PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
UNIVERSITY MOHAMMED SEDDIK BENYAHIA
JIJEL

Faculty of Exact Sciences and Computer Science
Department of Computer Science

MASTER THESIS

Submitted to the Department of Computer Science in Partial Fulfillment of the Requirements for the Degree of Master

Option : Computer System and Decision Support

Topic

The Application of SaaS-based Cloud Computing In An Online Collaboration Platform For Students

Presented by :

Mr. Ismail BENLAREDJ

Supervised by :

Mr. Tarek BOUTEFARA

Acknowledgment

Above all, I thank my Almighty God, who gave me strength, faith, health, will, and guidance to accomplish this modest work..

I wish particularly to acknowledge my Supervisor ***Mr. Tarek Boutefara*** *for having supervised, helped, guided, trust, and encouraged me throughout my work. Thanks to his instructions, the work has been adequately done.*

I would also like to thank the jury members who have agreed to review and participate in the jury of my work.

I take this opportunity to express gratitude to our teachers of the computer science department at the University of Jijel, who have ensured my formation during my university cycle.

I would like to extend my sincere thanks to all those who have contributed in one way or another to the realization of this thesis.

My deepest gratitude goes to my family, who have been able to approach me without letting go of the support during all these long years of study.

To all these contributors, I offer my thanks, respect, appreciation, and admiration.

Abstract

In recent years, there has been a growing interest in Cloud Computing and its services, with the advance and the development of related technologies, Cloud Computing has been widely adopted in many sectors. In this work we present The Cloud Computing fundamentals, we focus on one of the principal components of Cloud Computing which is Software as a Service (SaaS). SaaS has received significant attention among cloud technologies, it is a way of delivering complete software over the internet as a service, instead of installing and maintaining software, the users simply access it via the internet using a browser. Today modern education is highly dependent on new technologies that help to improve educational attainment. A key strategy for students to achieve a high-quality learning experience is online collaboration. Online collaboration refers to the use of computer tools to work with others online. In this thesis, we discuss online collaboration for students, and we propose a SaaS-based application that helps students collaborate online. This application is developed and deployed in a Cloud Computing environment.

Keywords : Cloud Computing, Software as a Service, SaaS, Online Collaboration for Student, Education, Web Application, Software Engineering.

ملخص

في السنوات الأخيرة، كان هناك اهتمام متزايد بالحوسبة السحابية وخدماتها ، مع تقدم وتطور التقنيات ذات الصلة ، تم اعتماد الحوسبة السحابية على نطاق واسع في العديد من القطاعات. في هذا العمل ، نقدم أساسيات الحوسبة السحابية ، ونركز على أحد المكونات الرئيسية للحوسبة السحابية وهو البرمجيات كخدمة Software as a Service . لقد حظيت (SaaS) باهتمام كبير بين التقنيات السحابية ، فهي طريقة لتقديم برامج كاملة عبر الإنترنت كخدمة ، بدلاً من تثبيت البرامج وصيانتها ، يقوم المستخدمون بالوصول إليها ببساطة عبر الإنترنت باستخدام متصفح.

يعتمد التعليم الحديث اليوم بشكل كبير على التقنيات الجديدة التي تساعد على تحسين التحصيل العلمي. من الإستراتيجيات الرئيسية للطلاب لتحقيق تجربة تعليمية عالية الجودة هي التعاون و العمل عبر الإنترنت. يشير التعاون عبر الإنترنت إلى استخدام أدوات الكمبيوتر للعمل مع الآخرين عبر الإنترنت. في هذه المذكرة ، نناقش التعاون عبر الإنترنت للطلاب ، ونقترح تطبيقًا قائمًا على (SaaS) يساعد الطلاب على التعاون عبر الإنترنت. تم تطوير هذا التطبيق ونشره في بيئة الحوسبة السحابية.

كلمات مفتاحية : الحوسبة السحابية ، البرمجيات كخدمة، Software as a Service ، التعاون عبر الانترنت، تطبيقات الويب

Résumé

Ces dernières années, l'intérêt pour le Cloud Computing et ses services s'est accru. Avec l'avancée et le développement des technologies associées, le Cloud Computing a été largement adopté dans de nombreux secteurs. Dans ce travail, nous présentons les fondements du Cloud Computing, nous nous concentrons sur l'un des principaux composants du Cloud Computing qui est le Software as a Service (SaaS). Le SaaS a fait l'objet d'une attention particulière parmi les technologies du Cloud Computing. Il s'agit d'une manière de fournir un logiciel complet sur Internet en tant que service, au lieu d'installer et de maintenir le logiciel, les utilisateurs y accèdent simplement via Internet en utilisant un navigateur. Aujourd'hui, l'éducation moderne est fortement tributaire des nouvelles technologies qui contribuent à améliorer le niveau d'instruction. La collaboration en ligne est une stratégie clé permettant aux étudiants de vivre une expérience d'apprentissage de haute qualité. La collaboration en ligne fait référence à l'utilisation d'outils informatiques pour travailler avec d'autres personnes en ligne. Dans cette thèse, nous discutons de la collaboration en ligne pour les étudiants, et nous proposons une application basée sur SaaS qui aide les étudiants à collaborer en ligne. Cette application est développée et déployée dans un environnement de Cloud Computing.

Mots clés : Cloud Computing, Software as a Service, SaaS, Collaboration en ligne pour les étudiants, Education, Web Application Collaboration en ligne pour les étudiants, Génie logiciel.

Contents

Introduction 12

Objectives 14

Thesis Structure 14

1 Cloud Computing Fundamentals **15**

1.1 Overview 15

1.2 Introduction 15

1.3 Definitions of Cloud Computing 16

1.4 Brief History 17

1.5 Key Components 19

1.5.1 Infrastructure 19

1.5.2 Virtualization 20

1.5.3 Internet and Web 2.0 22

1.5.4 Web Services and SOA/SOC 23

1.5.5 Open-Source Software 24

1.6 Characteristics of Cloud Computing 25

1.6.1 On-demand self-service: 25

1.6.2 Broad network access: 25

1.6.3 Resource pooling: 25

1.6.4 Rapid elasticity: 26

1.6.5 Measured service: 26

1.7 Cloud Computing Architecture 26

1.7.1 Basic Architecture 26

1.7.2 Cloud Computing Service Models 28
1.7.3 Cloud Computing Deployment Models 30
1.7.4 Cloud Computing Roles 31
1.8 Benefit of Cloud Computing 34
1.9 Cloud Challenges 35
1.10 Cloud Computing Platforms 37
1.10.1 Amazon Web Services AWS 37
1.10.2 Microsoft Azure 37
1.10.3 Google Cloud Platform GCP 38
1.11 Summary 38

2 Software as Service Model 39
2.1 Overview 39
2.2 Introduction 39
2.3 Architecture of SaaS Applications : 40
2.3.1 Level 1 – The Ad-Hoc/Custom 41
2.3.2 Level 2 – Configurable 41
2.3.3 Level 3 – Configurable Multi-Tenant-Efficient 41
2.3.4 Level 4 – Scalable, Configurable, and Multi-Tenant-Efficient . 42
2.4 Multi-Tenancy Architecture 42
2.4.1 Multi-Tenancy Concept 42
2.4.2 Achieving Multi-Tenancy 43
2.4.3 Configuration and Metadata-Driven 43
2.4.4 Database Modeling 45
2.5 SaaS Development and Deployment 47
2.5.1 Three-tier Model 47
2.5.2 Service Oriented Architecture 48
2.5.3 Microservices 48
2.5.4 Serverless 49
2.6 SaaS Applications Versus Similar Models 50
2.7 Summary 52

3 Online Collaboration For Students 53
3.1 Overview 53
3.2 Introduction 53
3.3 Definitions 54
3.3.1 Collaboration 54
3.3.2 Online Collaboration 54
3.4 Students Collaboration Principles 55
3.4.1 Working Together 55
3.4.2 Identifying a Group Goal 55
3.4.3 Equal Participation 55
3.4.4 Responsibility and interdependence 56
3.4.5 Communication 56
3.5 Factors of a Successful Online Collaboration 56
3.5.1 The Right Technology For The Task 57
3.5.2 Group Size 57
3.5.3 Individual accountability 58
3.6 Computer-Mediated Communication 58
3.6.1 Synchronous CMC 58
3.6.2 Asynchronous CMC 59
3.7 Cloud Collaboration 59
3.8 Different Types of Online Collaboration Tools 60
3.8.1 Early Generations Tools 60
3.8.2 Project Management Tools 61
3.8.3 Video Conferencing Tools 61
3.8.4 Social Media 62
3.9 Online Collaboration Platforms 62
3.9.1 Google Workspace for Education 62
3.9.2 Zoom 64
3.9.3 Slack 64
3.10 Advantages of Online Collaboration 65
3.11 Challenges and Issues 66

3.12 Summary 67

4 Planning and Analysis 68

4.1 Overview 68
4.2 Software Development Methodology 69
4.3 Development Phases 69
4.3.1 Analysis 69
4.3.2 Design 70
4.3.3 Implementation 70
4.4 Our Project 71
4.4.1 Background information 71
4.4.2 Purpose 71
4.4.3 Intended Audience 71
4.4.4 Scope of the Software 72
4.5 Our solution 72
4.5.1 Features 72
4.5.2 Basic Components 73
4.6 Software Requirements Specifications 74
4.6.1 Define Actors 74
4.6.2 Functional Requirements 74
4.6.3 Permissions and Rules 75
4.6.4 Use Cases 76
4.6.5 Use cases textual description 80
4.6.6 Non Functional Requirements 84
4.7 Summary 85

5 Design 86

5.1 Overview 86
5.2 Sequence diagram 86
5.2.1 Authentication 86
5.2.2 Guest 89

5.2.3 Edit text page . . . 90
5.2.4 Upload file . . . 91
5.2.5 Add member to a project . . . 92
5.2.6 Task management . . . 93
5.3 Class diagram . . . 93
5.4 Data Modeling . . . 95
5.5 Deployment architecture . . . 97

6 Implementation 99
6.1 Overview . . . 99
6.2 NoSQL Database . . . 99
6.3 Cloud Computing platform : . . . 100
6.3.1 Google App Engine . . . 101
6.3.2 Firestore . . . 101
6.3.3 Cloud storage . . . 102
6.3.4 Cloud Functions . . . 102
6.3.5 Identity platform . . . 102
6.4 Programming Languages and Libraries Used . . . 102
6.4.1 JavaScript . . . 102
6.4.2 Node js . . . 103
6.4.3 Express js . . . 103
6.4.4 React . . . 103
6.4.5 CSS . . . 103
6.4.6 Firebase Admin SDK . . . 104
6.4.7 Editor js . . . 104
6.5 Tools and extensions . . . 104
6.5.1 Google Cloud Console . . . 104
6.5.2 VS Code . . . 104
6.5.3 NPM . . . 105
6.5.4 Nodemon . . . 106
6.5.5 Webpack . . . 106

6.6 File Structure 107
6.6.1 Front-end files 107
6.6.2 Back-end 108
6.7 Code and Solutions 109
6.7.1 Authentication 109
6.7.2 Authorization 110
6.7.3 Files Access Authorization 111
6.8 Deployment 112
6.9 Application scalability 113
6.10 The Final application 114
6.11 Summary 121
Conclusion and Perspectives 122

List of Figures

1.1 NIST Cloud Computing definition 17
1.2 A hardware virtalized server hosting three virtual machines, each one running ditinct operating system 21
1.3 Basic Cloud Computing architecture 27
1.4 Cloud Computing Service Models and Control levels 29
1.5 The NIST Cloud Computing reference architecture 33

2.1 A multi-tenant application environment 44
2.2 Salesforce.com's metadata-driven architecture 45
2.3 This approach uses a different database for each tenant 46
2.4 In this approach each tenant has its own separate set of tables in a common database 46
2.5 In this approach each tenant has its own separate set of tables in a common database 47
2.6 3 three-tier architecture model 48
2.7 Sample Microservices architecture 49

4.1 Diagram Use case : Main application 76
4.2 Diagram Use case : Private space management 77
4.3 Diagram Use case : Project Managment 78
4.4 Diagram Use case : User in project space 79
4.5 Description table : Authenticate 80
4.6 Description table : Add memeber to a project 81

4.7 Description table : Edit text page 82
4.8 Description table : Task managment 83

5.1 sequence diagram: Sign up 87
5.2 sequence diagram:Sign in 88
5.3 sequence diagram: Guest visit a public page 89
5.4 sequence diagram:Edit text page 90
5.5 sequence diagram:Upload file 91
5.6 sequence diagram:Add member to a project 92
5.7 Sequence diagram: Task management 93
5.8 class diagram 94
5.9 classes 95
5.10 JSON DOC 1 96
5.11 JSON DOC 2 96
5.12 JSON DOC 3 97
5.13 Deployment architecture 98

6.1 Google Cloud Console 105
6.2 Front-end app files structure 107
6.3 Back-end app files structure 108
6.4 Authenticate Service 109
6.5 Part of the code of authenticate trigger function 110
6.6 Middleware source code 111
6.7 files access rules 112
6.8 screenshot from google cloud console shows app engine instances . . 113
6.9 Application screenshot : Sign up 115
6.10 Application screenshot : Sign in 116
6.11 Application screenshot : Main page 117
6.12 Application screenshot : Task table 118
6.13 Application screenshot : Todo page 119
6.14 Application screenshot : Calendar page and add event window 120

Introduction

Cloud Computing is an emerging technology trend, it's one of the most interesting topics talked about in the field of information technology, this new technology has known a massive growth and development during recent years. Cloud Computing is being transformed into a model consisting of services that are deliverable in a manner similar to utilities, such as water, electricity, and gas. Cloud Computing basically refers to the on-demand delivery of IT resources including, hardware, servers, storage, software, and IT infrastructure in general. All resources are available as services through the internet and offered on a pay-per-use basis.
Cloud Computing provides ways to design computing systems, develop applications, and leverage existing services for building software. It is based on the concept of "dynamic provisioning", which is applied not only to services but also to other information technology resources like infrastructure, compute capability, storage, networking.. etc[1].

Traditionally, software and applications are installed and used locally on the user's infrastructure, the user is responsible for providing the necessary equipment, manages and maintains it, updates and upgrades applications. However in recent years, the progress and the features that are presented by Cloud Computing are growing the interest in cloud technologies, this significant attention to Cloud Computing capabilities makes communities think about the way that they are dealing with software and information technology resources in general. Software as a Service (SaaS) is the principal component of Cloud Computing that has a direct effect on this change.
SaaS known as cloud-based software or cloud applications is application software that's hosted in the cloud and that we access and use via a web browser, a dedicated desktop client, or an API (Application Programming Interface).

Today, although most people don't recognize it, SaaS applications are more commonly used than they think and are probably the most used type of cloud services. SaaS is everywhere, web-based email services such as Gmail, Hotmail or text editors like Word, Google, file hosting services like Dropbox, entertainment platforms like Netflix... etc. All these tools are SaaS applications.

The capabilities offered by the SaaS model and Cloud Computing in general, led to massive development in almost all fields. One of those fields is education and learning methods.
Education is becoming increasingly collaborative, today teachers and students can skip time and space constraints of learning by moving an important part of the process to online collaboration. Online Collaboration or Digital collaboration is a group of two or more individuals work together, to learn or to achieve a goal from distance using digital technologies and the Internet, students and instructors can communicate, share documents, solve problems and work together on their projects from anywhere at any time.
Online collaboration is only partially enabled by the underlying concept of Cloud Computing [2], online collaboration tools must have some characteristics and specific capabilities to enable teams to work comfortably and effectively, capabilities like availability, less downtime, high performance, and unlimited scale. Since online collaboration tools are web-based applications, there is no better way than Cloud Computing to develop and host applications, Cloud Computing is the best choice to provide all functionalities and capabilities to fulfill all needs of any high-quality application.

Objectives

Our objectives in this project are defined in the following points :

- Study the Cloud Computing technology and get to know the basics of it services, it delivery and deployment models. By the end of this study we should be familiar with the basics of Cloud Computing concept
- Study the Software as a Service model, understand it architectures, characteristics and get know how to develop and deploy a SaaS application
- The final objective is to use Cloud Computing technologies and services to build a SaaS-based application that helps students collaborate and work together online.

Thesis Structure

This thesis consists of 6 chapters, we start by Cloud Computing Fundamentals chapter, in this chapter we will present the concept of Cloud Computing and it building blocks, like architecture, models, key enabling components.. etc
In the chapter 2 we will discuss Software as a Service (SaaS) model, we will talk about SaaS application architectures, developments and deployments models.
Next, in Online Collaboration for Students chapter, we will defined online collaboration for students, present some principles and factors that play an important role to an effective and success collaboration among student and explain mode of online communication also we will introduce popular tools,some platforms and applications used to collaborate online.
In the 4th and 5th chapters we will walk through the process of planning, analyse and design our SaaS application.
In the last chapter we will show the programming process, how we develop and deploy the application, technologies used and some coding solutions. In the end we will present the final application.

Chapter 1

Cloud Computing Fundamentals

1.1 Overview

In this chapter, we'll present the concept of Cloud Computing with its different definitions, we'll explain when and why the cloud concept was created through a brief history section, next we'll introduce Cloud Computing architecture with different deployment and delivery models, it characteristics, and it actor's roles, we also discuss the key enabling components and technologies, in the other side we'll address barriers challenges and issues of Cloud Computing, and at the end of this chapter we present the big three cloud platforms.

1.2 Introduction

Cloud Computing is an emerging technology trend, it's one of the most interesting topics talked about in the field of information technology, this new technology has known a massive growth and development during recent years. Like any utilities such as water, electricity, and gas. Cloud Computing basically refers to the on-demand delivery of IT resources including, hardware, servers, storage, software, and IT infrastructure in general. All resources are available as services through the internet and offered on pay-per-use basis.

Today, customers use Cloud Computing resources from anywhere at any time with the quantity needed to run applications, store data, and develop software, without having to think about the maintenance, the management, and the investment in these resources.

1.3 Definitions of Cloud Computing

Cloud Computing has several definitions based on the mode of implementation, various actors and academic spheres try to define what really Cloud Computing is, and clarify its features and characteristics[3]. Luis M. Vaquero el al said that the term of cloud is continually changing and they Proposed a Definition : “Cloud is a parallel and distributed computing system consisting of a collection of inter-connected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on service-level agreements (SLA) established through negotiation between the service provider and consumers” [4].

Buyya et al defined it as follows:
“Cloud is a parallel and distributed computing system consisting of a collection of inter-connected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on service-level agreements (SLA) established through negotiation between the service provider and consumers.”[5]

The most acceptable and widely cited definition is that given by National Institute of Standards and Technology. According to NIST : “Cloud Computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.” [6] As shown in Figure 1.1, NIST determined the structure of this cloud model and defined it as a model compose of five essential characteristics, three models of service delivery, and four main deployment models.

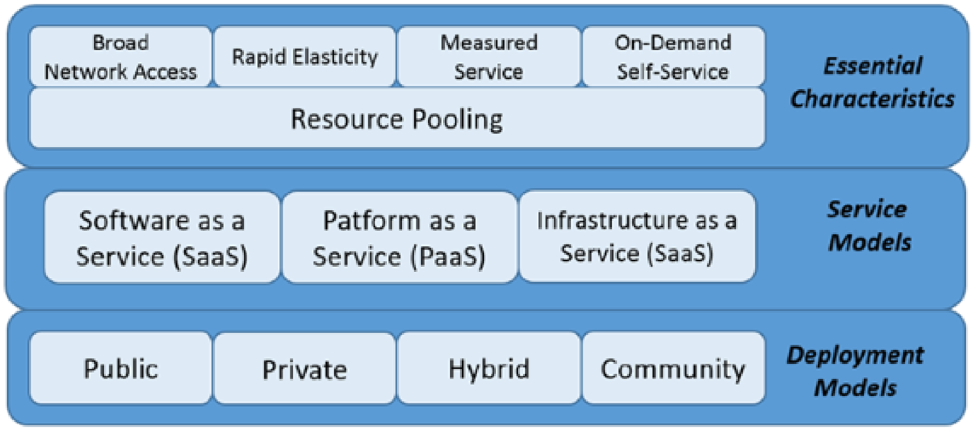


Figure 1.1 – NIST Cloud Computing definition

1.4 Brief History

Even though Cloud Computing is a pretty new technology, the concept of the cloud was introduced way back in the 1960s by John McCarthy [7], an American computer scientist (He coined the term "artificial intelligence" (AI)). In those days computing was a very hard task, time and money consuming. In 1959 John McCarthy was visiting the Massachusetts Institute of Technology, at that time MIT had an IBM 704 mainframe computer, submit input and wait for all day to get the results of computation, so McCarthy came up with the idea of time-shared computing resources, he wrote a note to the Director of the Computer Centre suggesting, expressing the need to the time-share computers, and the idea was revolutionary. [8] In a talk in 1961 (given to celebrate MIT's centennial) McCarthy suggest that computer time-sharing technology might result in a future in which computing power and even specific applications could be sold through the utility business model (like water or electricity). He said : "If the computers of the kind I have advocated become computers of the future then computers someday will be organized as a public utility just as a telephone system is a public utility. The computer utility could become the basis of a new and important industry." [9]

That was perhaps the first appearance of the concept of utility computing (Utility computing merely means "Pay and Use") and it was a fashion idea at that time, even though infrastructure and networks of the time were not developed enough for that challenge.
In 1966 in book titled "The Challenge of the Computer Utility" the author **Douglas Parkhill** set the modern-day characteristics of concept of Cloud Computing for the first time, characteristics like elasticity through a utility computing. [7]
In 1969, **Leonard Kleinrock** a chief scientist of the Advanced Research Projects Agency Network (ARPANET), project that seeded the Internet, talk about computer utilities, he said that it is the Directions of the future : “As of now, computer networks are still in their infancy, but as they grow up and become sophisticated, we will probably see the spread of ‘computer utilities”. [10] By the 1970s, the development of Virtualization makes great progress in computer utility, at that time the main problem that faces researchers and organizations is the high costs of computers, they could not justify the high investment in hardware that was used only on specific computing needs and during intermittent times. The solution to these concerns came up with the idea of "rent time", taking advantage of Virtualization, this solution allows organizations and users to rent computing resources with much lower subscription prices, instead of invest in very high-cost computers, and hardware providers can benefit form renting proceeds. [9]

The term "cloud" appear a few decades after, in the 1990s from the telecommunications world, the beginning of using virtual private network (VPN) services for data communication, bandwidth save and balanced utilization across the network, providers called it ’telecom cloud’, the similarity between telecom cloud and the concept of Cloud Computing led to release this term.[11]

The rapid developments of the internet and information technology at this period, led to more progress in productivity and technological innovations, especially the web, the emergence of cloud companies began in 1999, a company called **Salesforce** released one of the first real Cloud Computing implementations, they provided enterprise Cloud Computing services and began delivery customer relationship management software through a web site over the internet on a subscription basis. This successful use of Cloud Computing was the big step toward a new industry and establish the idea of delivering IT resources to the end-users, using the internet[11] [12] [9].

In 2002 Amazon web services start to provide some computing resources including storage and servers[11].

In 2006 Amazon web services began offering IT infrastructure services to businesses and organizations as services through the internet. AWS released Elastic Compute Cloud (EC2) for small companies and individuals[13]. In the same year, Google launched the Google Docs services, which include Web applications such as Gmail, Docs, and Calendar with 2GB space storage.

1.5 Key Components

Many key components that play a major role in the evolution of Cloud Computing, the innovations and the exploitation of technical components such as Virtualization, high–performance networks, and data center (hardware), enabled Cloud Computing to emerge and led IT to a new era [14].

1.5.1 Infrastructure

Cloud Computing is a group of data centers distributed across the world, each data center host hundreds of thousands of servers, connected and networked via hundreds of switches/routers, these servers consist of multiple high performance CPUs, hard disks, networks, etc... The accomplishment of modern-day Cloud Computing characteristics (see next section), requires high-quality hardware with less probability of failures, with best network systems and management models[15]. So that Cloud Computing requires the most advance and sophisticated technologies. Obviously, the infrastructure is the basis of the Cloud Computing foundation, so there is no doubt that the progress of Cloud Computing depends on the development of infrastructures and related technologies.

1.5.2 Virtualization

In computing, Virtualization is the creation of a virtual (rather than actual) instance or version of something including virtual hardware, network resources, operating system, and storage devices. In simple words, Virtualization is a computer program, which allows to use one physical computing resource and share it with multiple users at the same time, there are several types of Virtualization: Server Virtualization, Network Virtualization, hardware Virtualization... [16][17].

As we mentioned in the brief history section, Virtualization is an old technology, it began in the 1960s and 1970s, as a method of logically dividing computing resources of mainframe computers.

Virtualization helps to share the same hardware with multiple consumers without the intervention of the provider even with the new consumers. Virtualization enable the user to creates a virtual space known as virtual machine see Figure 1.2

A virtual machine is a digital version of a physical computer created by a software called a hypervisor, VM can perform exactly like a physical computer with almost all of its functions, like running operating systems, install applications store data, and connect to networks. Multiple VMs can run on a single physical machine, computing resources of the host machine (hardware that runs VMs) are accessible and distributed among VMs as needed.

1.5.2.1 Virtualization in Cloud Computing :

Virtualization plays a major role in Cloud Computing, it is the solution to many problems, with Virtualization we can gain many things:

— **Isolation and Multi-tenancy:** In cloud, customers use the same infrastructure but one user data should be isolated from other users, Virtualization helps achieve this requirement easily.

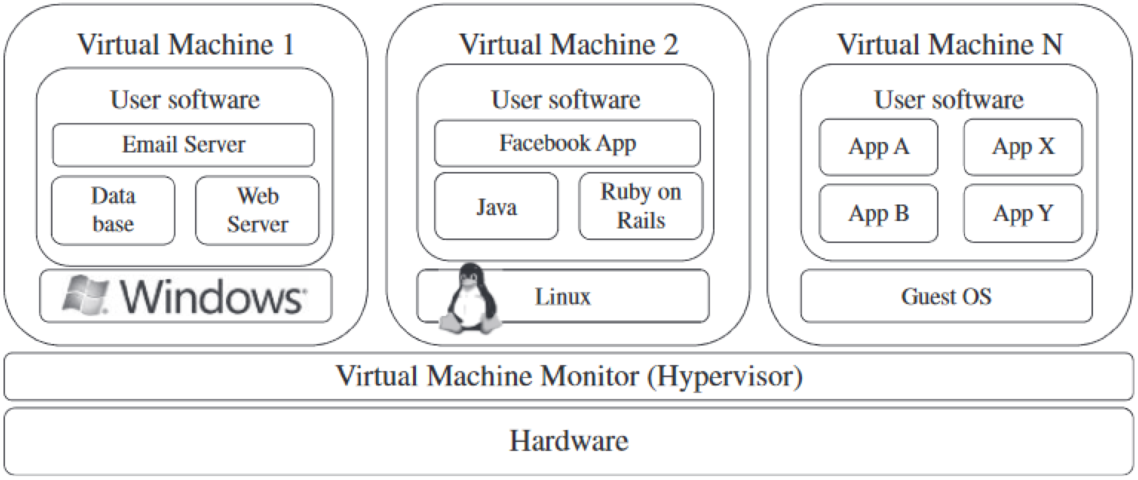


Figure 1.2 – A hardware virtalized server hosting three virtual machines, each one running ditinct operating system

- **Resource sharing :** Cloud Computing is a group of data centers, which means massive resources and a big number of servers, Virtualization helps to manage hardware and share it with multiple users by dividing it into smaller virtual resources.
- **Scalability:** Its helps cloud to scale[18].
- **Less complexity:** Virtualization allows multiple VMs to run on a single physical machine and therefore allows to run multiple OS, applications, share computing and storage with multiple customers, without extra complexity or more management effort[9].
- **Economical:** Virtualization is the best technique for cost-saving and hardware-reducing, providers do not need to set up new hardware for each new client instead they can re-allocating resources very easily, also it is eco-friendly and less energy-consuming.

