ITP Cloud

Build Your Imagination

Software Requirements Specification

Version 1.0.1

Prepared by Aaron Mkandawire

Zambia University College of Technology

10th August, 2023

Contents

1. Introduction

1.1. Purpose

This document specifies the requirements of the ITP Cloud platform. It describes only two main subsystems of the entire ITP Cloud platform. These subsystems are: the ITP Core System and the ITP Engine System. Together these systems form the ITP Cloud Platform. This document further serves as a guide for the developers (author) on one hand and a software validation document for the prospective reviewers (lecturers) on the other.

1.2. Document Conventions

This document was written using the Times New Roman font family. All subheadings have a larger font size compared to the paragraphs. Additionally, important pieces of information have been made bold for emphasis' sake. This document is prepared by following IEEE conventions for software requirement specifications.

1.3. Intended Audience and Reading Suggestions

This document is intended for students, lecturers, professors, developers, project managers, marketing staff, software engineers, DevOps engineers, cloud architects, testers, and documentation writers. The rest of this document contains the fundamental elements of a standard SRS as outlined in the tables of contents. The author recommends reading this document from top to bottom, section after section without skipping any part.

1.4. Product Scope

ITP Cloud is a web hosting platform designed for students. It provides a hosting environment where students with websites and webapps can deploy their projects. The goal is that by providing such a platform, students will be stimulated to engage in project-based learning and consequently become more skilled in software engineering skills.

1.5. References

[1] A. S. T. Maarten van Steen, DISTRIBUTED SYSTEMS, 4th edition ed., Maarten van Steen, 2023, pp. 2 - 54.

[2] C. B. Thomas Connolly, Database Systems A Practical Approach to Design, Implementation, and Management, England: Pearson Education Limited, 2015.

[3] Y. d. Liang, Introduction to Java™ Programming and Data Structures, Comprehensive Version, Harlow: Pearson Education Limited, 2019.

2. Overall Description

2.1. Product Perspective

The system being specified in this Software Requirements Document originates from the obligation to fulfil the requirements of graduation from the undergraduate program “Bachelor of Software Engineering” from the Zambia University College of Technology. The author, being a graduating student, was required to specify the technical requirements of the system in a Requirements

Document. The system being documented from here-and-on, is a custom web hosting platform, which will be made open source soon. Seeing that most web hosting platforms are closed source, the designs and implementation of the system in context, were acquired by systematically reverse engineering existing web hosting platforms work based on how they work.

As earlier mentioned in section 1.1, the ITP Cloud Platform is made up of mainly two subsystems. These subsystems are: the ITP Core System and the ITP Engine System. Together these systems form the ITP Cloud Platform. Below, is a brief High-level Design Overview of the entire system. It shows the major subsystems making up the whole system. It also shows the external systems that the subsystems interact with.

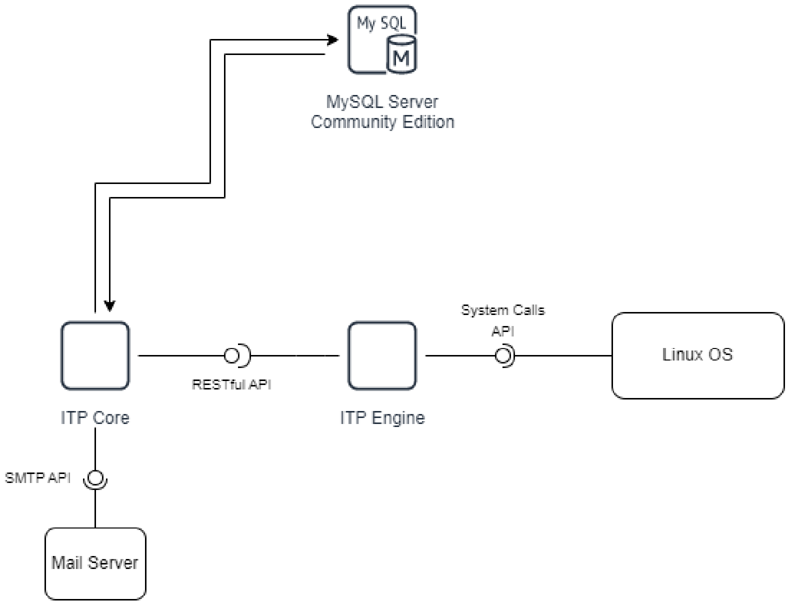


Figure 1 - High-Level Design Overview

In figure 1, ITP Core interacts with three entities: ITP Engine, a Mail Server and a Database Server. Public users of the system will mainly interact with ITP Core. ITP Core will then interact with a Database Server to store application and system data persistently. Occasionally, the ITP Core will send out emails via the Mail Server using the SMTP Protocol. These emails will mainly be for authentication purposes and any system notifications that may require the end user's attention. Lastly, ITP Core will interact with ITP Engine.

