

**Egerton University**

**Software Design Document**

**For**

**Online Work Management System**

**Prepared By**

**Elvis Mutende**

**SP13/00820/15**

**Project Supervisor: Mr. Jones Katiku.**

**Project Coordinator: Dr. Ing W. Gikaru.**

**Date: 17th May 2019**

**Version 1.0**

Overview.

This document describes the system requirements, operating environment, system and subsystem architecture, files and database design, input formats, output layouts, human machine interfaces, detailed design, processing logic, and external interfaces for the project.

Table of Contents

[1 Introduction 1](#_Toc9089669)

[1.1 Purpose and Scope 1](#_Toc9089670)

[1.2 Project Executive Summary 1](#_Toc9089671)

[1.2.1 System overview 1](#_Toc9089672)

[1.2.2 Design Constraints 2](#_Toc9089673)

[1.2.3 Future Contingencies 2](#_Toc9089674)

[1.3 Document Organization. 2](#_Toc9089675)

[1.4 Points of Contact 3](#_Toc9089676)

[1.5 Glossary 3](#_Toc9089677)

[2 System Architecture 4](#_Toc9089678)

[2.1 System Hardware Architecture 4](#_Toc9089679)

[2.2 System Software Architecture 4](#_Toc9089680)

[2.2.1 Database 5](#_Toc9089681)

[2.2.2 Messaging 5](#_Toc9089682)

[2.2.3 Payments 5](#_Toc9089683)

[2.2.4 MySQL 5](#_Toc9089684)

[2.2.5 UML 5](#_Toc9089685)

[2.2.6 Django Framework. 5](#_Toc9089686)

[2.2.7 Python 5](#_Toc9089687)

[2.2.8 VS CODE 6](#_Toc9089688)

[2.3 Internal Communication Architecture 6](#_Toc9089689)

[3 Files and Database Design. 7](#_Toc9089690)

[3.1 Database Management System files. 7](#_Toc9089691)

[3.1.1 ERD 7](#_Toc9089692)

[3.2 Non-Database management system files 7](#_Toc9089693)

[3.2.1 Static Files 7](#_Toc9089694)

[3.2.2 Media Files. 7](#_Toc9089695)

[4 Human-Machine Interfaces 8](#_Toc9089696)

[4.1 Inputs 8](#_Toc9089697)

[4.2 Outputs 10](#_Toc9089698)

[4.3 The admin panel. 10](#_Toc9089699)

[5 Detailed Design. 13](#_Toc9089700)

[5.1 Hardware Detailed Design 13](#_Toc9089701)

[5.2 Software Detailed Design. 13](#_Toc9089702)

[5.2.1 flow chart 13](#_Toc9089703)

[5.2.2 Class Diagram 13](#_Toc9089704)

[5.3 Internal Communication Detailed Design 13](#_Toc9089705)

[6 External Interfaces 14](#_Toc9089706)

[7 System Integrity Control. 15](#_Toc9089707)

# Introduction

## Purpose and Scope

The System Design Document describes the system requirements, operating environment, system and subsystem architecture, files and database design, input formats, output layouts, human machine interfaces, detailed design, processing logic, and external interfaces.

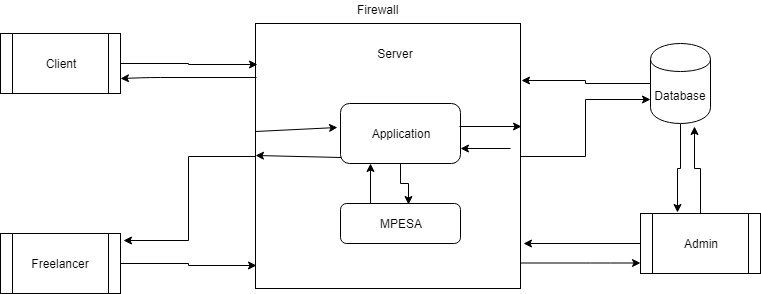
## Project Executive Summary

Online worksite is a web-based system implementing the client-server architecture build with Django framework. This will try to fix gaps in other system to achieve maximum number of users for being more reliable. Online worksite has 3 categories of users whose problems are being solved. The admin being the overseer of all the processes in the system, is the superuser who determines which people qualify to use the system and control what they need to access. The client posts their tasks to the platform, these tasks will be visible to freelancers whose accounts are activated. After completing the task, they return back to the owner of the tasks, who’s the client and wait for payments. In case of issues the system will provide a messaging platform to raise their concerns to specific users. The clients can further rate the freelancers as to how they did their task.