1.5.3 Internet and Web 2.0

Internet is a main component of Cloud Computing, it contributed to the emergence of Cloud Computing. The evolution of the internet and the growth of the number of its users open new opportunities, creates new markets, and makes Cloud Computing much easier to access it and more ways to use its services (different connected devices: phones, tab... etc). As the result, increase the number of customers, growth in financial returns, and expansion of the Cloud Computing market, which led to putting more investment and effort in the field of Cloud Computing to increase productivity, create more complex services and develop applications... etc
All these massive benefits and progress in Cloud Computing, would not happen without a rich platform that facilitates interaction and brings dynamism and flexibility into Web pages.

Web or specifically Web 2.0 is considered as the primary interface through which Cloud Computing delivers its services [1]. It's a complementary of the internet to more progress in cloud evolution.

Web 2.0 is a term coined by Tim O'Reilly [19]. It refers to Web design resulting in an interactive transport mechanism, rather than conventional static screens. Web 2.0 is viewed as a platform for running software applications instead of install and running them on-promise (desktop PCs, phones..).

Web 2.0 applications are extremely dynamic and they have almost the same functionality as desktop applications. These capabilities are obtained by integrating a collection of standards and technologies, such as XML, Asynchronous JavaScript, Web Services, and others, Which makes complex cloud applications run on the front end part smoothly and flexible [1] [12].

1.5.4 Web Services and SOA/SOC

Web services play an important role in the concept of Cloud Computing, which is based on delivering IT resources in form of services through networks, web services help Cloud Computing on better communication and data transactions between providers and different consumers using XML-based messages.
According to W3C (World Wide Web Consortium), a web service is a software system designed to support interoperable, machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP-messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.

When we talk about web services on Cloud Computing we should talk about Service-Oriented Architecture (SOA) and Service-Oriented Computing (SOC).
SOA is an architectural style that supports service orientation, is a way of thinking in terms of services and service-based development and the outcomes of services, it provides an architectural style for building software applications, that promotes loose coupling between components. [20]

SOC is a paradigm that refers to the set of concepts, principles, and methodologies that represent computing in Service-Oriented Architecture (SOA), in which software applications are designed and developed based on services as fundamental elements and with standard interfaces, in other words, SOC and Cloud Computing will coexist, complement, and support each other.
SOC constitute a fundamental building block for Cloud Computing systems and applications since the main objective of Cloud Computing is to deliver everything as a service, from infrastructure to applications, SOA is a powerful approach because it encompasses design principles to structure, compose, and deploy software systems in terms of services[1].

1.5.5 Open-Source Software

Open Source is a term that refers to computer software, algorithms, and programs that are released under a license, which give the right to anyone for any purpose to use, access, change, edit and distribute the source code[21].
Open-source has an enormous impact in the field of Cloud Computing, which is based on an important number of free license software like **Linux operating system** that has become a major building block of largest cloud environments [12]. **Xen** that it's using in Cloud Computing for a full Virtualization (completely emulate all hardware devices to the virtual machines) [22].
Another popular open-source software called Eucalyptus, it used to implements IaaS-style Cloud Computing using the Linux-based infrastructure, found in many modern data centers [19].

The use of open-source software in Cloud Computing can provides numerous benefits, the following some of it [12] :

- **Cost savings :**providers use open-source software for free without have to pay high prices for software licenses or upgrades fees.
- **Modify and customize :** the ability to customize software's source code according to personal and specific needs
- **Time saving**: open-source provides ready solutions, no one needs to start from scratch.

1.6 Characteristics of Cloud Computing

There are numerous essential characteristics of Cloud Computing, that defined the outlines of how Cloud Computing should be and its basic features, which require to be available on each cloud platform. To get a common understanding and standards of Cloud Computing many groups and organizations try to set important characteristics of cloud, as we mentioned before NIST defined Cloud Computing with five essential characteristics, following the five essential characteristics of Cloud Computing defined by NIST [6]:

1.6.1 On-demand self-service:

User should be able to provision computing resources including CUP time, storage, network access, server time, applications... etc, automatically as needed without human influence with each service provider.

1.6.2 Broad network access:

Computing resources are available online over standard networks as services, user can access, use it from anywhere at any time through heterogeneous devices (smart phone, laptops, workstations.. ect)

1.6.3 Resource pooling:

Use a multi-tenant model to serve multiple consumers from one pool of computing resources include storage, processing, memory, and network bandwidth. which allow customers to use the same computing resources separately on the logical level, physical and virtual resources can be dynamically assigned and reassigned according to user demand, generally the customer has no control or knowledge over the exact location of the provided resources, but they may be able to choose a specific location by country, state or data-center.

1.6.4 Rapid elasticity:

Computing resources can be elastically provisioned and released, in some cases automatically, and fast scaling up and down depending on demand, therefore, consumer considers that the capabilities are unlimited and available at any time with any quantity needed.

1.6.5 Measured service:

Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service.

1.7 Cloud Computing Architecture

1.7.1 Basic Architecture

Cloud Computing architecture refers to the components and sub-components required for Cloud Computing. The most basic architecture of Cloud Computing components is divided into the following two parts: front end and back end, both are connected via the internet(Figure 1.3). [7] [23].

Front-end: is the side of the end-user, it presents all components and user interfaces that the client sees and interacts with, like button, media, content...etc. Client access those interfaces using mobile or computer devices. User from front-end part interacts with back end via applications and protocols, commonly web browsers.

Back-end : is the server-side, it refers to data storage, hardware management, respond to the requests, load balance, application, and software source code, all important computing happen on this part, end users can not see component of the back end they just can interact with from front end part.

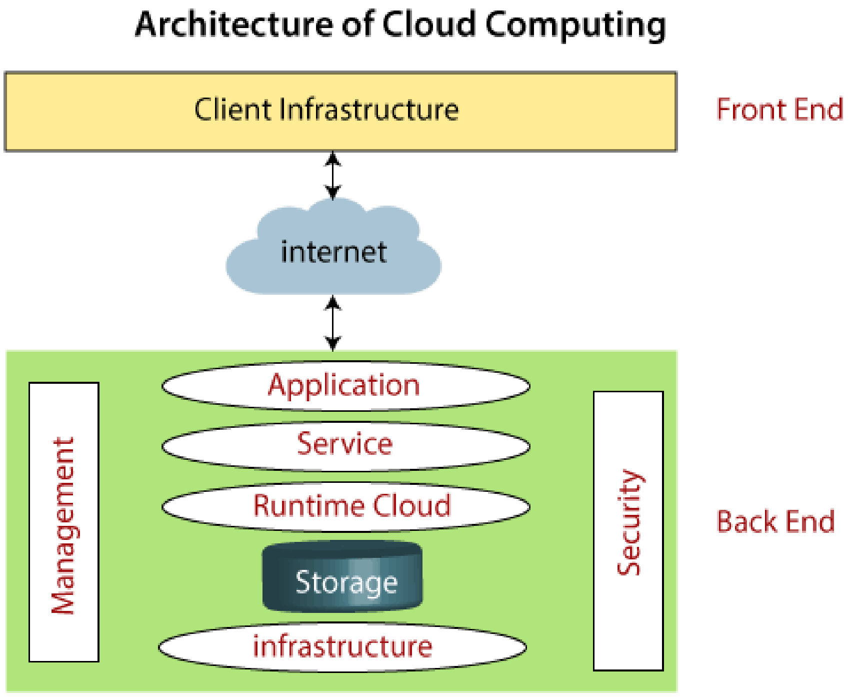


Figure 1.3 – Basic Cloud Computing architecture

1.7.2 Cloud Computing Service Models

Cloud provides several services, according to NIST there is tree main types of cloud services [6]:

1.7.2.1 Infrastructure-As-A-Service (IaaS)

Infrastructure as a Service (IaaS) is the lower layer of this service model. IaaS referred to everything as a service, basic computing resources like servers, storage, networking resources, and Virtualization technology are provisioned to consumers in a virtualized environment where they do not have to manage or control the underlying resources [24]. Instead, consumers have full control over operating systems, runtime, middleware, and deployed applications, they can freely deploy, run software and OS needed. the famous example of IaaS: Amazon EC2.

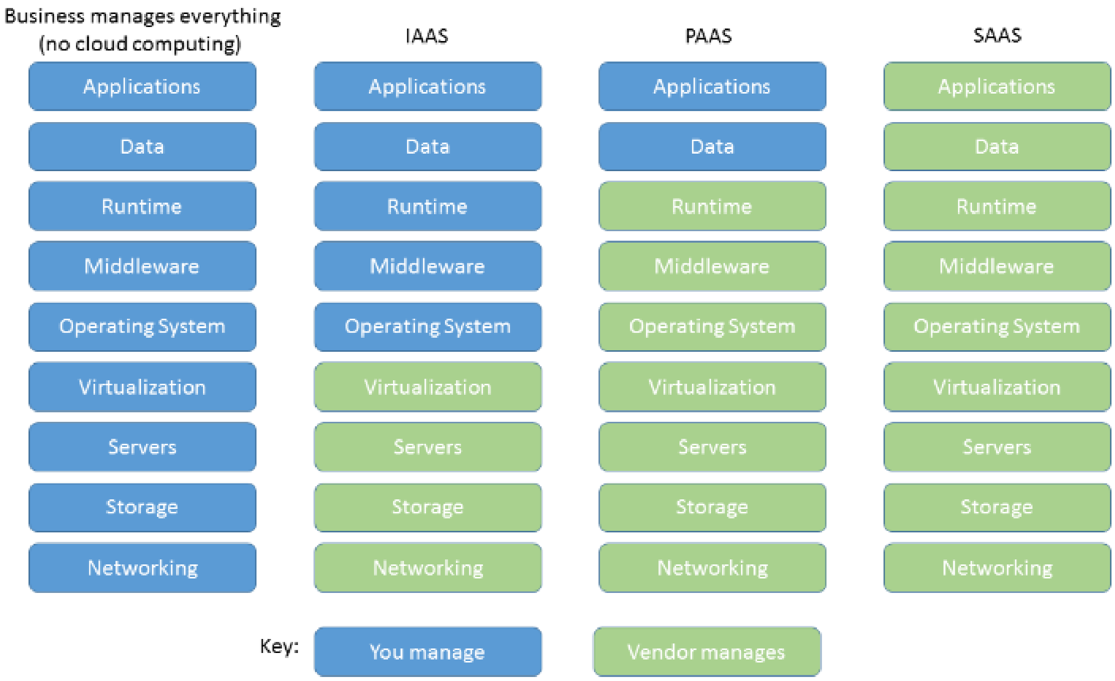
1.7.2.2 Platform as a Service (PaaS)

Platform as a service (PaaS) is the layer above IaaS, which means less control over IT resources, using IaaS, PaaS provides a computing platform as a service [7], the consumer (usually a developer) use PaaS to develop, test, deploy their cloud services and applications (e.g. SaaS applications) [25], PaaS provides everything required for the life cycle of software and Web applications like (developer tools, frameworks, programming languages, testing environments.. etc) [26]. Consumers do not manage and control underlying resources include: servers, storage, OS, runtime, build tools.. etc), they focus only on the development process of their solutions, but they can control the deployment of applications and the configuration settings of the platform in general, example of PaaS: Google App Engine, Heroku...

1.7.2.3 Software as a Service (SaaS)

Software as a Service(SaaS) is the top layer of the cloud model and is the most used service model of Cloud Computing. Almost everyone has already used a SaaS-based application at some point [26]. SaaS refers to the final application that is

delivered to the end-user, on the opposite of On-premises software where users need to install software locally, update it manually, maintain and manage the hardware, SaaS is a full ready software hosted and running on cloud infrastructure, users can access this software from multiple client devices through either a thin client interface, such as a web browser or a program interface, without the need to manage and control any cloud resources or even maintain and update the software.
SaaS is built on the multi-tenant architecture that makes users share the same single application, with a separation between them on the logical level [27], example of SaaS: web-based email, Google Doc, Microsoft 360...



Business manages everything (no cloud computing)	IAAS	PAAS	SAAS
🟦 Applications	🟦 Applications	🟦 Applications	🟩 Applications
🟦 Data	🟦 Data	🟦 Data	🟩 Data
🟦 Runtime	🟦 Runtime	🟩 Runtime	🟩 Runtime
🟦 Middleware	🟦 Middleware	🟩 Middleware	🟩 Middleware
🟦 Operating System	🟦 Operating System	🟩 Operating System	🟩 Operating System
🟦 Virtualization	🟩 Virtualization	🟩 Virtualization	🟩 Virtualization
🟦 Servers	🟩 Servers	🟩 Servers	🟩 Servers
🟦 Storage	🟩 Storage	🟩 Storage	🟩 Storage
🟦 Networking	🟩 Networking	🟩 Networking	🟩 Networking

Key: 🟦 You manage 🟩 Vendor manages

Figure 1.4 – Cloud Computing Service Models and Control levels

1.7.3 Cloud Computing Deployment Models

According to NIST there are four main cloud deployment models [6]:

1.7.3.1 Private Cloud

Private cloud known as internal cloud refers to internal data centers intended for a single organization for just local use, which not made available to the general public [28]. Private cloud build and managed by the organization itself or by an external provider or a combination of both, it may be on or off-premise, a private cloud provides full control of resources, high performance, and high security [29].

1.7.3.2 Community Cloud

A community cloud is a private cloud but instead of one organization, its provisioned for a specific group or community of consumers from multiple organizations and companies that have common interests (e.g. mission, security requirements, policy, and compliance considerations). It's not made available for users outside the community, community cloud build and manage in a partnership between group members, by an external provider, or a combination of both, it may be online or off promise.

1.7.3.3 Hybrid cloud

A hybrid cloud is a combination of two or more clouds with different models (private, community, or public), which remain unique entities bound together by standardized or proprietary technology, in such a way to improve information, data, and application portability between these entities (e.g., cloud bursting for load-balancing between clouds).
Hybrid clouds offer many benefits it has more flexibility than one cloud model alone, for example, it enables the user to use a public cloud for massive computing resources and stores sensitive data on a private cloud.

1.7.3.4 Public Cloud

The cloud infrastructure is provisioned to the general public as services, on a commercial basis (pay-as-you-go model) [28], everyone can use public cloud from anywhere using the internet, It may be owned, managed, and operated by a business, academic, or government organization, or some combination of them, public cloud available on the promise of the cloud provider.

1.7.4 Cloud Computing Roles

Through the growth of Cloud Computing and the diversity of its complex services, more opportunities are added to integrate more individuals and actors, which gave rise to various new roles and responsibilities, NIST propose a Conceptual Reference Model called, "the NIST Cloud Computing reference architecture" which identifies the major actors, their activities and functions in Cloud Computing, Figure 1.5 shows that NIST defines five major roles: cloud consumer, cloud provider, cloud carrier, cloud auditor and cloud broker.

1.7.4.1 Cloud Consumer

A Cloud Consumer is a person or entity that maintains a business relationship with service from Cloud Providers, Cloud Consumer use provided cloud resources after sets up a contract or arrangement with the providers, to allow him to purchase, exploit, or rent cloud services. Cloud Consumer is considered as the main stakeholder for Cloud Computing service.

Cloud consumer activities and usage scenarios differ depending on the services. it can be different among cloud consumers. For example, SaaS consumer can be an organization that provides their employees access to SaaS-based applications(ex: ERP software), so employees are considered as the end-users of the application but their activities differ depending on their responsibilities on the organization (ex: administrator, seller..).

Another example of the difference among cloud consumers, SaaS consumer may be a person and generally is the final user, he just consumes cloud resources, but in the other side PaaS consumer generally is a developer use PaaS to build and deploy a SaaS application, which makes him a cloud PaaS consumer and cloud SaaS provider at the same time, though they both cloud consumers but the services consumed makes their activities difference.

1.7.4.2 Cloud Provider

Cloud provider is a person or an organization that offer IT resources as services to interested entities. Cloud provider is responsible for the infrastructure management, maintain and runs the cloud software, cloud security and makes arrangement to deliver the cloud services to the Cloud Consumers through network.

1.7.4.3 Cloud Auditor

A cloud auditor is a party that can perform an independent examination of cloud service, controls with the intent to express an opinion thereon. Audits are performed to verify conformance to standards through a review of objective evidence. A cloud auditor can evaluate the services provided by a cloud provider in terms of security controls, privacy impact, performance... etc

1.7.4.4 Cloud Broker

As Cloud Computing evolves, the integration of cloud services can be too complex for cloud consumers to manage. A cloud consumer may request cloud services from a cloud broker, instead of contacting a cloud provider directly. A cloud broker is an entity that manages the use, performance and delivery of cloud services and negotiates relationships between cloud providers and cloud consumers.

1.7.4.5 Cloud Carrier

A cloud carrier acts as an intermediary that provides connectivity and transport of cloud services between cloud consumers and cloud providers. Cloud carriers provide access to consumers through networks, telecommunication, and other access devices. For example, cloud consumers can obtain cloud services through network access devices, such as computers, laptops, mobile phones, mobile Internet devices (MIDs)..etc.

The distribution of cloud services is normally provided by network and telecommunication carriers or a transport agent, where a transport agent refers to a business organization that provides physical transport of storage media such as high-capacity hard drives. Note that a cloud provider will set up SLAs with a cloud carrier to provide services consistent with the level of SLAs offered to cloud consumers, and may require the cloud carrier to provide dedicated and secure connections between cloud consumers and cloud providers.

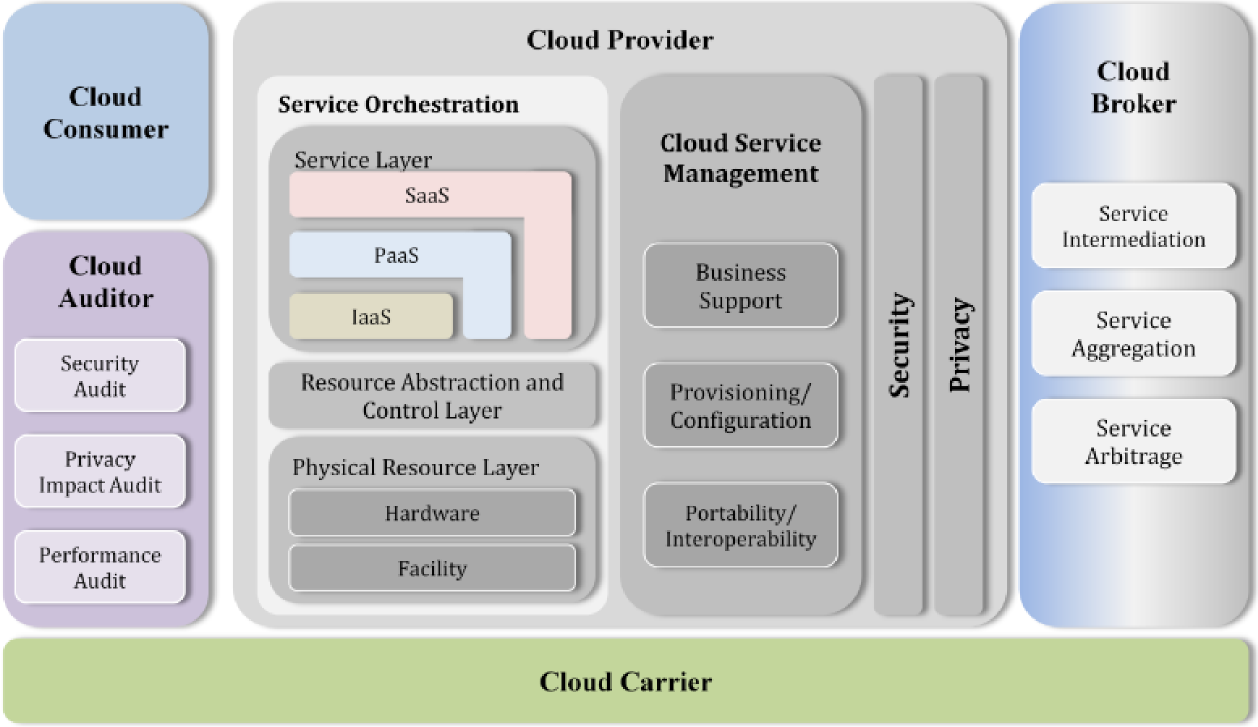


Figure 1.5 – The NIST Cloud Computing reference architecture

1.8 Benefit of Cloud Computing

Cloud Computing offers many important benefits to individuals and enterprises, in almost all fields, Cloud Computing gives opportunities to small companies to grow and offers its business many benefits, which led to more productivity and development especially in fields that depend on technology like economics, education, healthcare...

Following some benefits and advantages offered by Cloud Computing:

— **Reduced IT costs**

Cloud Computing may reduce the cost of managing and maintaining IT resources. Rather than investing in expensive systems and equipment for businesses and then put effort to maintain and manage them, we can reduce costs by using the resources and services provided by Cloud Computing.

— **Less time to market**

Ideas owner, developers, small companies and groups, with Cloud Computing and SaaS models, they can focus on developing their ideas without thinking about capitals or project financing, because of Cloud Computing billing that allows users pay only when they use resources.

— **Collaboration efficiency**

Collaboration in a cloud environment provides the ability to communicate, share and collaborate on documents and projects more easily outside of the traditional methods. this ability for many users is one of the most important advantages of the cloud.

— **Security**

Many organizations have security concerns when it comes to adopting a cloud-computing solution. but researches show that 94% of business claimed saw an improvement in security after switching to the cloud. 91% said that the cloud makes it easier to meet government compliance requirements [30].

— **Business Model**
Cloud Computing offers a good business model to small companies and individuals, the pay-as-you-go model allows them to pay according to their size.

— **Performance**
Cloud Computing offer a massive Computing Power and almost Unlimited Storage Capacity.

1.9 Cloud Challenges

With the growth of Cloud Computing and the popularity that has gained, new challenges and issues have to be faced, many interesting problems and obstacles are regularly being posed to the cloud community and practitioners.
Like any new technology, there is no clear definition and intelligible concepts, so that both actual consumers and new consumers may be uncomfortable about some issues like security, costing model, maturity... etc which makes the use or the transformation to the cloud may seem ambiguous. The following are some major issues:

1.9.0.1 Cloud Security

Based on a survey conducted in 2008 [25][31], and based on a second survey conducted by **Morgan Stanley research** [32], Securing the cloud and data security, ranks at the top of the list of Cloud Computing challenges.
The survey conducted by **Morgan Stanley research** 2011 [32] show that 43% of respondents in the conversations with industry participants (companies), cited security as one of the top three obstacles to cloud converting. This is a logical fear, moving to a public cloud means move and store sensitive data, run private software, put all major company's activities on the infrastructure of someone else, and that makes companies more careful about the cloud environment, especially they have less control over security which is the responsibility of the providers.
(Security is not just secure cloud from Hackers and Malicious but security can include data protections ex: data backup in hardware damage, Availability, Scalability...)

1.9.0.2 Cost management

As we know the Cloud Computing can significantly reduce infrastructure cost, software, management, and maintenance fees. In addition, Cloud Computing use the pay-as-you-go models which means customer pay only for what he uses (computing resources). However charging model of Cloud Computing Considered as a big issue of cloud and as the second most troubling obstacle for moving workloads to the cloud [32], because of the on-demand and auto-scaling nature of Cloud Computing services and the pricing model that is based on CPU cycles, network traffic, and storage volumes, make it sometimes difficult to define and predict costs and the estimation of the real financial benefits, this uncertain cost and benefits makes companies unsure about Cloud Computing finance.

On the other side providers are overburdened by this problem, they faced a big challenge and they make a massive effort to keep cloud costs in check by conducting better financial analytic, reporting, and articles, develop programs and calculators for cost calculation and better benefits estimation.

1.9.0.3 Other Issues

Performance : when companies move their core business to a cloud or a third party vendor, their business performance starts to depend on the provider as well, which make performance one of the main challenges of Cloud Computing, providers should always meet the basic expectations, like availability, auto-scale, less downtime...

Migration : Another big challenge of Cloud Computing is migration. is the process of moving data and software from on-premise infrastructure to cloud or from a cloud to another one, companies faced many problems and difficulties like complicated techniques problems, Slow data migrations, Failure risks... etc

1.10 Cloud Computing Platforms

In this section we present the biggest three platforms in the field of Cloud Computing:

1.10.1 Amazon Web Services AWS

Amazon Web Services (AWS) is a subsidiary of Amazon is the world's most inclusive and large adopted cloud platform it is considered an early leader in the market for cloud infrastructure, offering over 200 products and fully featured services from data centers globally, including computing, networking, storage, databases, analytics, and application services in the form of web services. AWS platform provides a variety of basic abstract infrastructure, distributed computing building blocks, and tools. AWS provides infrastructure as a service, platform as a service, and hosts private Cloud Computing. Some of AWS products: Amazon EC2,AWS Lambda, DynamoDB...etc

AWS began offering IT infrastructure services to businesses In 2006. Today, Amazon Web Services provides a highly reliable, scalable, low-cost infrastructure platform in the cloud that powers hundreds of thousands of businesses in 190 countries around the globe, it is considered the most profitable technical technology company in the world. wherein Q4 of 2020 AWS continuing to capitalize on its position as the leader in Cloud Computing market with 32-34% [33], exceeding its two largest competitors, Microsoft and Google, exceeding even the combined market share of these two competitors.

1.10.2 Microsoft Azure

Microsoft Azure or Azure is a private and public Cloud Computing platform offered by Microsoft to develop, deploy, and manage applications with different programming languages, Azure provides a range of cloud services including compute, analytics, storage, and networking with different models Software as a Service (SaaS), platform as a service (PaaS) and infrastructure as a service (IaaS).

Azure was announced at Microsoft's Professional Developers Conference (PDC) in October 2008.

1.10.3 Google Cloud Platform GCP

GCP is provided by Google, is a collection of Cloud Computing services and resources including computing, data storage, data analytics, APIs, and machine learning. Those services and infrastructure are the same that Google uses internally for its end-user and products, like YouTube, Gmail, Google Search... etc. GCP provides various environments and models like infrastructure as a service, platform as a service, and serverless computing. The beginning of the Google Cloud Platform was in April 2008 with the release of the first Cloud Computing service from Google the App Engine, a platform as a service to develop and host web applications in Google data centers, since this announcement Google added many cloud services and products, today there are over 100 products available on the platform like Compute Engine, Cloud Functions, VMware engine...etc

1.11 Summary

In this chapter, we explored the standard definition of Cloud Computing with its characteristics, as well as introduced Cloud Computing Architecture and its service models, deployment models, and roles of its actors. we also discussed the timeline of Cloud Computing, when and why the concept of the cloud was created, add to that we talk about some key components that help the cloud arises and some challenges that we must deal with it. Cloud Computing has successfully changed the way that consumers use information technology resources, also changed the form that these resources are delivered, instead of buying and managing physical hardware Cloud Computing offer on-demand delivery of IT resources including computing power, storage, databases, and software. Organizations of every type, size, or individual can access these technology services over the Internet, on an as-needed basis with a pay-as-you-go pricing model.

Chapter 2

Software as Service Model

2.1 Overview

In this chapter we will discuss the Software as a Service (SaaS) model, it is a main model of Cloud Computing, we see what exactly the SaaS model is, we will talk about SaaS application architectures, developments, and deployments models, we also compare SaaS with other similar models and see the major differences.

2.2 Introduction

Traditionally software and applications are installed and used locally on the user's infrastructure, the user is responsible for providing the necessary equipment, manage and maintain it, update and upgrade applications. However in recent years, the progress and the features that are presented by Cloud Computing are growing the interest in cloud technologies, this significant attention to Cloud Computing capabilities, makes communities think about the way that they are dealing with software and information technology resources in general.

From its name, Software as a Service refers to the model of delivery software in the form of a service, precisely SaaS is a cloud-based and web-based software delivered through the internet, usually accessed via a thin program such as a browser that runs on local client's devices. These software or applications are developed and hosted by a service provider, using the Cloud Computing paradigm, the provider is responsible for all tasks that make the service available on-demand for multiple users. By purchasing a monthly or yearly subscription, organizations and individuals can use software needed online rather than installed on local computers and without the need to invest and maintain the infrastructure or update and manage the software.

In the early 2000s, the first generation of SaaS applications was simple and inflexible, designed to solve small tasks and single business problem, but with the development of related technologies such as infrastructure, networks, web 2.0... etc, SaaS has evolved significantly.
Today, although most people don't recognize it, SaaS applications are more commonly used than they actually think and is probably the most used type of cloud services, SaaS is everywhere, from web-based email services such as Gmail, Hotmail or text editors like Word, Google doc or file hosting service like Dropbox to even must sophisticated applications like enterprise applications, such as relationship management (CRM), enterprise resource planning (ERP) and Image Processing like Photoshop and Canva.

