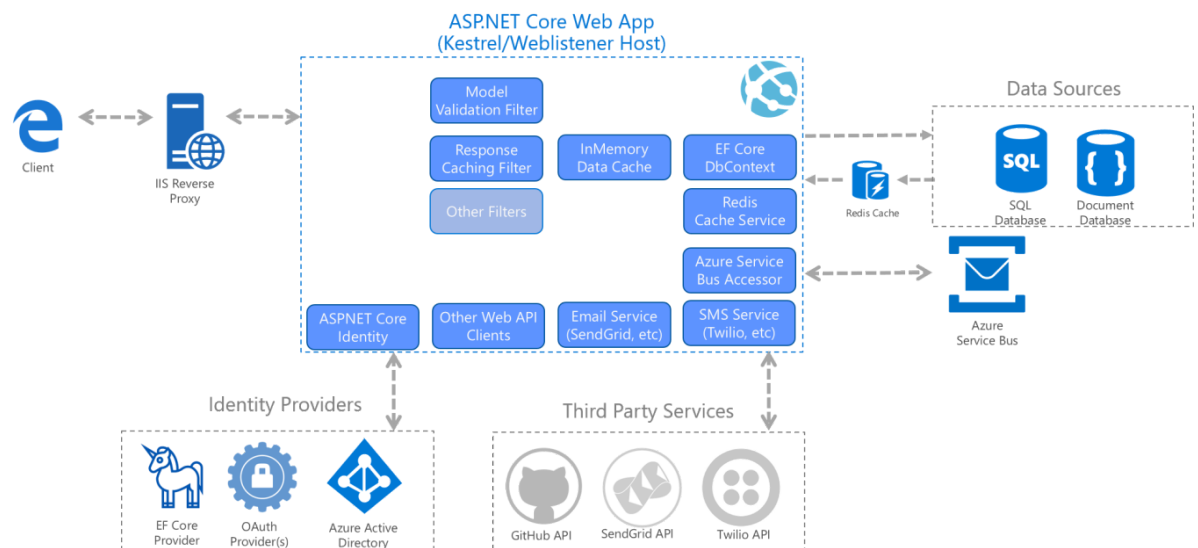
One of these limitations relates to the graphical user interface and the presentation of the content. Using browser applications to develop desktop applications, such as rich user interfaces, is not easy. Earlier, plugins (such as JAVA applets and Macromedia (post-Adobe) Flash) became popular. These patented technologies have their technical problems and will become less and less later as CT development is based on standard HTML, CSS, and JS technologies. However, developing rich graphical user interfaces for browser-based applications is more complex, challenging, and laborious than desktop applications. Besides, the browser-based client component can be developed into complete applications without the need for server-side components (Kraenzel et al., 2017). Therefore it can be used as a standalone desktop application. They can be placed on a client computer system without a hosting server and can run in an internet browser.



2) TT for non-browser client components on desktops or laptops, one way to develop non-browser client components is to use custom technologies that enhance Adobe Flash / Flex or JAVA devices. Another approach is to use a standard framework development course development application, such as .Net or JAVA. They also include utilities to develop web-compatible features that can communicate with web servers. As mobile applications evolve, an approach called “hybrid application” is becoming increasingly popular. These composite applications either use a browser container within the original application or use web-based applications. Or the original application itself communicates directly with the webserver (Ianevski et al., 2017). However, the behavior of a native desktop application with

a browser container is almost similar to that of another browser, so it is not appropriate in the context of this discussion. Today, most mobile applications are considered client components of web-based applications because they use web and web services to perform advanced functions such as local services, email, and the like. The modern concept of the Internet of Things has become a reality and is capable of implementing network usage and communication of devices such as clocks and vehicles. The various sensors in these device systems read the parameters, and the client components embedded in the systems communicate with the service components to handle these parameters and provide many advanced functions. A detailed discussion of the concept of the Internet of Things is avoided here and the focus is exclusively on the development of CT client components. Special CT points are available for the development of client components of IoT systems, such as frameworks, platforms, standards, and so on (Rowe et al., 2018).