### System overview

Online worksite tries to solve the issue of unemployment situation experienced by most people mostly being youths, fresh graduates and those with low levels education but have skills in some specific fields. The system allows client to post tasks, pay for the tasks to a freelancer who have skills required for the task. When the freelancer finishes a task, they post it back to the client who further recommends if it is satisfying or not. The client and the admin must have their account activated to for them to undertake all those activities.

The system architecture is as below:



### Design Constraints

The system is designed to make sure all the needs of each clients are fulfilled and incase of anything the admin will be readily available or even call the technical team to improve or add more functionalities. For the user to use the system, it is assumed that they will have to read the system manual which will be available online before they signup. The manual will have details of how each user will interact with the system at their different levels. They are further assumed to have basic knowledge of using computers, since the system will not provide any guidance of using a computer. During design, it is assumed the system will work in a network-based environment taking the client-server architecture, with internet access.

### Future Contingencies

If the new requirements arise after the design, the system is expected to adapt them and be added to the next version of the system. The system is open to scale up as long as the requirements of the users are being fulfilled.

## Document Organization.

|  |  |  |
| --- | --- | --- |
| Chapter | Topic | Description |
| 1 | Introduction | Introduces what is going to be discussed in the whole document |
| 2 | System Architecture | Describes the system and/or subsystem architecture for the project. |
| 3 | File and Database Design | Shows the final design of the database, ER-Diagrams and Relational Diagram with all the entities |
| 4 | Human Machine Interface | Shows the detailed design including input and output interfaces of the system and subsystem. |
| 5 | Detailed Design | Data that is necessary for system implementation |
| 6 | External Interface Design | Describes how the system will connect with other systems. |
| 7 | Integrity Control |  |

## Points of Contact

|  |  |
| --- | --- |
| Person | Role |
| System Manager | This is the person who should fully understand the system and takes care of queries, feedback and concerned raise by other people. |
| Quality Assurance | Test the system and ensure that the functionality meets the market standards before actual deployment. |
| Security Manager | This person ensures the data in the system is protected and is not used for malicious purposes. |
| User Organization. | The users of the system. |
| Configuration Manager | Ensures all the subcomponents of the system communicate with each other perfectly. |

## Glossary

|  |  |
| --- | --- |
| Term | Meaning |
| Client | A person who posts jobs to the system. |
| Freelancer | A person who takes and does job for the client |
| API | Application Programming Interface |
| WSS | Web socket security |
| WS | Web socket |
| CRUD | Create Read Update Delete |
| HTML | Hyper Text Markup Language |
| CSS | Cascading Style Sheet |
| HTTP | Hyper Text Transfer Protocol |

# System Architecture

## System Hardware Architecture

The system is web-based system that will implement client server architecture, in this architecture; client PC which are the users of the system connects and sends requests to the server. The server processes the requests and if necessary, communicate with the database and sends the feedback back to the clients.



## System Software Architecture

Online worksite will implement the Model-View-Controller (MVC) which basically separate the key functionalities of the software into 3.

1. Model – acts a data store. Communicates with the view and the controller.
2. View – represents the visualization of the data. (User Interface). Communicates with the model.
3. Controller – manipulates data. Controls the data flow in the system and updates the view with any changes.



### Database

The system will connect the database use MySQL connector, which contain drivers necessary to connect to the database using Django framework.

### Messaging

The system shall apply web sockets to ensure users can communicate message while on the platform

### Payments

The system shall extend MPESA API that will be used to process payments, but later on other payments methods will be integrated to support more users and also for flexibility purposes.

