System Title

SRS

Introduced By Author or Company Name

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Written By | Reviewed By | Approved By | Date |
| 0.X |  |  |  |  |
| 1.X |  |  |  |  |

# Introduction

## Executive Summary

Overview of product.

## Document Overview

Overview of the document.

## Abbreviations and Terminologies

Table of abbreviations and terminologies.

## References

List of external references.

# System Description

## Introduction

Provide detailed system description.

Use block or context diagram to illustrate the system architecture.

## Users

Provide a list of system users with brief explanations.

## Modules

Provide a list modules with brief explanations.

# 

# System Users

## User Description

Provide user role description in details.

## User Description

Provide user role description in details.

## User Description

Provide user role description in details.

# 

# System Modules

## Module Description

Provide module description.

Use block or context diagram to illustrate external and sub-modules.

Use activity diagram, state machine diagram, data flow diagrams to illustrate module operations.

## Module Description

Provide module description.

Use block or context diagram to illustrate external and sub-modules.

Use activity diagram, state machine diagram, data flow diagrams to illustrate module operations.

## Module Description

Provide module description.

Use block or context diagram to illustrate external and sub-modules.

Use activity diagram, state machine diagram, data flow diagrams to illustrate module operations.

# 

# System Functions

## [1.1] Shop Owner Functions

### [1.1.1] Add product Function

**Description**: This function make the shop owner add his products to the website.

**Inputs**: name of product, its type, salary .

**Outputs**: publish product to website .

**Pre-conditions**: shop owner must be logged in .

**Post-conditions**: description function must be created .

### [1.1.2] Edit Product Function

**Description**: Edit the product provided .

**Inputs**: The product you want to edit .

**Outputs**: Publish product after editing .

**Pre-conditions**: The product must be created first & shop owner must be logged in .

**Post-conditions**: none .

### [1.1.3] Delete Product Function

**Description**: Delete product of yours.

**Inputs**: The product you want to edit .

**Outputs**: return to the Home page .

**Pre-conditions**: The product must be created first & shop owner must be logged in .

**Post-conditions**: none .

## [1.2] customer Functions

### [1.2.1] Get My Favourite Function

**Description**: Add product to Favorite list.

**Inputs**: Favorite product.

**Outputs**: product tagged as Favorite .

**Pre-conditions**: The product must be created first & customer must be logged in.

**Post-conditions**: none

### [1.2.1] Edit Favorite Function

**Description**: Edit favorite list.

**Inputs**: Favorite list.

**Outputs**: list after edition .

**Pre-conditions**: The list must be created first & customer must be logged in.

**Post-conditions**: none

### [1.2.1] Delete Favorite Function

**Description**: Delete Product from favorite list.

**Inputs**: Favorite product.

**Outputs**: favorite list without that product .

**Pre-conditions**: The product must be tagged as favorite & customer must be logged in.

**Post-conditions**: none

## [1.3] User (Profile Module) Functions

### [1.3.1] Get info Function

**Description**: Create account to the user .

**Inputs**: email , password , username.

**Outputs**: account on website .

**Pre-conditions**: user must enter his personal info to create an account.

**Post-conditions**: none

### [1.3.2] Edit info Function

**Description**: Edit account information .

**Inputs**: User Account .

**Outputs**: account after edition .

**Pre-conditions**: User must be logged in .

**Post-conditions**: none

## [1.4] Product (Build Your Computer Module) Functions

### [1.4.1] Get CPU Function

**Description**: Getting CPU for product .

**Inputs**: Select CPU.

**Outputs**: CPU selected .

**Pre-conditions**: selected list must be created first.

**Post-conditions**: Build your Product function must be created after that.

### [1.4.2] Get GPU Function

**Description**: Getting GPU for product .

**Inputs**: Select GPU.

**Outputs**: GPU selected .

**Pre-conditions**: selected list must be created first.

**Post-conditions**: Build your Product function must be created after that.

### [1.4.3] Get RAM Function

**Description**: Getting RAM for product .

**Inputs**: Select RAM.

**Outputs**: RAM selected .

**Pre-conditions**: selected list must be created first.

**Post-conditions**: Build your Product function must be created after that.

### [1.4.4] Build your product Function

**Description**: Build a product of specifications you selected first .

**Inputs**: GPU,CPU,RAM.

**Outputs**: Product .

**Pre-conditions**: GPU,CPU & RAM must be selected first .

**Post-conditions**: none

## [1.5] Subject Functions

### [1.5.1] Get Label Function

**Description**: make a name for your product.

**Inputs**: product.

**Outputs**: Product named .

**Pre-conditions**: The product must be created first & shop owner must be logged in.

**Post-conditions**: none

### [1.5.2] Get Content Function

**Description**: make a Description for your product.

**Inputs**: product.

**Outputs**: Product descripted .

**Pre-conditions**: The product must be created first & shop owner must be logged in.

**Post-conditions**: none

### [1.5.3] Edit Subject Function

**Description**: edit your product’s name and description .

**Inputs**: product.

**Outputs**: Product edited .

**Pre-conditions**: The product must be created first & shop owner must be logged in.

**Post-conditions**: none

# System Models

## Using Class Diagram

## 

### Description

In the previous Diagram there is the parent class named User . this class has the responsibility to create accounts and edit personal data of the user .

The next class is Customer class which inherits from user all the data it has, so this class for the customers who will visit the website and the class has the responsibly to add and edit his favorite list .

