Contents

[1. Introduction 1](#_Toc419070508)

[1.1 Document Purpose 1](#_Toc419070509)

[1.2 Product Scope 1](#_Toc419070510)

[1.3 Intended Audience and Document Overview 2](#_Toc419070511)

[1.4 References 3](#_Toc419070512)

[1.5 Definitions, acronyms, and abbreviations 3](#_Toc419070513)

[2 Overall Description 5](#_Toc419070514)

[2.1 Product Perspective 5](#_Toc419070515)

[2.2 Product Functionality 5](#_Toc419070516)

[2.2.1 Functional Requirements 5](#_Toc419070517)

[2.2.2 Non-functional Requirements 6](#_Toc419070518)

[2.3 Users and Characteristics 7](#_Toc419070519)

[2.4 Design and Implementation Constraints 7](#_Toc419070520)

[2.5 Assumptions and Dependencies 8](#_Toc419070521)

[3 Specific Requirements 9](#_Toc419070522)

[3.1 External Interface Requirements 9](#_Toc419070523)

[3.1.1 User Interfaces 9](#_Toc419070524)

[3.1.2 Hardware Interface 12](#_Toc419070525)

[3.1.3 Software Interface 12](#_Toc419070526)

[3.2 Use Case View 13](#_Toc419070527)

[4 Other Nonfunctional Requirements 14](#_Toc419070528)

[4.1 Software Quality Attributes 14](#_Toc419070529)

[4.1.1 Maintainability 14](#_Toc419070530)

[4.1.2 Simplicity 14](#_Toc419070531)

[4.1.3 Extensibility 14](#_Toc419070532)

[4.1.4 System Security 15](#_Toc419070533)

# Introduction

This section gives a scope description and overview of everything included in this SRS document. Also, the purpose for this document is described, and a list of abbreviations and definitions is provided.

## Document Purpose

The purpose of this document is to give a detailed description of the requirements for the “Go Paperless” website. It will illustrate the purpose and complete declaration for the development of system. It will also explain system constraints, interface and interactions with other external applications. This document is primarily intended to be proposed to the investor for its approval and a reference for developing the first version of the system for the development team.

The current version of the document is version 1, which covers the requirement specifications of the entire project its which include two Sub System and two databases that are required for the operation of the system. The product is a Website as mentioned before and its design and technical requirements will be covered in this document. To understand the full scope of the project and its operational plan please refer to the Software Product Management Plan.

## Product Scope

The product GoPaperless is designed to provide a very easy but powerful paperless environment software. Designed to as to provide an implementation of paperless environment in an affordable manner for small and large organizations looking to shift to a more environmental friendly environment.

The product is specifically marketed to firms that have large internal paper usage by giving them a simple digital environment where forms and memos can be replicated and used without the need to print and reprint them multiple times. This offers a huge financial benefit as reduced paper usage and costs relating to them (such as archiving records) is greatly reduced. It also means that the reduced paper usage helps the environment as reduced paper requirement will mean a reduction in paper production and related environmental consequences (deforestation, chemical processing, carbon dioxide build up etc.).

To achieve this goal the proposed web applications will provide a powerful was to digitize and firms form based process without the need to have specialized software development or much technical knowledge.

## Intended Audience and Document Overview

This document is intended to provide the invertors an idea of the overall working and more specifically the requirements that will be met by the finished system. The project manager and upper management can review this document for future review and monitoring of the project before delivery or on critical milestones to ensure adherence to the specifications.

The project team will use this document to understand the requirements of the software so that mentioned requirements are farther analyzed and refined for implementation, these deign decisions will be contained in the Software Design Document (SDD). The final system will be developed based on these requirements and tasks will be monitored to ensure all requirements are met in the specified manner.