### MySQL

This is a database management software that will be used to design and create database for the website. It is an easy-to-use to use software for database management.

### UML

Unified Modeling Language, is a software that is used to design details of a system using standard UML diagrams. It will be used to model the system diagram showing how each component will communicate with the others.

### Django Framework.

Is a python web development framework used for web development implementing model-view-template which is basically MVC, that separate database, views and controller making it easier to develop or edit and existing software. Its primary goal is to ease the creation of complex, database-driven websites by encouraging code reuse and ‘plug ability’ of components.

### Python

Python is a powerful, high-level, interpreted, multi-purpose programming language that is simple with organized syntax. It is dynamically typed and supports multiple programming paradigms such as procedural, object-oriented and functional programming.

### VS CODE

Visual Studio code is an open source text editor that can be customized by installing plugins for editing and writing source code of a software. We will use this software for writing the source code of the system.

## Internal Communication Architecture

Web browsers and servers will communicate via TCP/IP. Hypertext Transfer Protocol (HTTP) is the standard application protocol on top of TCP/IP supporting web browser requests and server responses. The HTTP and WS will use port 80 while WSS will use port 443. The database will communicate using port 3306.

For payment purpose the system will use APIs

The system will typically be deployed on three servers:

1. A web server
2. Storage for various application configuration files (e.g., CSS, images, etc.);
3. A MYSQL database server to host the database

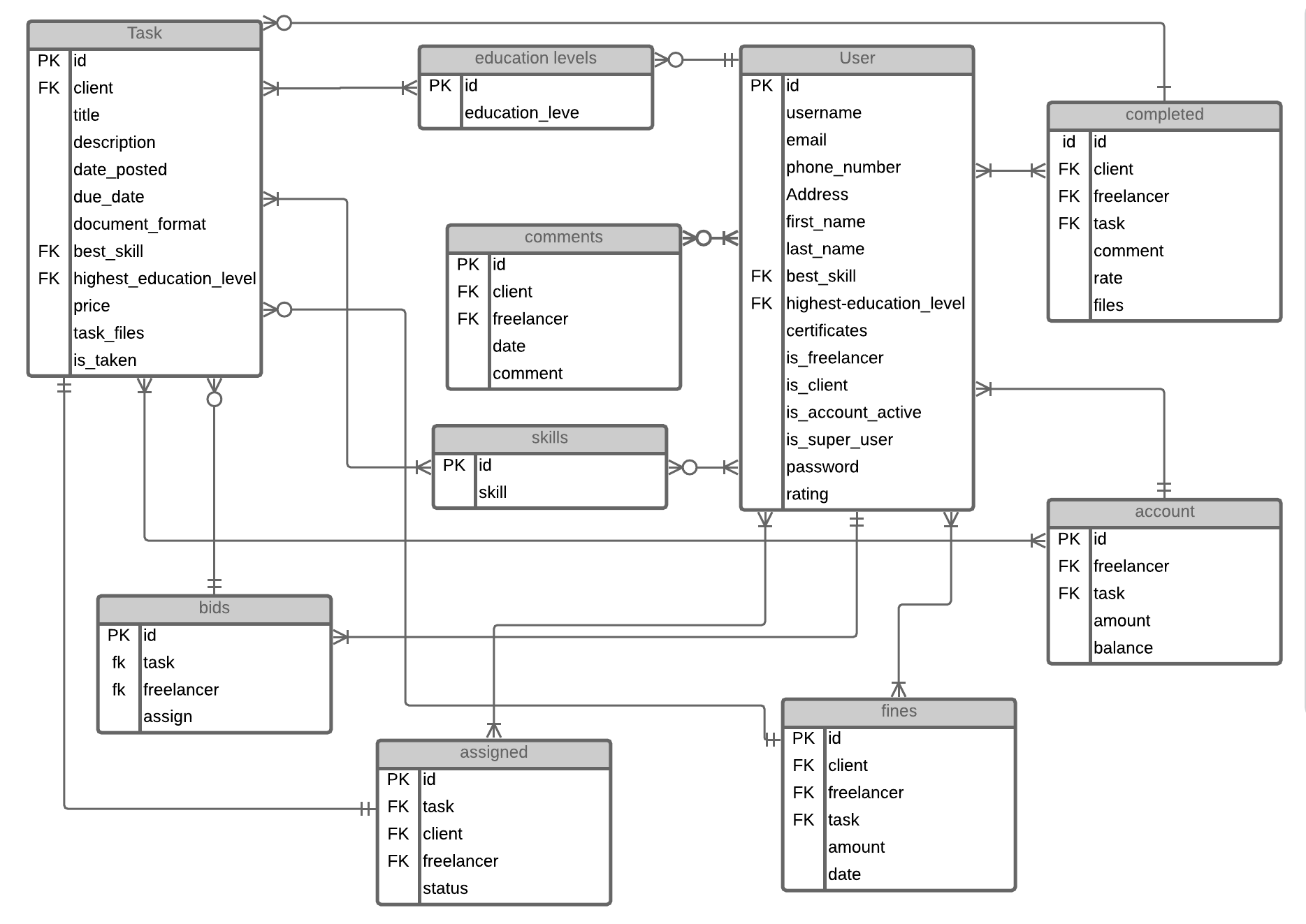
# Files and Database Design.