Software as a Service model gain more and more popularity, because of its business benefits and capabilities that fulfill needs whether the consumer is a big organization, startup, or just individuals, in addition to that, the SaaS market is on the rise, according to Gartner report, SaaS remains the largest market segment and is forecast to reach $122.6 billion in 2021.

2.3 Architecture of SaaS Applications :

SaaS architecture is divided into levels base on the maturity aspect, many models are proposed to define the levels or 'maturity 'of SaaS applications. The most cited SaaS maturity model is the model proposed by Microsoft, which is a sort of incremental development model through integration between functional features, is classified SaaS architecture into one of four maturity levels along with three attributes: scalability, multi-tenancy, and configurability. Each level is distinguished from the previous one by the addition of one of these three attributes. [34] [35]

2.3.1 Level 1 – The Ad-Hoc/Custom

Ad-hoc or custom level is the first level of maturity, at this level, every customer (tenant) is provided a unique, customized version of the hosted application, the application runs its instance on the host's servers. this is the only way to support multiple tenants and it can be highly customized to client needs but customized by writing custom code, not through configuration and without scalability and multi-tenancy aspect, even though the software is technically delivered as a service. This level is useful when migrating from or converting an existing client-server or a traditional non-networked application, also this level is the least development efforts and reduced cost-effective [34, 36, 35].

2.3.2 Level 2 – Configurable

At this level, instead of customized by writing custom code, the application can be customized by changing its configuration and each customer can use separate instances of the same application, but it is still not scalable and cannot support multi-tenancy. This is done by configuring unique metadata, which helps cloud provider recognize the different users and their specific needs, because of that maintain and update the common codebases are made easy and simple for the users [34, 36, 35].

2.3.3 Level 3 – Configurable Multi-Tenant-Efficient

This level includes multi-tenancy concepts, which means that a single application instance can serve multiple customers, a customer can customize the same shared instance through a self-service tool. This level is limited in its ability to scale massively, it can be scaled up but only by moving it to a more powerful server, which isn't cost-effective. Still, the inefficient need for server space to accommodate many instances is eliminated and costs can be greatly reduced compared with level 2 of the SaaS maturity model [34, 36, 35, 37].

2.3.4 Level 4 – Scalable, Configurable, and Multi-Tenant-Efficient

The final maturity level adds scalability to configurability and multi-tenancy of level 3, all key attributes are available, this level allows the creation of new instances, adds them to a dynamic pool of instances and run it in the hundreds or even thousands of servers with the help of with a load balancer, dynamically increased or decreased to match load demand by adding or removing servers which maximize the infrastructure use [34, 36, 35, 37].

One more key attribute, which may be used in SaaS architectures, is Virtualization which replaces multi-tenant attribute. The advantage of adding up Virtualization over multi-tenant application in the architecture is that the system's capacity can be increased with no further programming. If both attributes are present, SaaS provides greater flexibility for tuning the system for best performance [36].

2.4 Multi-Tenancy Architecture

2.4.1 Multi-Tenancy Concept

Multi-tenancy concept or many tenants sharing the same resources is fundamental to Cloud Computing and it is a defining characteristic of SaaS applications [3]. Multi-tenancy is an architectural approach defined as a single instance of the software running on the vendor's servers, serving multiple tenants (users). With a multi-tenant architecture, a software application is designed to virtually partition its data and configuration, so that each tenant works with a customized virtual application instance, tenants use and customize an application as though they each have

a separate instance, their data and customization remain secure and isolated from the activity of all other tenants [38, 39].
Multi-tenancy has a tremendous impact on SaaS applications it decreases the cost of application delivery, allow for highly scalable, makes only one set of hardware resources adequate to meet the needs of all users, administration can efficiently manage only one stack of software and hardware, and developers can build and support a single code base on just one platform (operating system, database, etc.) rather than many. Not only for the application provider multi-tenancy also provides interesting benefits to the tenant it securely saves and isolates data from other tenants, improves software quality, makes the user feel that he uses the software lonely, facilitate collaboration by allows any user of any application varied access to specific sets of data.[38]

2.4.2 Achieving Multi-Tenancy

There are three different methods for achieving multi-tenancy: using a database, using Virtualization, or through physical separation [40].
Multi-tenancy via physical separation is rarely used because it relies on giving each tenant his own dedicated hardware resources. Second Virtualization, as we explain in the Cloud Computing section Virtualization is a technology for achieving multi-tenancy, however, achieving Multi-tenancy by Virtualization is more used for infrastructure sharing or infrastructure as a service (IaaS) platforms. In the case of Software as a Service (SaaS), multi-tenancy is almost always achieved via a database and configuration, with isolation provided at the application layer [40].

2.4.3 Configuration and Metadata-Driven

Multi-tenancy is practical only when it can support applications that are reliable, customizable, upgradeable, secure, and fast[40]. Software Architects **Microsoft** and **Salesforce.com** have said that metadata driven architectures are the core logic that gets applied within multi-tenant SaaS applications [39].

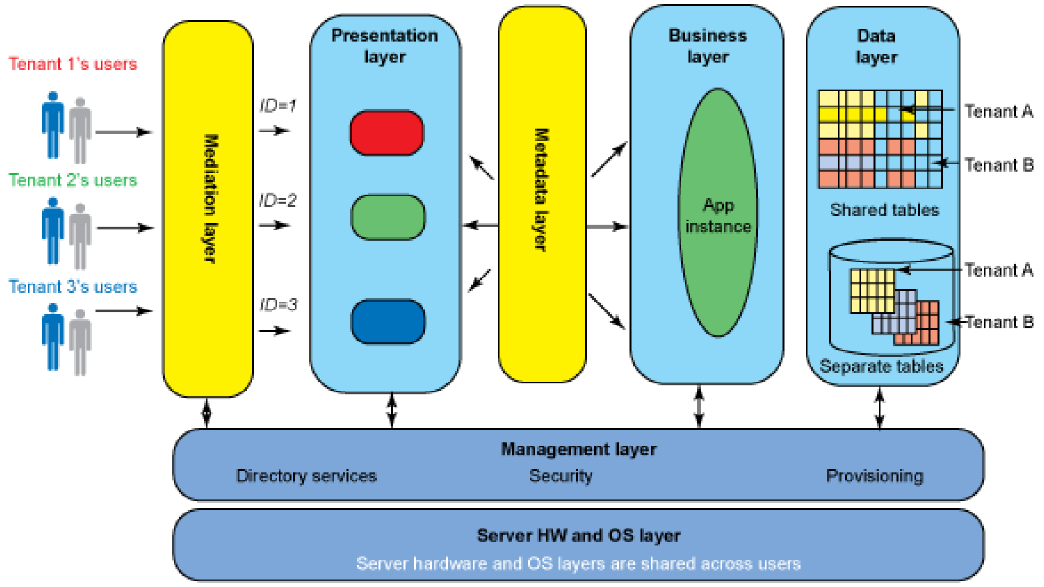


Figure 2.1 – A multi-tenant application environment

As shown in Figure 2.2, this model consists of a polymorphic application on the front end, a runtime engine to generate tenant specific UI page in runtime from the metadata according to his customization, a shared database contains metadata of each tenant customization, common metadata that describes the base functionality of the application and data of all tenants. In this architecture, there is a clear separation of the compiled run-time engine (kernel), application data, the metadata of an application, and the metadata that corresponds to each tenant's data and customization [38].

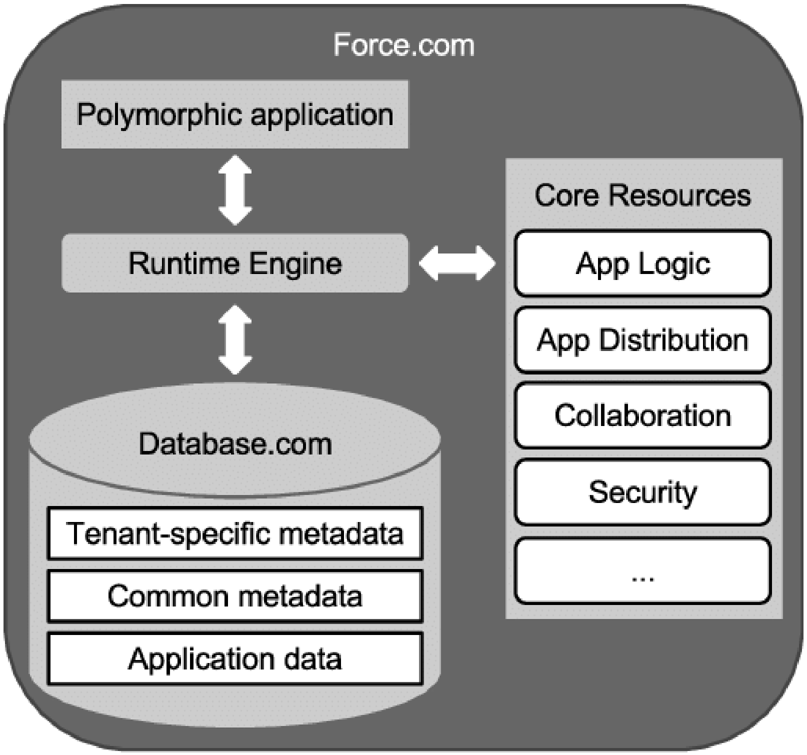


Figure 2.2 – Salesforce.com's metadata-driven architecture

2.4.4 Database Modeling

There are three approaches to managing multi-tenant data in SaaS application [41]

2.4.4.1 Separate Databases :

Storing tenant data in separate databases is the simplest approach to data isolation. application code is generally shared between all the tenants on a server, but each tenant has its own set of data that remains logically isolated from data that belongs to all other tenants. Metadata associates each database with the correct tenant, and database security prevents any tenant from accidentally or maliciously accessing other tenants' data, Giving each tenant its own database makes it easy to extend the application's data model, this approach tends to lead to higher costs for maintaining equipment and backing up tenant data. Hardware costs are also higher than they are under alternative approaches.

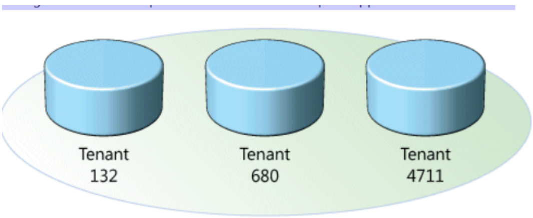


Figure 2.3 – This approach uses a different database for each tenant

2.4.4.2 Shared Database, Separate Schemas:

In this approach, tenants share the database, but the schema is different for each tenant. This approach is applicable for data separation and sharing databases, this approach is also known as the semi-multi- tenancy approach [3].

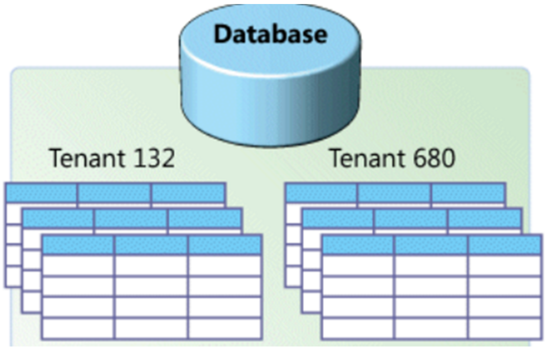
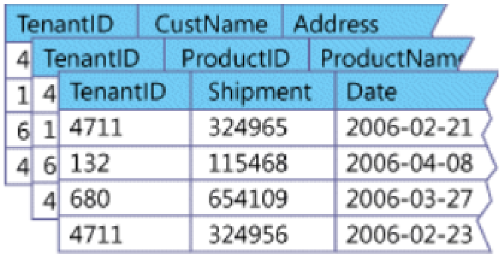


Figure 2.4 – In this approach each tenant has its own separate set of tables in a common database

2.4.4.3 Shared Database, Shared Schema

This third approach involves using the same database and the same set of tables to host multiple tenants' data. A given table or (document in NoSQL database) can include records from multiple tenants stored in any order, A tenant ID column associates every record with the appropriate tenant. This one is a highly preferred approach and called as pure-multi- tenancy approach[3].



TenantID CustName Address

TenantID ProductID ProductNam

TenantID	Shipment	Date
4711	324965	2006-02-21
132	115468	2006-04-08
680	654109	2006-03-27
4711	324956	2006-02-23

Figure 2.5 – In this approach each tenant has its own separate set of tables in a common database

2.5 SaaS Development and Deployment

SaaS applications are computer software, SaaS can be developed following different development pattern and models like others software, However SaaS model is different from other models, there are many concepts and technical matters must take into consideration.

2.5.1 Three-tier Model

SaaS applications follow the client-server pattern, in which a client makes requests and a server responds to the requests of many clients, A SaaS server generally follows the three-tier model pattern, which separates the responsibilities of different SaaS server components and enables horizontal scaling to accommodate millions of users Three-tier architecture is a client-server software architecture pattern shown figure in which the user interface (presentation), functional process logic ("business rules"), computer data storage and data access are developed and maintained as independent modules, most often on separate platforms [42].

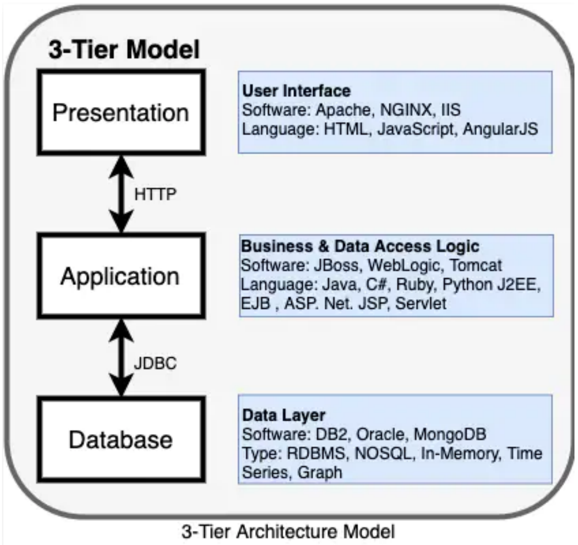

3-Tier Model
Presentation
User Interface
Software: Apache, NGINX, IIS
Language: HTML, JavaScript, AngularJS
HTTP
Application
Business & Data Access Logic
Software: JBoss, WebLogic, Tomcat
Language: Java, C#, Ruby, Python J2EE, EJB , ASP. Net. JSP, Servlet
JDBC
Database
Data Layer
Software: DB2, Oracle, MongoDB
Type: RDBMS, NOSQL, In-Memory, Time Series, Graph
3-Tier Architecture Model

Figure 2.6 – 3 three-tier architecture model

2.5.2 Service Oriented Architecture

Development of SaaS architectures can be done following SOA style, SOA requires loose coupling of services, which requires extensive planning and analysis to be done by the software architect to ensure the right SOA components and constructs to get a high degree of performance, scalability, and security [39].

2.5.3 Microservices

In the case of large and complex SaaS applications, the 3-tier model may need to update and modify. Microservices are an architectural style that structures the application as a collection of services. Each service can be written in a different programming language and tested separately. They are independently deployable and organized around business capabilities. As we mention SaaS applications are a group of services and data decoupled from each other and the core application. So that Microservices is one of the best choices to develop and deploy a complex SaaS application.

One of the best choices for creating and running microservices applications is by using containers. Containers encapsulate a lightweight Virtualization runtime environment for SaaS applications and allow to move the application from the developer's desktop to production deployment. Containers present a consistent software environment, and the developer can encapsulate all dependencies of the application as a deployable unit.
To manage containers developers use an open-source platform named Kubernetes (Kube), Kube is a container orchestration platform for scheduling and automating the deployment, management, and scaling of containerized applications. [43]

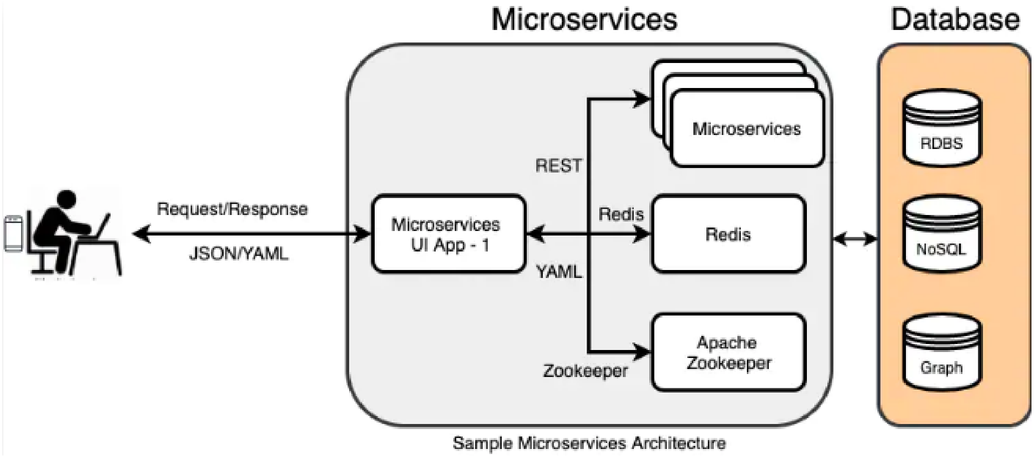


Figure 2.7 – Sample Microservices architecture

2.5.4 Serverless

Another option available to develop and deploy SaaS applications is Serverless, Serverless is a new way for the development and deployment of cloud applications, is a Cloud Computing execution model where developers don't need to manage the infrastructures, servers, or anything besides the final application code, the provider only focus on the application code and develop new features and functionality, Serverless is usually Event-driven means it runs code on-demand only, typically in a stateless container, generally called cloud functions or Function as a service (FaaS), on a per-request basis, automatically scale up and down as needed and load balancing without the interaction of the developers [44].

Serverless characteristics: [45]

— The serverless model requires no management and operation of infrastructure, giving developers more time to optimize code and develop innovative new features and functionalities.

— Serverless computing runs code on-demand only, typically in a stateless container, on a per-request basis, and scales transparently with the number of requests being served.

— Serverless computing enables end users to pay only for resources being used, never paying for idle capacity.

— It enables developers to focus on code, not infrastructure. (It's also a polyglot environment, enabling developers to code in any language or framework - Java, Python, node.js - with which they're comfortable.)

2.6 SaaS Applications Versus Similar Models

SaaS applications are cloud applications, so that each one must have Cloud Computing standards and characteristics discussed in the Cloud Computing characteristics section in the previous chapter. However, in some cases, we can't recognize other similar models if they are SaaS or not, following some points of difference that help distinguish SaaS applications from other models:

2.6.0.1 SaaS vs on-promise

On-promise software refers to traditional software that is installed locally, on user computers or servers, On-premise software provider sells a licensed copy of the software to the customer, to run the software, the customer need to manage and maintain the infrastructure such as hardware, networks... etc, and usually, he updates and upgrades software manually, On-premise is the easier model that we can distinguish it from SaaS is completely the opposite of SaaS, SaaS applications are hosted by the provider and run on his infrastructure, not on the client infrastructure, the user only needs a thin application like browser and pay for a subscription or just login to use the application.

2.6.0.2 SaaS vs ASP

An application service provider (ASP) is a business providing computer-based services to customers over a network, such as access to a particular software application that resides on the vendor's system and is accessed by users through a web browser, or by special purpose client software provided by the vendor[46].
From ASP definition we can see the similarity between ASP and SaaS, However SaaS model have advantage and different points such as:

- Usability: ASP applications are difficult to use where a customized version of an already complex system requires a lot of training and orientation, but SaaS Usability is easy where Internet-based SaaS applications bank on their intuitive usability and claim that users can start using the system within no time [47].
- Implementation Time: for the ASP model, to install and customize a commercial application, built by some other company, may take too long time and Lengthy cycle, for SaaS applications are Availability for all paid customers on-demand [47].

2.6.0.3 SaaS vs Web Application

A web application is application software that runs on a web server, accessed by the user through a web browser with an active network connection, as we mention before SaaS applications are web-based applications, but not all web applications are SaaS, in some cases it hard to differentiate between SaaS and traditional web app, following some differences between this two models:

— Hosting: The fundamental difference between cloud applications and web applications is hosting. SaaS applications are hosted in a cloud environment, in multiple data centers around the world, while other web applications are hosted in single data centers or in a server. which makes Web applications limited by availability and scalability.

— Type of application: SaaS is complete software delivered over the internet, while the web application is simple requests and responses between a server and a client, like wikis applications or some eCommerce applications [48].

2.7 Summary

Software-as-a-service (SaaS) is one of three principal components of Cloud Computing, is a new delivery model where users access the software via a Web browser, software and associated data are hosted and running in a cloud environment, SaaS applications have the same characteristics of Cloud Computing, like high scalable, availability, accessed through the internet... etc. SaaS architecture is divided into four levels base on the maturity aspect, each level is distinguished from the previous one by the addition of one of three attributes: scalability, multi-tenancy, and configurability. The metadata and multi-tenant are fundamentals components in SaaS architectures.

Chapter 3

Online Collaboration For Students

3.1 Overview

In this chapter we will talk about Online collaboration, we will define it, present some principles and factors that play an important role in effective and successful collaboration among students and explain models of online communication, also we will introduce popular tools, some platforms, and applications used to collaborate online.
In this chapter you will notice that we use the term "online collaboration" and we talk about online collaboration in general. However, in research process, we don't find any difference between online collaboration for students or for others, in general definitions, rules, and characteristics of online collaboration are the same in all fields.

3.2 Introduction

Teamwork and collaboration with others have become key instructional among education strategies. Online Collaboration is one of the best ways to achieve high quality learning experiences. Work with a collaborative group can help the learner improve his social skills, develop the way of thinking, and absorb more knowledge. Recently with massive progress of technologies especially the WEB, has changed the

way that students learn and interact with the education environment[49, 50].
Education is becoming increasingly collaborative. Today teachers and students can skip time and space constraints of learning by moving an important part of the process to online collaboration, students and instructors can communicate, share documents, solve problems and work together on their projects from anywhere at any time [51].

Furthermore, online collaboration and the ability to work as part of a team from distance is becoming more important to the advancement of knowledge, success as a student, and even as a future employee since collaborating and learning using technology is a required skill for today's jobs [50].
Many tools are available and full of features that make online collaboration possible, easy to use, and almost with same as face-to-face collaboration.

3.3 Definitions

3.3.1 Collaboration

Collaboration is defined as "an activity that is undertaken by equal partners who work jointly on the same problem rather than on different components of the problem "[52]. Also, Collaboration can be defined as the interdependence of the group participants as they share unique ideas and experiences. The result is better than any one of the individuals could have obtained by working alone. The group members all work on the whole problem, and the individual contributions of the participants often cannot be clearly separated in the final project [49].

3.3.2 Online Collaboration

Online Collaboration or Digital collaboration is groups of two or more individuals work together to learn or to achieve a goal, from distance, using digital technologies and the Internet.

3.4 Students Collaboration Principles

Not each group work together is collaboration, working effectively and efficiently as a team especially online is not that easy so that many principles that must be the foundation of in any collaborative group, Stephen b Richard [53], identified several principles and practices that impact collaborative processes, following some major principles of collaboration:

3.4.1 Working Together

The obvious principle of collaboration is that each member of the group must work with other members. In the collaboration concept, this principle may seem axiomatic, but it may be broken very easily, if two students work on the same project but they disagree about the roles or the goals of the team, this may leads the group down the wrong path, and makes members work against one another rather than work together, individuals must work for the group and not for the personal interest.

3.4.2 Identifying a Group Goal

The first purpose of creating a collaborative group is to work together to achieve a common goal, this should be the first consideration for a collaboration to be successful, individuals must specify and agree on group goals.

3.4.3 Equal Participation

For a group to be collaborative, all the members of the group need to do the same amount of work and effort, for example, some students when they work in a group they may do less effort and show less motivation to participate, the unequal contribution from members will affect the team spirit and collaboration process.

3.4.4 Responsibility and interdependence

Each collaborative group must share responsibilities and divide tasks between members, this gives each individual a sense of importance and emphasizes and encourages to take responsibility, another factor is interdependence, interdependence influences the individual behavior and outcomes of the group. It makes each member focus on his roles and tasks without extra thinking about the rest parts of the project. Take responsibility and interdependence are two complementary principles that make members search for learning and not for obstructing or ignoring each other.

3.4.5 Communication

The core of any collaboration is communication, especially in online collaboration, researches found that traditional teams tend to communicate more effectively than virtual teams, members of an online collaborative group must focus on effective communication with the necessary canals, communication helps to connect with one another, share opportunities and resources, exchange meaningful information, build relationships, create group culture, and the more important thing is to get job done[49, 53, 54].

3.5 Factors of a Successful Online Collaboration

A group of people works together doesn't mean that the collaboration will succeed, in addition to principles in the previous section that define the basis of collaboration, many factors, and considerations play an important role to collaborate successfully, Ingerm et all [49] describe some factors that may prompt and optimize online collaboration or inhibit it:

3.5.1 The Right Technology For The Task

A project can be divided into small, different tasks, each task needs to use the appropriate technology and the right treatment. There are two models of communication in the online collaboration context (we will cover them in the next section) that can affect the treatment of the task, some tasks need to get done in real-time communication, which means that those involved with this task should work together at the same time, like using videoconferencing or chat rooms, usually these type of task need to make the right decisions from too many ideas. Another type of task needs to be done in asynchronous time, where those involved with this task should take enough time to solve problems or read and study the messages before they respond, tools like email can be useful in this case

Another factor that can affect the quality of online collaboration, is technologies and tools available to all members of the group, individuals with slow computers and internet connections can obstruct the process of communication, even some Text-only communication may lose vital information and may not allow participants to effectively work together, in this case, more developed tools like video conference or real-time document editors will be a mandatory choice, so that the right choice among those technologies and models can improve or deteriorate the results.

3.5.2 Group Size

The size of the group in online collaborative situations is influential, the size of the group must be varied according to the project and to the goals of the group, even in big projects divide the large group into small teams will be the right decision, because in large groups members may feel less responsible, moreover research show that social loafing and the size of the group is directly correlated, groups should be small enough to allow the contribution of every member to be noticed by the instructor and the group [55].

3.5.3 Individual accountability

An important factor is accountability, which means that every member should bears the consequences of his performance or actions. Also, it is important for all members to feel that they are providing a unique and visible contribution to the group effort, this feature can be more clear in online collaborations, a member can notices contributions of the other members at any time, he can compare his work with others and so that he can improve his work and try to learn more. Members need to notice that their contributions: will be noticed by other members and is valuable to the outcome of the group. These points can benefit individuals and groups as a result.

3.6 Computer-Mediated Communication

Basically, online collaboration is online communication between individuals of a group. When we talk about online communication we must talk about Computer-Mediated Communication (CMC). CMC is a process of human communication via computers, involving people, situated in particular contexts, engaging in processes to shape media for a variety of purposes [56].

CMC can be divided into two modes: synchronous CMC and asynchronous CMC, these different types of CMC support different types of communication and mode of work so that they have different implications for online collaboration [49].

3.6.1 Synchronous CMC

Synchronous CMC refers to the communication that happens in real-time, this means that all parties are engaged in the communication together at the same time (live) and it occurs almost instantly, like a videoconferencing, a phone call, direct chat rooms... etc

3.6.2 Asynchronous CMC

Asynchronous CMC is communications that does not happen in real-time. This means that not require the simultaneous interaction of individuals, they are independent of time and space. In this case, the sender might not receive a response immediately from the receiver after sending the message, the receiver takes his time to analyze and treats the information before he sends back a response. example about this mode: email, commenting in a blog post, write a review or feedback.

3.7 Cloud Collaboration

Cloud collaboration refers to cloud-based applications that enable collaboration to take place on a virtual dimension, regardless of time and location by allowing students to share documents and files, communicate, manage their projects, and real-time working [57].