## References

* **Software requirements specification IEEE 830**

## Definitions, acronyms, and abbreviations

|  |  |
| --- | --- |
| **Table** | **Definitions** |
| User | A person who directly interact with the software |
| Administrator | System administrator for the website |
| Subscriber | User that has the main account linked to the payment |
| Customer | the person(s) who orders the travel packages |
| HCI | Human Computer Interaction |
| Html | Hyper Text Markup Language |
| CSS | Cascading Style Sheets |
| JSP | Java Server Pages |
| Bootstrap | is an open-source JavaScript framework developed by the team at Twitter |
| Web-forms | allows a user to enter data that is sent to a server for processing |
| GUI | Graphical user Interface |
| Front-end users | User site |
| Back-end users | Administrator site |
| Dynamically | web page whose construction is controlled by an application server processing server-side scripts |
| DB | Database |
| PC | Personal Computer |
| JavaScript | Front end scripting language. |
| Paperless | An environment that reduces paper usage or eliminated it all together |

# Overall Description

## Product Perspective

In GoPaperless web application is completely new self-contained product, being developed to provide a simple and very fast way for companies to shift from traditional paper form based process to a completely digital environmentally friendly environment without the need to much development experience or a huge financial and time investment to have a new software developed. The final product will offer a way to easily emulate the paper based processes in a web based environment without the ability or need to write code or develop a database.   
As this is a web based application it has various systems and subsystems that make up the final working software. Section 2 and 3 will go over various details of the final software.

## Product Functionality

GoPaperless is being developed as a web application there are various functional requirements that will be met by the final system. These requirements are discussed below. The functions in order of operations that will be done by a brand new user to the system.

### Functional Requirements

* The main webpage should be easy to use and will be used to provide information on the systems.
* A user will be able to subscribe to the services that are offered (packages).
* A Subscribed user and its organizational details will be stored
* An organization can have multiple administrator accounts.
* Admin users can create forms.
* The forms will be created using a simple to use drag and drop interface.
* The forms will not require any technical knowledge of web design, programming or database design to create.
* The forms once created will be submitted to the database pending review and “publishing” by the Subscriber or Super Admin user.
* All forms created are tied to the organization.
* The forms one published will generate code and database structures automatically behind the scenes.
* The admin user can create user/employee accounts.
* Users are notified of the account creation.
* Published forms are available to the users of an organization.
* Published forms can be filled with information by the users.
* The inserted data can be reviewed latter.

### Non-functional Requirements

* Web interface should be completely responsive (Bootstrap).
* The web interface should be very simple to use.
* The system should work on devices with various screen sizes.
* The drag and drop interface should to intuitive to use and should not lead to confusions.
* The web interface should be modern as well as minimalist to be attractive to the largest population of people.

## Users and Characteristics

This system will be used by many different users, operating under different roles. The functionality of the system will vary depending on the roles and below are the different types of users.

* **Subscriber**: this user creates the original account that is tied to the organization and payment details. It can access all features of the software and is responsible for the publishing of forms once they are complete so they can he used by simple users.
* **Admin**: this user is responsible for the creation and management of the different forms, this user will have access to the form building sub system. This user will also create user accounts.
* **User**/**Employee**: the user or employee account is the one used by most of the users that will be filling the forms. This is the most basic role with the least amount of freedom but is essential.

## Design and Implementation Constraints

There are various design and implementation constricts that the project team should consider when developing the project.

* The software is being developed using Java web technologies specifically JSP.
* The software will require a server capable of hosting a JSP based web application.
* The software will require a server that is capable of hosting MySQL based databases.
* The development team will need you have source control software to have access to the software repository.
* The development team will require Eclipse IDE for back end development, MySQL WorkBench for DB management and Sublime along with a modern web browser for front end development.

## Assumptions and Dependencies

These are the assumptions that the project team will take and base there design upon.

* All systems and database components being developed are unique to this system.
* The Website is simple to use and a user will have prior knowledge of who websites are generally laid out and used.
* The Admin users while not required to know the developmental details should still have some idea of form development such has field selections so that they can make “good” forms.
* All users have access to an internet capable device.
* All users use a modern web browser.
* Users will require access to the system 24 hours a day.

# Specific Requirements

## External Interface Requirements

### User Interfaces

Since this project’s primary objective is to provide users with an easy to use and understandable website through which they can perform the task; it is imperative that the user-interface be as clear (in terms of color scheme of the website) and perfectly readable (in terms of font size, color and type). The main focus would be on making a user-friendly theme which will be classy as well as informative.