## Database Management System files.

MySQL is a database management software that will be used to manage the database. It enables data definition, data manipulation, data updating, data retrieval and user administration easily. MySQL supports CRUD operation easily.

### ERD

Shows how tables and entity are related to each other. The diagram below shows full relationship of the tables.



## Non-Database management system files

### Static Files

These are files which are not stored in the database but are part of the files of the system. The static files are stored in static directory at the root directory of the application. Some of these files include JavaScript and CSS files.

### Media Files.

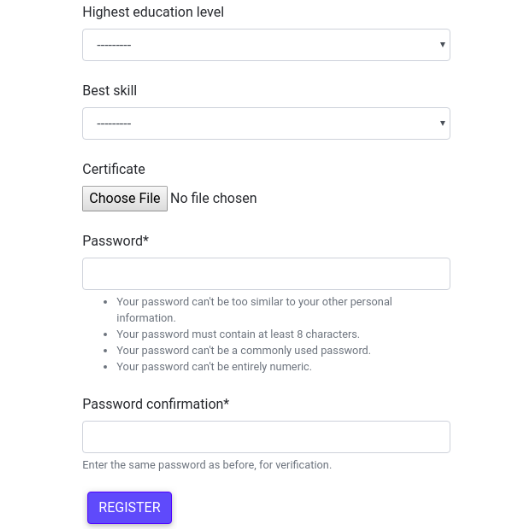
These are files that are uploaded to the system. The name and storage directory might be stored in database but the real files are stored in media directory at the root of the application.

# Human-Machine Interfaces

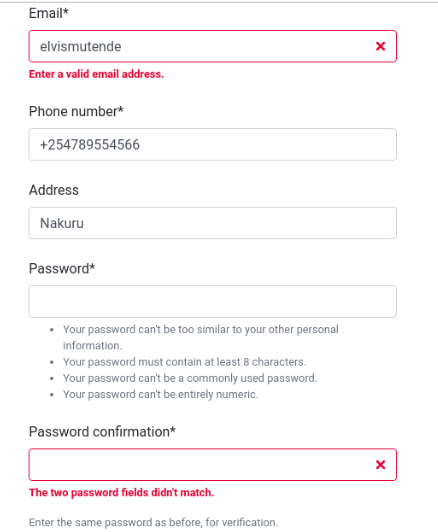
## Inputs

Data input in online worksite will be mostly through forms. These forms consist input fields, password fields, selection list and file fields. The data is captured validated and stored in the database. The stored data can be retrieved updated or deleted depending on the user needs. In case of any errors that occur during input of data, they are thrown to the users for guidance of what is expected. When the user provides required or expected they are redirected to the next page or a success message is displayed.