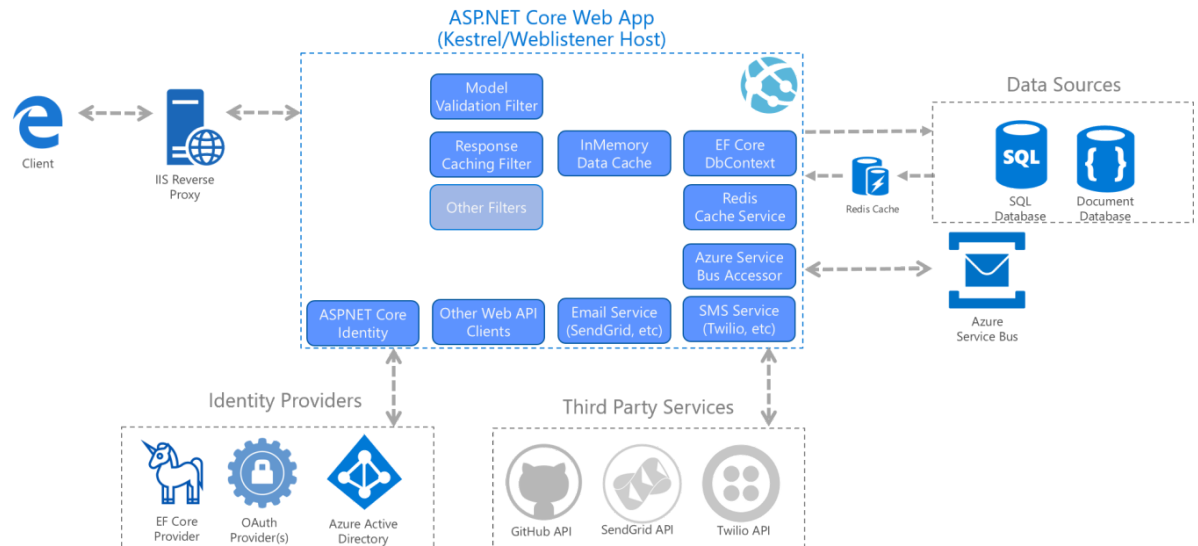
ASP.NET Core Architecture



TT for web service application components

Similarly, the development of the web server component is well connected to the webserver and the host operating system on which the webserver is installed. The server component can also be a "network" that is a special client application that can be developed using a server-side language and another dedicated CT (such as SOAP, REST, etc.). They are implemented as separate application components (Aslam et al., 2016). And providing an application programming interface (API) to the client component so that the client component can communicate with it via a separate HTTP-based TT. The scope of the concept of web service is independent and huge and it will not be repeated here. Service-Oriented Architecture (SOA) and cloud computing are advanced concepts related to Web services that make it possible to extend the functionality of web-based application service components. These concepts extend the basic architecture of the client and server to the layered architecture. The server-side TT, it is understood that they are independent of the client-side spiral CT. It should be noted, however, that compared to standards based on web application components, web services may provide better compatibility services for non-web client components (Lam et al., 2016).

ASP.NET Core Architecture



Elements of a TT application interface

The internal specifications for CT-based application communication are not discussed here, but their basic functions are discussed here. For connectors related to network application connections, it is necessary to consider several aspects that are discussed below. For the interface component used in the communication: the client includes an interface component to send the request and the purpose of receiving the response; the server-side has a receiver and a corresponding component (Huang et al., 2017).

Interface - This is a shared interface on which the system components exchange information and connect.

Data - This is information management and data flow.

Main tasks to be performed in the system design

Original definition

Design and define technologies that shape and implement system elements and their physical interfaces.

Identify which technologies and elements of the system are at risk of exit or development during the system phase. Plan your ability to replace them.

Document the project definition strategy, including the requirements for all permitted systems, products, or services, and the requirements that implement the project.

2. Create design properties

Define design features related to architectural features and see if they can be implemented.

Define interfaces that are not defined in the system architecture processor that require complete interfaces as development information evolves.

Define and document the designs of each system 2

3. Evaluate alternative methods to obtain systemic elements

Evaluation planning

Choose the most appropriate option.

If the development of system elements is decided, the project definition process and the rest of the implementation process will be used. If you decide to purchase or reuse system items, you can use the purchase process to obtain system items.

4. Administrative planning

Capture and preserve all the reasons for the choice, including features, options, and design and architecture decisions.

Evaluate and manage the development of project resources.

Factors that influence technical commitments in system design

Product classification

Time

Sometimes learning new technologies takes time. In this case, the commitment is based on the battery/technique used before the specified delivery date. If a change in a new battery/technology causes a major change in the delivery date and causes significant inconvenience to stakeholders, you can postpone the transition to the right time.

Cost

On a larger scale, technical decision making is based on which method is most economical, and based on that, it is possible to compare which method is most effective for purchasing a free system, modifying it, or creating a new system.

Efficiency

It is also based on which technology is most effective in technical companies, such as a choice between React and AngularJ for user interface applications.

Resistance

Durability, In this case, it is easy to save the product to correct errors, correct errors, and add other features. The commitment decision is based on the sustainability of the technology

Reliability

In this case, the compromise is constantly based on high-speed technology updated to a more efficient version.

Scalability

It is also based on more scalable technology and is a compromise between technologies that can effectively deal with increasing loads without compromising system efficiency.

Advantages of the MVC design model

Multiple developers can work on templates, controllers, and views at the same time.

The MVC can logically group the associated functions in the controller. Scenes from specific models are also collected.

Low switching - The core of the MVC framework is low switching between models, views, or controllers.

The model can have multiple views.

Easy to adapt - due to the difference in tasks it is easier to develop or adapt in the future

Cons

Understanding several techniques has become the norm. Developers using MVC need to be proficient in multiple technologies.

CHAPTER 4

IMPLEMENTATION

Since this is a web-based app, this is a very easy tool to use. It does not require to be installed in the local machine but to run the URL where the tool is located.

<https://dsu-class.com/infos730/ruben17/>

Also, because of the same reasons, it does not use local resources. The database should be stored in the cloud to avoid using local resources as well.