ITP Engine acts as a resources manager and an adapter between ITP Core and the Linux Operating System. ITP Engine interacts mainly with ITP Core by receiving requests pertaining to resource allocation and management. The Engine then interacts with the Linux Operating System to achieve the tasks.

2.2.Product Functions

The system will allow the users (students) to perform the following functions. The functions have been grouped based on category.

Authentication

- Create an account using their details
- Activate their newly created account using email verification

- Log into their account
- Log out from their account
- Be verified and approved or rejected

Resource Management: Databases

- Create a database
- Create a database user
- View the details of a selected database
- View the details of a selected database user
- Delete a database
- Delete a database user
- Access phpMyAdmin

Resource Management: Websites

- Create a website instance
- Setup and configure the website instance settings
- Edit an existing website instance's configuration
- Delete a website instance

Resource Management: DevOps

- Deploy a website using GitHub
- Deploy a website using an SFTP client
- Access their Linux account via SSH

The system will also allow the **moderators (lecturers)** to perform the following functions.

User and Content Management

- Log into the system
- Log out from their account
- Verify pending student account registrations
- Moderate content being published on the platform
- Ban users who violate the terms of use
- Remove websites that do not conform to the regulations set forth

2.3.User Classes and Characteristics

The ITP Cloud Platform will have mainly two user classes. These include students and lecturers. Students are the primary users of the system. About 80% of the system functions are designed and made for students. While the remaining 20% for the lecturers.

2.4.Operating Environment

As earlier mentioned, the ITP Cloud Platform is made up of two main subsystems. ITP Core and ITP Engine. ITP Core is a web application written using PHP 8 and the CodeIgniter 4 framework. ITP Engine on the other hand is also a web application written using Java 8 and the Spring Boot framework. Because of this, the optimum operating environment for the ITP Cloud platform would consist of the following Software Components and Operating System.

Software Requirements

- Ubuntu Server 22.04 LTS
- Java 8 Runtime and JDK 8
- PHP 8.2 (with all pertinent extensions enabled)
- MySQL Server Community Edition
- Composer 2 (a PHP dependency manager)
- CodeIgniter 4
- Git
- Nginx
- OpenSSH
- An FTP Server
- Python 3.10+
- Node.js 18+
- gcc, g++ and CMake

Hardware Requirements

Test Environment

The ITP Cloud platform can be setup and configured either on a virtual machine or on an actual machine, provided it meets the specifications below.

- 4 GB RAM DDR3
- 2 Core CPU @ 2.0GHz or better
- 50 GB SSD Secondary Storage
- Air Conditioning and Cooling infrastructure
- Network Adapter

Production Environment

- 256 GB RAM DDR4
- 64 Core CPU @ 5.0GHz or better
- 8 TB SSD Secondary Storage
- Network Adapter
- Air Conditioning and Cooling infrastructure
- UPS Power Supply System

The ITP Cloud platform can also be configured to run in a distributed environment. However, additional software and hardware requirements will be need. In addition to this, more configurations may be needed to ensure effective and efficient operations.

2.5.Design and Implementation Constraints

The following *issues are expected to limit the options available for the developer to* design, develop and implement the ITP Coud platform. These have been categorized as follows:

1. Hardware Limitations

The nature of the system being built is massively hardware intensive. This owes to the fact that the Cloud platform is some kind of runtime environment that hosts other applications that each may have their own hardware requirements. Thus, the specific hardware requirements for the ITP Cloud Platform are largely dependent on the specific hardware requirements of each application that is hosted on the platform. It can be said that the hardware requirements of the ITP Cloud Platform are a cumulative summation of all possible application specific hardware requirements.

One suggested solution to this, is actually building a distributed computer system, that is globally distributed and pools computing resources like Memory and Compute. However, according to Maarten and Andrew [1] distributed and decentralized systems are inherently complex and difficult to build and maintain.

2. Interfaces to Other Applications

The ITP Engine will mainly interact with the Linux Operating System. This will be achieved in two ways. First, making system calls from Java to the Linux OS. And secondly, by directly interacting with the Linux Terminal programmatically. The latter approach of interacting with the Linux OS is complex and prone to errors. This too will pose a significant limitation to making the two systems, that is, the ITP Engine and the Linux OS, communicate effectively.

3. Databases To Be Used

The ITP Cloud Platform will be accessed by many users simultaneously. This simultaneous interaction will naturally give rise to database issues partly highlighted by Thomas Connolly and Carolyn Beggsuch [2]. These include:

- The lost update
- The uncommitted dependency (or dirty read) problem
- The inconsistent analysis problem

4. Parallel Operations

The ITP Cloud Platform will be accessed by many users simultaneously. This means that a number of parallel operations will be executed at a time. Such parallel interaction may result in race conditions, and deadlocks [3], if a single shared resource is being accessed by multiple parallel threads or processes. In addition to this, the simultaneous interaction may also naturally give rise to database issues partly highlighted by Thomas Connolly and Carolyn Beggsuch [2] in the previous section above.