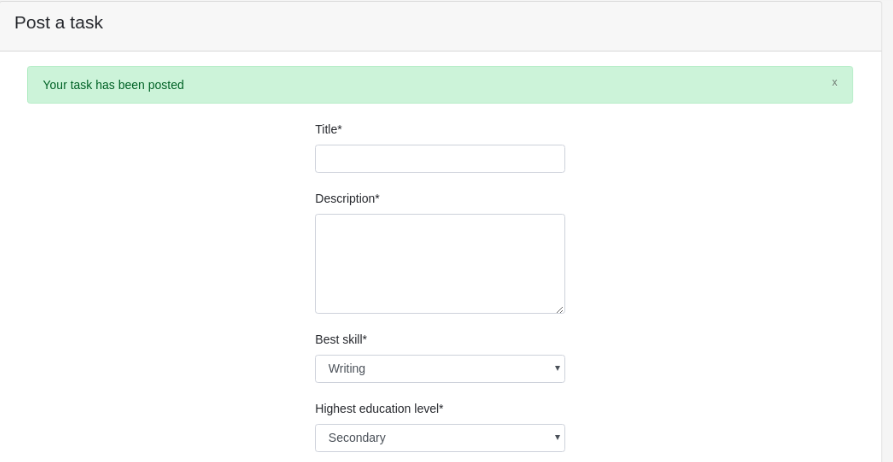
1. Input fields example.



1. Fields with errors



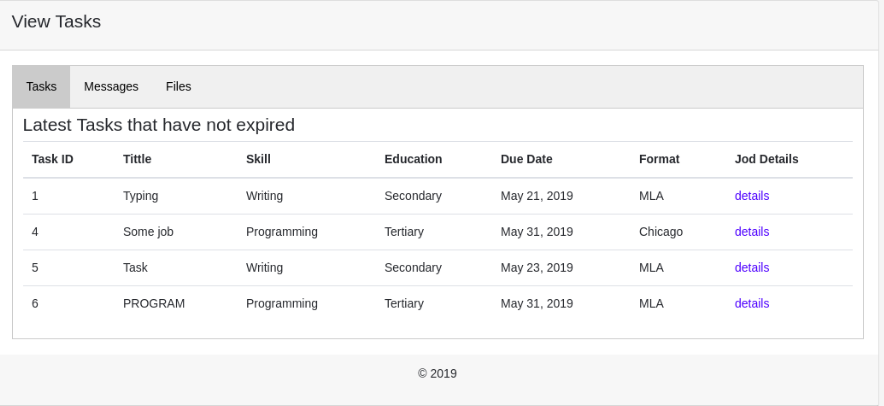
1. Fields with success message



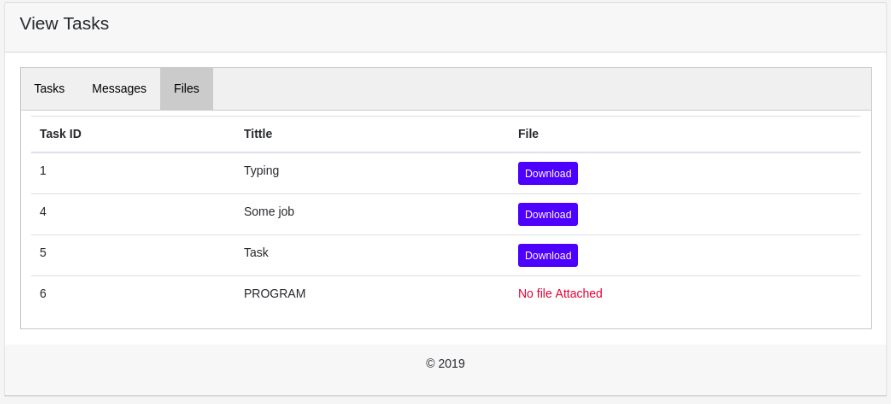
## Outputs

data output is summarized in tables. This is the best way for the users to easily visualize and understand whatever output is being displayed. Where the user’s attention is required, a different color is used to get the users’ attention.