There are 3 major types of user interfaces that are in the system. The 3 major GUI representations are pictured below. *All images are based on the templates being considered before development and the final project might look different in terms of design but will flow similar design language.*

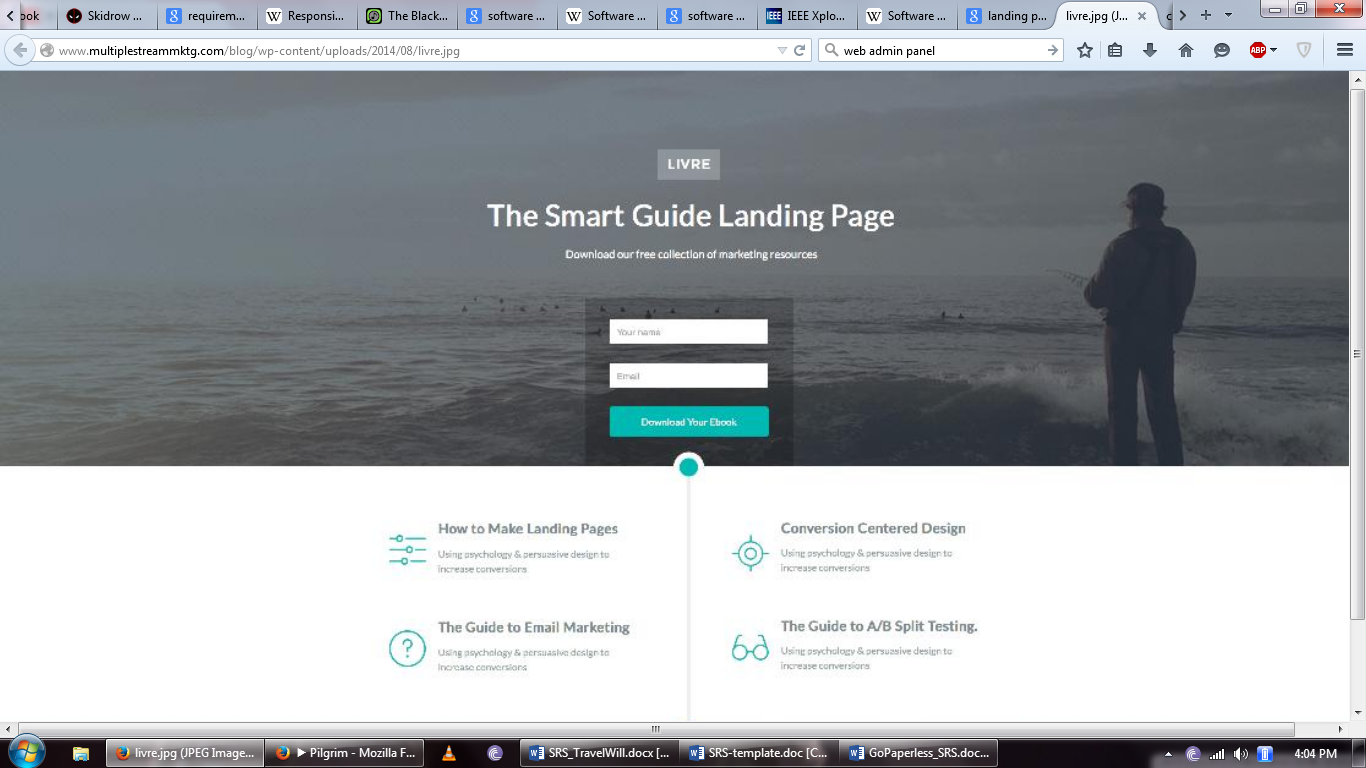


Figure 1 Landing page

The landing page is the first page the user will see and can be consider the main page. It will be relatively simple contain one subscription form or button and only information about the GoPaperless software along with information about its team.

The second part of the application will be a user portal/dashboard where any registered user will spend most of their time will using the application, using forms or for Admin managing and creating new forms.

