**NOTE: This template is shareware downloaded from** [www.processimpact.com](http://www.processimpact.com/)**. All shareware payments are donated to the Norm Kerth Benefit Fund to help a consultant who is disabled with a brain injury. Please visit** <http://www.processimpact.com/norm_kerth.html> **to make a shareware payment ($10 suggested). Thank you!**

Software Requirements Specification

for

TrendyFit

Table of Contents

1. Introduction 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Project Scope 1

1.5 References 1

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Features 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 2

2.6 User Documentation 2

2.7 Assumptions and Dependencies 3

3. System Features 3

3.1 System Feature 1 3

3.2 System Feature 2 (and so on) 4

4. External Interface Requirements 4

4.1 User Interfaces 4

4.2 Hardware Interfaces 4

4.3 Software Interfaces 4

4.4 Communications Interfaces 4

5. Other Nonfunctional Requirements 5

5.1 Performance Requirements 5

5.2 Safety Requirements 5

5.3 Security Requirements 5

5.4 Software Quality Attributes 5

6. Other Requirements 5

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

The mian purpose of this document is to describe our project requirements like software, hardware, functional and non functional. Project is about the online web application where user can check with designer to get ideas and suggestions about the outfits.We have chosen the name of the project as “TrendyFit”.

## Document Conventions

We have followed the SRS template format in this document. Fonts used in this document is Times New Roman. Also under every category, we have divided into subcategories based on the priority**.**

## Intended Audience and Reading Suggestions

This document is intended for team members , instructor and TA’s. Basically this document contains overview of the project, Features, Functional, Non functional requirements and external interface requirements. Explained the functionalities and technologies using in this project in detail.

## Project Scope

Software we are using in this project is Python, Django for developing the online web application. Main goal of this application is to have a connection between designer and the user for getting suggestions related to the fashion and outfits.

## References

We have followed the SRS template for documenting the project requirements in a specified way.

# Overall Description

## Product Perspective

It's an online web application where users can interact with the fashion designer to make suggestions in choosing the outfits which suit the user. This application will help to get suggestions related to fashion from known fashion designers. Main functionalities in this application are Admin, User, Designer. We have interfaces like Software, Hardware and User interfaces.

## Product Features

Major features of this web application are user management, view profiles, chat access, manage subscription,feedback. In this web application, three main modules are Admin, Designer, User where everyone can register to the application. Each one is having their own functionalities. Admin has overall access like Login, View , Manage and aceept designers. User can login and check the designers and chat with them. Designers can login and subscribe to the website and chat with users. Detailed functionalities will be explained in below.

## User Classes and Characteristics

Not applicable to our project**.**

## Operating Environment

Software we are using in this project is Python, Django. No particular hardware devices we are using, we can use this website in any browser in mobile or desktop. All the versions and operating systems details are mentioned in external requirements**.**

## Design and Implementation Constraints

For developing this application we are using Pycharm Tool and the framework we are suing is Django.

## User Documentation

Not applicable for this project.

## Assumptions and Dependencies

As of now we are not having any dependencies.

# System Features

# Functional Requirements

Functional requirements are product features that focus on user demands and describe what a software system must accomplish and how it must work. It is a summary of the service that the software must provide. It refers to a software system or a component of one. Here in our application, we offer a separate dashboard for admin, designer and user, and the functional requirements are:

Admin:

* Login: Admin is provided with a facility to log into the application using the credentials that are registered with our application.
* Accept Designer: Admin is given access to accept requests from the newly registered designers with the application.
* View Users: Admin can view registered users where admin will verify details like their id, status and subscription plan.
* Add plans: Admin can add plans for the designer like a silver plan for three months subscriptions, gold for six months and diamond for an annual subscription.
* Manage Subscriptions: Admin has the feature to manage subscriptions on request from the registered users.
* View Designer: Admin will be able to view details like status, plan and the id of the registered Designers.

Designer:

* Registration: The designer will register using their details like email id, password, and plan for the subscription.
* Login: The designer can log into the application using the credentials obtained from registration.
* View Users: The designer will also be able to view users for helping them with their knowledge in designing.
* Chat: The designers are provided with a feature to chat with the users to interact with them, which helps users choose their designer
* Change Password: The designers will have an option to change their passwords in case of an emergency.

User:

* Chat: The users are allowed to chat with the designers to choose the designer of their choice.
* View Stylish: The users can view designers and check their profiles to know their past work and interact with them.
* Feedback: After a chat with the designer, the users can give feedback based on the conversation.
* My Details: The users can view the details given during registration, like their email or chosen plan.
* Change Password: The users will have an option to change their passwords in case of an emergency.