1. List of tasks in table form



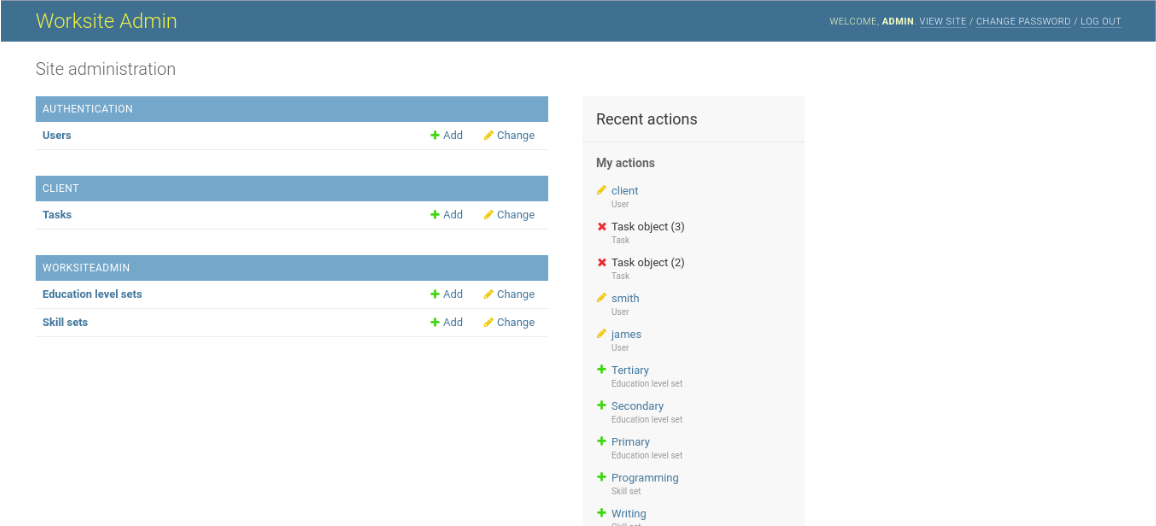
1. Point of focus with different color to attract users’ attention.



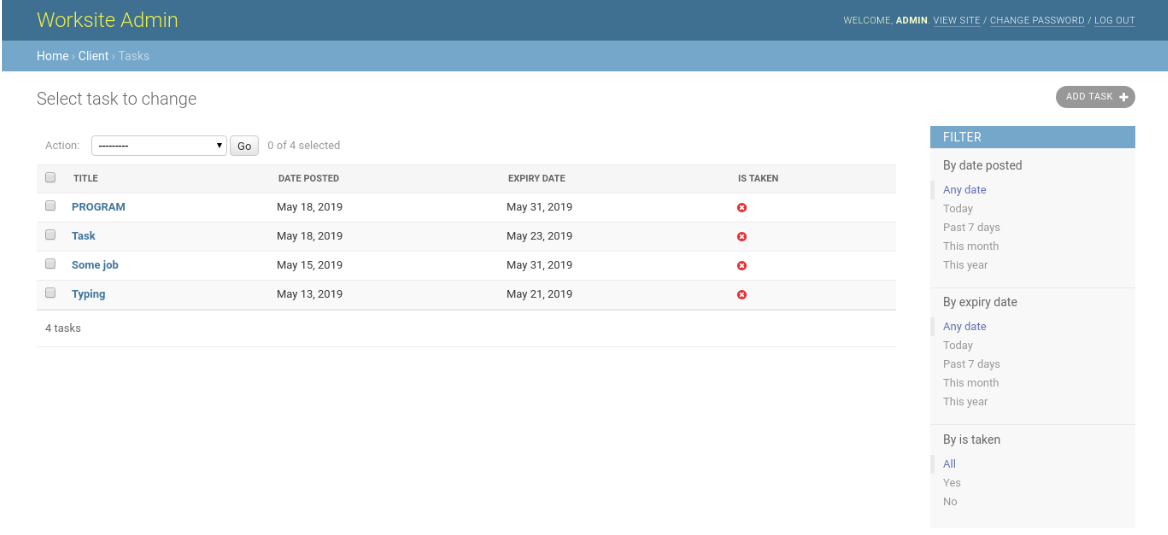
## The admin panel.