Online collaboration is only partially enabled by the underlying concept of Cloud Computing. Online collaboration tools must have some characteristics and specific capabilities to enable teams to work comfortably and effectively, basically capabilities like availability, less downtime, high performance, and unlimited scale. Since online collaboration tools are web-based applications, there is no way best than the way of Cloud Computing to develop and host applications. Cloud Computing is the best choice to provide all functionalities and capabilities to fulfill all needs of any high quality application [2].

Before Cloud Computing, most of the online collaboration tools were limited to basic methods and functionalities, but the migration to Cloud Computing environments led to more progress in the online collaboration field and allow to create more rich applications, with almost all features and functionalists needed. Today almost all online collaboration applications are Software as a Service based applications, like Dropbox, Google Docs, Zoom, Office 365... etc.

3.8 Different Types of Online Collaboration Tools

The development of cloud, web 2.0, and related technologies allow to develop many applications with rich functionalities in all life sections, online collaboration is one of the fields that know many new tools that enable more coordinated workflow by providing a unified collaborative platform for team discussion, file sharing, online storage, document editors, task assignments, and real-time project management, there are various types of tools each one has different functionalities and ways of communicating, following some types of tools used to collaborate online:

3.8.1 Early Generations Tools

Early generation tools for online collaboration represented in email, wiki and blogs [58], email is a method of exchanging messages ("mail") between people using electronic devices, email is an old technology used in early computers, it is a default tool to cooperate and online collaborate of the past. Even today, email is still the basic tool for collaboration in both academic and industry fields.

Wiki is another early collaboration tool especially for students, A wiki is a collaborative tool that allows students to contribute and modify one or more pages of course related materials. Wikis are collaborative and facilitate community-building within a course. Essentially, a wiki is a web page with an open-editing system. However, wikis are not full-scale collaboration tools and they have a very low barrier to entry. They are based on free or cheap software that can be managed behind the corporate firewall so they don't suffer from the same security concerns that management may worry about with online [59].

3.8.2 Project Management Tools

Project Management Tool is a combination of various features and functionalities including scheduling, information and file sharing, messaging, tasks assigning, document editing.. etc, all in one application for collaborative work, it helps to manage projects and knowledge, track team productivity and tasks progress, communicate with members, and organize workflow.

Project Management is one of the best tools for online collaboration for students because in general students lack experience in running projects and organizing groups, especially online, so that they need solid tools to manage projects and increase team effectiveness. Almost all PM platforms have all needs of a student in one place, from messaging, documents sharing to task assigning, to-do list, calendar.. etc, also PMs help team members work smarter and track all tasks progress and schedules.

3.8.3 Video Conferencing Tools

Video Conferencing is the tool for the transmission and reception of video and audio signals by users in different locations, for communication. This technology allows multiple people to meet and collaborate face to face long distances in different cities or even different countries by transmitting audio, video, text, and presentations in real-time through the internet [60].

Video Conferencing is synchronous CMC mode, it is a very effective tool for collaboration not only for education but for all academic and industry sectors, it is widely used by the biggest companies to collaborate and for their core business activities, it is used even by international organizations and governments especially in crisis periods like COVID-19 pandemic.

Students can use this tool to collaborate on the better way with each other or with teachers for their studies in general, an important feature available in almost all Video Conferencing applications is allowing for one-on-one chat conferences or with a group of participants, it's easy to use and available for free.

The global video conferencing market size prove the widely use and its effectiveness, the market is expected to grow from USD 9.2 billion in 2021 to USD 22.5 billion by 2026 [61].

3.8.4 Social Media

Even though social media websites are very distracting, especially for young people and students, it stays the easier way to communicate and collaborate if it well used. a feature like Facebook groups is a very effective way for students to collaborate online.

3.9 Online Collaboration Platforms

There are numerous websites and applications for online collaboration available on the internet, for free or paid versions, each product has many features that can help students collaborate at ease, here is some popular applications and platforms:

3.9.1 Google Workspace for Education

Google Workspace is a suite of Cloud Computing SaaS applications and tools for productivity and collaboration all in one place with a single account, all these applications are developed by Google and hosted in Google cloud one of the biggest Cloud Computing in the world, Google Workspace includes across apps like Gmail, Chat, Calendar, Drive, Docs, Sheets, Meet and more, it provides all kind of tools for online collaboration like email, videoconferencing, documents editing and sharing, file storage, chat... etc. Google Workspace for education, previously known as G Suite for Education, offers flexible tools so students can better collaborate and create safe learning environments.

Because moving from tools to another one can make communication and projects organization a hard task, Google Workspace helps students to centralized all applications needed to collaborate just in one platform, Google workspace apps are developed by google the largest tech company this means high quality product and best user experience, also apps are hosted in google cloud which makes it available at any time and scalable. Following some Google workspace application [62, 63]:

3.9.1.1 Docs

Google docs is an online text editor is a smart editing and styling tool helps user format text and paragraphs, hundreds of fonts are available, add links, images, drawings.. etc, using Google Docs everyone on the team can write, edit text files and work together in the same document at the same time, comment and chat about the document, also users can share documents with anyone with a direct link, docs provide all functionalities to not worry about file formats, user can convert Word files to Google Docs and vice versa and many more features, google docs is a powerful tool for students for online documents editing and sharing available for free on the internet.

3.9.1.2 Chat

Google chat is a 1:1 and group messaging, it allows for direct messages and group conversations, Google Chat helps teams collaborate fluidly and efficiently from anywhere. it very easy to create dedicated virtual rooms for teams conversations to keep projects organized and on track. Shared chat including threaded conversations plus shared files and tasks like Google Docs. Also, Google Chat supports external users, 28 languages, and 8,000 members per room and it makes communication between groups members flow very smoothly. with many features like easily find past conversations or files with Google's powerful search. Google Chat simply helps students get more done and faster.

3.9.1.3 Jamboard

Jamboard is a virtual whiteboard, a team member can sketch his ideas while benefiting from the access and connectivity of an interactive canvas, on mobile, or right from a web browser. Drop images, add notes and pull assets directly from the web while collaborating with other team members from anywhere. It saves work to the cloud automatically, uses the easy-to-read handwriting and shapes recognition tool, and draws with a stylus or with a finger, just like a real whiteboard. Jamboard unlocks students team's creative potential with real-time co-authoring. Experience unhindered productivity, whether the team is in the same room using multiple Jamboards, or across the world using the Jamboard app on mobile. Jamboard helps students to collaborate in brainstorm cases and sketch ideas, pull in work from Docs, Sheets, and Slides, even add photos stored in Drive.

3.9.2 Zoom

Zoom is a collaborative, cloud-based videoconferencing service offering features including video telephony, online meetings, group messaging services, and secure recording of sessions [64].

Zoom simplified video conferencing and messaging across any device PC, Tablet, and Phone, it supports HD video and audio meetings with up to 1000 video participants and 49 videos on screen. Zoom is a powerful app for collaboration, where multiple participants can share their screens simultaneously and co-annotate for a more interactive meeting, Chat with groups, searchable history, integrated file sharing are available.
Zoom is a preferred method for collaboration compared to other videoconferencing platforms. it is easy to use with the best user experience. [64]

3.9.3 Slack

Slack is a popular collaboration tool that can be used on both desktop and mobile devices, is a messaging app for business that connects people to the information they need. By bringing people together to work as one unified team, Slack transforms the way teams communicate.

Slack is a popular app for online collaboration among students because of the way that it structures the blogs and posts and the way of communication and interaction of users with each other, it organizes chats into channels, say, for a specific project, department, company, topic.. etc.

This tool stands out from the crowd, its support for voice and video calling through its desktop or mobile app or website, allowing for more direct contact when needed. There's even your space inside the app, where you can draft messages, list your to-dos, or keep links. Slack supports asynchronous work when work is organized in channels. In Slack, everyone in a team has access to the same shared and searchable information. When teams work together in channels, information can be shared with everyone at once, helping keep teams stay aligned and make decisions more quickly.

3.10 Advantages of Online Collaboration

Online collaboration is rising and it is becoming more and more approved and accepted as a method of work especially in the education sector, and it is a high demand skill for many jobs, online collaboration has many benefits for students following advantages of it:

- The Ability to use online collaboration technologies is a valuable skill for students for future jobs, online collaboration technologies are considered as one of the most important skills required for future jobs[50].
- An online collaborative learning environment can positively affect student's performance on problem solving and complete projects. Researches suggested that computer-mediated collaborative groups had positive attitudes toward learning collaboratively and performed significantly better than participants who worked alone [51].
- Online collaboration leads to higher academic achievement, improved relationships, and better social skills [49].
- Online collaboration allows students to work even with international students not just with local colleagues, this can open new opportunities to students,

gain more knowledge, and learn about different cultures and people [50].

- Help students organize their project and save their documents, easy access to information from anywhere at anytime.
- The ability to learn and complete projects from outside schools and universities and outside work time
- Allow students to communicate better and help each others

3.11 Challenges and Issues

It's important to be aware that online collaboration will not necessarily solve or improves issues of communication and learning in student teams, as we mentioned early, many factors play an important role in effective work, moreover online collaboration as a method has limitations in some cases, and probably will not work. These some challenges and issues of online collaboration:

- **Discipline**: online collaboration can make some students lose discipline that they may have in classes or face-to-face work. this can delay the project's deadlines and not achieving goals.
- **Students do not contribute equally:** A big challenge for online collaboration is the equal contribution, because of the first issue 'discipline', some students can contribute more then others and this can negatively affects the benefits of collaboration and learning in a group [65]
- **Finding a mutually suitable time to work together :** Online collaboration allow students to work at anytime, so that finding the time suitable for all group members to work together especially on synchronous tasks is a challenge
- **Communication :** Although online collaboration are advanced and developed well, but communication through internet still have some limitations, students should deal with these limitations and do more effort to stay connected. [66]
- **Technical problems :**Online collaboration is totally dependent on electronic devices and software, some times technical problems can obstruct the collabo-

ration process, like slow internet connection, low performance computers, applications with bad user experience... etc.

3.12 Summary

Online collaboration is an effective learning method, it provides a model of learning in which students are encouraged and supported to work together to create knowledge, to explore ways to innovate, interact with a learning environment, and communicate with others. In this chapter, we identified that any online collaboration should be characterized by some principles and factors to be successful and for a good outcome. With the rapid development of technologies especially Cloud Computing many tools are developed and available for students to use very easily, we present some platforms and applications that students can use to learn and complete their projects. From our research about online collaboration and its tools, we found that online collaboration is a suite of features and functionalities including real-time communications, online text editors, storage and documents sharing, tasks assignment, to-do lists, calendars... etc.

Online collaboration for students has various benefits and advantages, it increases productivity, provides higher academic achievement, improves communications skills, and prepares students for future jobs.

Chapter 4

Planning and Analysis

4.1 Overview

In this chapter, we start developing our solution which is a cloud-based application for students, to collaborate and work together online. But before diving into the development process and like any project in any domain the first stage that must pass by is the planning stage, we will define a roadmap and the methodology to follow.

In this chapter, analysis is described, which involves discovering and listing both functional and non-functional requirements and analysis of similar applications and potential competition in general.

The analysis started with a brainstorming session, then we proposed our solution according to our research in the online collaborations for students in chapter 3, and based on some ideas from products that we presented in the same chapter, this will help to outline the key points of the application, we also defined actors of the application with the basic functionality and different authorization levels. Next, with the help of UML we translated and presented basic use cases of the application in form of diagrams, this will help us understand better our application.

4.2 Software Development Methodology

Software development methodology refers to structured processes involved when developing software. It is a procedure that organizes and structures a set of activities to create a software product.

Software Development Life Cycle (SDLC) is the most important element in software development, it defines different practices that are necessary to bring a project from its initial idea to deployment and later maintenance. SDLC is a conceptual model for software development that consists of multiple phases. Building a software product is a process consisting of several distinct phases, each phase can be thought of as a building block for the next phase and each phase has its own deliverables and is bound by a specific time frame. There are different SDLC models, such as the classic "Waterfall Model," "Spiral", as well as many modified Waterfall models. Conventional software processes usually address four development phases: analysis, design implementation, and testing/ maintenance [67].

4.3 Development Phases

There are many models that defined and structure SDLC phases, each model has its own process division and phases but in general, all models have the same goals and they are very similar. However, in our case, we chose basic models and phases that fit our project and situation, following the detailed development phases that we will pass it one by one:

4.3.1 Analysis

The first phase in our development process identifies and defines the system to be developed. In this phase, we will brainstorm about our solution, suggestions, and the shapes of the software to be developed. After that, we will analyze the requirements of the software to gain a clear understanding of what is required, also we will gather all the specific details required for the software as well as determining functional, non-functional requirements, and define the technical aspects of the requirement. To understand how the software will be used, who the users will be, we will define actors of the application, analyze the software behaviors by defining software use cases and expected user interaction, for that we will use use-cases descriptions and UML diagrams.

4.3.2 Design

Once the analysis phase is completed the next phase is design, in this phase we will start building the software components, chose the right architecture that fit cloud-based application and fit our project. Transfer ideas and requirements to practical solutions that satisfy the requirements identified in the previous phase and that can be used as an input to system development in the next phase. In general, our goal in this phase is to transform the requirements into complete and detailed system design specifications.

4.3.3 Implementation

Its development stage is the part where developers transfer diagrams, algorithms, and textual solutions to machine language to build the application according to the earlier design documents and outlined specifications. In this phase, we will choose the technologies, design patterns, deploy environment, and tools that we will use in the development process, also, we will explain some code and programming solutions.

4.4 Our Project

4.4.1 Background information

This is a SaaS-based application for students to collaborate and work together online to get more knowledge and complete their homework, school projects, personals projects... etc, Students create, write and edit documents, organize workflow by creating to-do lists, tasks tables, and schedules projects, communicate and share ideas. Students access the application over the internet using a browser.

4.4.2 Purpose

The purpose of our project is to provide a tool that helps students work together from distance, organize projects, documents, tasks and provide ways to communicate in a learning environment.
Why our project is relevant? : Because recent decades have seen a dramatically accelerating pace in the development and adoption of new technologies, today almost all students are familiar with the technology. Exploit student's technology skills and provide a tool that helps them increase knowledge, is a good idea. in addition to the many benefits of online collaboration for students, those tools can be very effective especially in crises periods. Like the COVID-19 pandemic that has led to explosive growth in student collaboration tools that play an important role to continue learning and working with other students.

4.4.3 Intended Audience

The target audience and users of our software are the students and teachers in different levels with basic skills of use of the technology.

4.4.4 Scope of the Software

The scope of the project means identifies the boundaries for the project, what the software will do and provides, and what didn't. The software is a tool to create and share content (text and media), structure files, documents, and projects, help students organize their teams, communicate and facilitate working together online. However the software is not a spreadsheet application like Excel and it doesn't provide a service like that or do any kind of calculations, is not a storage service to store files and documents, also it is not for daily direct messaging or communication tool like email.

4.5 Our solution

Based on chapter 3 and tools that we discuss, we recognize some features that may help students collaborate online, we propose the following features and functionalities:

4.5.1 Features

Text editor : The software provides a complete text editor that works on the browser, users can create and edit content easily.

Private space : A student can work on his own space, only the owner can access the private space.

Team space : The software provides a team space where a group of students work together in the same place and share the same documents.

Project management : The software helps groups organize projects by assigning tasks and define goals by creating to-do lists, also the software allows for controlling groups by admins.

Communication : In the project space, students can communicate in different ways, by commenting on pages or by creating posts and replies to other posts.

4.5.2 Basic Components

Pages

Pages are the smallest components in the application, it where users do their work (write content, notes, create to-do lists... etc), pages are classified into four types:

- Text page : is where users write and edit text content
- To-do list page : is where users create a list of tasks that they need to complete
- Tasks page : is a pages contain tables of different project's tasks each tasks assign
- Calendar : is a page where user schedules events and set reminders

Page comments

Is a space that exists on every page, where the user can comment and write notes about the content of the page.

Folders

As a normal computer folder, a folder is a way to organize different files, is a storage space, or container where the user put and classified many files like documents, text pages, and images.

Projects space

Projects are spaces where many users can work together and share the same files, it contains pages, folders, and a communication section.

Private space

Beside projects each user have his own private space, where he creates pages store files.. etc, the private space can accessed only by the actual user.

Group Channels Is a space where group members communicate, a user can create posts and share them with other members to ask for help or general questions, other users can comment and interact with posts.

Group channels are available only in the project space.

4.6 Software Requirements Specifications

4.6.1 Define Actors

Define actor of the application helps us specify use cases and functional requirements, following actors of the application:

- **Normal User :** It is the main actor, it is a person that uses the application, he needs to Sign Up to get access to the app.
- **Admin** : It is a user that has some extra permissions in projects space, a user can be a project admin if he is the creator or if he is added by other project admins, set an admin actor is important because we need some organization in project space.
- **Guest** : A guest is a visitor to a public page, he can only read the content without sign-up or authentication, it's important to allow the user to share his content with others even if they are not users of the application. this can helps users of the application freely work with anyone.
- **Unknown :** A person who wants to use the App and applies for a registration or authentication

4.6.2 Functional Requirements

Following functional requirements classified based on actor permissions:

Unknown

- Sign in
- Sign up

Guest

- Visit and read content of a public page

User

- Page management : create, delete, rename, edit, share text page.

- Folders and files management : folders (create, delete, rename), file(upload, download, delete).
- Create projects
- Comment on a text page
- Create a posts in group's channels
- comment on posts

Admin

- Create channels
- Add members to a group
- Delete members
- Assign tasks
- Create tasks, To-dos

4.6.3 Permissions and Rules

Each team needs a leader or a decision maker, in our application project spaces, we need to set a superior user to control the space and its members, which may guarantee some organization.
In a project space, only admin have permission to:

- Create task, to-do, calendar pages, create folders
- Create channels
- Add and delete members
- assign tasks

pages rules:

- Only the creator of the page can edit page if it locked
- Only the creator can delete and rename page
- User can edit other member page if it unlock

4.6.4 Use Cases

A use case is a written description and UML diagrams of how users will perform tasks on the application, following use cases diagrams:

Use case :Main application

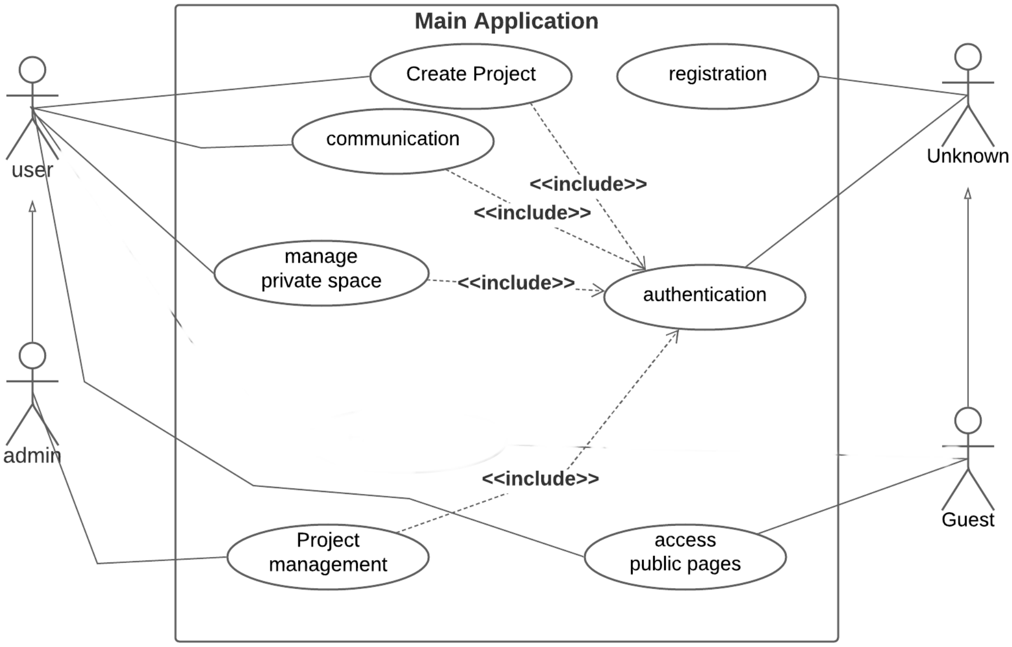


Figure 4.1 – Diagram Use case : Main application

Use case :Private space management

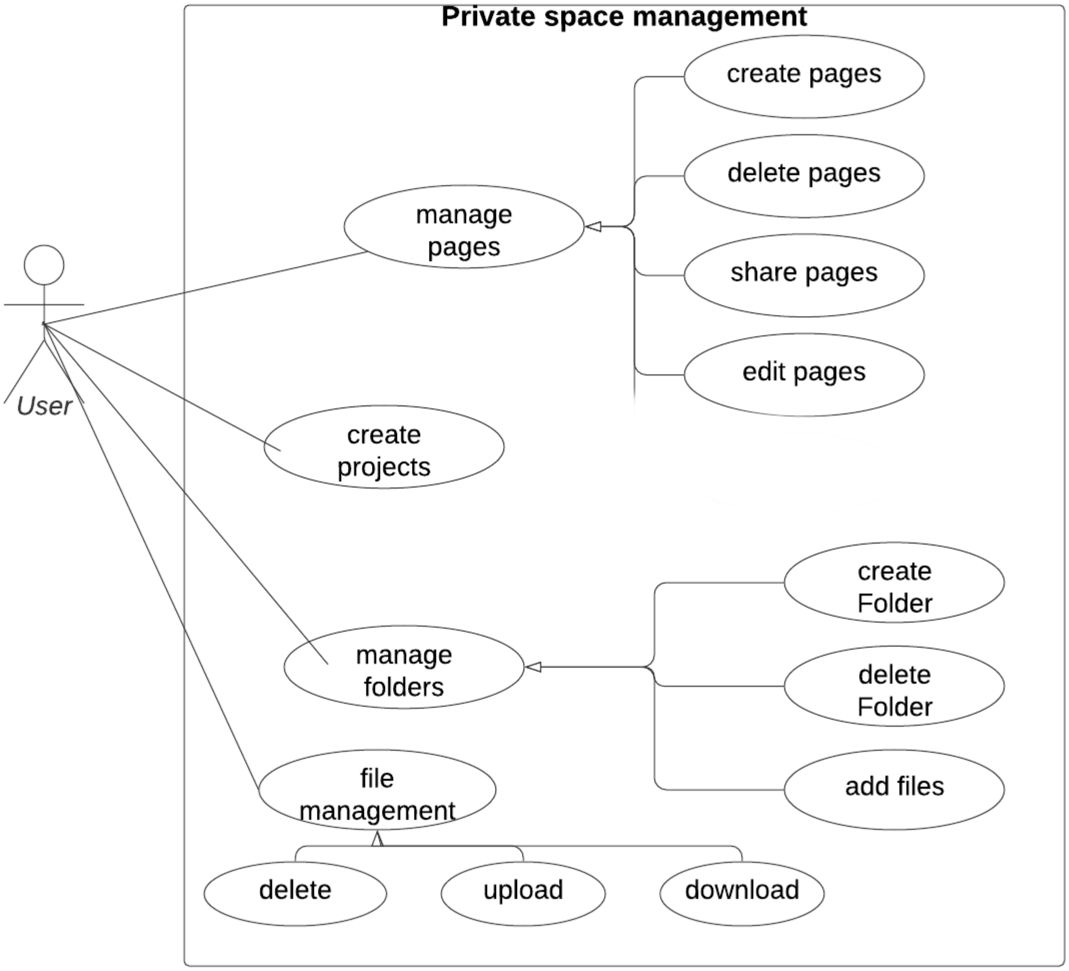


Figure 4.2 – Diagram Use case : Private space management

Use case :Project Managment

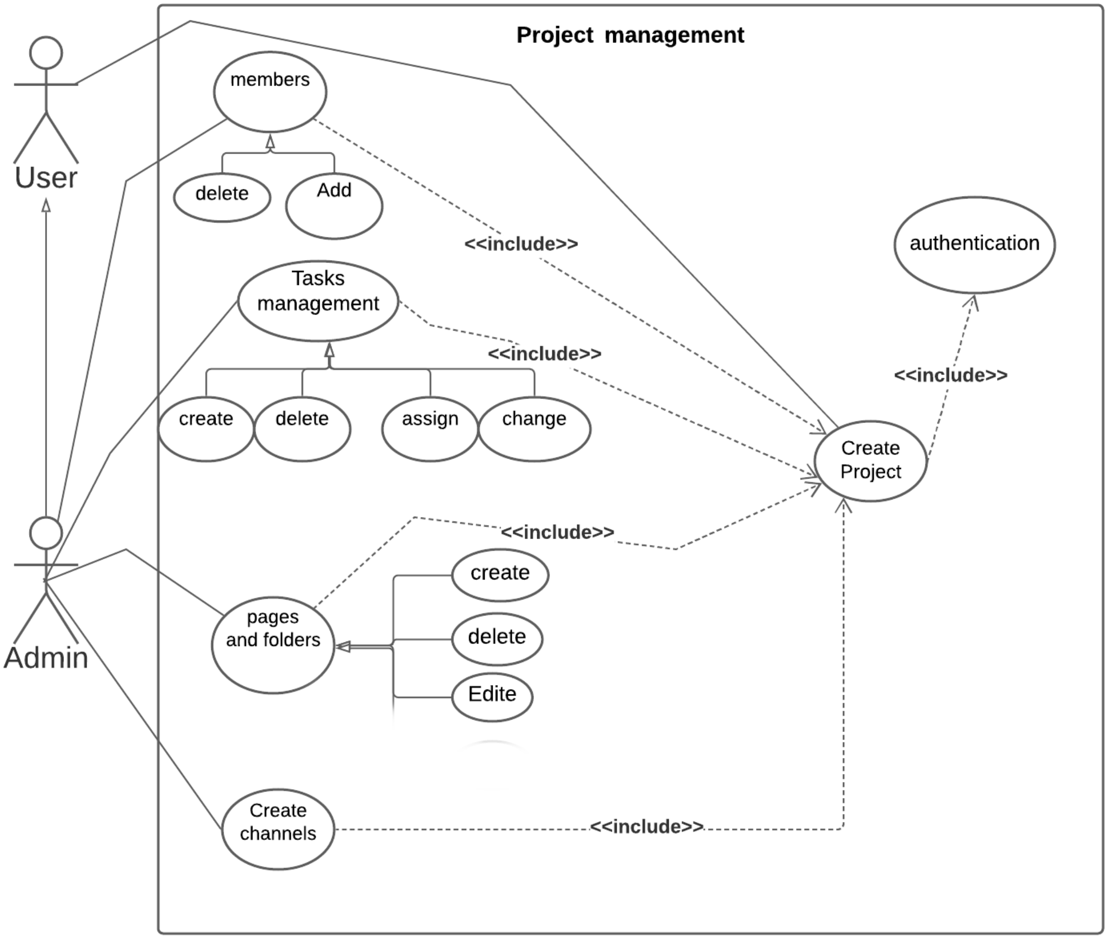


Figure 4.3 – Diagram Use case : Project Managment

Use case :User in project space

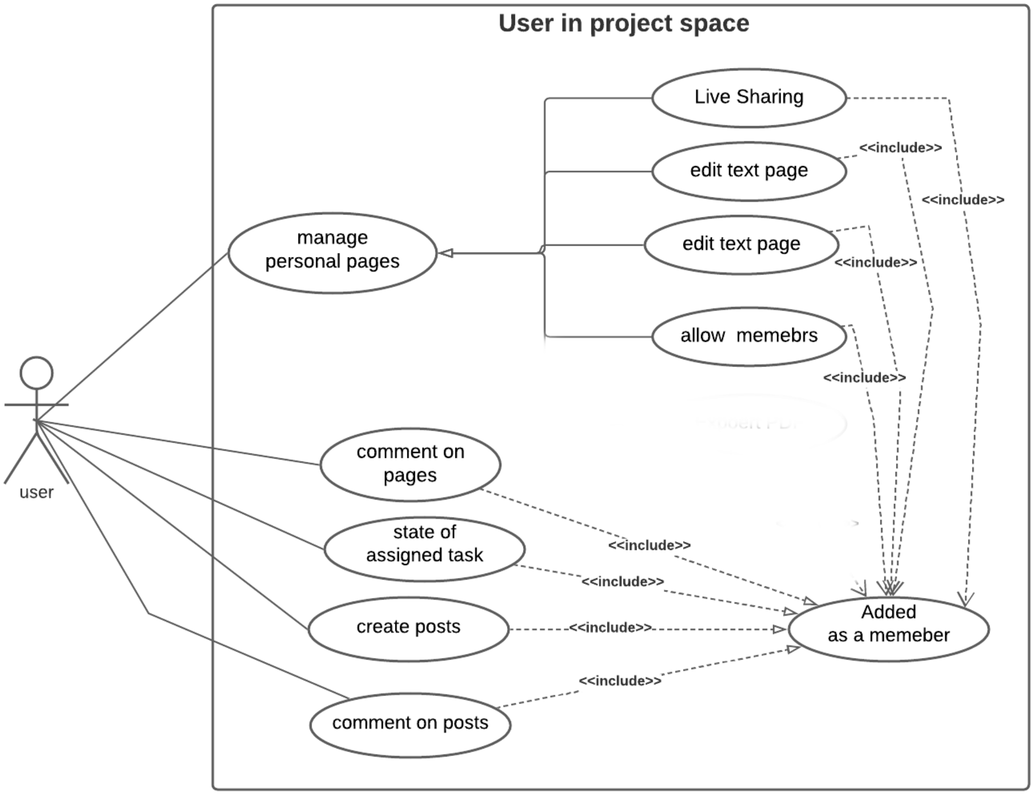
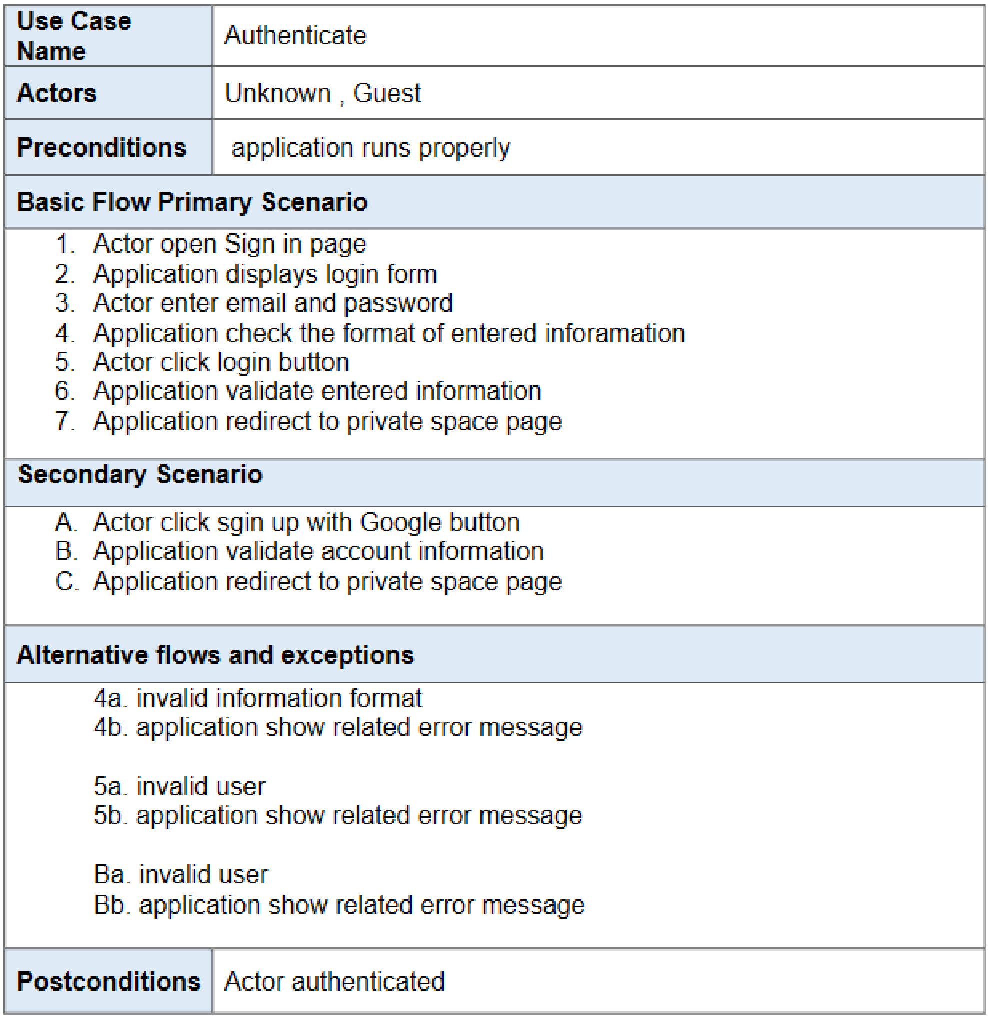


Figure 4.4 – Diagram Use case : User in project space

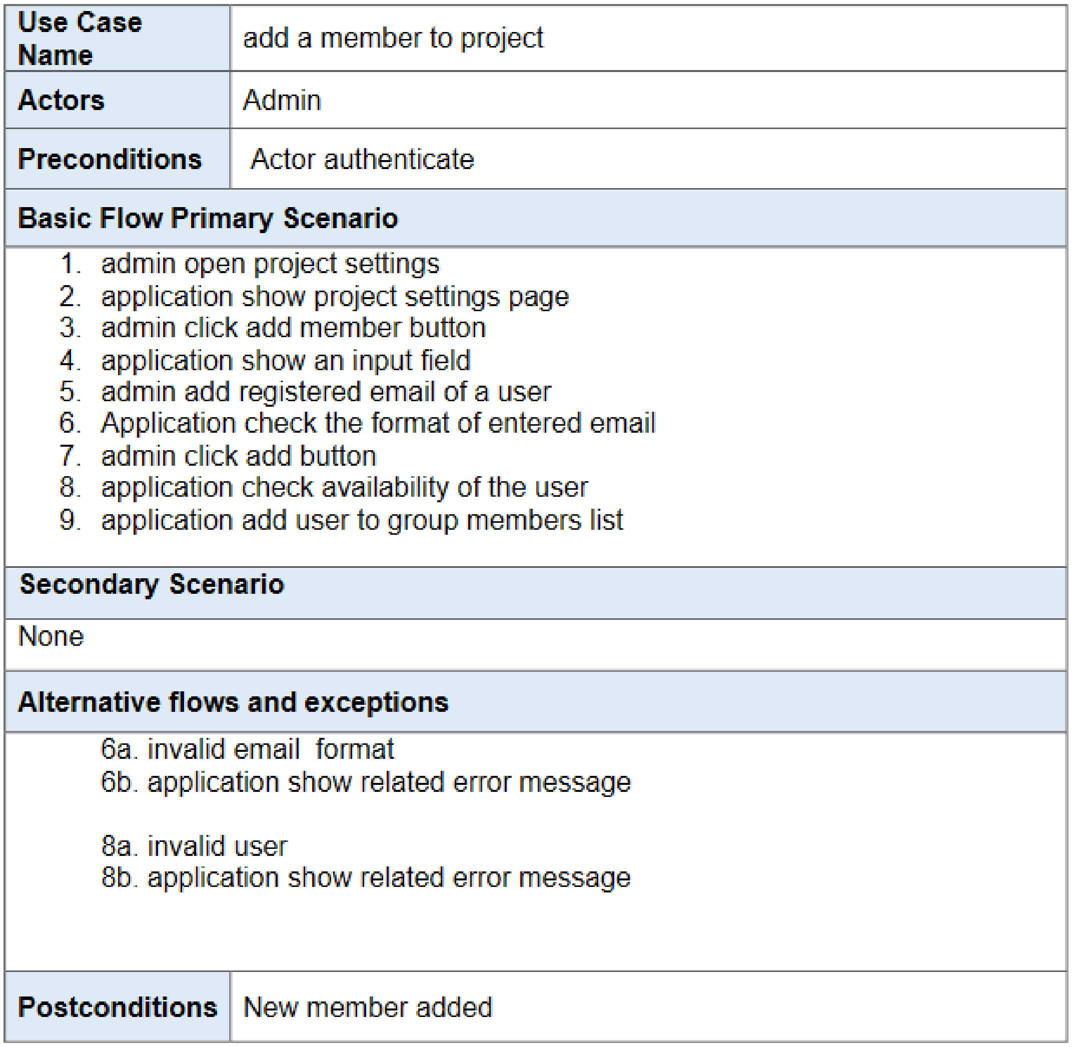
4.6.5 Use cases textual description

For more details we use Use case textual description tables:



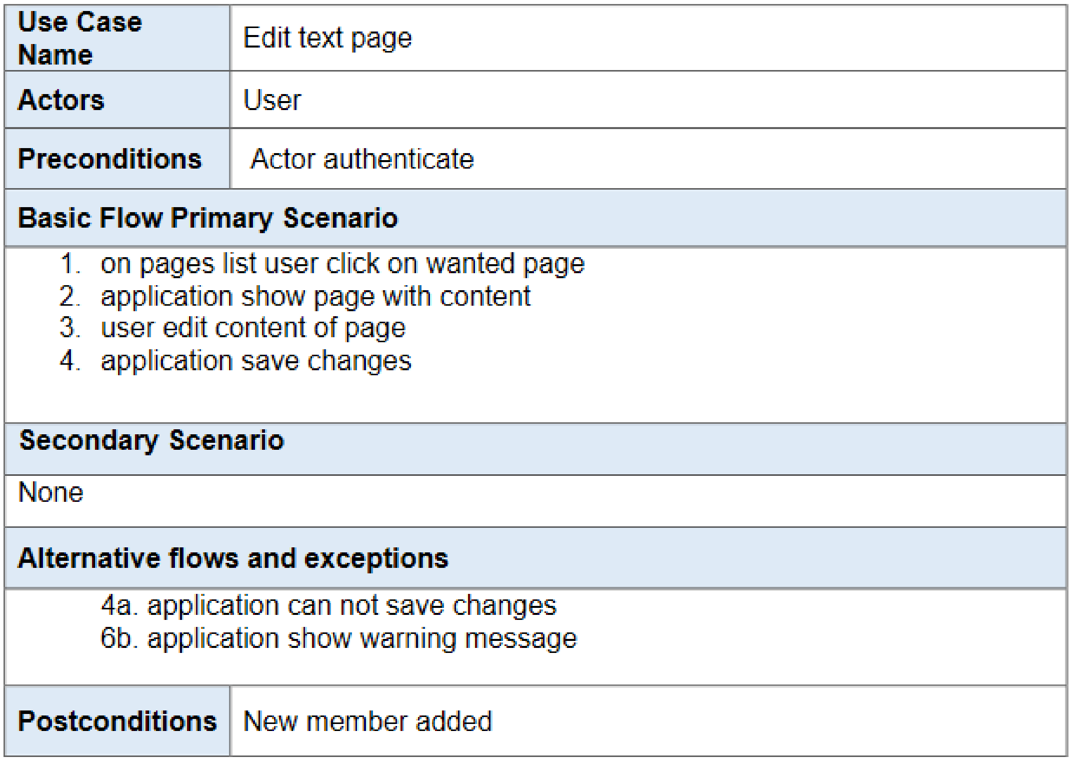
Use Case Name	Authenticate
Actors	Unknown , Guest
Preconditions	application runs properly
Basic Flow Primary Scenario	
1. Actor open Sign in page 2. Application displays login form 3. Actor enter email and password 4. Application check the format of entered inforamation 5. Actor click login button 6. Application validate entered information 7. Application redirect to private space page	
Secondary Scenario	
A. Actor click sgin up with Google button B. Application validate account information C. Application redirect to private space page	
Alternative flows and exceptions	
4a. invalid information format 4b. application show related error message 5a. invalid user 5b. application show related error message Ba. invalid user Bb. application show related error message	
Postconditions	Actor authenticated

Figure 4.5 – Description table : Authenticate



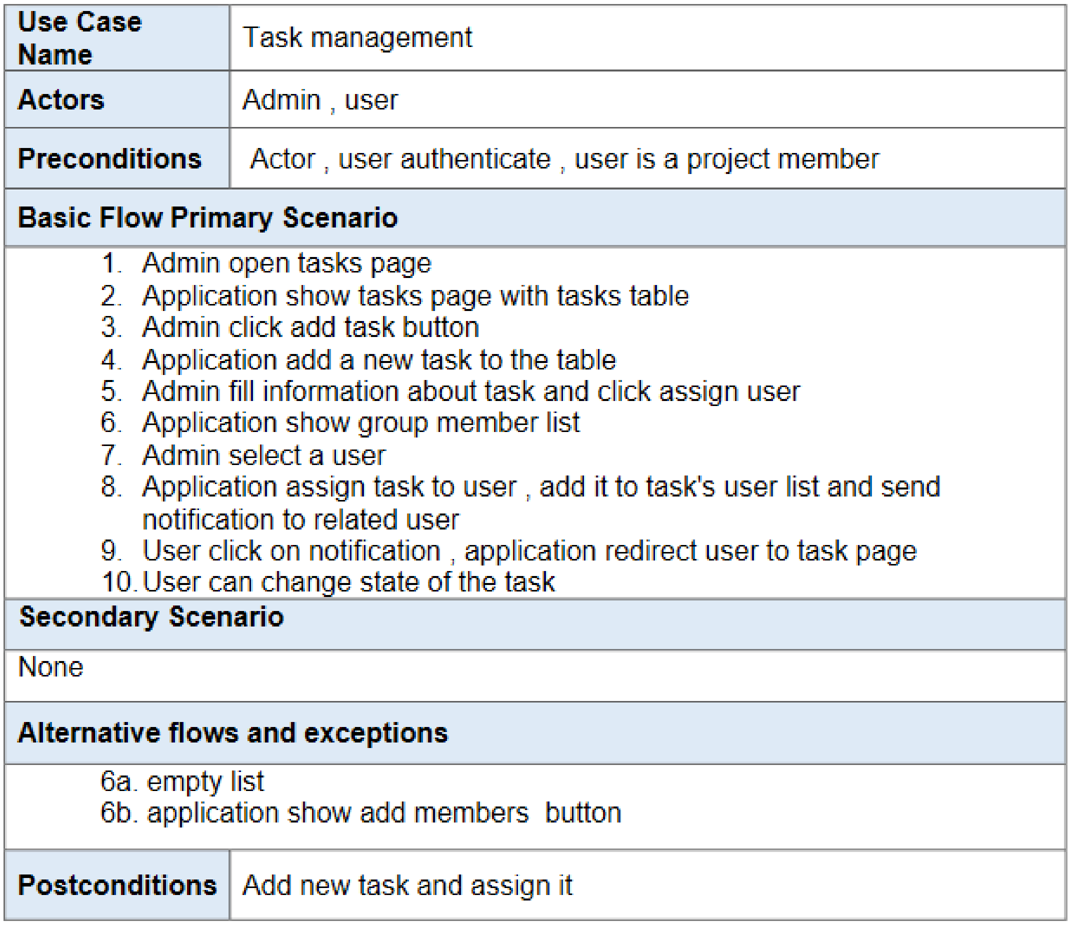
Use Case Name	add a member to project
Actors	Admin
Preconditions	Actor authenticate
Basic Flow Primary Scenario	
1. admin open project settings 2. application show project settings page 3. admin click add member button 4. application show an input field 5. admin add registered email of a user 6. Application check the format of entered email 7. admin click add button 8. application check availability of the user 9. application add user to group members list	
Secondary Scenario	
None	
Alternative flows and exceptions	
6a. invalid email format 6b. application show related error message 8a. invalid user 8b. application show related error message	
Postconditions	New member added

Figure 4.6 – Description table : Add memeber to a project



Use Case Name	Edit text page
Actors	User
Preconditions	Actor authenticate
Basic Flow Primary Scenario	
1. on pages list user click on wanted page 2. application show page with content 3. user edit content of page 4. application save changes	
Secondary Scenario	
None	
Alternative flows and exceptions	
4a. application can not save changes 6b. application show warning message	
Postconditions	New member added

Figure 4.7 – Description table : Edit text page



Use Case Name	Task management
Actors	Admin , user
Preconditions	Actor , user authenticate , user is a project member
Basic Flow Primary Scenario	
1. Admin open tasks page 2. Application show tasks page with tasks table 3. Admin click add task button 4. Application add a new task to the table 5. Admin fill information about task and click assign user 6. Application show group member list 7. Admin select a user 8. Application assign task to user , add it to task's user list and send notification to related user 9. User click on notification , application redirect user to task page 10. User can change state of the task	
Secondary Scenario	
None	
Alternative flows and exceptions	
6a. empty list 6b. application show add members button	
Postconditions	Add new task and assign it

Figure 4.8 – Description table : Task managment

4.6.6 Non Functional Requirements

4.6.6.1 Performance

It’s important to focus on the performance of the applications, which should run smoothly with optimal response times. For a good performance:

- write clean code without duplicate code lines and files.
- code logic of the application must be in line with the right approaches.
- use module bundles to minify files and dependencies.

4.6.6.2 Availability

A main requirement of our application is availability, the application should available to use all time with fewer downtimes.

4.6.6.3 Usability

User Experience (UX) and User Interface (UI) components design should be clear and clean, help user to quickly find what he wants, easy and familiar interface, clear navigation and visual logic that provide meaningful and relevant experiences to users.

4.6.6.4 Security

The application should provide a secure environment, protect user data and private spaces.

- Strong password: obligate users to enter long passwords with different types of characters and symbols.
- Provide Automatic sign-in and sign-out with other accounts like Google account.
- use multi-tenant architectures to separate users data.
- set rules for authorization (ex :an image in page in project space, only group member can see it).

4.7 Summary

In this chapter we start build our application, we first present our idea, then we defined the software requirements, with also present use case diagrams of some system's use case.

Chapter 5

Design

5.1 Overview

After the analyses phase, we pass to the design phase. In this phase, we transform the requirements as described in the analyses phase to system functionalities. In this phase, we define the basic system classes, data modeling, and deployment model.

5.2 Sequence diagram

A sequence diagram is a Unified Modeling Language (UML) diagram that illustrates object interactions arranged in time sequence in a use case in the field of software engineering. It depicts the objects involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.
we use sequence diagram to detail the scenarios, following some use cases :

5.2.1 Authentication

Authentication is an important component in the application, the user can access data and the application only when he authenticates, first he needs to create an account in our system (Sign up), then the Auth services give the user a unique ID token, to track him when he opens the application on the browser used to sign up or sign in, so whenever the user opens the application, the Auth service verify his ID token if the token is valid the user access the application, if not, he will be redirected to the sign-in page.

5.2.1.1 Sign up

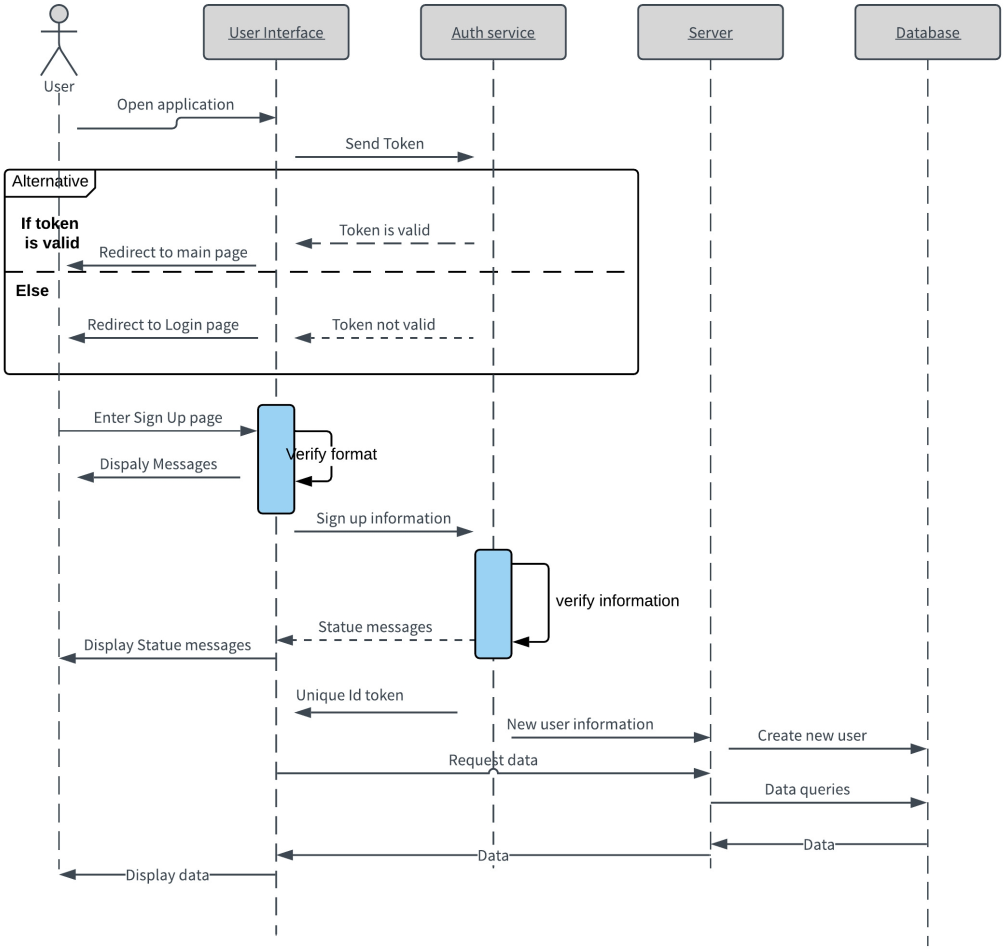


Figure 5.1 – sequence diagram: Sign up

5.2.1.2 Sign in

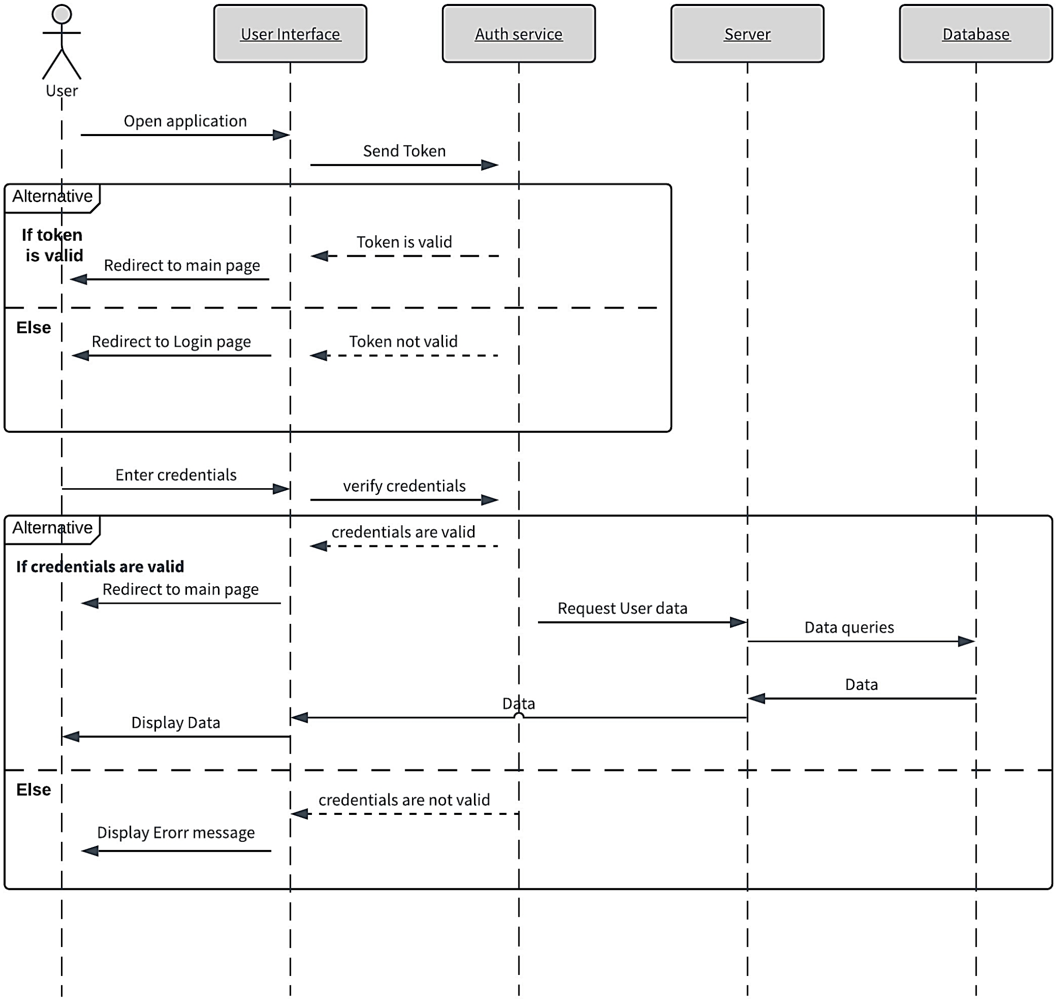


Figure 5.2 – sequence diagram:Sign in

5.2.2 Guest

Without authentication, a guest can see the content of a page if only it is a public page.