5. Communications Protocols

The primary communication protocols to be used are as follows:

- Hyper Text Transfer Protocol (HTTP)
- Secure Shell (SSH)
- File Transfer Protocol (FTP)

- Web Sockets (WS)
- Simple Mail Transfer Protocol (SMTP)

It will therefore be imperative that only secure versions of the above protocols are used. This will reduce any security risks that may exist. In addition, the two subsystems will make use of these protocols to interact with each other. Thus, the data formats to be used, are supposed to be known on each side and must be consistent and standardized.

6. **Security Considerations**

The ITP Cloud Platform is a network centric system. This means that most communication will happen over the internet or on the network where the Platform operates. Because of this, it will be imperative to ensure that all data communication protocols are secured. In addition, the ITP Engine will operate on the same environment as the end user's applications. This in itself may pose additional security risks. Thus, the task at hand is to ensure that all user's data is completely isolated, and that one system user cannot access the data of another system user.

7. **Design Conventions**

Due to the nature of this project, the design conventions to be followed while building the ITP Engine using Spring Boot will have to be changed to suit the requirements of this project.

2.6.User Documentation

In addition to building the ITP Cloud Platform, User Documentation in the form of YouTube video tutorials and GitHub ReadMe Docs, will be made available.

2.7.Assumptions and Dependencies

Some assumptions being made are that the ITP Cloud Platform will not in its initial stage be a fully-fledged Cloud Platform with all the services that a standard Cloud Provider may offer. Instead, it starts out as a Shared Web Hosting provider, that does not yet have Nameservers to be used to link stand-alone domain names to websites hosted on the Platform. On the other hand, some critical dependencies that the ITP Cloud Platform has are as follows:

- Java 8 Runtime Environment
- Java 8 Development Kit
- Nginx
- Linux OS

3. External Interface Requirements

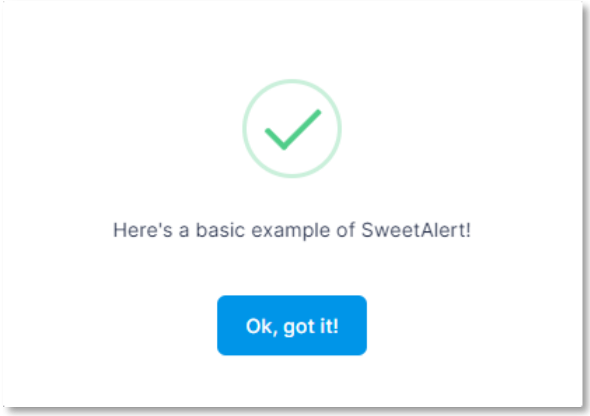
3.1.User Interfaces

As earlier mentioned, the ITP Cloud Platform has two main subsystems, which are the ITP Core and the ITP Engine. Since end users will mainly interact with the Platform via ITP Core, only it, will require a user interface. At the time of writing this document, no user interface designs were present. The user interface will be based on an existing HTML5, and Bootstrap 5 dashboard template. Due to the fact that

Bootstrap 5 will be used for the user interface design and implementation, all style guidelines designated for this project will be inherited from the Bootstrap 5 specification. This specification defines a standard set of colours, sizing, spacing and layout to be applied to the user interface.

Dialogs

Inevitably, as the user interacts with the Platform, errors will be expected. Such errors will be displayed to the end users via popup boxes powered by sweetalert2 (https://sweetalert2.github.io/). Not only will errors be displayed using the sweetalert2 library, but also success and info messages. All in all, the error, success and info messages will have a generally polite tone, and communicate to the end user what exactly is happening in an unambiguous way. An example of a SweetAlert is shown below in Figure 2



Here's a basic example of SweetAlert!

Figure 2 - Sweet Alert

In addition, Bootstrap 5 Modals will also be used for constructing dialogs and prompting the user for additional information.

Navigation

The UI dashboard, as earlier mentioned will be based on an existing HTML5 and Bootstrap 5 template. On this dashboard, a left-aside kind of navigation will be present on all pages. The navigation bar will have links to the main sections of the Platform. These sections will correspond to the main functional requirements as defined in section 4 of this document. Besides that, breadcrumbs will also be present on all pages. Breadcrumbs are a navigational aid that allows users to keep track of their current location on a website or interface. A breadcrumb trail displays the page a user is currently on and its relation to the pages they visited before it, or the hierarchy of higher-level parent pages above the current page.

Other UI Components

As for the rest of the UI components to be used on the dashboard, the author will either reuse existing Bootstrap 5 Components or create custom ones to fit a use case.

3.2.Hardware Interfaces

The ITP Cloud Servers and other Clients will primarily use the following hardware interfaces:

1. **Ethernet Interfaces**
   Ethernet interfaces are fundamental for connecting servers to local area networks (LANs) and the wider internet. Gigabit Ethernet, 10 Gigabit Ethernet, and 25/40/100 Gigabit Ethernet interfaces will be suitable for the ITP Cloud Platform.