The admin has a different User interface that enables him/her see all the data in the database. Since they oversee everything, they need to get both summery and full details for each and every record.

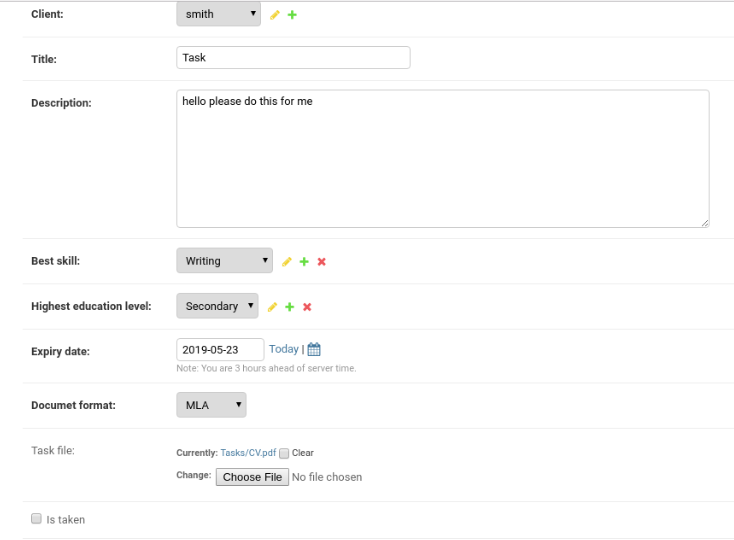
1. Admin panel general view



1. Admin list view



1. Admin detailed view



# Detailed Design.

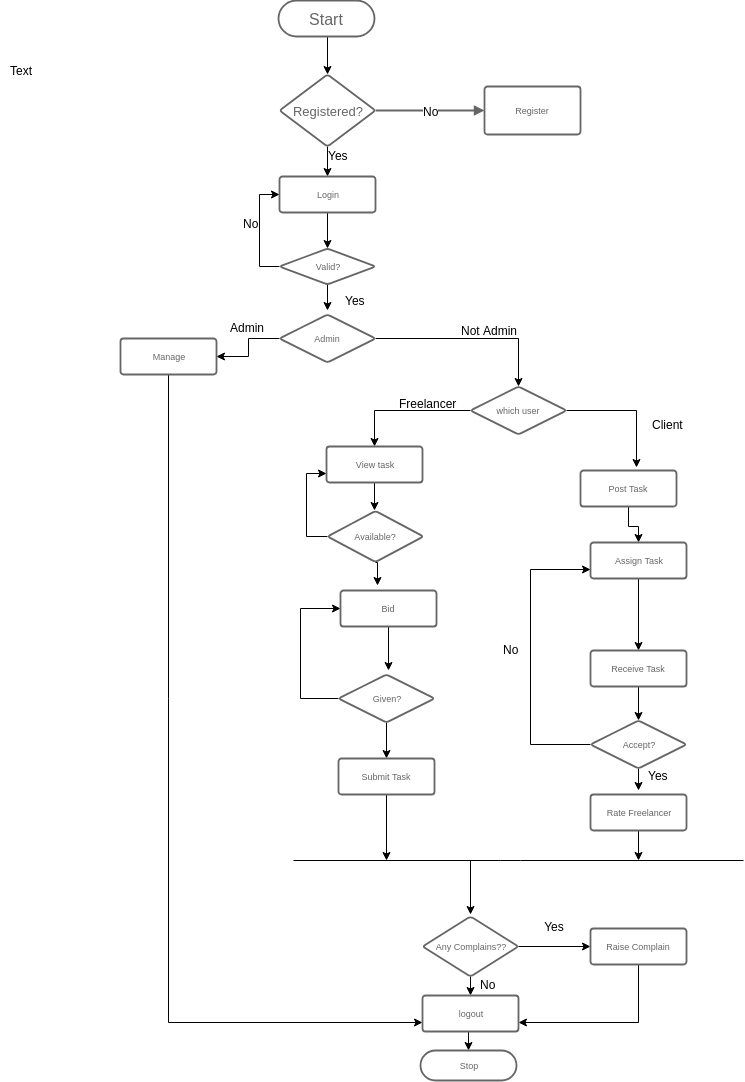
## Hardware Detailed Design

Online Worksite is a web-based system and the hardware parts of the user system are PC which can run web browser, all other hardware specifications are for deployment server.