# External Interface Requirements

## User Interfaces

## Login Interface:

## In this interface, there will be a register Button and a login button. If users have not registered to the application, they will use the register button and fill in details like email id, password, Subscription Plan and type of user (Designer/normal user) and register for our application. After registering successfully, the user will be redirected to the login page, and they have to enter the credentials to log into the application

1. Register Interface:

In this interface, there will be a form requesting details for registering users to the application. The details requested are to be filled in by the user and click on register to submit the form. A new user profile will be created, and the details will be stored in the database.

1. Admin Dashboard:

In Admin Dashboard, he will have access to different functionalities like login to dashboard, Accept Designers requests, View Designers, user’s profiles, and can manage subscriptions. Admin can add plans like in our application we provide three subscription plans silver for three months, gold for six months and diamond for annual subscription.

1. Designer Dashboard:

In Designer Dashboard, First, they have to register into the application then login to the dashboard using their credentials, Here Designers will be having functionalities like choosing their plan for subscription, change their password, view their details, view users profile and can also chat with the users

1. User Dashboard:

Customers must first register for the application and then connect to the dashboard using their credentials. From here, users can choose their subscription plan, change their password, examine their details, view Designers' profiles, and interact with them.

Chat Interface:

We also provide a chat interface in our application for designers and users to connect with one another and for users to submit feedback to designers based on the chat.

4.2 Hardware Interfaces

Processor                                      :          Intel Pentium(R) Dual-Core Processor

Speed                                            :          2.9 GHz

RAM                                             :          6 GB RAM

Hard Disk                                     :          40GB

4.3 Software Interfaces

Operating System                         :        Ubuntu 18 Lts

Browser                                        :        Chrome/Firefox

Front End                                     :        HTML,CSS,JavaScript,Bootstrap5

Back End                                     :        Python, Django

Database                                      :        SQLite

4.4 Communications Interfaces

The application uses the server-side rendering concept, which describes an application's ability to produce a web page on the server rather than in the browser. We are using the Django framework for server-side rendering (SSR) and the rest protocol and javascript for client-side rendering. And our application does require access to an email id for login purposes. The system requires a wifi connection for accessing the application.

# Other Nonfunctional Requirements

It defines a software system's quality attribute, which evaluates the software system based on its responsiveness, usability, security, portability, and other non-functional criteria crucial to its success.

## Performance Requirements

The product is based on the web and runs on a web server. The web application loading time depends on the strength of the internet connection and the media from which the application is run. The client's/customer hardware components will determine the performance of our application.

## Safety Requirements

## Our application doesn't require many safety requirements as we are using SQLite as our database and also performing backup periodically using cron job scheduler, and overall security will be taken care of by the docker, which makes sure that the internal ports are not exposed to external.

## 5.3Security Requirements

Our application does prevent unauthorised users from accessing our application as it is intended only for the authorised users, and it also prevents DOS attacks as we are using cross-site request forgery (CSRF) token, which is difficult to guess by intruders.

* The system will make use of a secure database.
* Except for their personal information, normal users can only view information and cannot edit or modify it.
* There will be various categories of users in the system, each with their own set of access restrictions.

## Software Quality Attributes

1. Usability:

Usability refers to how readily a user can complete a task with the product; it is the outcome of deliberate, research-based, and user-tested design decisions made with one purpose in mind: to make it simple for users to complete their tasks. In the case of our application, we provide a separate dashboard based on the type of users like admin, designer and users according to their functionality. Every user can easily register to our application using details like email id, password and a plan of choice as we provide three different subscription plans like silver for three months subscription, gold for six months and diamond for an annual subscription.

1. Security:

If your site deals with monetary transactions, users' financial information, or sensitive data, security is critical. Our application protects you from unauthorised users who try to log in using someone else's credentials without their permission. Cross-site request forgery (CSRF) protection is used in our application to prevent such attacks from jeopardising user security.

1. Availability and maintainability:

The ability of the system to adapt to changes with ease is known as maintainability. When adding or updating the application's functionality to solve issues or satisfy new business requirements, these changes may have an influence on components, services, features, and interfaces. Maintainability can also affect how long it takes to get a system back online after a breakdown or when it's taken offline for an upgrade. Improved system maintainability can boost availability while also reducing the impact of run-time errors.

* The application is simple to maintain. As long as the device is in good functioning order, the programme will be available on the admin's system at all times.
* The application's functionality will be determined by any external services, such as internet connectivity, that are required.

1. Data Integrity:

The data collected from the systems are identical to what is stored on the server. The information is supplied as it is found on the server.

1. Real-Time:

The application will give you up-to-date information.

# Other Requirements

Not applicable to this project.