2. **USB Ports**
   USB ports on servers are used for connecting peripherals like keyboards, mice, external drives, and other devices for management and maintenance purposes.

3. **VGA/DVI/HDMI Ports**
   These video interfaces are used to connect monitors for server management, configuration, and troubleshooting. Many servers offer remote management through these interfaces.

4. **PCIe (Peripheral Component Interconnect Express) Slots**
   PCIe slots are used to expand server capabilities with add-in cards such as network interface cards (NICs), graphics cards, storage controllers, and more.

5. **RJ-45 Console Ports**
   RJ-45 console ports are used for serial console access to servers. They provide a means to access server consoles remotely for configuration and troubleshooting.

6. **Power Connectors**
   Servers require various power connectors, such as standard power cables, redundant power supplies, and power distribution units (PDUs), to ensure continuous operation.

3.3.Software Interfaces

The ITP Cloud Platform will interact with a database management system (DBMS) - MySQL Server Community Edition or MariaDB, where all user and system data will be stored. The communication between the Cloud Platform and MySQL Server will go through the standard port (3306) and using the IP Address of the machine where they will both operate (localhost). The Cloud Platform will run on top of Ubuntu Server 22.04 LTS.

Part of the Cloud Platform (ITP Core) will be written using CodeIgniter 4 (a PHP framework). CodeIgniter 4 is a lightweight PHP framework for building websites using the Model View Controller design principle. CodeIgniter 4 can retrieved and scaffolded using Composer version 2. Composer will also be needed to manage all PHP dependencies that will be used by and within the project.

ITP Core will communicate with the ITP Engine using the HTTP protocol and the RESTful Service Oriented Architecture. RESTful Service-Oriented Architecture (SOA) is an architectural approach for designing and building distributed software systems and Cloud Platforms that use the principles of Representational State Transfer (REST) to enable communication and interaction between different components or services. It combines the concepts of SOA with the REST architectural style, which is often associated with web services and HTTP.

RESTful architecture is a valuable design choice when building a cloud platform, as it provides a straightforward and scalable way to interact with services and resources in a cloud environment. Here's how the RESTful architecture will be used in the context of building the ITP Cloud Platform:

- **Resource Modelling**
  In our cloud platform, various resources need to be managed, such as databases, websites and storage. Each of these resources will be modelled as RESTful resources with their own unique URIs.

- **HTTP Methods**
  Different HTTP methods can be used to perform operations on these resources. For example:
  - GET: Retrieve information about a resource (e.g., status, configuration).
  - POST: Create a new resource (e.g., provision a new database).
  - PUT: Update the state or configuration of a resource (e.g., modify firewall rules).
  - DELETE: Remove a resource (e.g., delete a website).

- **Data Representation Formats**
  When using the RESTful architecture, information is usually provided to clients in various formats, such as JSON or XML. Clients can specify their preferred representation using the Accept header in their requests. In our case, the ITP Core is the client, while the ITP Engine is the server.

By embracing the RESTful architecture in the design and development of this miniature cloud platform, we can create a flexible, scalable, and user-friendly environment that allows developers, administrators, and users to interact with cloud resources in a standardized and intuitive manner. Additionally, RESTful principles promote the use of standard HTTP communication, making it easier to integrate with various client applications and services.

3.4.Communication Interfaces

The ITP Cloud Platform will use a wide range of data communication protocols. These include: HTTP, SSH, FTP, SMTP, and Web Sockets. For each of these data communication protocols, the secure versions of each will be preferred over the standard versions. The primary approach that end-users will use to interact with the Cloud Platform is the web browser. Thus, the HTTP protocol will be used heavily. The authentication aspect of the Cloud Platform (ITP Core) will require email verification; therefore, ITP Core will consequently have to interact with a mail server to send out emails to end users. The preferred data communication protocol to be used will be the Simple Mail Transfer Protocol (SMTP).

ITP Core and the ITP Engine will interact with each other using the HTTP protocol, and more specifically using the RESTful Service Oriented Architecture. As a result, the JavaScript Object Notation (JSON) standard will be used as a primary way of formatting the data to be exchanged between the subsystems. In addition, ITP Core will communicate within itself, that is, the frontend JavaScript code with the backend PHP code. This communication will also use the RESTful Architecture from time to time.

End users (the developers) will also be able to upload their files directly to the server using the File Transfer Protocol. They will also be given access to a controlled Secure Shell (SSH) account. Such an account will allow for secure communication and remote access to the ITP Cloud servers over an unsecured network, such as the internet. SSH provides encrypted and secure connections, making it a fundamental tool in system administration, file transfers, and secure communication between devices.

Ultimately, due to the sensitive nature of the data being exchanged, it will be imperative to enforce Secure versions of the data communication protocols being used. SSL and TLS certificates will have to be initiated, verified and issued, to ensure that data exchange is secure and encrypted.