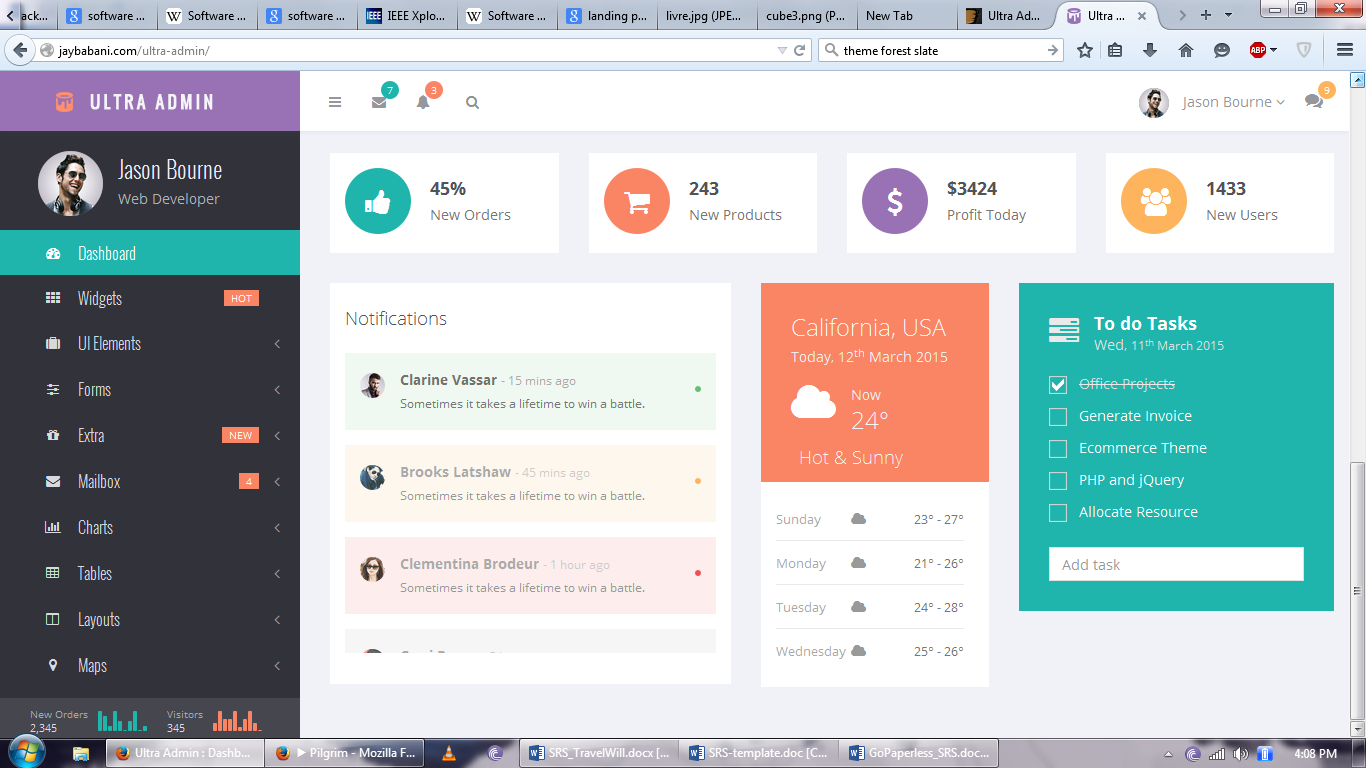


Figure Dashboard

The last part in the form building interface which will be incorporated into the user dashboard. A mockup of the drag and drop interface can be seen below.