Below is a picture of the lookup



Sales Reps Look up Tool

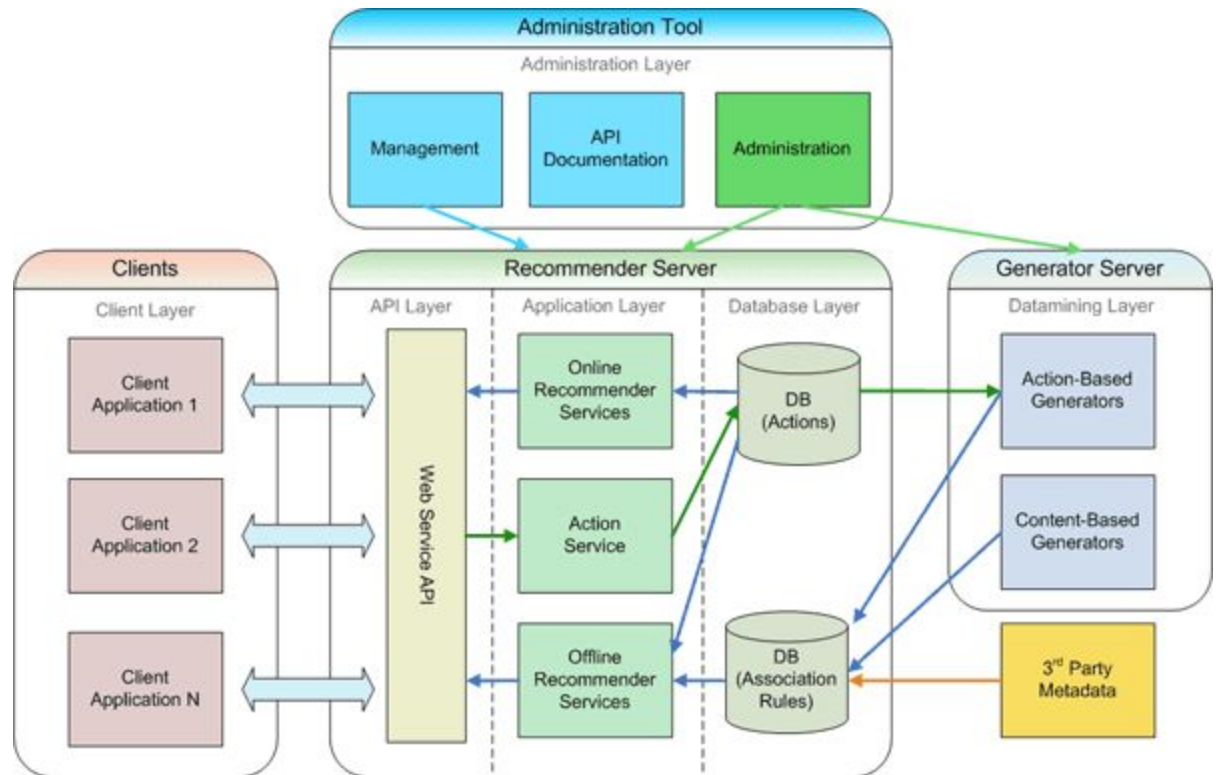
Please enter a zip code or City

Figure 2: Lookup tool

In browser client components, the browser itself launches the first application addressed to the server to load the original client component (usually a web page) of the system into the browser. The functions of the downloaded client component can initiate

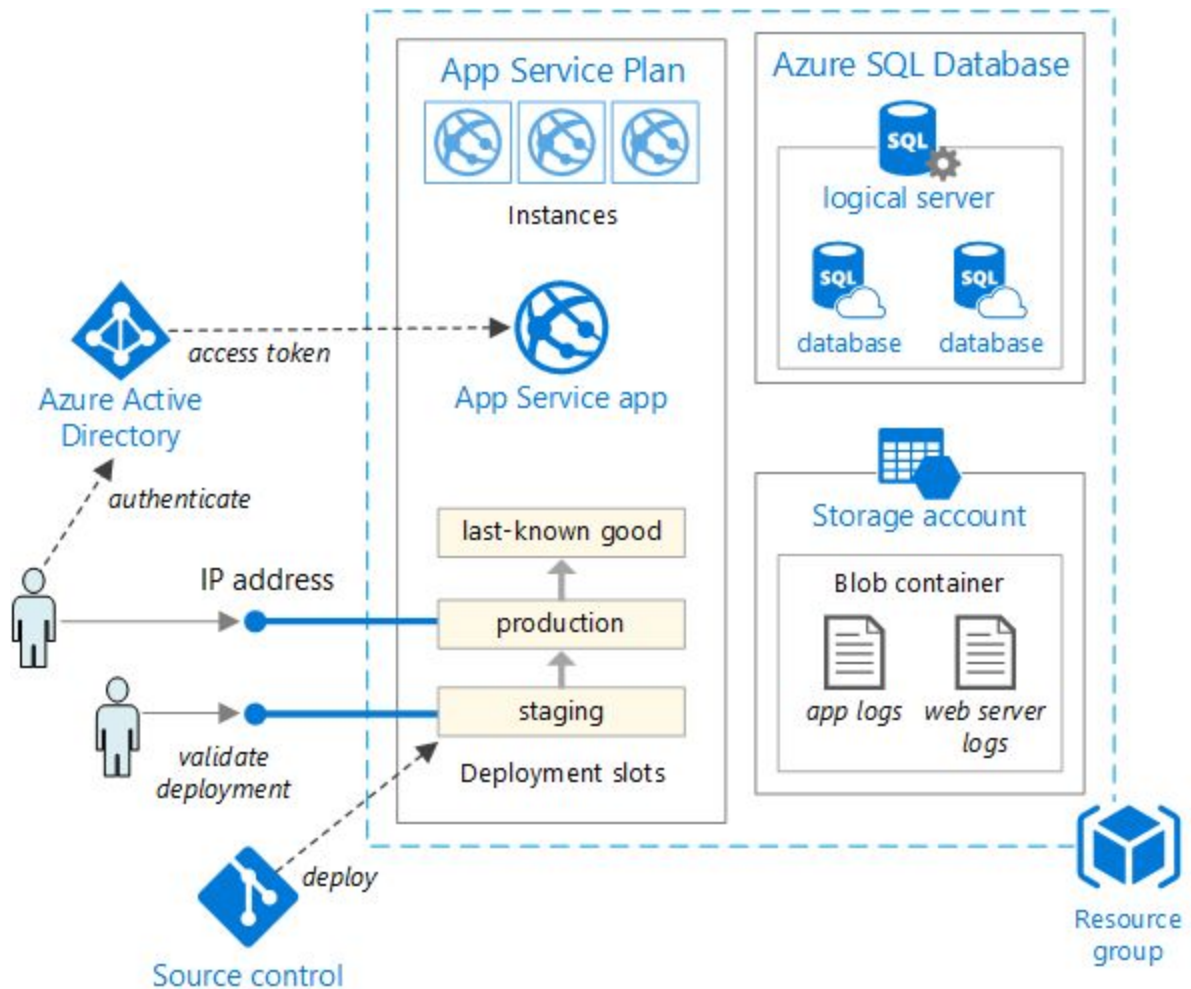
additional communication with the webserver based on user interaction. These features use mechanisms such as HTML hyperlinks, JS redirects, or form submission to begin application development (Calvo et al., 2016). With non-client components, you can use available native network utilities to develop send and reply requests on separate Web sites (e.g. SOAP, XML, or JSON). 2) TT is a network application server, the server-side interface component that waits for the component's connection to receive a request and send a response containing the right resources back to the client (Keshavan et al., 2018).