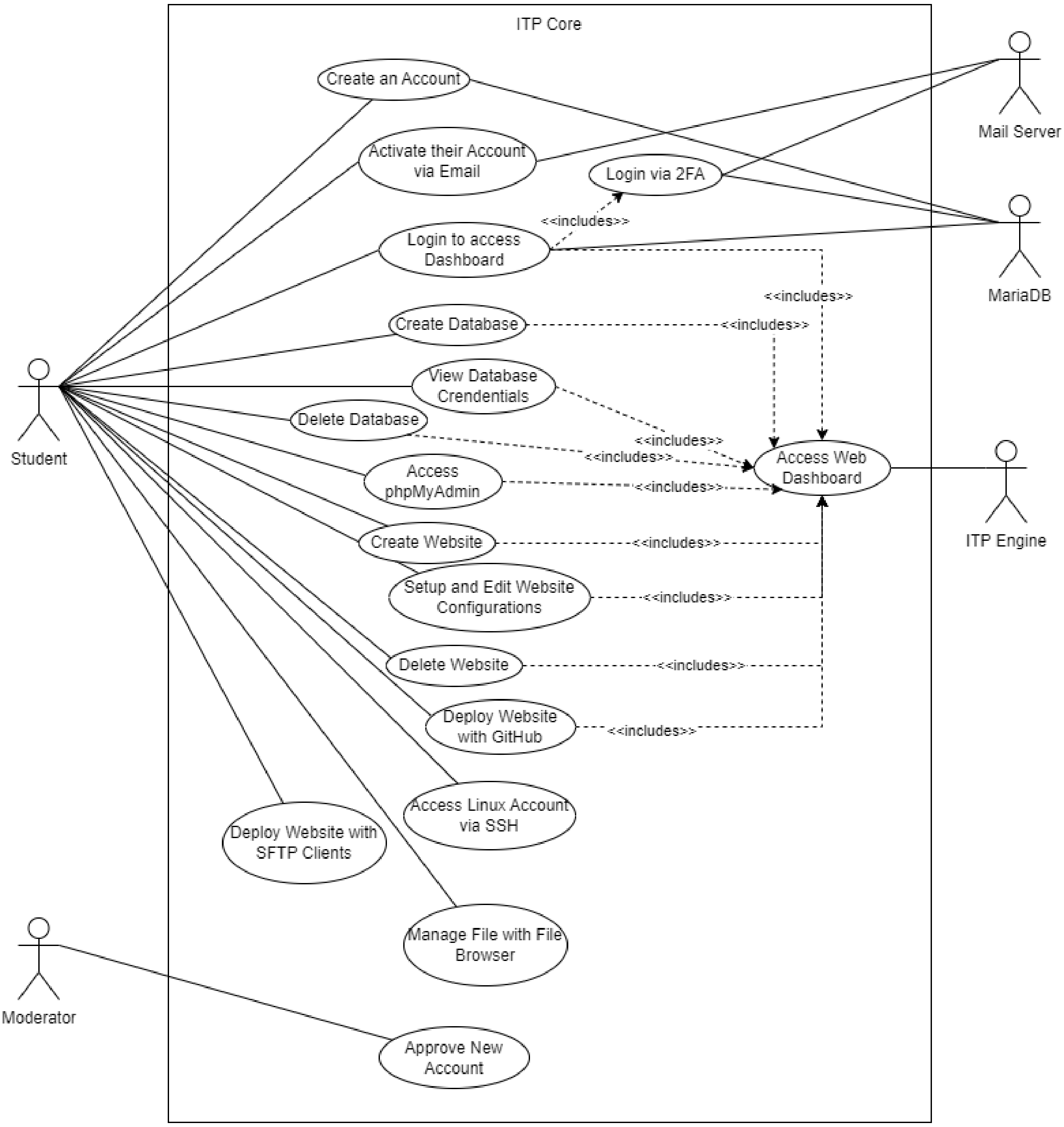
Intentionally left blank to mark start of next section.

4. System Features

The functional requirements to follow will be grouped based on the subsystems making up the ITP Cloud Platform. Each system feature will be preceded by a use case diagram describing the actions that the actors will be able to perform with regard to the subsystem. In addition, primary and secondary actors will be identified, along with the specific actions they will be allowed to perform.

4.1.ITP Core

ITP Core is the main website and web dashboard component of the ITP Cloud Platform. A number of features have been made available to end users. These are described next using a use case diagram.


ITP Core
Create an Account
Activate their Account via Email
Login via 2FA
Login to access Dashboard
<<includes>>
Create Database
View Database Crendentials
Delete Database
Access phpMyAdmin
Access Web Dashboard
Create Website
Setup and Edit Website Configurations
Delete Website
Deploy Website with GitHub
Access Linux Account via SSH
Deploy Website with SFTP Clients
Manage File with File Browser
Approve New Account
Student
Moderator
Mail Server
MariaDB
ITP Engine

Actors

a. **Primary Actors**: Student, and Moderator
b. **Secondary Actors**: Mail Server, MariaDB, and ITP Engine

Functional Requirements - Students

The functional requirements will be described based on the category to which the requirement belongs.