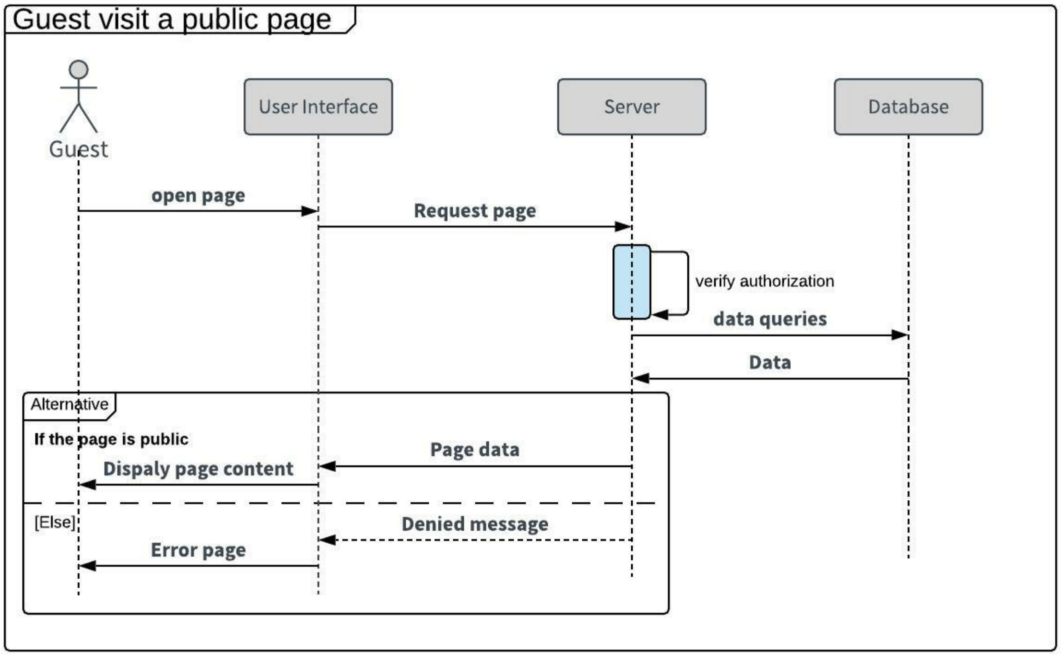


Figure 5.3 – sequence diagram: Guest visit a public page

5.2.3 Edit text page

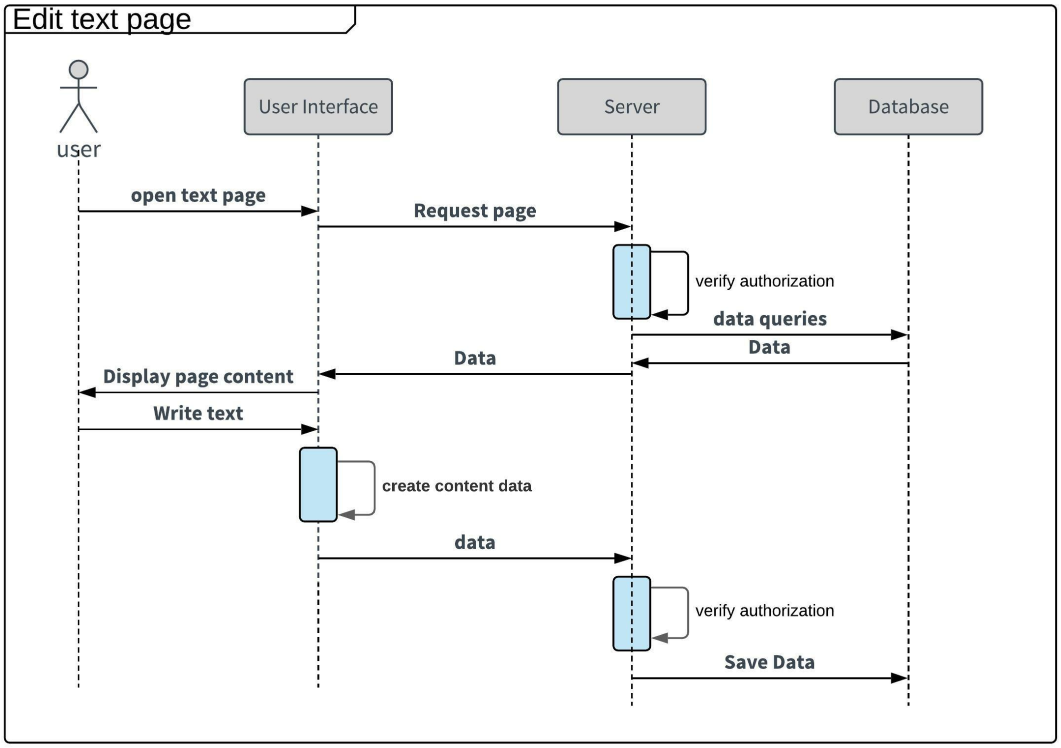


Figure 5.4 – sequence diagram:Edit text page

5.2.4 Upload file

Users can upload files only on a folder

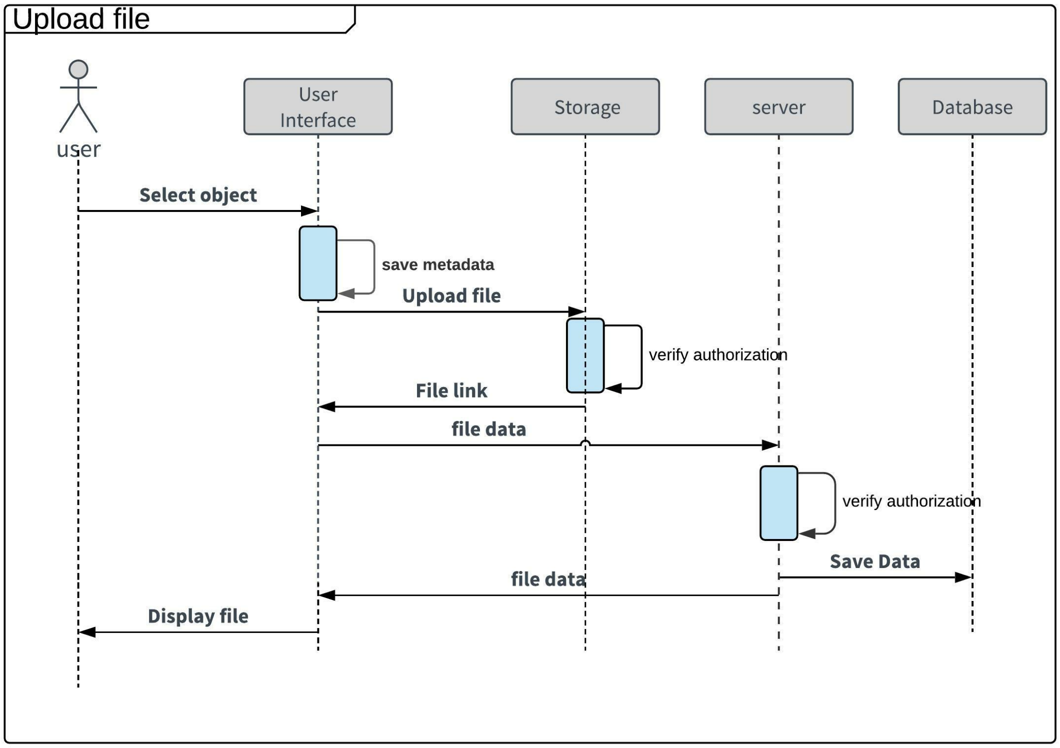


Figure 5.5 – sequence diagram:Upload file

5.2.5 Add member to a project

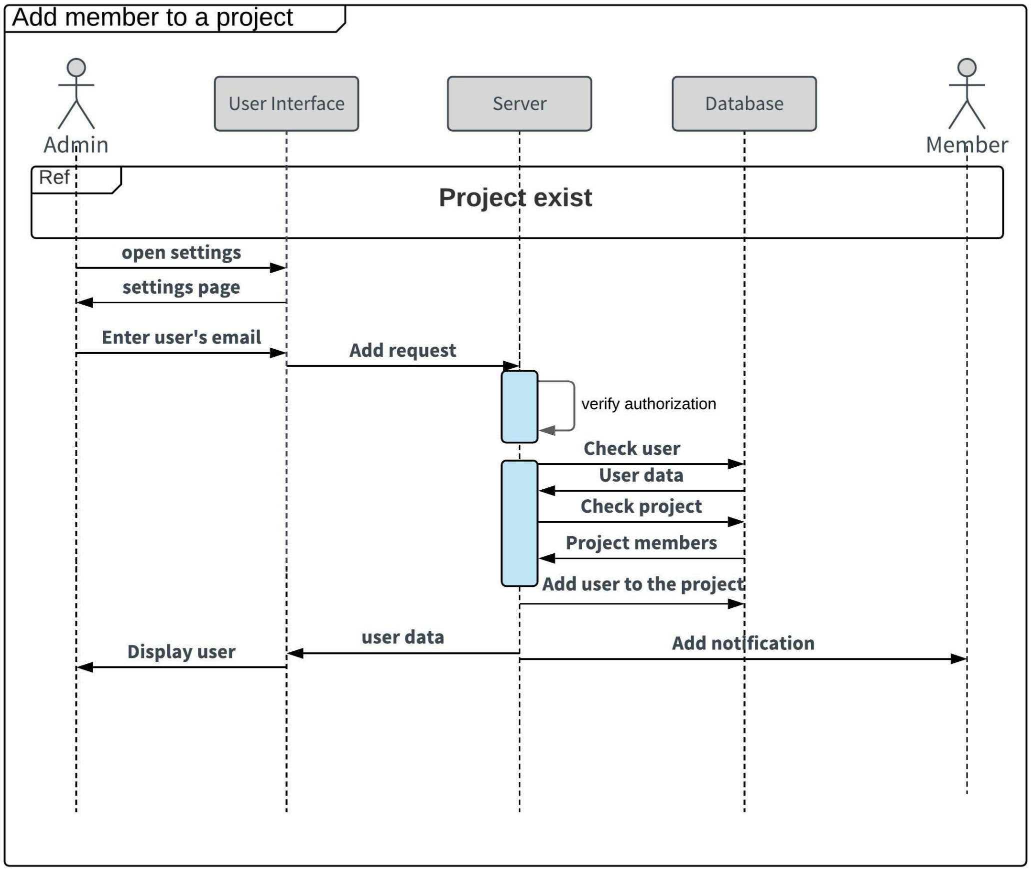


Figure 5.6 – sequence diagram:Add member to a project

5.2.6 Task management

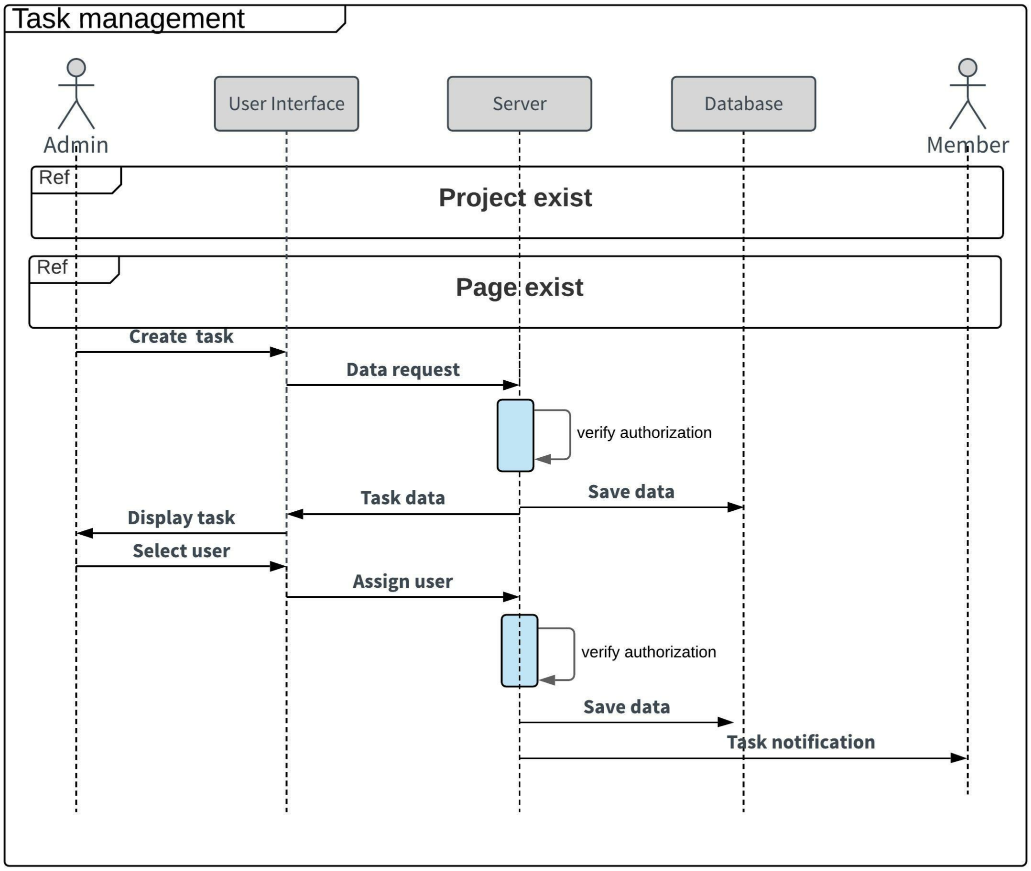


Figure 5.7 – Sequence diagram: Task management

5.3 Class diagram

In the Unified Modeling Language (UML), a class diagram is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations, and the relationships among objects.

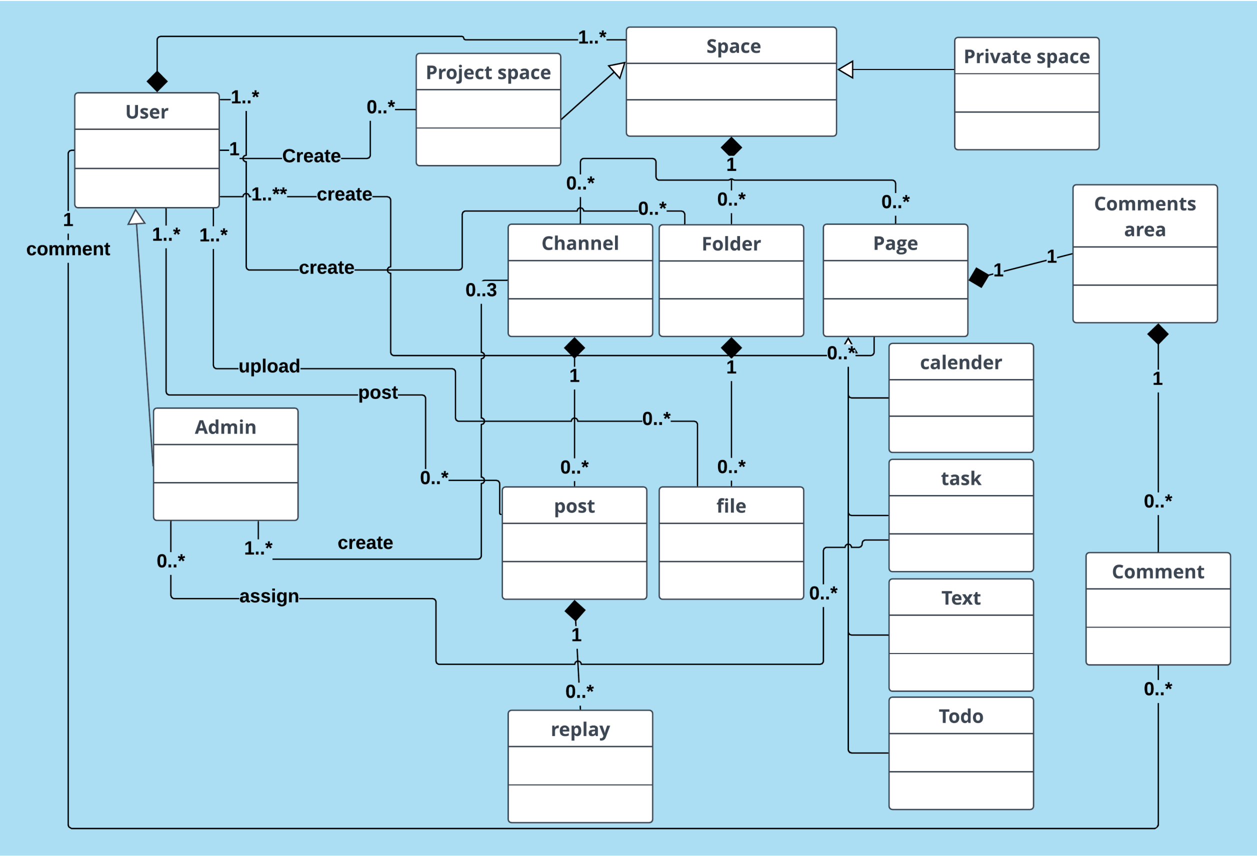


Figure 5.8 – class diagram

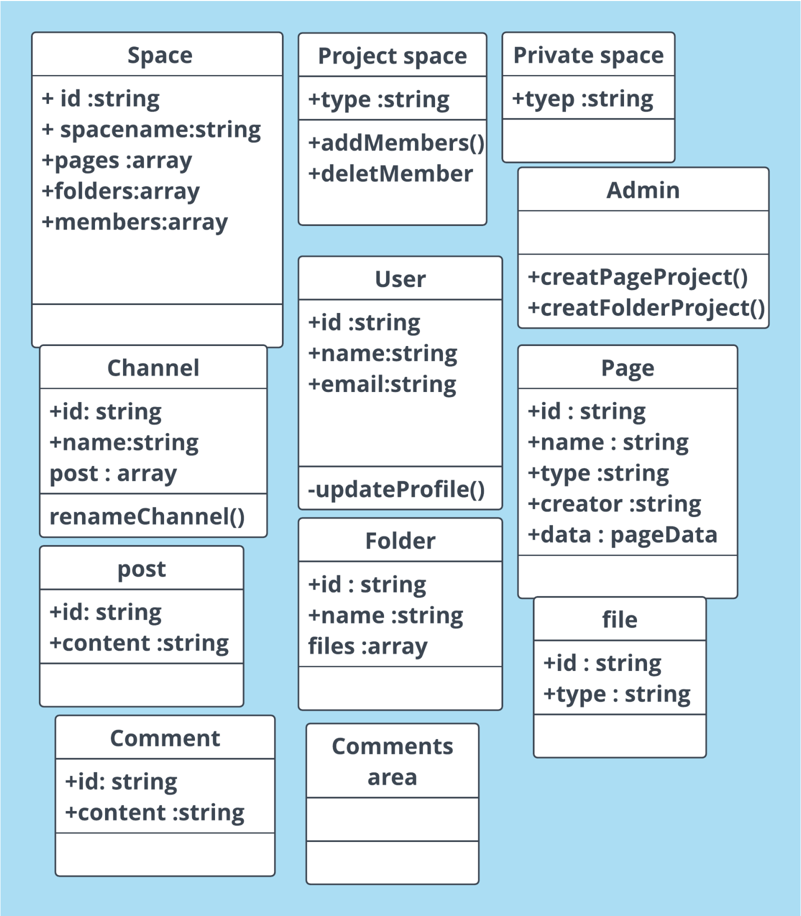
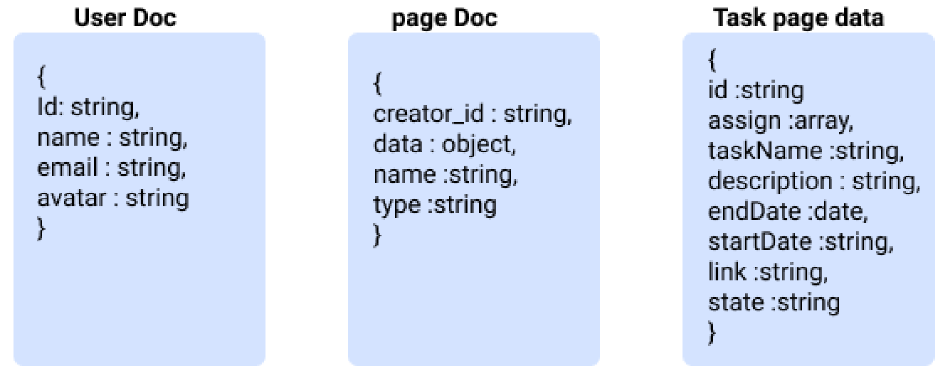


Figure 5.9 – classes

5.4 Data Modeling

We use NoSQL Document database to save application data, we use the shared database shared schema model, following JSON docs describe how we structure data:



User Doc

```
{
Id: string,
name : string,
email : string,
avatar : string
}
```

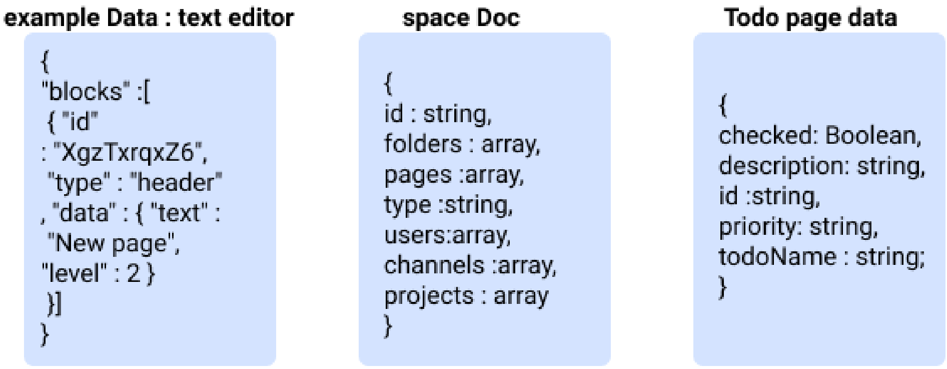
page Doc

```
{
creator_id : string,
data : object,
name :string,
type :string
}
```

Task page data

```
{
id :string
assign :array,
taskName :string,
description : string,
endDate :date,
startDate :string,
link :string,
state :string
}
```

Figure 5.10 – JSON DOC 1



example Data : text editor

```
{
"blocks" :[
 { "id"
: "XgzTxrqxZ6",
 "type" : "header"
, "data" : { "text" :
 "New page",
"level" : 2 }
 }]
}
```

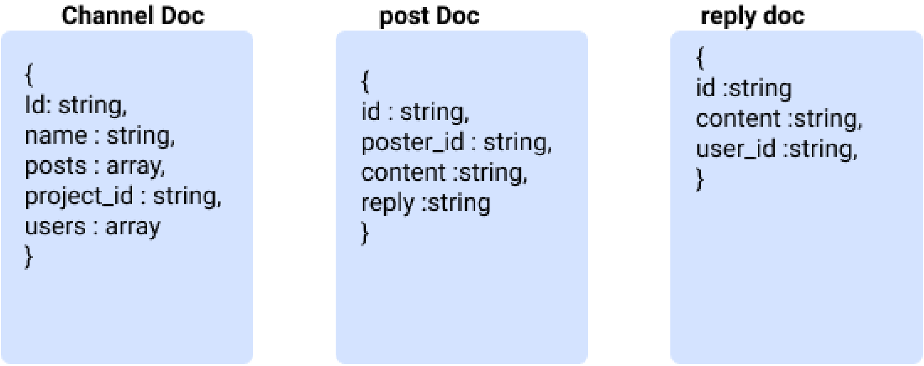
space Doc

```
{
id : string,
folders : array,
pages :array,
type :string,
users:array,
channels :array,
projects : array
}
```

Todo page data

```
{
checked: Boolean,
description: string,
id :string,
priority: string,
todoName : string;
}
```

Figure 5.11 – JSON DOC 2



Channel Doc

```
{
Id: string,
name : string,
posts : array,
project_id : string,
users : array
}
```

post Doc

```
{
id : string,
poster_id : string,
content :string,
reply :string
}
```

reply doc

```
{
id :string
content :string,
user_id :string,
}
```

Figure 5.12 – JSON DOC 3

5.5 Deployment architecture

We use microservices to develop and deploy the application, microservices is an approach to developing a single application as a suite of small services, each running in its own process and communicating with lightweight mechanisms. built around business capabilities and independently deployable by fully automated deployment machinery. In our case, we deploy our application to the main service and other services used and called on-demand, following deployment architecture:

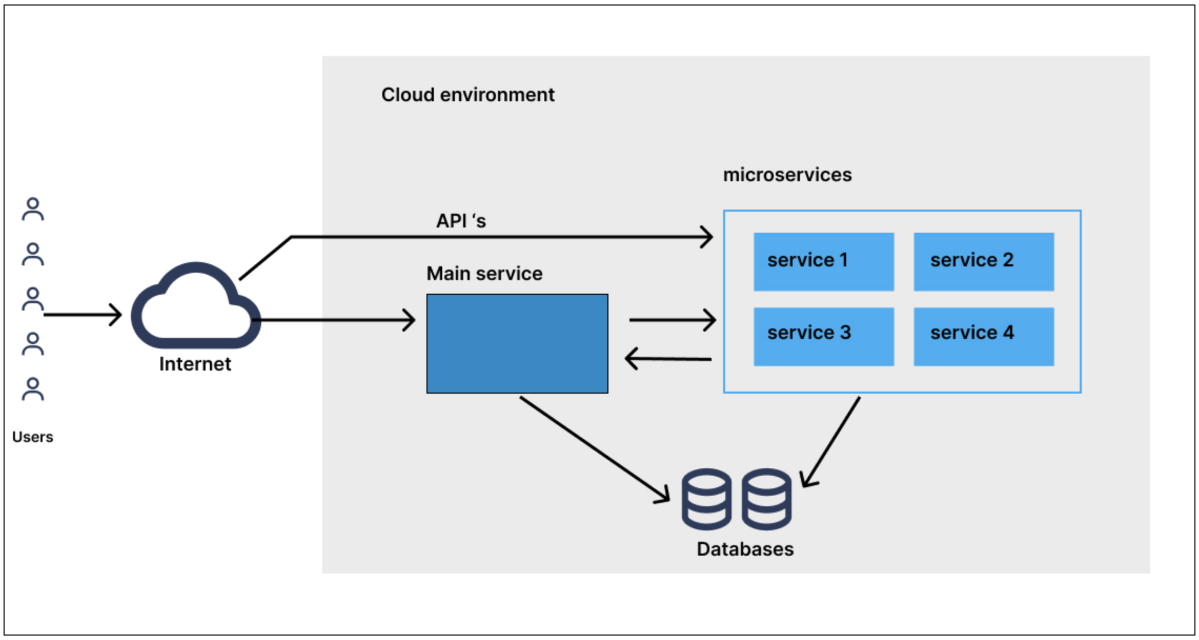


Figure 5.13 – Deployment architecture

Chapter 6

Implementation

6.1 Overview

In the implementation phase we move our application to the production stage, in this phase we start coding and building our application, in this chapter we present technologies that we used, configurations, and how we connect services. Also in this chapter, we show how cloud technologies work and how we deal with them.

6.2 NoSQL Database

NoSQL databases are non-relational databases, non-tabular, and store data differently than relational tables. NoSQL databases come in a variety of types based on their data model. The main types are document, key-value, wide-column, and graph. They provide flexible schemas and scale easily with large amounts of data and high user loads.

NoSQL Document databases are a way of store data in documents similar to JSON (JavaScript Object Notation) objects. Each document contains pairs of fields and values. The values can typically be a variety of types including things like strings, numbers, booleans, arrays, or objects, and their structures typically align

with objects developers are working within code. Because of their variety of field value types and powerful query languages, document databases are great for a wide variety of use cases and can be used as a general purpose database. They can horizontally scale out to accommodate large data volumes.
following are some points that led us to chose the NoSQL document database:

- Development process with NoSQL databases can be much faster than with a SQL database
- NoSQL database fit with our development methodology and fit with modern Agile development practices
- NoSQL databases are often better suited to storing and modeling
- Our application data are unstructured data (Multimedia content, Text files, various values types.. etc). NoSQL databases are the ideal choice since they can more easily adapt their data model with a flexible schema

6.3 Cloud Computing platform :

We develop a SaaS application so we need a Cloud Computing environment to deploy our solution, we decide to use Google Cloud Platform, GCP is a public cloud vendor provided by Google it is a suite of Cloud Computing services that runs on the same infrastructure that Google uses internally for its end-user products, such as Google Search, Gmail, file storage, and YouTube.
GCP provides all product needed to build and deploy high performance and scaled cloud-based application like database, storage, serverless platforms, load balancing, auto-scale.. etc, we chose Google cloud platform because :

- Very rich and detailed documentations with a massive community are available
- It provides high quality products that we need in our development process
- Easy to understand and deploy applications compare to AWS and Azure
- It provides a console that allows for easy access to all GCP projects and products. A customizable project dashboard, with an overview of Google Cloud resources, billing, and a filterable activity listing.

— Free use plan and free trial that includes $300 in credits to spend over all products

Following Google cloud product and services that we use :

6.3.1 Google App Engine

We use App Engine to deploy our main application. Google App Engine is a Cloud Computing platform as a service (PaaS) for developing and hosting web applications in Google-managed data centers and run across multiple servers. App Engine is a fully managed, serverless platform we can choose from several popular languages, libraries, and frameworks to develop our apps, then App Engine takes care of provisioning servers and scaling the app instances based on demand.
for auto-scaling App Engine can automatically create and shut down instances as traffic load changes, basically An instance is a virtual copy of the main server or a virtual server that runs the same code that we deployed, also we can specify the number of instances to run regardless of the amount of traffic. To determine how and when new instances are created, we specify a scaling type for the application.
An App Engine application is made up of a single application resource that consists of one or more services. Within each service, we can deploy different versions of that service. Each version then runs within one or more instances, depending on the traffic or our configuration.

6.3.2 Firestore

Cloud Firestore is a cloud-hosted NoSQL document database, it is flexible, scalable for mobile, web, and server development from Firebase and Google Cloud. It keeps data in sync across client apps through real-time listeners and offers offline support for mobile and web, we can build responsive apps that work regardless of network latency or Internet connectivity. Cloud Firestore offers seamless integration with other Firebase and Google Cloud products, including Cloud Functions. Cloud Firestore also features richer, faster, and simple queries to manage data.

6.3.3 Cloud storage

Cloud Storage is a service for storing objects in Google Cloud. An object is an immutable piece of data consisting of a file of any format. we store objects in containers called buckets. All buckets are associated with a project, and we can group projects under an organization.
Cloud Storage is a flexible, scalable, and durable storage option we can read and write files to Cloud Storage buckets from almost anywhere, so we can use buckets as common storage between our services and main application in App Engine. all user's files will be saved on cloud storage.

6.3.4 Cloud Functions

Google Cloud Functions is a serverless execution environment for building and connecting cloud services. With Cloud Functions, we write simple, single-purpose functions that are attached to events emitted from or services. A function is triggered when an event being watched is fired. Our code executes in a fully managed environment. There is no need to provision any infrastructure or worry about managing any servers. we can deploy a complete service using Cloud Function.

6.3.5 Identity platform

Identity Platform is a customer identity and access management (CIAM) platform that helps organizations add identity and access management functionality to their applications, protect user accounts, and scale with confidence on Google Cloud.