These functions look like the server itself, but we're focusing here on the layer above the webserver. When the webserver receives the request, it finds the intended resource in the application through the application and then passes the execution to the appropriate resource, which can be a web page, a file script, or any other item (Maxbauer et al., 2016). The target resource contains one or more server-side link components that handle the request and will be able to forward the information. Once the results have been prepared and the server-side connector receives, it prepares the response and releases it to the network server for transmission. Customer. These interface components are typically developed on the same TT as the development service components.



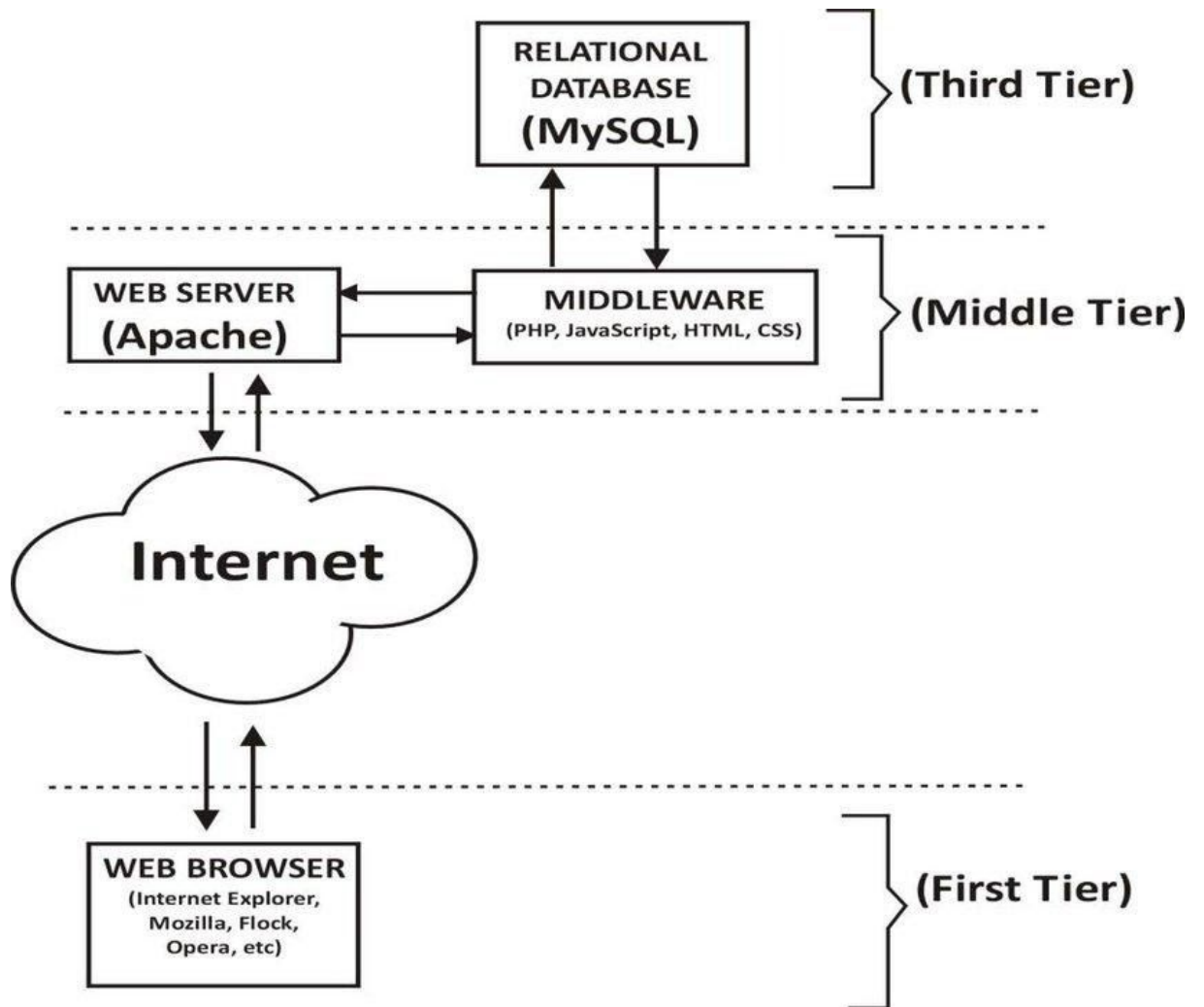
It can improve connectivity semantics and communication efficiency by reducing unnecessary processing. As previously mentioned, a request-response model is used for communication with a client-server, and communication is managed primarily via HTTP. However, other protocols such as SOAP or REST could also be included (Prokhorenko et al., 2016). The REST style uses a way to construct URLs to identify any relevant information to be sent to the server. The server-side components, with the REST functions, can be used in traditional web applications, the network without specific protocols (such as the SOAP), which can be considered the main advantage of the others. E. TT of Data Application Elements In browser applications, most responses are web pages that contain applications or other files, such as images or text files, and are downloaded from the browser and displayed to the user (Braga et al., 2017).