The 3rd class called shop owner which has the responsibility to make the owners of the products upload the products and publish it on our website and edit it and so on.

This class also inherits from the user class .

The 4th class is product class which has the responsibility to build a product to the user who will use this feature.

The last one called subject and this one to give the product a name and describe it.

# Non-Functional Requirements

## [2.1] <Security> Requirements

### [2.1.1] <Security> Requirement

The system must be secure in storing passwords in Database, so Hashing the Passwords is required.

## [2.2] <Usability> Requirements

### [2.2.1] <Usability> Requirement

System should be usable for all kind of people to make Shopping easier to them , so Items should be viewed in organized way. Also You must make comparing between items very clear to the visitor.

## [2.3] <Performance> Requirements

### [2.3.1] <Performance> Requirement

You should treat with the load on the website to make the server always working and avoiding the server dropping . It is expected that the number of visitors will be average in the range :

(1500-3000)visitor daily .

## [2.4] <Technology> Requirements

### [2.4.1]< Technology> Requirements

The customer wants this website to performed in python – Django , so you need to setup an IDE for python and Supports Django .

### [2.4.2]<IDE>Requirements

We suggests to work by Pycharm IDE . It’s an excellent IDE having a lot of features and easy to use .

### [2.4.3]<Python>Environment

You also need to setup Python environment on your Computer ,so we suggests Python 3.7.

It is the latest version until now.

### [2.4.4]<Deployment>Requirement

Also you need website for your deployment to deploy your python – Django website. We suggests pythonanywhere as Deployment way for python.

## [2.5] <Development> Requirements

### [2.5.1]<Development > Language

You should be aware of Python language to do this website.

### [2.5.2]<Web Development>Framework

You must learn how to use Django to make this website , so choose any source and learn quickly .

## [2.6] <Delivery> Requirements

This project must be delivered before the end of January.

Deliver it by include the source code of all the Phases of this project.

# Domain Requirements

## [3.1] <Fund > Requirements

### [3.1.1] <Server> Requirement

You should have the required fund to deploy the website on a good server to avoid any server down problems.

## [3.2] <ISP Network> Requirements

### [3.2.1] <network speed> Requirement

Network speed must be stable and high to access the website quickly.

## [3.3] <Security> Requirements

### [3.3.1] <Security> Requirement

You must take care of any external device to the device you store the backup data to avoid steeling the personal data of the users.

## [3.4] <Buying > Requirements

### [3.4.1] <buying> Requirement

You should notice that the website isn’t responsible for any product . the website just a linker between the customer and the shop owner.

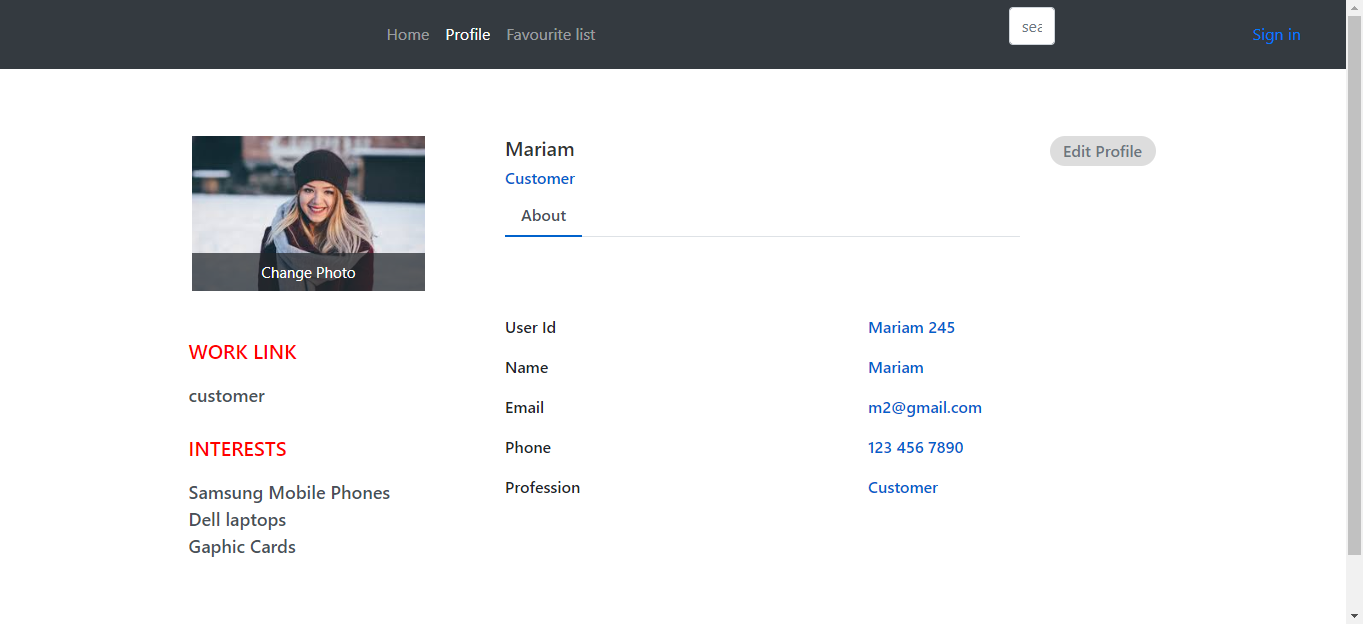
# System Interfaces

## User Interfaces

### Login Screens

### Signup

### Profile



Favorite list



### Home