6.4 Programming Languages and Libraries Used

6.4.1 JavaScript

JavaScript (JS) is a lightweight, interpreted, or just-in-time compiled programming language with first-class functions. While it is most well-known as the scripting language for Web pages, many non-browser environments also use it, such as Node.js, Apache CouchDB, and Adobe Acrobat. JavaScript is a prototype-based, multi-paradigm, single-threaded, dynamic language, supporting object-oriented, imperative, and declarative (e.g. functional programming) styles.

6.4.2 Node js

Node.js is an open-source, cross-platform runtime environment that allows developers to create all kinds of server-side tools and applications in JavaScript. The runtime is intended for use outside of a browser context (i.e. running directly on a computer or server OS). As such, the environment omits browser-specific JavaScript APIs and adds support for more traditional OS APIs including HTTP and file system libraries. Node js is so powerful and very popular an used by many big companies like : Netflix, PayPal, Uber... etc

6.4.3 Express js

Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications, Express provides a thin layer of fundamental web application features, without obscuring Node.js features.

6.4.4 React

React is a free and open-source front-end JavaScript library for building user interfaces and UI components. It is developed by Facebook and developers community. React makes it painless to create interactive UIs. Design simple views for each state in the application, and React will efficiently update and render just the right components when application data changes.

6.4.5 CSS

Cascading Style Sheets (CSS) is a simple mechanism for adding style (e.g., fonts, colors, spacing) to web documents.

6.4.6 Firebase Admin SDK

The Admin SDK is a set of server libraries that lets us interact with Firebase from privileged environments to perform actions like:
- Read and write with full admin privileges.
- Programmatically send Firebase Cloud Messaging messages using a simple, alternative approach to the Firebase Cloud Messaging server protocols.
- Generate and verify Firebase auth tokens.
- Access Google Cloud resources like Cloud Storage buckets and Cloud Firestore databases associated with google cloud projects.

6.4.7 Editor js

Editor.js is a block-styled editor for rich media stories. It outputs clean data in JSON instead of heavy HTML-markup. And more important thing is that Editor.js is designed to be API extendable and pluggable.

6.5 Tools and extensions

6.5.1 Google Cloud Console

Google cloud console (GCC) is a powerful web admin UI, GCC is a web-based interface that helps us to quickly find resources and connect to instances via SSH in the browser. access and manage google cloud products, GCC helps us to have full management control over resources that we use.

6.5.2 VS Code

Visual Studio Code is a lightweight but powerful source code editor which runs on desktop and is available for Windows, macOS, and Linux. It comes with built-in

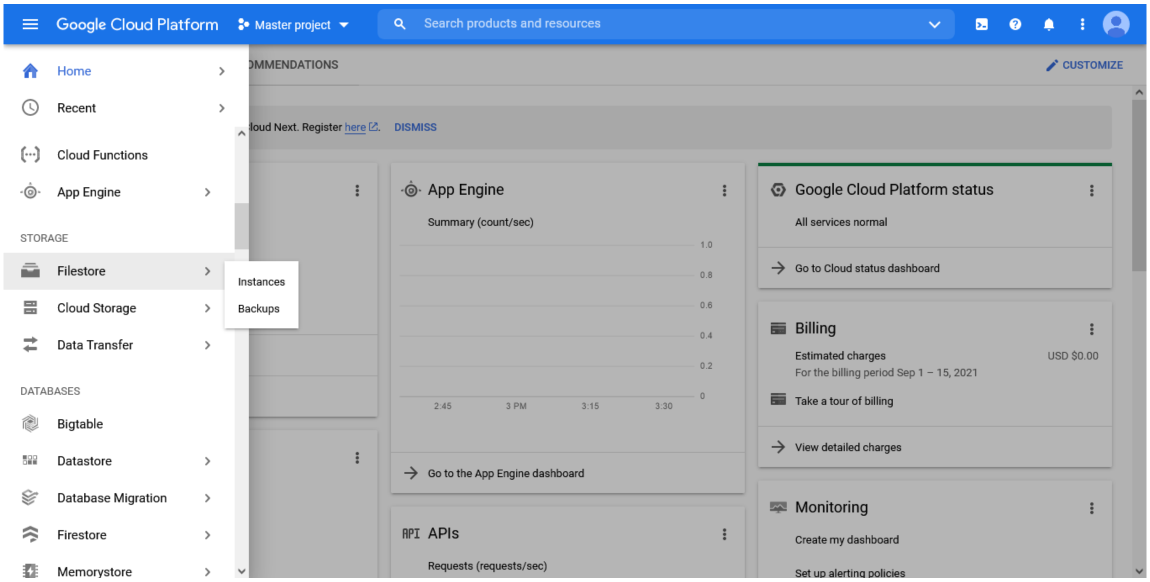


Figure 6.1 – Google Cloud Console

support for JavaScript, TypeScript, and Node.js and has a rich ecosystem of extensions for other languages (such as C++, C, Java, Python, PHP, Go) and runtimes (such as. NET and Unity).
One of the best features of VS Code is the ability to install extensions that were very useful when we write code, it helps us coding very faster and it makes the code very clean and super organized.
extensions:
VS Code ES7 React/Redux/React-Native/JS snippets, auto close tag, Bracket Pair Colorizer, beautfiy.. etc

6.5.3 NPM

Npm is the world's largest software registry. Open source developers from every continent use Npm to share and borrow packages. Npm is a package manager for the JavaScript programming language maintained by Npm, Inc. Npm runs with Node.js. all packages and libraries we use in this project are installed and managed using Npm and its CIL (command line interface) that execute in the windows terminal.

6.5.4 Nodemon

Nodemon is a tool that helps develop node.js based applications by automatically restarting the node application when file changes in the directory are detected. its a very useful tools, it helps us a lot in the coding process

6.5.5 Webpack

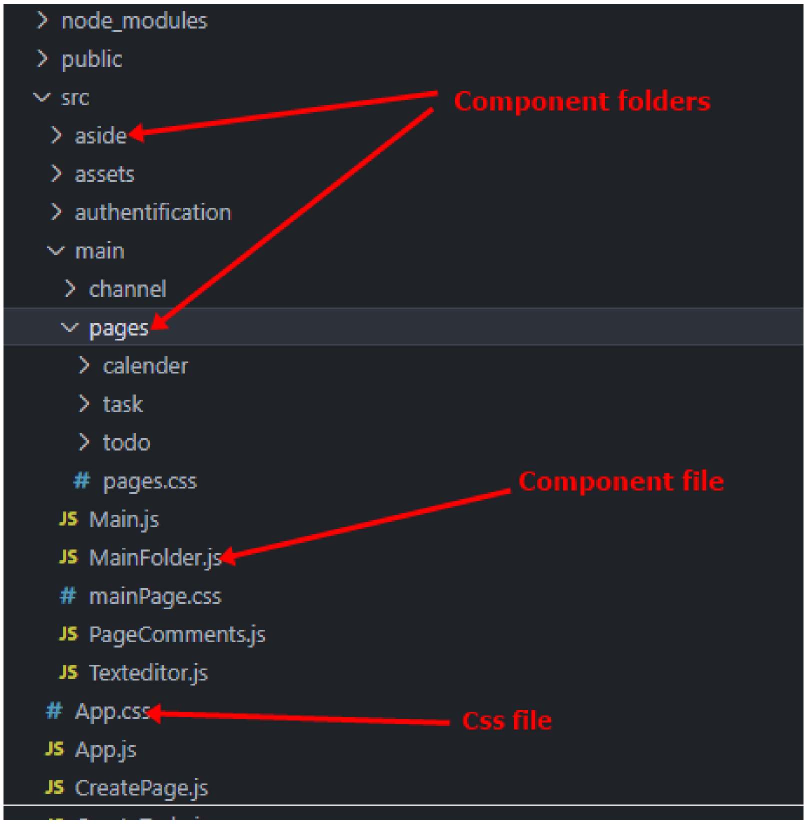
Webpack is a static module bundler for modern JavaScript applications. When webpack processes the application, it internally builds a dependency graph from one or more entry points and then combines every module your project needs into one or more bundles, which are static assets to serve content from. in other words webpack is a tool to bundle, minify and squeeze all files, modules, dependencies and create just a few files, webpack is very powerful, and have many benefits: improve performance, Automatic Dependency Collection...

6.6 File Structure

file structure is the way that we structure application files, the well structured and organized files helps us in the development process.

6.6.1 Front-end files

front-end applications files are grouped based on their type as shown in Figure 6.2



```
> node_modules
> public
v src
  > aside
  > assets
  > authentification
  v main
    > channel
    v pages
      > calender
      > task
      > todo
      # pages.css
    JS Main.js
    JS MainFolder.js
    # mainPage.css
    JS PageComments.js
    JS Texteditor.js
  # App.css
  JS App.js
  JS CreatePage.js
```

Component folders

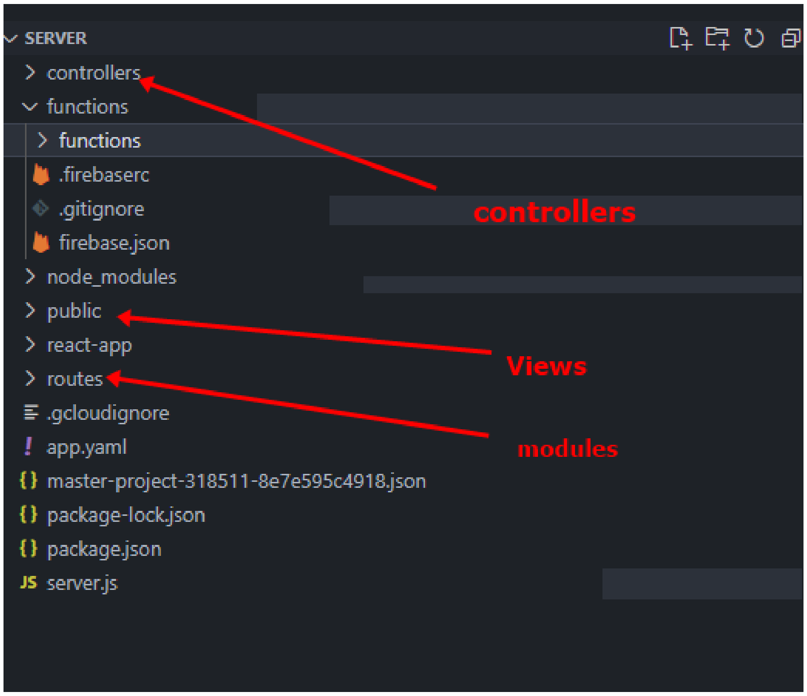
Component file

Css file

Figure 6.2 – Front-end app files structure

6.6.2 Back-end

In the main back-end we follow the MVC methodology (Model–view–controller) it is a way of structuring our code and files, In the MVC approach application files are divided into three interconnected parts, Model, View, and Controller, this approach keeps code more modular, reusable and easier to read.



```
SERVER
  > controllers
  v functions
    > functions
      .firebaserc
      .gitignore
      firebase.json
  > node_modules
  > public
  > react-app
  > routes
    .gcloudignore
    app.yaml
    master-project-318511-8e7e595c4918.json
    package-lock.json
    package.json
    server.js
```

Figure 6.3 – Back-end app files structure

6.7 Code and Solutions

6.7.1 Authentication

When a user opens the application, the authentication service code executes, if the browser contains a valid ID token, the user will be redirected to the main page of the application, if the ID token is unavailable or expire, the user will be redirected to the sign-in page. On the sign-in page, if the user enters the right credentials, the authentication service will save a new valid ID token on the browser storage, and allows the user to access the application

If the user hasn't an account, he needs to switch to the Sign-up page to create a new account, after he enters his information and clicks the sign-up button, the authentication service will create a new account, then a Cloud Function will invoke, this function can access the new user information, it will create new start data and save it with the user information (name) in Firestore database. The figure below summarized this process.

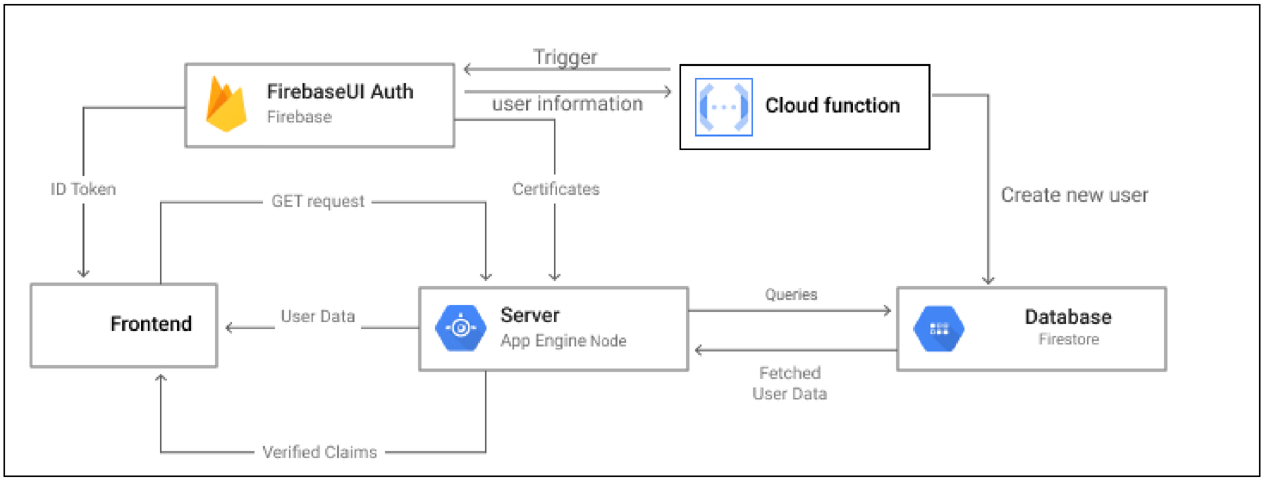
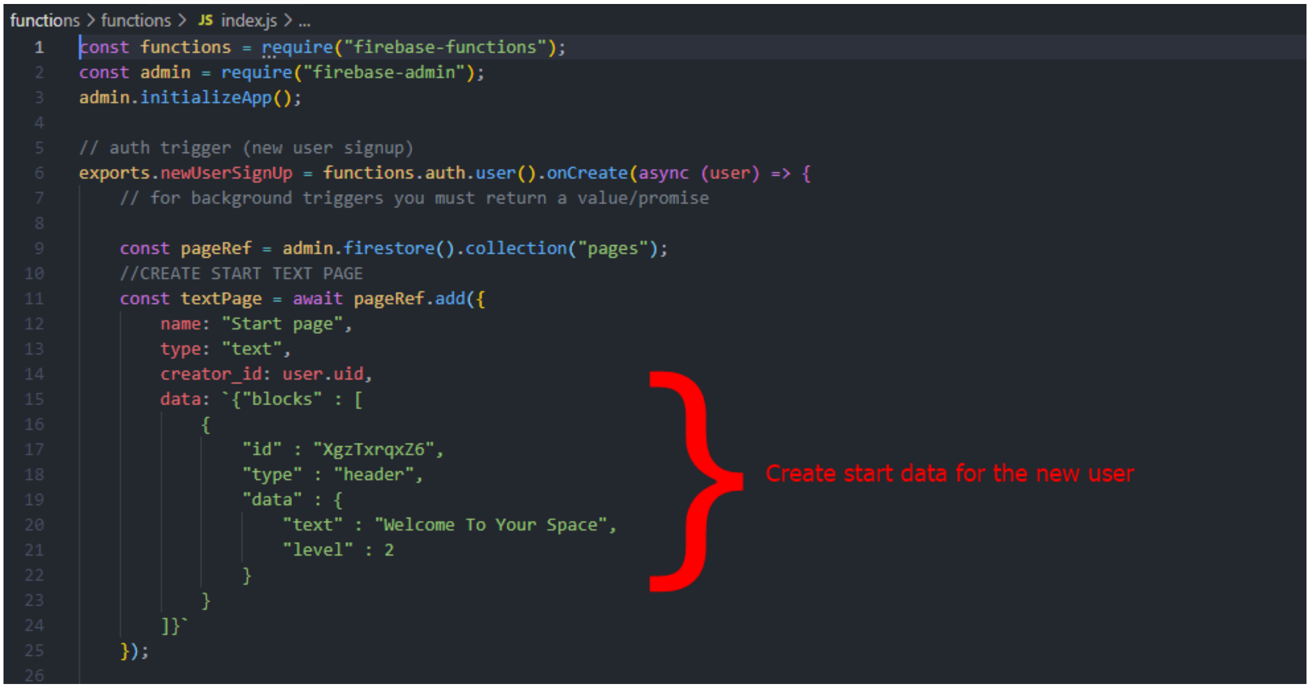


Figure 6.4 – Authenticate Service



```
const functions = require("firebase-functions");
const admin = require("firebase-admin");
admin.initializeApp();

// auth trigger (new user signup)
exports.newUserSignUp = functions.auth.user().onCreate(async (user) => {
    // for background triggers you must return a value/promise

    const pageRef = admin.firestore().collection("pages");
    //CREATE START TEXT PAGE
    const textPage = await pageRef.add({
        name: "Start page",
        type: "text",
        creator_id: user.uid,
        data: `{"blocks" : [
            {
                "id" : "XgzTxrqxZ6",
                "type" : "header",
                "data" : {
                    "text" : "Welcome To Your Space",
                    "level" : 2
                }
            }
        ]}`
    });

```

Create start data for the new user

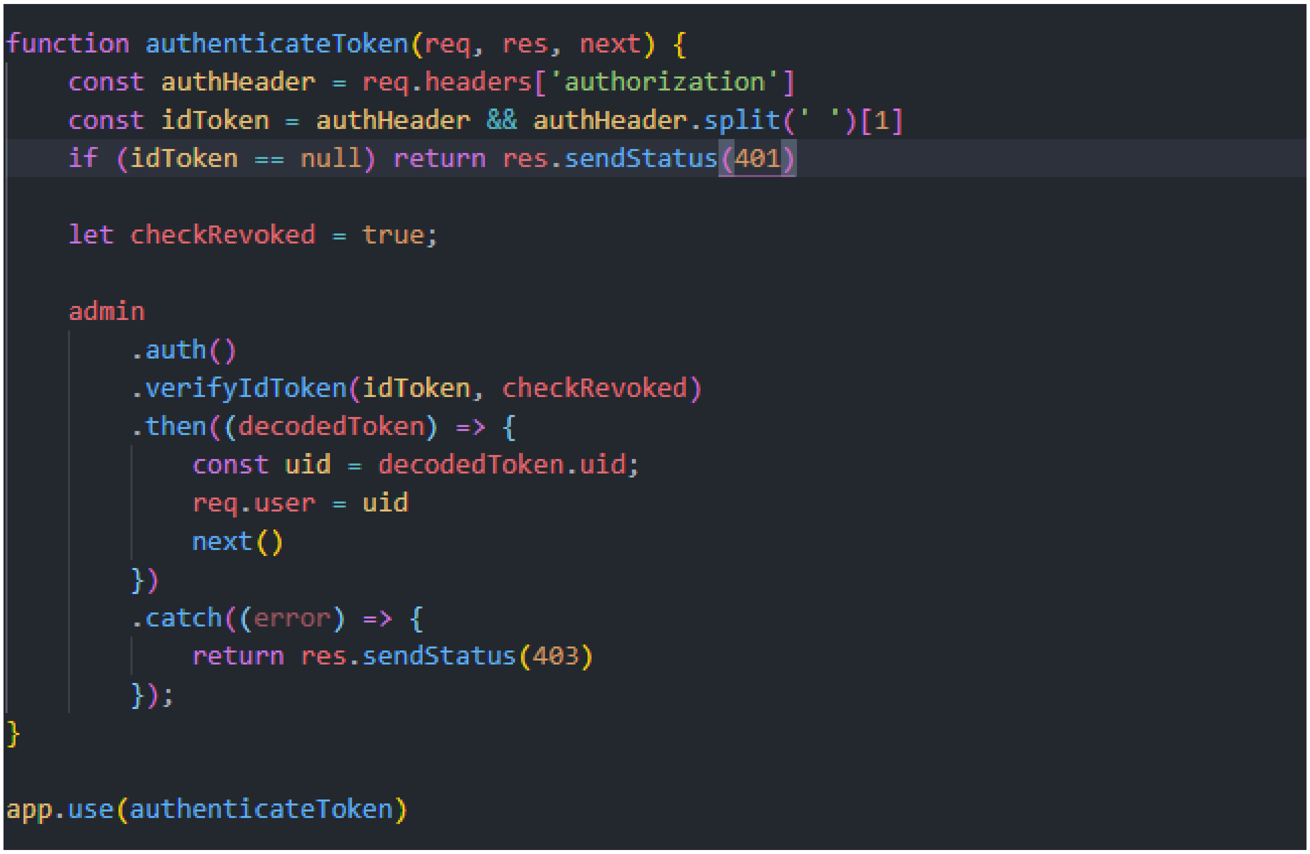
Figure 6.5 – Part of the code of authenticate trigger function

6.7.2 Authorization

As we saw in the analyses and design phase, to use our applications all users must authenticate (sign-up or sign-in), or all requests will be denied. When a user or device successfully sign up or sign in, the authentication service creates a corresponding ID token that uniquely identifies and grants users access to the application. The front-end application put an authorization header that contains the unique ID token in every POST and GET request sent by the user. In the back-end when it receives a request, first of all, it verifies the value of the authorization header (validation of ID token) using a Middleware function.

A Middleware is a function that invokes before the controller receives requests. In our back-end a Middleware function checks the validation of tokens using the authentication service, if the ID token is valid the Middleware passes the request to the controller to treat it, if not the Middleware sends back a forbidden response.

Following the source code of the Middleware



```
function authenticateToken(req, res, next) {
    const authHeader = req.headers['authorization']
    const idToken = authHeader && authHeader.split(' ')[1]
    if (idToken == null) return res.sendStatus(401)

    let checkRevoked = true;

    admin
        .auth()
        .verifyIdToken(idToken, checkRevoked)
        .then((decodedToken) => {
            const uid = decodedToken.uid;
            req.user = uid
            next()
        })
        .catch((error) => {
            return res.sendStatus(403)
        });
}

app.use(authenticateToken)
```

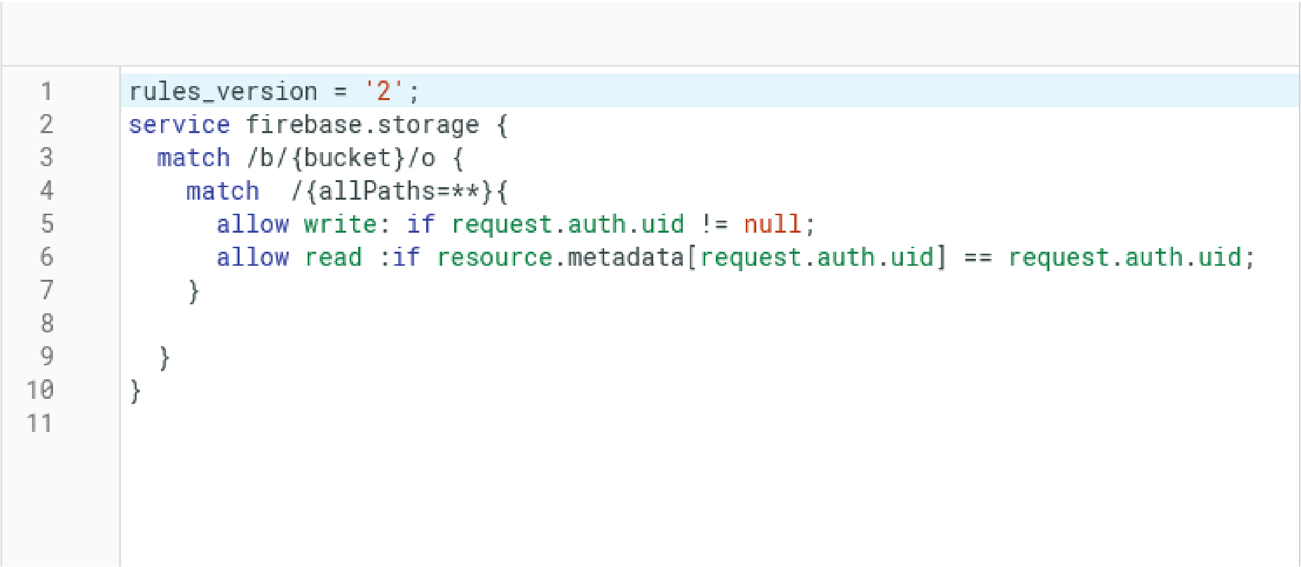
Figure 6.6 – Middleware source code

6.7.3 Files Access Authorization

Files access authorization was a hard task in the development process, when a user uploaded a file, the file saved by cloud storage in cloud buckets and return a link takes to the actual file, this link will be saved in the Firestore database as a string, the issue was with that link, anyone with this link can access to the file and this is a big security and privacy problem.

The solution to this problem is to use cloud storage rules and file metadata, Objects stored in Cloud Storage have metadata associated with them. Metadata identifies

properties of the object, as well as specifies how the object should be handled when it's accessed. So whenever a user uploads a file, the application in the front-end will save the user ID in the metadata of the file. If the file is uploaded in a project space the application will save all users IDs in the metadata (user ID are provided by authentication service).
On the other side, we configure Cloud storage rules to authorize access to the requested file only if the UID (user ID) matches the one saved in the metadata of the file. rules configuration in Figure 6.7



```
rules_version = '2';
service firebase.storage {
  match /b/{bucket}/o {
    match  /{allPaths=**}{
      allow write: if request.auth.uid != null;
      allow read :if resource.metadata[request.auth.uid] == request.auth.uid;
    }

  }
}
```

Figure 6.7 – files access rules

6.8 Deployment

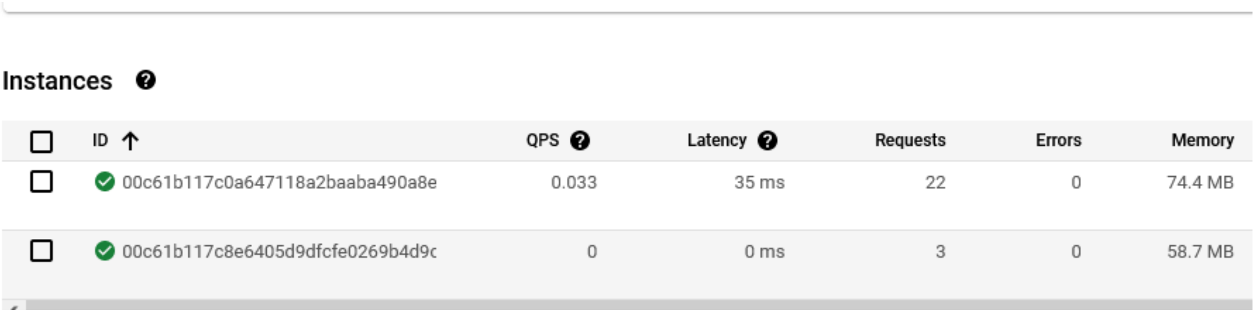
To deploy the application we use Cloud SDK CIL (or gcloud CIL). The gcloud command-line tool is the primary CLI tool to create, manage and deploy Google Cloud resources. The deployment is done according to the service and platform used.

6.8.0.1 Deploy main application to app engine

First of all, we pass our react application (front-end application) to the production mode, to do so we open the terminal in the root file of the application and then execute this command: **npm react-scripts build** , this command will invoke webpack, then the build file of our front-end application will be created. Next, we copy this file and past it in the public folder of the Node js application (main server), then in the root of the server folder, we execute the gcloud command: **gcloud app deploy**

6.9 Application scalability

The application automatically scales up and down, app engine will create new instances according to the traffic, it can scales up and hold even to a million requests without any overload problems in Figure 6.8 below show instances created automatically by App Engine, the images is from google cloud console of our project. Also, other services are auto-scale up and down without managing them.