Metadata understood by the machine is likely to lead to the next wave of technological development. The development of metadata technology is slowing down, but now that the underlying technology has been integrated, the situation has changed.



Defining SOAP-based Web services is only part of achieving true interoperability. Creating a SOAP request requires information about the service - the name of the method to be called, its parameters and data type, and the semantics of the response (Li et al., 2019). This information can be obtained from human-readable documentation, but it does not achieve the actual goals of interoperability.

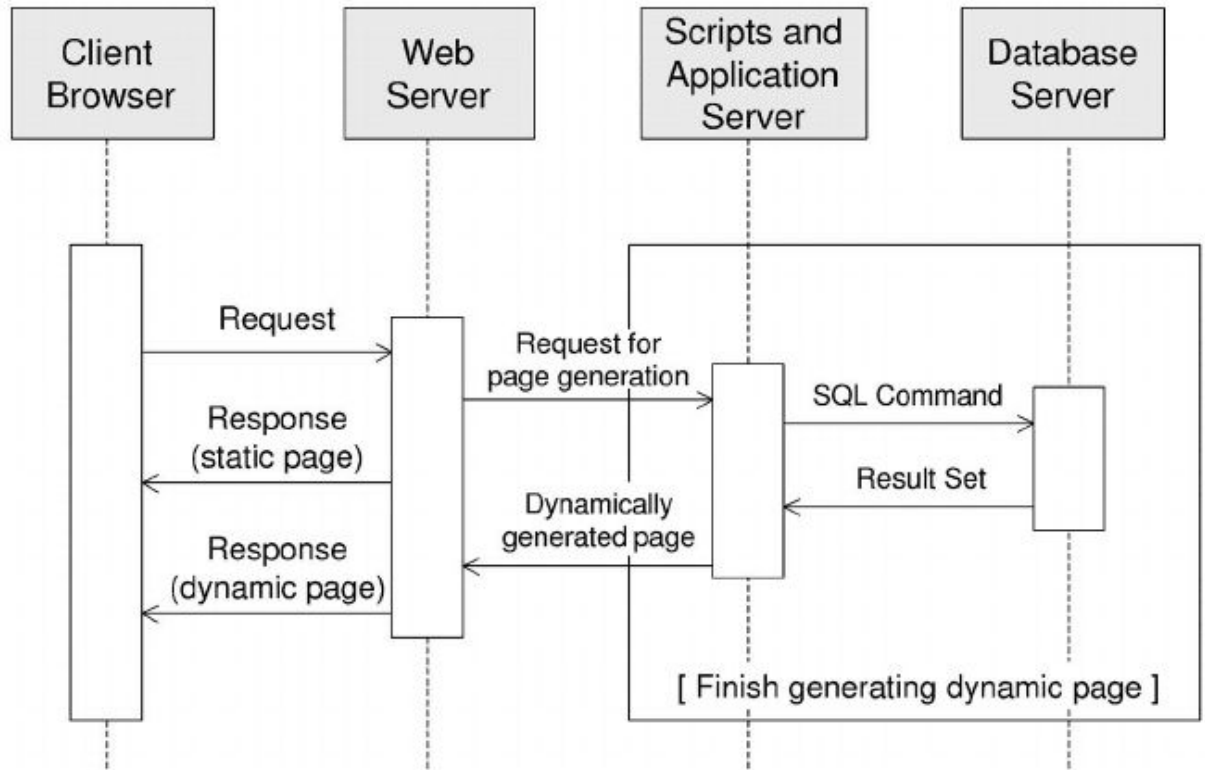
The component includes an element (to give an easy understanding of the service) and an element associated with an element called WeatherServiceBinding, which in turn (by its type-ID) is associated with an element called WeatherServicePortType. An element contains an element whose location property specifies a URL that can be used to name the service.



Web applications that provide separate functions and show functionality to other Web applications are distributed as clearly defined according to standard Internet protocols. The end-user uses a browser to send an HTTP request to the webserver (which includes the URL as well as the query parameters, titles, and optional body). The web server sends an HTTP

response (containing the header and body) to the browser to send to the user. The body of the response is any human-readable content, such as HTML pages, images, or sounds. Web services work the same way, except that the recipient of the response is another web page. Because the receiver is software and not a person, the answer must be understood to be mechanically (Ofoegbu et al., 2019). Therefore it must follow a protocol understandable to the machine (i.e., the computer that manages the network application). If you want to write an application that is used only in a very limited environment, you can decide how to run it yourself.