A. **Authentication**
Students will be able to:
   i. Create an account using their details
   ii. Activate their newly created account using email verification
   iii. Log into their account
   iv. Log out from their account
   v. Be verified and approved or rejected

B. **Resource Management: Databases**
Students will be able to:
   i. Create a database
   ii. Create a database user
   iii. View the details of a selected database
   iv. View the details of a selected database user
   v. Delete a database
   vi. Delete a database user
   vii. Access phpMyAdmin

C. **Resource Management: Websites**
Students will be able to:
   i. Create a website instance
   ii. Setup and configure the website instance settings
   iii. Edit an existing website instance's configuration
   iv. Delete a website instance

D. **Resource Management: DevOps**
Students will be able to:
   i. Deploy a website using GitHub
   ii. Deploy a website using an SFTP client
   iii. Access their Linux account via SSH
   iv. Manage files using File Browser (https://filebrowser.org)

Functional Requirements - Moderators

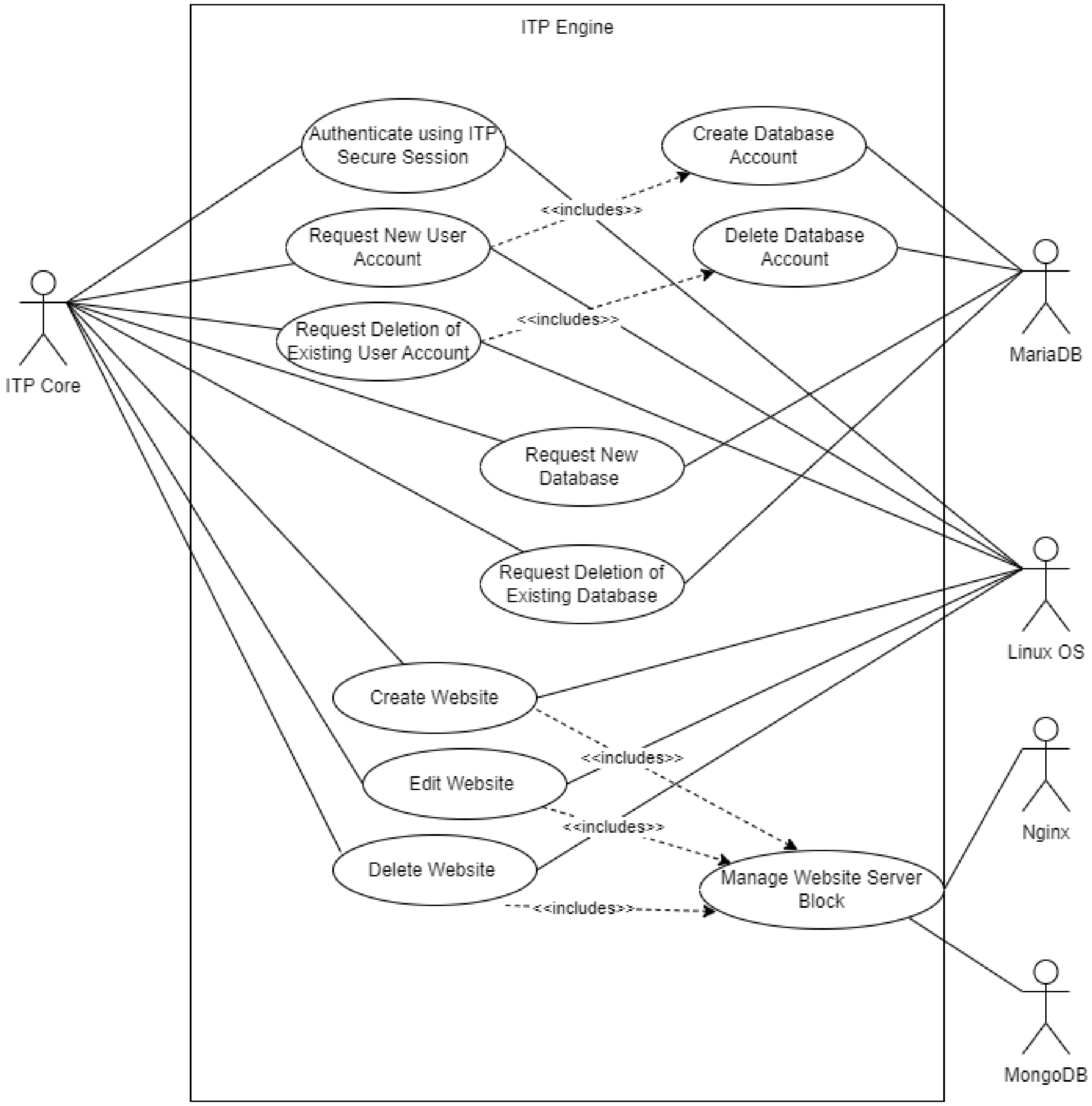
The system will also allow the **moderators (lecturers)** to perform the following functions.