Instances

□	ID ↑	QPS	Latency	Requests	Errors	Memory
□	✓ 00c61b117c0a647118a2baaba490a8e	0.033	35 ms	22	0	74.4 MB
□	✓ 00c61b117c8e6405d9dfcfe0269b4d9c	0	0 ms	3	0	58.7 MB

Figure 6.8 – screenshot from google cloud console shows app engine instances

6.10 The Final application

The final application is SaaS cloud-based application, multi-tenet architecture, high availability, automatically scale, and accessible via the internet. Following are some screenshots from the application:

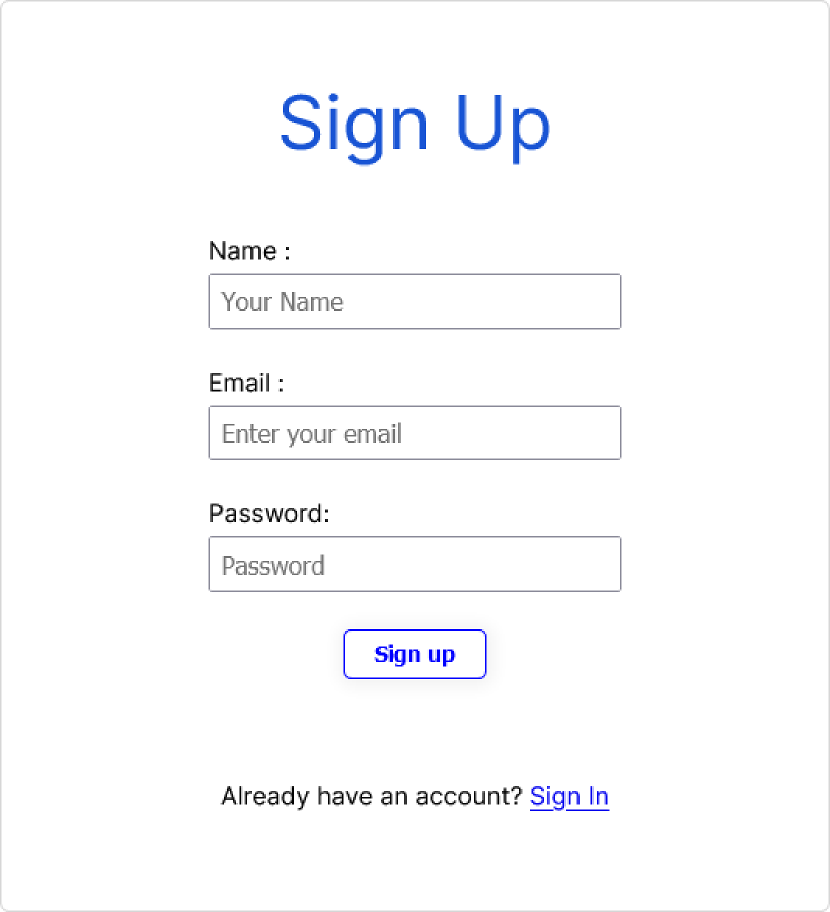
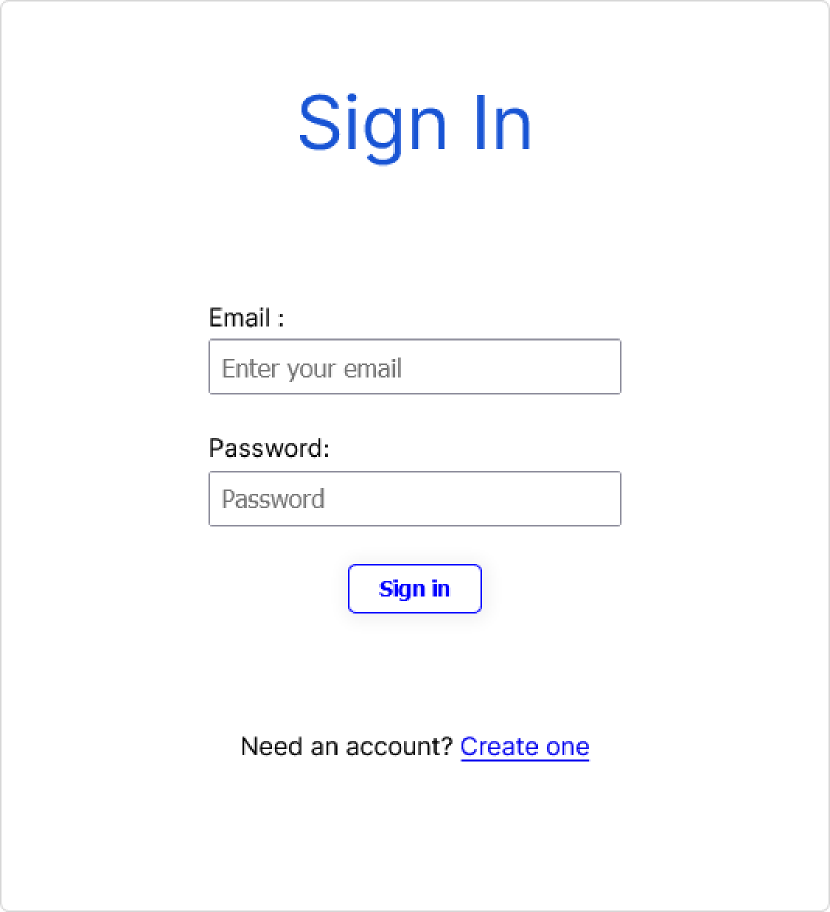
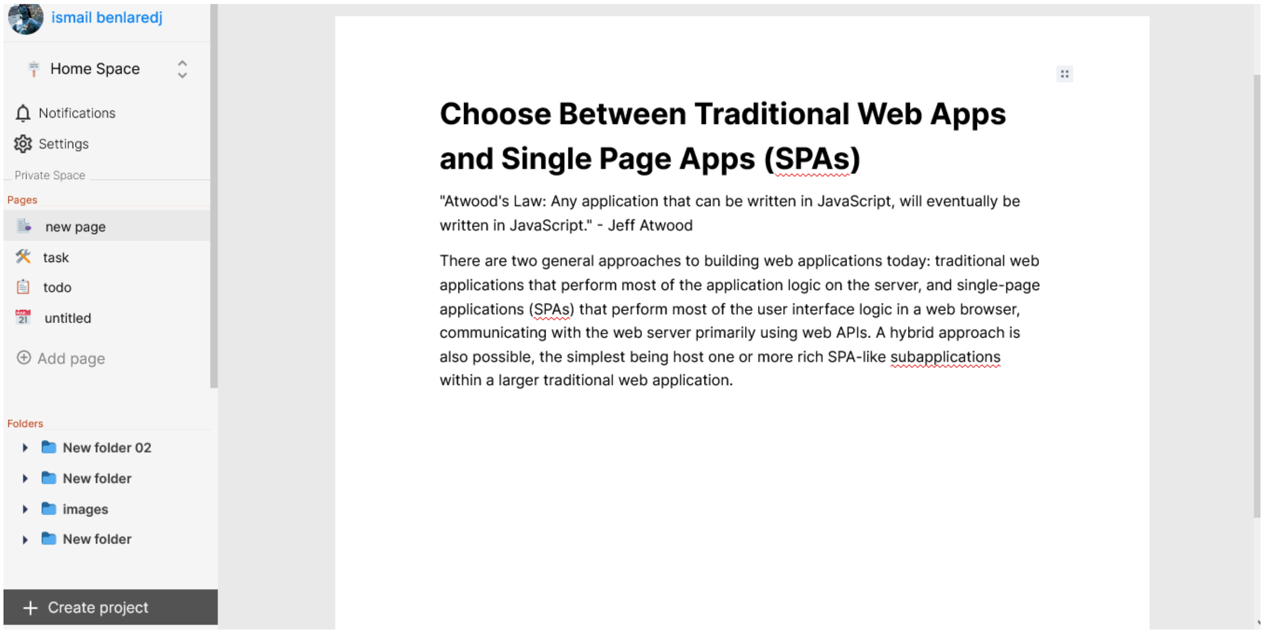


Figure 6.9 – Application screenshot : Sign up



Need an account? Create one

Figure 6.10 – Application screenshot : Sign in

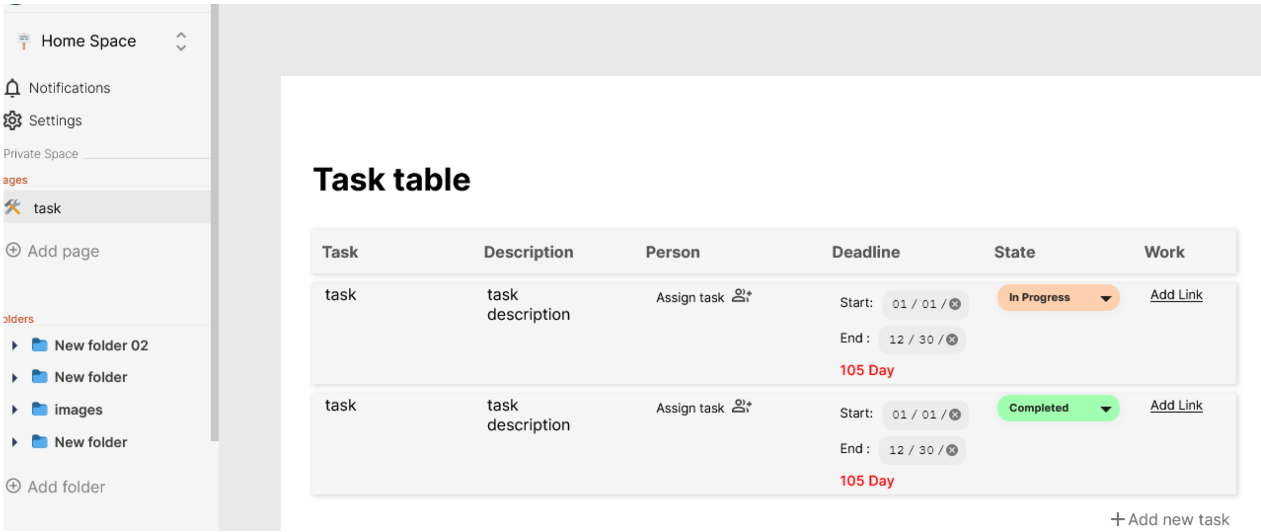


Choose Between Traditional Web Apps and Single Page Apps (SPAs)

"Atwood's Law: Any application that can be written in JavaScript, will eventually be written in JavaScript." - Jeff Atwood

There are two general approaches to building web applications today: traditional web applications that perform most of the application logic on the server, and single-page applications (SPAs) that perform most of the user interface logic in a web browser, communicating with the web server primarily using web APIs. A hybrid approach is also possible, the simplest being host one or more rich SPA-like subapplications within a larger traditional web application.

Figure 6.11 – Application screenshot : Main page



Task table

Task	Description	Person	Deadline	State	Work
task	task description	Assign task	Start: 01 / 01 / End : 12 / 30 / 105 Day	In Progress	Add Link
task	task description	Assign task	Start: 01 / 01 / End : 12 / 30 / 105 Day	Completed	Add Link

+Add new task

Figure 6.12 – Application screenshot : Task table

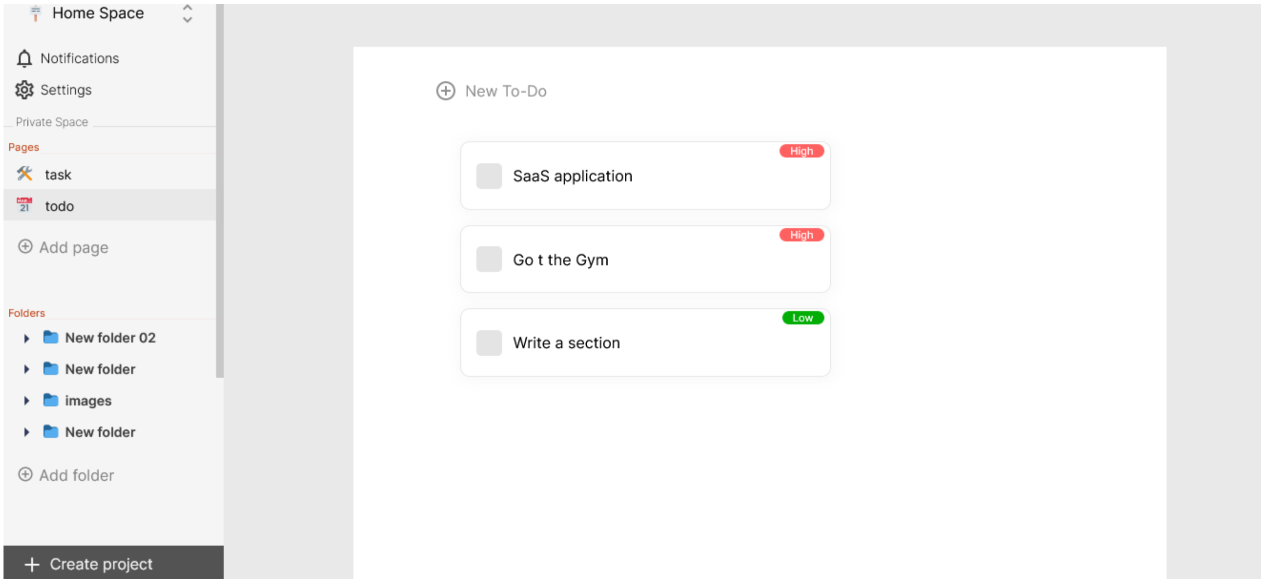
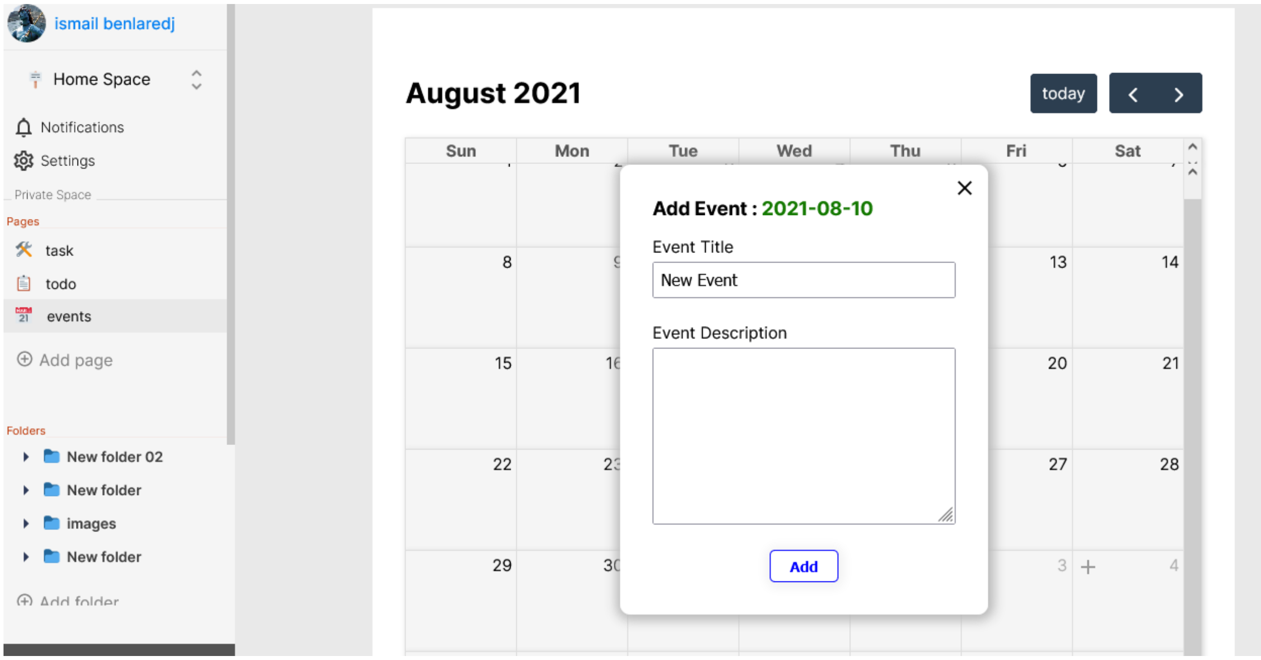


Figure 6.13 – Application screenshot : Todo page



Add Event : 2021-08-10

Event Title: New Event

Event Description

Figure 6.14 – Application screenshot : Calendar page and add event window

6.11 Summary

In this chapter, we start the programming process, we show how we develop and deploy the application, we present some technologies used and some coding solutions. In the end, we present the final application.

Conclusion and Perspectives

In this study, we have tackled the concept of Cloud Computing, which is an emerging technology trend, it has known a massive growth and development during recent years. Cloud Computing basically refers to the on-demand delivery of IT resources including, hardware, servers, storage, software, and IT infrastructure in general. All resources are available as services through the internet and offered on a pay-per-use basis. we presented its architectures, characteristics, models, and building blocks.

One of the main models of Cloud Computing is Software as a Service, SaaS refers to the model of delivery software in the form of a service, precisely SaaS is a cloud-based and web-based software delivered through the internet, usually accessed via a thin program such as a browser that runs on local client's. In our study, we presented the SaaS model, we explained its architectures, development, and deployment strategies. Our main goal in this project is to create a SaaS application to help students collaborate online. To get know more about online collaboration fields, we talked about it in chapter 3, we discussed the online collaboration, its principals and key factors of a successful one, this helps us to create a general idea about the application that we created.

In the end, as we saw in the last chapter, created a SaaS application for online collaboration between students.

As a perspective, we aim to add more functionalities to the application like a journal to record activities of group members, add more features to the text editor, this will help users create rich content, also we aim to make the application responsive, which means the user can access the application using a mobile, or tablet browser and the application will be displayed properly. Regarding the cloud computing technologies, we aspire to use more products and services like memorystore, Cloud Tasks, Cloud Run..., these services will help us improve performance and make it easy for us to create more functionalities and features.

Bibliography

[1] R. Buyya, C. Vecchiola, and S. T. Selvi, *Mastering Cloud Computing: Foundations and Applications Programming.* San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1st ed., 2013.

[2] S. Mark-Shane, “Cloud computing and collaboration,” *Library Hi Tech News*, vol. 26, pp. 10–13, 10 2009.

[3] G. F. P. B. C. M. Jadhav, V. V. Bandgar, “An approach for development of multitenant application as saas cloud,” 20014.

[4] L. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner, “A break in the clouds: Towards a cloud definition,” *Computer Communication Review*, vol. 39, pp. 50–55, 01 2009.

[5] R. Buyya, C. S. Yeo, and S. Venugopal, “Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities,” *Proceedings - 10th IEEE International Conference on High Performance Computing and Communications, HPCC 2008*, 09 2008.

[6] R. Bohn, J. Messina, F. Liu, J. Tong, and J. Mao, “Nist cloud computing reference architecture,” pp. 594–596, 07 2011.

[7] Y. Jadeja and K. Modi, “Cloud computing - concepts, architecture and challenges,” 03 2012.

[8] J. McCarthy, “Reminiscences on the history of time-sharing,” *IEEE Ann. Hist. Comput.*, vol. 14, p. 19–24, Jan. 1992.

[9] A. Oludele, E. Ogu, S. Kuyoro, and C. Umezuruike, “On the evolution of virtualization and cloud computing: A review,” *Journal of Computer Science and Applications*, vol. 2, pp. 40–43, 12 2014.

[10] T. Erl, R. Puttini, and Z. Mahmood, *Cloud Computing: Concepts, Technology, & Architecture*. The Prentice Hall service technology series from Thomas Erl, Prentice Hall, 2013.

[11] L. M. Kaufman, “Data security in the world of cloud computing,” *IEEE Security Privacy*, vol. 7, no. 4, pp. 61–64, 2009.

[12] R. Krutz and R. Vines, *Cloud Security: A Comprehensive Guide To Secure Cloud Computing*. Wiley India Pvt. Limited, 2010.

[13] “About aws.” `https://aws.amazon.com/about-aws/`. Retrieved: 2021-07-30.

[14] W. V. Leslie P. Willcocks and E. A. Whitley, “loud sourcing and innovation: slow train coming? a composite research study,” 2013.

[15] K. Vishwanath and N. Nagappan, “Characterizing cloud computing hardware reliability,” pp. 193–204, 01 2010.

[16] M. Naeem, H. Mahar, F. Memon, M. Siddique, and A. R. K. Rajput, “An overview of virtualization cloud computing,” *Science international journal lahore*, 07 2016.

[17] “Virtualization.” `https://en.wikipedia.org/wiki/Virtualization`. Retrieved: 2021.

[18] K. Devi, G. Subramanian, and D. R, “Virtualization in cloud computing,” *IJARCCE*, vol. 7, pp. 104–108, 11 2018.

[19] D. E. Y. Sarna, “Implementing and developing cloud computing applications,” 2010.

[20] “What is soa??.” `https://collaboration.opengroup.org/projects/soa-book/pages.php?action=show&ggid=1314`. Retrieved: 2021.

[21] “Open-source software.” `https://en.wikipedia.org/wiki/Open-source_software`. Retrieved: 2021.

[22] "Dive into the xen hypervisor." https://developer.ibm.com/articles/cl-hypervisorcompare-xen/. Retrieved: 2021.

[23] "Cloud computing architecturer." https://en.wikipedia.org/wiki/Cloud_computing_architecture. Retrieved: 2021.

[24] S. Okai-Ugbaje, M. Uddin, A. Arshad, and A. Shah, "Cloud computing adoption model for universities to increase ict proficiency," *SAGE Open*, vol. 4, 07 2014.

[25] D. Bharti and R. Goudar, "Cloud computing–research issues, challenges, architecture, platforms and applications: A survey," 12 2012.

[26] P. Neto, "Demystifying cloud computing."

[27] I. Ashraf, "An overview of service models of cloud computing," *International journal of multidisciplinary and current research*, vol. 2, 2014.

[28] M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," *Commun. ACM*, vol. 53, pp. 50–58, 04 2010.

[29] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: State-of-the-art and research challenges," *Journal of Internet Services and Applications*, vol. 1, pp. 7–18, 05 2010.

[30] "12 benefits of cloud computing." https://www.salesforce.com/products/platform/best-practices/benefits-of-cloud-computing/. Retrieved: 2021.

[31] T. Dillon, C. Wu, and E. Chang, "Cloud computing: Issues and challenges," in *2010 24th IEEE International Conference on Advanced Information Networking and Applications*, pp. 27–33, 2010.

[32] K. H. E. G. Adam Holt, Keith Weiss, "Cloud computing takes off market set to boom as migration accelerates," 05 2011.

[33] "Cloud market ends 2020 on a high while microsoft continues to gain ground on amazon." https://www.srgresearch.com/articles/cloud-market-ends-2020-high-while-microsoft-continues-gain-ground-amazon. Retrieved: 2021.

[34] S. Satyanarayana, “Cloud computing : Saas,” *GESJ: Computer Science and Telecommunications*, vol. 4, 12 2011.

[35] S. Kang, J. Myung, J. Yeon, S.-W. Ha, T. Cho, J.-M. Chung, and S.-g. Lee, “A general maturity model and reference architecture for saas service,” vol. 5982, pp. 337–346, 04 2010.

[36] A. LIMBÄ‚ÅžAN and R. Lucia, “Implementing saas solution for crm,” *Informatica Economica*, vol. 15, pp. 175–183, 01 2011.

[37] “What is the saas architecture maturity model?.” `https://www.forbes.com/sites/forbesbusinesscouncil/2019/11/20/what-is-the-saas-architecture-maturity-model/?sh=13cac27e4b40`. Retrieved: 2021.

[38] J. Fiaidhi, I. Bojanova, J. Zhang, and L.-J. Zhang, “Enforcing multitenancy for cloud computing environments,” *IT Professional*, vol. 14, pp. 16–18, 01 2012.

[39] W. N. T. Alwis and C. Gamage, “A multi-tenancy aware architectural framework for saas application development,” *Engineer: Journal of the Institution of Engineers*, vol. 46, p. 21, 2013.

[40] J. Fiaidhi, I. Bojanova, J. Zhang, and L.-J. Zhang, “Enforcing multitenancy for cloud computing environments,” *IT Professional*, vol. 14, pp. 16–18, 01 2012.

[41] G. C. Frederick Chong and R. Wolter, “Multi-tenant data architecture,” 2006.

[42] A. Fox and D. Patterson, *Engineering Software as a Service: An Agile Approach Using Cloud Computing.* Strawberry Canyon LLC, 2013.

[43] “Kubernetes.” `https://www.ibm.com/cloud/learn/kubernetes`. Retrieved: 2021.

[44] I. Baldini, P. Castro, K. Chang, P. Cheng, S. Fink, V. Ishakian, N. Mitchell, V. Muthusamy, R. Rabbah, A. Slominski, and P. Suter, “Serverless computing: Current trends and open problems,” 2017.

[45] “Serverless computing.” `https://www.ibm.com/cloud/learn/serverless`. Retrieved: 2021.

[46] "Application service provider." https://en.wikipedia.org/wiki/Application_service_provider. Retrieved: 2021.

[47] L. I. P. Ltd, "Difference between asp saas models." https://www.luitinfotech.com/.

[48] V. Parthasarathy and V. Kumar, "Design comparison of traditional application and saas," 03 2014.

[49] L. G. Hathorn and A. L. Ingram, "Online collaboration: Making it work," *Educational Technology*, vol. 42, no. 1, pp. 33–40, 2002.

[50] C. Moore, "The future of work: What google shows us about the present and future of online collaboration," *TechTrends*, vol. 60, 03 2016.

[51] H. Tseng, C.-H. Wang, H.-Y. Ku, and L. Sun, "Key factors in online collaboration and their relationship to teamwork satisfaction. quarterly review of distance education, 10(2).," *Quarterly Review of Distance Education*, vol. 10, pp. 195–206, 06 2009.

[52] D. P. Brandon and A. B. Hollingshead, "Collaborative learning and computer-supported groups," *Communication Education*, vol. 48, no. 2, pp. 109–126, 1999.

[53] M.-K. S. J. M. A. Stephen B. Richards, Catherine Lawless Frank, *Collaboration Among Professionals, Students, Families, and Communities Effective Teaming for Student Learning*. Routledge, 2016.

[54] A. Powell, G. Piccoli, and B. Ives, "Virtual teams: A review of current literature and directions for future research," *DATA BASE*, vol. 35, pp. 6–36, 12 2004.

[55] . W. K. D. Karau, S. J., "Social loafing: A meta-analytic review and theoretical integration.," *Journal of Personality and Social Psychology*, 1993.

[56] C. Thurlow, L. Lengel, and A. Tomic, *Computer Mediated Communication*. SAGE Publications, 2004.

[57] "Cloud collaboration." https://en.wikipedia.org/wiki/Cloud_collaboration. Retrieved: 2021.

[58] R. Godwin-Jones, “Emerging technologies: Blogs and wikis: Environments for on-line collaboration,” *Language Learning Technology*, vol. 7, pp. 12–16, 05 2003.

[59] E. Harrin, *Collaboration Tools for Project Managers: How to Choose, Get Started and Collaborate with Technology.* Project Management Institute, 2016.

[60] “Video conferencing definition.” https://www.lifesize.com/en/video-conferencing/what-is-video-conferencing/. Retrieved: 2021.

[61] “Video conferencing market by component (hardware, solutions, and services), application (corporate communication, training and development, and marketing and client engagement), deployment mode, vertical, and region - global forecast to 2026.” https://www.marketsandmarkets.com/Market-Reports/video-conferencing-market-99384414.html. Retrieved: 2021.

[62] “Google workspace for everyone.” https://blog.google/products/workspace/google-workspace-everyone/. Retrieved: 2021.

[63] “Introducing google workspace to help you get more done.” https://blog.google/products/workspace/introducing-google-workspace-help-you-get-more-done/. Retrieved: 2021.

[64] M. Archibald, R. Ambagtsheer, M. G. Casey, and M. T. Lawless, “Using zoom videoconferencing for qualitative data collection: Perceptions and experiences of researchers and participants,” *International Journal of Qualitative Methods*, vol. 18, 2019.

[65] A. Nolte, C. Alvarez, R. Hishiyama, I. Chounta, M. Rodríguez-Triana, and T. Inoue, *Collaboration Technologies and Social Computing: 26th International Conference, CollabTech 2020, Tartu, Estonia, September 8–11, 2020, Proceedings.* Lecture Notes in Computer Science, Springer International Publishing, 2020.

[66] A. Powell, G. Piccoli, and B. Ives, “Virtual teams: A review of current literature and directions for future research,” *DATA BASE*, vol. 35, pp. 6–36, 12 2004.

[67] P. Lavappa, "Guidelines for planning and development of software for buildings and building systems," 2008-06-01 2008.