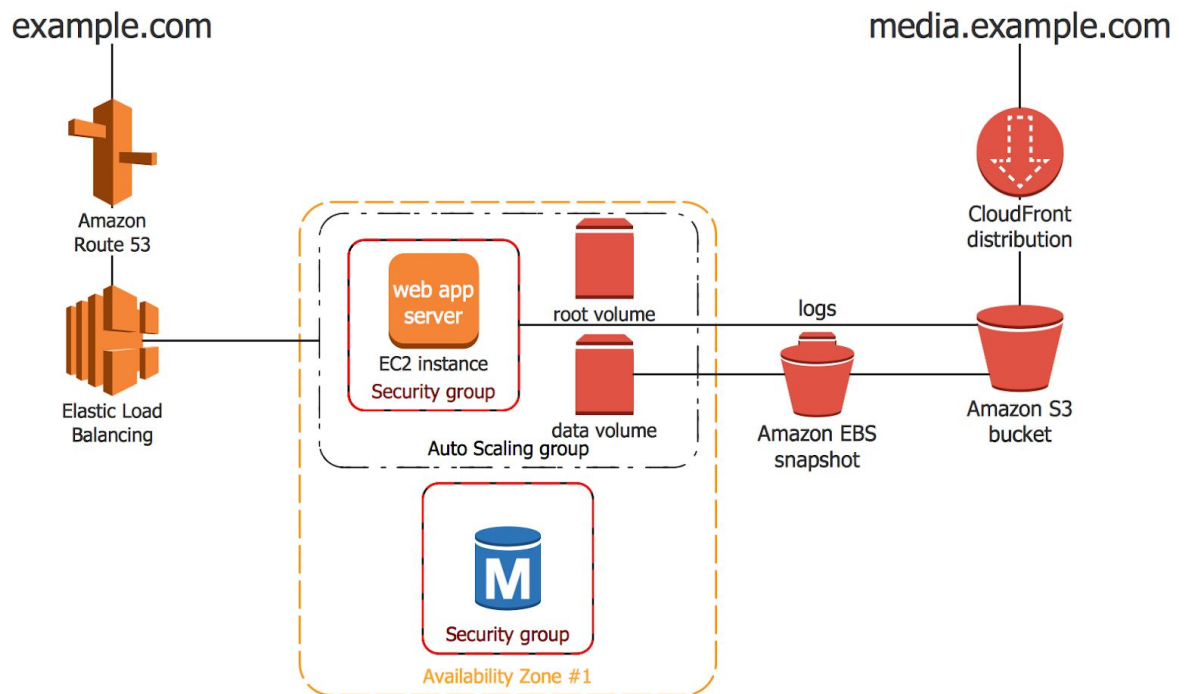
It refers to a set of protocols used to define standardized descriptive services, a mechanism to reveal its existence, and a mechanism to create, sends, and process web services. Together, these protocols provide consistency, scalability, and interoperability, allowing online services to operate in many environments (including Sun J2EE and Microsoft .NET) (Kraenzel et al., 2017). The rapid development of Internet technology is not free of cost. Technical incompatibilities and inconsistencies are putting pressure on the process of developing network applications.



After login, they can use the program's profile setting feature to change the password to the chosen password (Weaver et al., 2018). This is a safer way to register new users than the method included in the program. In its current state, the application allows anyone to enter an incorrect or valid email address, without penalty. New users will need to provide a valid email address and will receive a message containing the login password.

Although it is strongly emphasized that the application without the operating interface is incomplete, the download control interface was not implemented. The application package includes SQL queries to add partners and lists to the database, but there is no mechanism to update the database to add partners or lists (other than running SQL queries manually). If you want to use a fixed administrative service interface that is shared by all servers, you can add a password (preferably an encrypted password) to the application file. Providing database tables

(similar to VRLS CUSTOMER profile information for server authentication) is more efficient (and more secure). Ideally, the operating interface should be a separate application and set a unique Servlet context that is not related to the main application. Of course, you need custom features and visual components. Figure 10.11 provides a sample for the control interface of the struts-config.xml file. It defines how to log in, add and edit partners,

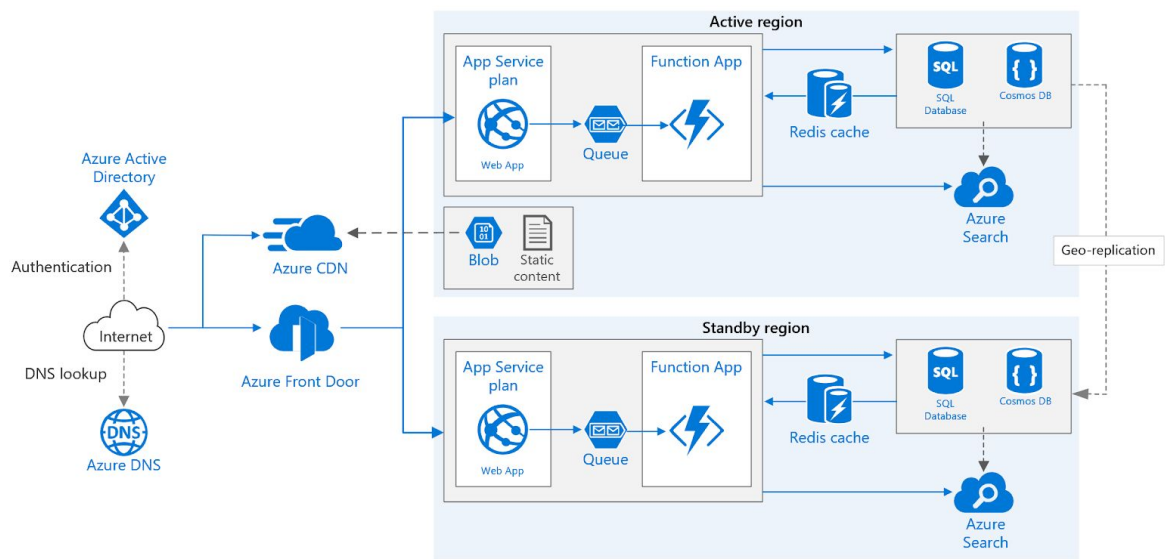


This application has two separate tasks and is combined into one: unknown users (during the registration process) create new customer profiles and registered users edit

existing

customer

profiles.



CHAPTER 5

CONCLUSIONS

Applications that use web services have evolved into very complex systems. The term “web application” probably covers these terms, and the definition of “web application” has not been updated to address these modern and complex issues. A complex system. This article suggests the general term “Web-based applications” to cover systems that use Web services. This article also includes a definition of this, focusing on the abstract architectural form of these applications/systems. This article discusses the limitations of web-based applications from a contemporary perspective, as well as the development of TT for web-based applications to adapt to the abstract architectural components of these applications. The discussion dealt with some modern concepts, such as cloud computing and the concept of web objects introduced. Currently, with the popularity of stronger RIA popularity, the development of standard web-based applications remains stagnant. It is a hope to further expand the concept of web applications to adapt to RIA and related concepts and develop a TT to achieve the abstract architectural forms of RIA.