A. **User and Content Management**
   i. Log into the system
   ii. Log out from their account
   iii. Verify pending student account registrations

iv. Moderate content being published on the platform
v. Ban users who violate the terms of use
vi. Remove websites that do not conform to the regulations set forth

4.2.ITP Engine

ITP Engine is the actual ITP Cloud Platform manager. It is responsible for all resource management and allocation activities. It has one primary actor, the ITP Core subsystem. Effectively, all user requirements prescribed for the ITP Core, are actually implemented by the ITP Engine. A number of features have been made available to ITP Core and these correspond closely to those described under ITP Core. The use case diagram to follow highlights the main functional requirements ITP Engine offers.



Actors

a. **Primary Actors**: ITP Core
b. **Secondary Actors**: MariaDB, Linux OS, MongoDB and Nginx

Functional Requirements – ITP Core

The functional requirements will be described based on the category to which they belong.

A. **Authentication**
The ITP Core subsystem will be able to:
   i. Authenticate using ITP Secure Session

B. **Resource Management: Databases**
The ITP Core subsystem will be able to:
   i. Create a database
   ii. Create a database user
   iii. Delete a database
   iv. Delete a database user

C. **Resource Management: Websites**
The ITP Core subsystem will be able to:
   i. Create a website instance
   ii. Edit an existing website instance's configuration
   iii. Delete a website instance

5. Other Nonfunctional Requirements

5.1. Performance Requirements

When talking about performance in IT systems, we normally determine it by how fast a system responds to user's requests. The cloud platform about to be developed is expected to consistently provide fast response times (at least within 2000 milliseconds) while maintaining an uptime of at least 75.0%. The goal for the platform is to try accommodating a minimum of 50 concurrent users without performance degradation. In addition, the platform should efficiently manage resources, scale seamlessly to handle increased loads, and employ caching mechanisms for improved response times.

5.2. Safety Requirements

Safety requirements are crucial for ensuring the security and well-being of users, data, and the system itself. Since the cloud platform being developed is a sandbox for students to enhance their skills, some form of data protection will be implemented. However, the majority of student projects deployed on the platform will have to be secured by the students themselves. On the other hand, the cloud platform is expected to ensure that user data relative to it is secured. This will involve data encryption and implementing authentication and authorization mechanisms throughout the system. In case data loss does occur, all system users will be notified and instructed to change their authentication details. Other than that, a detailed report of the breach will be written and shared. This will add to the learning process.

5.3.Security Requirements

As briefly discussed in the previous section, user authentication and authorization will be implemented to ensure security on the cloud platform. Authentication will require that the end user provide an email and password for them to log into the cloud platform dashboard. In addition, users will be given the option to enable Multi-Factor Authentication, via email.

Other security considerations involve not saving end user credentials to provisioned resources like databases accounts, ftp accounts, etc., on the any storage form. This includes files, and databases. The end users need to save the credentials when they get displayed to them. After this point, the system will not allow them to see their password. However, a provision will be availed to reset the password in case they forget theirs.

Authorization on the platform will be facilitated through the use Role Based Access Control and Linux Operating System Permissions. At specified intervals, permission audits will be carried out to ensure that all systems users have access to the correct resources.

Usually, when talking about data security, three security objectives are addressed: confidentiality, integrity, and availability:

1. **Confidentiality** describes the state in which data is protected from unauthorized disclosure, e.g., a loss of confidentiality occurs when the content of a communication or a file is disclosed.

2. **Integrity** means that the data has not been altered or destroyed, which can be done accidentally (e.g., transmission errors) or with malicious intent (e.g., sabotage). Integrity more specifically is expected to be implemented on a database level by using transactions.

3. **Availability** refers to authorized persons accessing data and systems within an appropriate period of time. Reasons for loss of availability may be attacks or instabilities of the system.

5.4.Software Quality Attributes

Availability

ITP Cloud Platform will be available 24/7 with a fully functional end user support team that will respond to queries when they can. Also, a group of volunteers, the system maintenance team, will be responsible for making sure the system is working properly.

Interoperability and Portability

The ITP Cloud Platform is intended to be portable and interoperable with any standard Linux Operating System Distribution. If modifications are needed, these are minor, such as, with regards to the Linux Terminal Commands. Other hand that, as long a standard set of dependencies is installed as prescribed by the installation guide, everything should work right off the back.

Testability

Software testability indicates how well a software-driven system allows Software Testing professionals to conduct tests in line with predefined criteria. This attribute also assesses the ease with which Software Quality Assurance engineers can develop test criteria for a said system and its various components. Engineers can assess the testability of a system by using various techniques such as encapsulation, interfaces, patterns, low coupling, and more. All of the aforementioned is expected to be achieved with reference to the ITP Cloud Platform.

Usability

Every software-driven system is designed for ease of use to accomplish certain tasks and the ITP Cloud Platform is not an exception. The system is purposely designed to make website deployment and management very easy for students. Students can easily get started on the ITP Cloud Platform. Besides all the above, the user interface will be designed in an intuitive way so that users easily navigate the website and access various resources and administer various tasks.