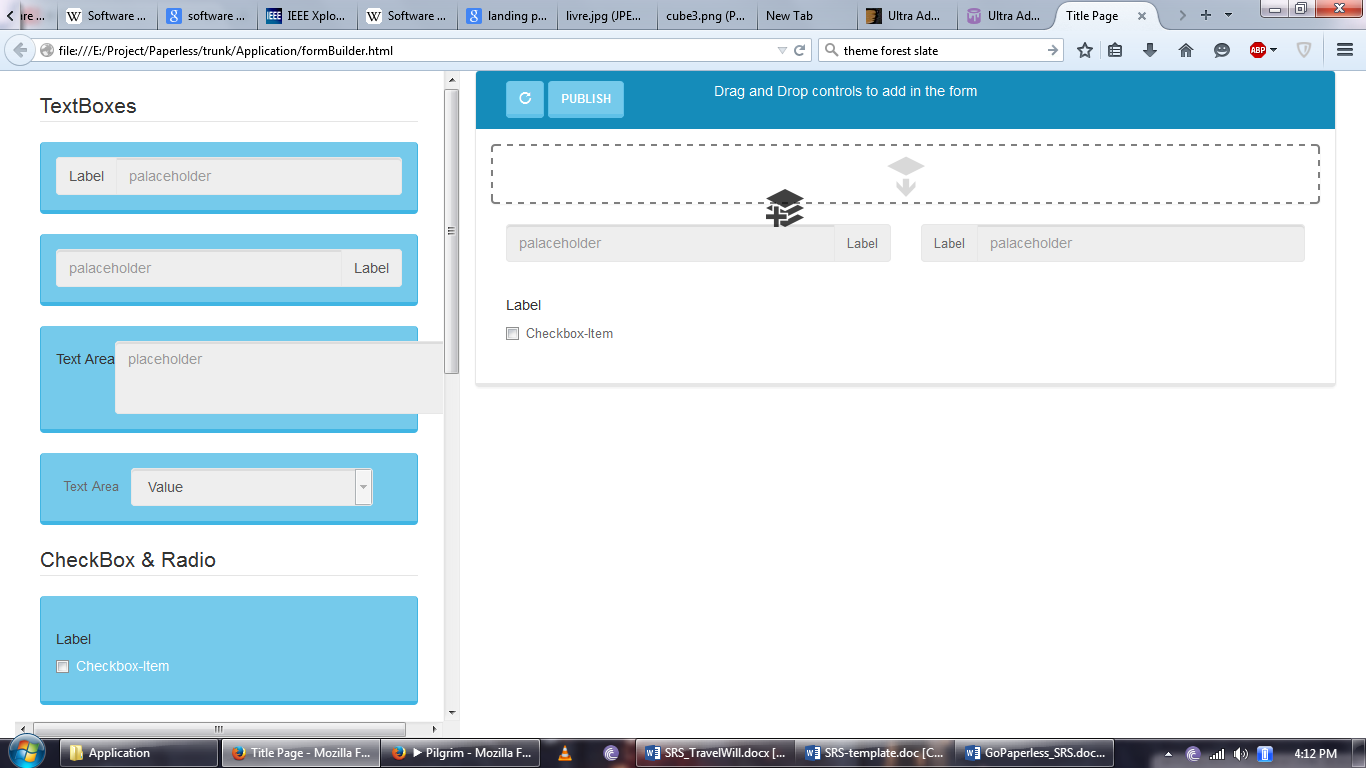


Figure 3 Drag and Drop operation

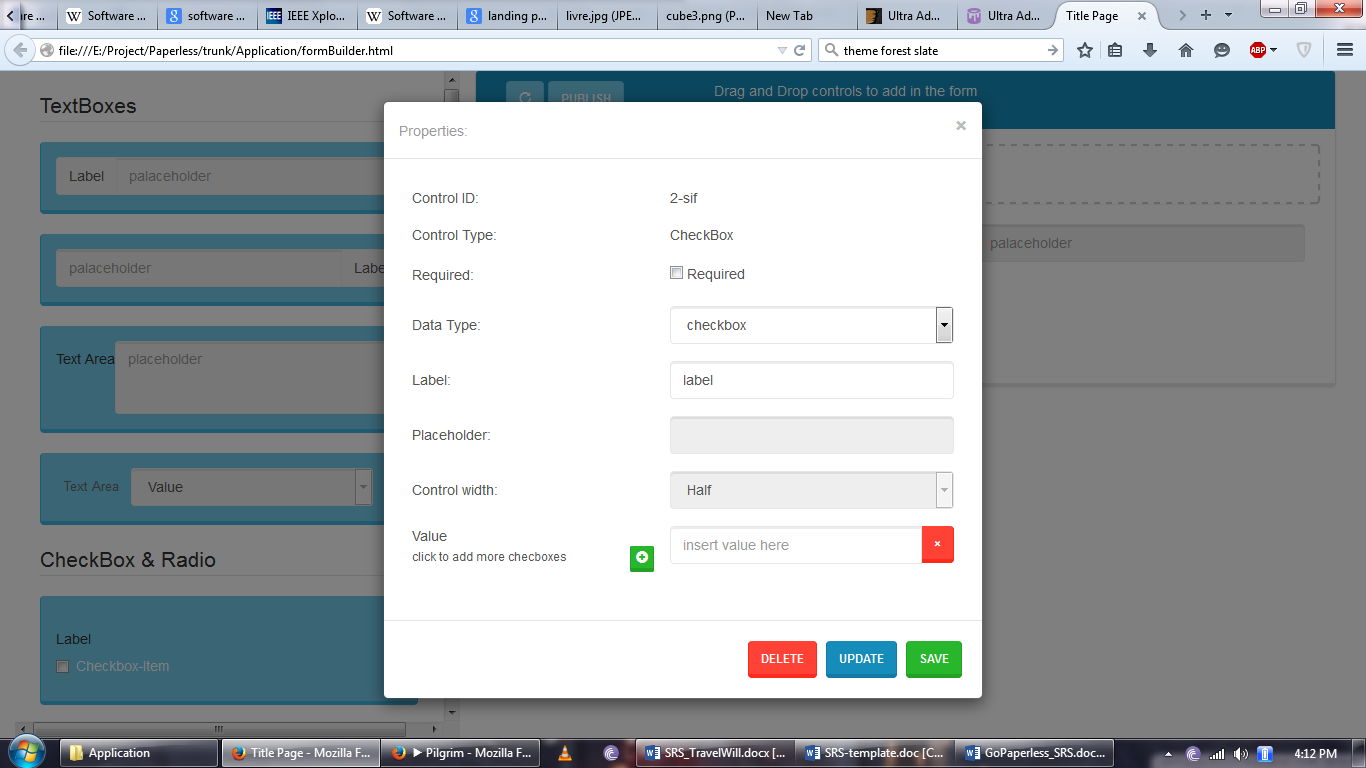


Figure 4 Field Editing

All the designs will feature flat and minimalist design language.

### Hardware Interface

For the time being, the only hardware interface which needs to be considered is the PC, touch screen of the smart phone devices. It is not one of the primary aspects of the project because it is a website along with a bootstrap which can easily open in any device. Though, other factors to consider would be the screen resolution and dimensions so that the users’ viewer experience of the interface would not be compromised.

### Software Interface

The software interface(s) refers to the back-end workings of the web application which is one of the primary focuses of this project. Hence, first off the main coding language which will be used Java (JSP), and Java Scripts will be used as well. Alongside this, HTML5 and CSS3 will be thoroughly used for creating the interface. This said there are chances of incorporating certain elements of other languages into the project as well given the circumstances based on the progress of the project. The database which will be used is MySQL5.

### Use Case View



Figure Go Paperless Usercase

# Other Nonfunctional Requirements

## Software Quality Attributes

### Maintainability

Maintainability is the type of modifiability representing the degree of easiness with which a business enterprise, system, application, or component can be modified between major releases when not required by changes to requirements. In this case, there is ample room for modification because this is a responsive project so changes in one sector will not concurrently require changes in the remaining sectors.

### Simplicity

Simplicity is referred to as the ease with which a person can use the program for its intended purpose, or in some cases even unanticipated purposes. Such issues can make or break its success even regardless of other issues. This involves a wide range of textual, graphical and sometimes hardware elements that improve the clarity, intuitiveness, cohesiveness and completeness of a program’s user interface.  
As the project is deigned to be used by many different types of people the design and operations should be simple and intuitive.

### Extensibility

Extensibility in terms of quality requirements of an application refers to the type of modifiability representing the degree of easiness of enhancing a system, application, or component to meet future goals and significantly changing requirements. As mentioned earlier, this system has a high maintainability ratio which also works as its extensibility factor as well. In fact, this system will constantly add new modules and features to keep user satisfied as well as attract new users to the software. And thing from a new control in the form building sub system to a whole new subsystem can be added without disturbing the functionality of the other modules.

### System Security

System security of the application depends on two major factors namely the efficiency in terms of performance and the robustness of the system. Efficiency in terms of the performance is the amount of system resources a program consumes (processor time, memory space, slow devices such as disks, network bandwidth and to some extent even user interaction): the less, the better. This also includes correct disposal of some resources, such as cleaning up temporary files and lack of memory leaks. This application is light without too many complex mechanisms running at the back-end which can attribute to the security of the system.

The second aspect deals with the robustness of the system which is basically how well a program anticipates problems not due to programmer error. This includes situations such as incorrect, inappropriate or corrupt data, unavailability of needed resources such as memory, operating system services and network connections, and user error. This project contains specially designed validators as well as security measures for important data such as digital signatures.