REFERENCES

Kraenzel, C.J., Immerman, J.D., Mills, W.A. and Lu, J.J., International Business Machines Corp, 2017. System and method for developing and administering web applications and services from a workflow, enterprise, and mail-enabled web application server and platform. U.S. Patent 9,805,337.

Weaver, S., Shank, S.D., Spielman, S.J., Li, M., Muse, S.V. and Kosakovsky Pond, S.L., 2018. Datamonkey 2.0: a modern web application for characterizing selective and other evolutionary processes. *Molecular biology and evolution*, 35(3), pp.773-777.

Ianevski, A., He, L., Aittokallio, T. and Tang, J., 2017. SynergyFinder: a web application for analyzing drug combination dose–response matrix data. *Bioinformatics*, 33(15), pp.2413-2415.

Rowe, E., Palsson, B.O. and King, Z.A., 2018. Escher-FBA: a web application for interactive flux balance analysis. *BMC systems biology*, 12(1), p.84.

Bello, S.I., Bello, R.O., Babatunde, A.O., Olugbebi, M. and Bello, B.O., 2017. A University Examination Web Application Based on Linear-Sequential Life Cycle Model.

Aslam, B., Shaikh, M.P. and Olson, K., Microsoft Technology Licensing LLC, 2016. Automatic recognition of web application. U.S. Patent 9,244,709.

Lam, F., Lalansingh, C.M., Babaran, H.E., Wang, Z., Prokopec, S.D., Fox, N.S. and Boutros, P.C., 2016. VennDiagramWeb: a web application for the generation of highly customizable Venn and Euler diagrams. *BMC bioinformatics*, 17(1), p.401.

- Huang, H.C., Zhang, Z.K., Cheng, H.W. and Shieh, S.W., 2017. Web application security: Threats, countermeasures, and pitfalls. *Computer*, 50(6), pp.81-85.
- Keshavan, A., Datta, E., McDonough, I.M., Madan, C.R., Jordan, K. and Henry, R.G., 2018. Mindcontrol: A web application for brain segmentation quality control. *Neuroimage*, 170, pp.365-372.
- Calvo, R., Vikingstad, O., Kakes, W., Edwall III, C.E., Pinzon, J.C., Frost, M., Chang, J.W., McLachlan, M. and Coelho, E., Apple Inc, 2016. Web application architecture. U.S. Patent 9,384,101.
- Maxbauer, D.P., Feinberg, J.M. and Fox, D.L., 2016. MAX UnMix: A web application for unmixing magnetic coercivity distributions. *Computers & Geosciences*, 95, pp.140-145.
- Gupta, S. and Gupta, B.B., 2017. Detection, avoidance, and attack pattern mechanisms in modern web application vulnerabilities: present and future challenges. *International Journal of Cloud Applications and Computing (IJCAC)*, 7(3), pp.1-43.
- Prokhorenko, V., Choo, K.K.R. and Ashman, H., 2016. Context-oriented web application protection model. *Applied Mathematics and Computation*, 285, pp.59-78.
- Braga, R.C., Alves, V.M., Muratov, E.N., Strickland, J., Kleinstreuer, N., Tropscha, A. and Andrade, C.H., 2017. Pred-skin: a fast and reliable web application to assess skin sensitization effect of chemicals. *Journal of chemical information and modeling*, 57(5), pp.1013-1017.

Fonseca, C.R., Hidalgo, V., Díaz-Delgado, C., Vilchis-Francés, A.Y. and Gallego, I., 2017. Design of optimal tank size for rainwater harvesting systems through use of a web application and geo-referenced rainfall patterns. *Journal of Cleaner Production*, 145, pp.323-335.

Li, W.B., Mao, X., Yi, L. and Zhu, K., International Business Machines Corp, 2019. System and method for creating an internationalized web application. U.S. Patent 10,229,115.

Rudik, A.V., Bezhentsev, V.M., Dmitriev, A.V., Druzhilovskiy, D.S., Lagunin, A.A., Filimonov, D.A. and Poroikov, V.V., 2017. MetaTox: web application for predicting structure and toxicity of xenobiotics' metabolites. *Journal of chemical information and modeling*, 57(4), pp.638-642.

Denis, F., Yossi, S., Septans, A.L., Charron, A., Voog, E., Dupuis, O., Ganem, G., Pointreau, Y. and Letellier, C., 2017. Improving survival in patients treated for a lung cancer using self-evaluated symptoms reported through a web application. *American journal of clinical oncology*, 40(5), pp.464-469.