## Software Detailed Design.

### flow chart

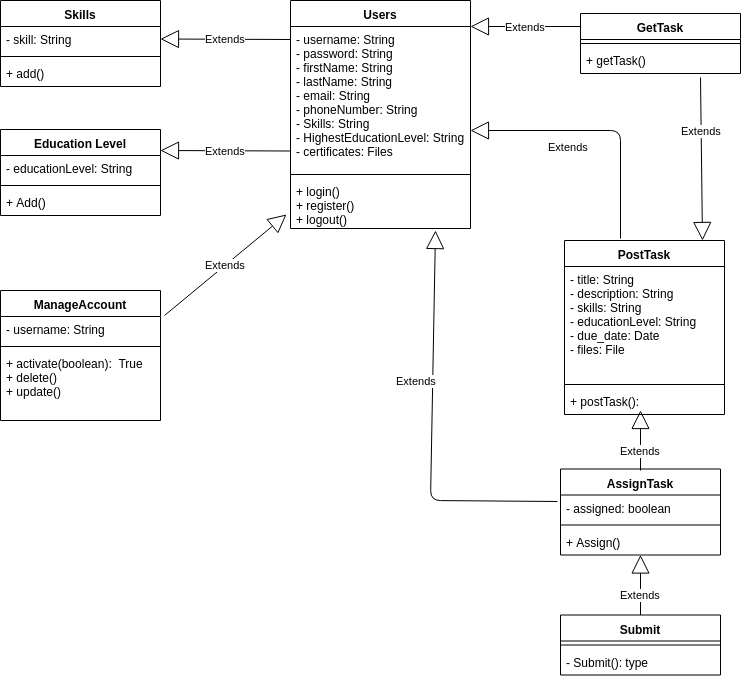
The flow chart shows the flow of activities from start to the end. The diagram below summarizes all the activities as they follow each other in order.



### 

### Class Diagram

How functionalities are achieved.

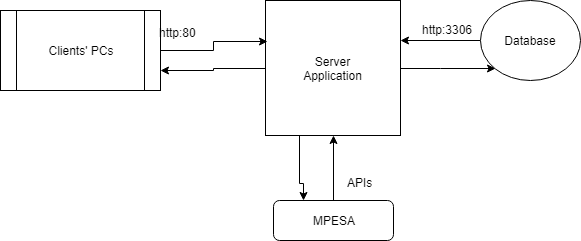


## Internal Communication Detailed Design

The system will implement client-server architectures, this means the client PCs will communicate to the server using HTTP protocol port 80. Since there is also a database which will be storing data for the system, the server will connect to the database using port 3306

Payments

The system will communicate with the MPESA via API to implement payment methods. Further details of how MPESA payments are implemented are documented on daraja API on Safaricom developer’s portal.



# External Interfaces

There are no other systems that are expected to connect with online worksite for exchange of data, in case there will be need to, then the design will have to change and scale up to accommodate the new system.

# System Integrity Control.

Data in any system is very important and needs to protect or control how it is accessed is very important. To implement this several things will have to be implemented to ensure data of online worksite is safe.

1. Authorization Control

only authenticated users can access data that belongs to their roles as per their limit of access.

1. Database Restriction

Users are limited to update data that they are supplied. Access of data affecting the whole system will be restricted unless it is the system administrator or the database administrator.

1. Integrity control

This will ensure that data stored in the database cannot be violated or altered by any unauthorized parties.

1. Confidentiality Control

This will ensure that unauthorized access of personal information will be prohibited.

1. Password Encryption

All password will be encrypted before they are stored in the database. In case someone accesses the database, they should not be able to get into users account and alter the accounts.

1. Data validation

Before data is stored in the database, it will be validated to ensure only expected data is stored e.g. email addresses.