In addition, setting up a custom instance of the ITP Cloud Platform on the Virtualized environment will equally made easy. This will be aided through the provision of comprehensive and easy to use **Installation Instructions or Documentation**. Additionally, user training will be provided in the form of YouTube video tutorials.

Consistency and Integrity

Long operations such as, Database Operations and Linux Terminal Operations, will need to be ATOMIC. This means that they must execute to completion or not at all. And each operation should leave the System in consistent state.

6. Domain Requirements

To be determined

Appendix A: Glossary

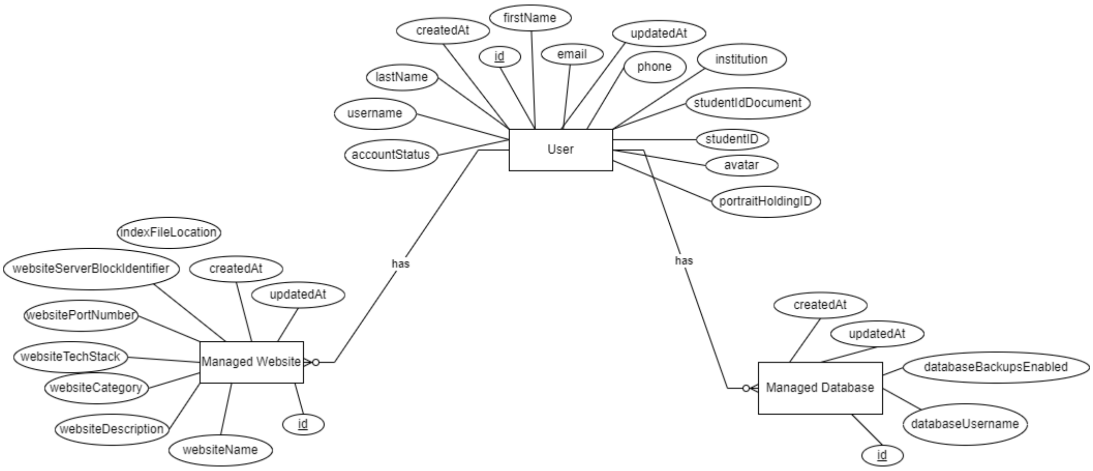
Term	Definition
ITP	Industrial Training Platform. This is the name of the system being developed.
Cloud computing	Cloud computing is the delivery of computing services—including servers, storage, databases, networking, software, analytics, and intelligence—over the internet (“the cloud”) to offer faster innovation, flexible resources, and economies of scale.
Customers	The everyday person who seeks to purchase a product for his house or a flight ticket to go on a holiday.
Students	Refer to the people who will deploy and manage their website projects on the Cloud Platform
System Administrator	The site manager and webmaster. Mainly responsible for managing students and all Cloud Resources.
FTP	File Transfer Protocol
HTTP	Hype Text Transfer Protocol
PHP	Hypertext Preprocessor
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
JDK	Java Development Kit
JRE	Java Runtime Environment
JSON	JavaScript Object Notation
REST	Representational State Transfer
SMTP	Simple Mail Transfer Protocol
XML	eXtensible Markup Language

SOA	Service-Oriented Architecture
HDMI	High-Definition Multimedia Interface
DVI	Digital Visual Interface
VGA	Video Graphics Array
USB	Universal Serial Bus
SSH	Secure Shell
RAM	Random Access Memory
CPU	Central Processing Unit
SSD	Solid State Drive
API	Application Programming Interface
Apache HTTP Web Server	Software that receives and processes incoming HTTP requests
MySQL	A Relational Database Management Software
SSL	Secure Socket Layer
TLS	Transport Layer Security (TLS) is a cryptographic protocol that protects Internet communications
IP	Internet Protocol
OS	Operating System
ATOMIC	An atomic transaction is an indivisible and irreducible series of database operations such that either all occurs, or nothing occurs. A guarantee of atomicity prevents updates to the database occurring only partially, which can cause greater problems than rejecting the whole series outright.
ERD	An ERD, which stands for Entity Relationship Diagram, is a visual representation of how different things in a database are related to each other. These things can be people, objects, or ideas. ERDs also

	show the characteristics or properties of these things. People sometimes refer to ERDs as ER diagrams or Entity Relationship Models.
IEEE	Institute of Electrical and Engineering.

Appendix B: Analysis Models

During the analysis of the system description, **entity-relationship modelling** was used to identify the fundamental entities involved in the cloud platform. At present, only three main resources will be made available to end users (students). These resources include: a **Linux user account (**and a ftp account with the same credentials**)**, one or more **database accounts** and one or more **managed website projects**.



Appendix C: To Be Determined List

Below is a list of items that will be determined later on as the project progresses. This due to the fact that some constraints and information will be available as the project evolves.

1. Domain Requirements