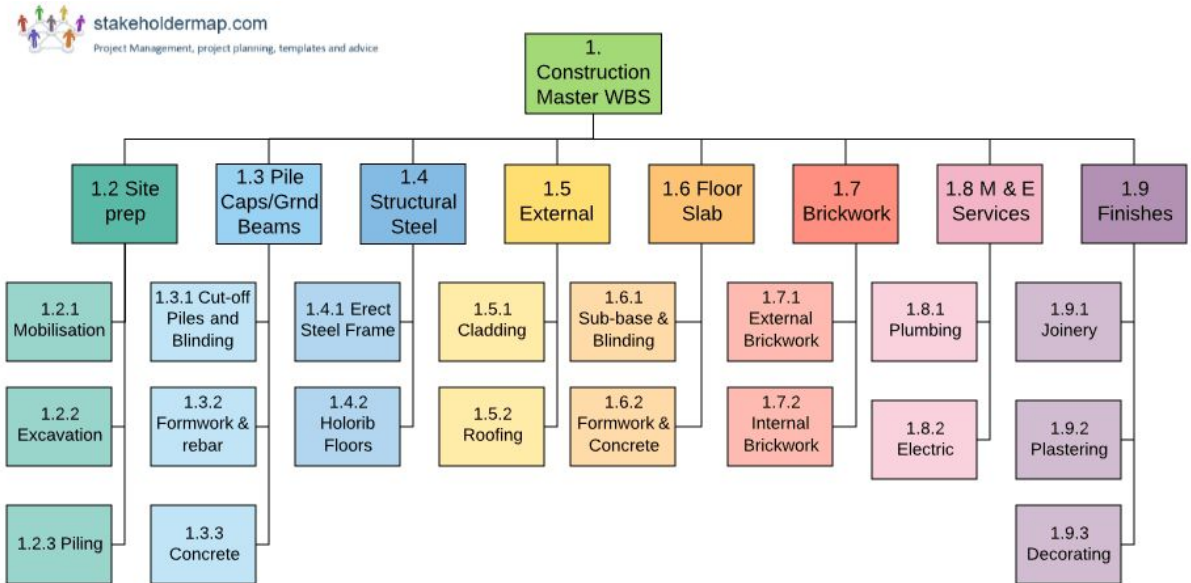
Kaplinger, T.E., Moore, V.S. and Nusbickel, W.L., International Business Machines Corp, 2019. Mobilizing an existing web application. U.S. Patent 10,216,855.

Ofoegbu, T.C., David, A., Kelley, L.A., Mezulis, S., Islam, S.A., Mersmann, S.F., Strömich, L., Vakser, I.A., Houlston, R.S. and Sternberg, M.J., 2019. PhyreRisk: a dynamic web application to bridge genomics, proteomics and 3D structural data to guide interpretation of human genetic variants. *Journal of molecular biology*, 431(13), pp.2460-2466.

APPENDICES

APPENDIX A: USERS' MANUAL

Work breakdown structure



Gantt chart

Discount Information Sharing Application

Task Name	ID	Start	Finish	Duration	2011/1/31	2011/2/7	2011/2/14	2011/2/21	2011/2/28	2011/3/7	2011/3/14	2011/3/21	2011/3/28	2011/4/4	2011/4/11	2011/4/18	2011/4/25	2011/5/2	2011/5/9	2011/5/16		
Welcome and Project Brief	1	2/1/11	2/8/11	1 Week	▶																	
Analysis of existing similar tools	2	2/7/11	3/7/11	4 Weeks		▶																
Related academic work	3	2/7/11	3/7/11	4 Weeks		▶																
Links to related news items in the tech media	4	2/7/11	3/7/11	4 Weeks		▶																
Interview with users or focus groups	5	2/7/11	2/14/11	1 Week		▶																
Main features of our application	6	2/7/11	3/7/11	4 Weeks		▶																
Mockups and ideas	7	2/28/11	3/7/11	1 Week					▶													
Scenarios and personas	8	2/28/11	3/7/11	1 Week					▶													
Web maps and storyboards	9	2/28/11	3/7/11	1 Week					▶													
UML diagrams	10	3/7/11	3/14/11	1 Week						▶												
Overview of standards and protocols	11	3/7/11	3/14/11	1 Week						▶												
Tech demos and code snippets	12	3/14/11	4/4/11	3 Weeks							▶											

1.8KB/S 78.7KB/S

APPENDIX B: SYSTEM TECHNICAL DOCUMENTATION

Mobile Web Requirements

Created by Mitch Davis, last modified just a moment ago

Target release	1.0
Epic	MDT-18 - Mobile optimized web app TO DO
Document status	DRAFT
Document owner	@Mitch Davis
Designer	@Cassie Owens
Developers	@Harvey Jennings
QA	@Kevin Campbell

Background and strategic fit

We all know mobile is on the rise. A [recent survey](#) to customers showed that 85% of users use their mobile on a daily basis. Most of our customers also use competitor apps, so this is something we need to have.

Customer research

- [Customer interview - Netflix](#)
- [Customer interview - Homeaway](#)
- [Customer interview - Bitbucket](#)

Requirements

#	User story title	User story description	Priority	Notes
1	Facebook Integration MDT-13 TO DO	A user wants to sign up via Facebook	Must Have	<ul style="list-style-type: none"> • We will need to talk to Cassie Owens. • There has also been some research done on this (see Facebook integration prototype)
2	Activity Stream MDT-14 TO DO	A user wants to view the latest updates via the mobile dashboard so that they can get a better understanding of what is in place	Must Have	
3	Post Updates MDT-15 TO DO	A user wants to be able to post status updates on the go	Must Have	<p>The key things we will need to support:</p> <ul style="list-style-type: none"> • Text status updates • Mentions • Support for images • Smart embedding for YouTube vids
4	API MDT-16 TO DO	A developer wants to integrate with the mobile app so that they can embed the activity stream on their website	Should Have	<ul style="list-style-type: none"> • We should chat to Team Dyno as they did something similar.

APPENDIX C: AUTO-COMPLETE FIELDS & HYPERLINKED EMAIL ADDRESS

Sales Reps Look up Tool

06101 or Hartford

80

80201

35801

32801

96801

Sales Reps Look up Tool

06101 or Hartford

I

Beverly Hills

Los Angeles

Little Rock

Huntsville

Pensacola

Orlando

Sales Reps Look up Tool

Select zip code or Select City

First Name	No	Email	City	Zip Code
Brit	7-3101	bdeverehunt2@pcworld.com	Hartford	06101

Select zip code

06101

80201

90209

90001

04302

APPENDIX D: DATABASE SCHEMA

