



---

# **SOFTWARE REQUIREMENTS SPECIFICATION**

**for**

**Online Postage Ordering System**

**Release 0.1.0**

**Version 0.1.0 approved**

**Prepared by Team Moon Moon**

---

Team Moon Moon

# Contents

<b>1</b>	<b>Introduction</b>	<b>7</b>
1.1	Purpose . . . . .	7
1.2	Project Scope and Product Features . . . . .	7
1.3	Defintions, Acronyms, and Abbreviations . . . . .	7
1.4	References . . . . .	7
1.5	Overview . . . . .	7
<b>2</b>	<b>Overall Description</b>	<b>8</b>
2.1	Product Perspective . . . . .	8
2.2	Product Functions . . . . .	8
2.2.1	Create An Account . . . . .	8
2.2.2	Print Postage . . . . .	8
2.2.3	Buy Postage . . . . .	8
2.2.4	List Postage Transactions . . . . .	8
2.2.5	Get Postage Balance . . . . .	8
2.2.6	Calculate Postage . . . . .	8
2.2.7	Track/Confirm Package . . . . .	9
2.2.8	Login . . . . .	9
2.2.9	Refund Request . . . . .	9
2.2.10	Validate Address . . . . .	9
2.3	User Characteristics . . . . .	9
2.4	Constraints . . . . .	9
2.5	Assumptions and Dependencies . . . . .	9
<b>3</b>	<b>Specific Requirements</b>	<b>10</b>
3.1	External Interface Requirements . . . . .	10
3.1.1	System Interfaces . . . . .	10
3.1.2	User Interfaces . . . . .	10
3.1.3	Hardware Interfaces . . . . .	10
3.1.4	Software Interfaces . . . . .	10
3.1.5	Communication Interfaces . . . . .	10
3.2	Functional Requirements . . . . .	10
3.2.1	Create an Account . . . . .	10
3.2.2	Print Postage . . . . .	11
3.2.3	Buy Postage . . . . .	11
3.2.4	List Postage Transactions . . . . .	11

3.2.5	Get Postage Balance . . . . .	11
3.2.6	Calculate Postage . . . . .	12
3.2.7	Track/Confirm Package . . . . .	12
3.2.8	Login . . . . .	12
3.2.9	Refund Request . . . . .	12
3.2.10	Validate Address . . . . .	12
3.3	Performance Requirements . . . . .	12
3.4	Logical Structure of the Data . . . . .	12
3.5	Design Constraints . . . . .	12
3.6	Software System Attributes . . . . .	13

# Revision History

1. Team Moon Moon Needs No Revisions.
2. Team Moon Moon Never Makes No Mistakes.
3. Correction. Team Moon Moon “fails fast.” Isn’t that the new buzzword or buzzphrase?
4. The skeleton has been updated to reflect the actual requirements provided by Dr. Seaman.

# 1 Introduction

Team Moon Moon has been tasked with developing a SRS for their Software Engineering class at Texas State University. If Team Moon Moon doesn't do this project, Team Moon Moon will fail D: Team Moon Moon wants to graduate already and doesn't feel like taking this class again.

## 1.1 Purpose

The purpose of this project is to get our C. Because C's get degrees!

## 1.2 Project Scope and Product Features

BLAH BLAH BLAH

## 1.3 Defintions, Acronyms, and Abbreviations

What is the airspeed velocity of an unladen swallow?

## 1.4 References

OMG OMG OMG

## 1.5 Overview

Aren't "Introduction" and "Overview" kinda the same thing? Just sayin'...

## 2 Overall Description

Many description. Much adjective. Wow.

### 2.1 Product Perspective

Such perspective. So edgy.

### 2.2 Product Functions

BRO DO YOU EVEN FUNCTION?

#### 2.2.1 Create An Account

Lol please shoot me now nobody does software development like this anymore.

#### 2.2.2 Print Postage

The Print Postage function shall allow the user to print postage that has been purchased, either to a file or directly to a printer.

#### 2.2.3 Buy Postage

TO THE MOON!!!

#### 2.2.4 List Postage Transactions

herpderp

#### 2.2.5 Get Postage Balance

The Get Postage Balance function shall allow the user to obtain their current amount in their postage buying account.

#### 2.2.6 Calculate Postage

The Calculate Postage function shall allow the user to enter a name, address, package type, mail class, and weight. The user shall then be able to see a calculated postage amount to ship their package.



### **2.2.7 Track/Confirm Package**

Here, you can incessantly check on the state of your package inbetween meals and trips to the loo! Never be out of the loop again. You really needed to know that your package left the seller's facility.

### **2.2.8 Login**

I can't let you do that Dave.

I can't let you do that, Dave.

Commas.

### **2.2.9 Refund Request**

It's been denied.

### **2.2.10 Validate Address**

Validating addresses is actually a royal PITA. Just let Google Maps API handle it.

## **2.3 User Characteristics**

Our users are in the 19-29 demographic. They love iProducts of all sorts and can't wait to buy postage stamps with our hip new website!

## **2.4 Constraints**

Batman has no constraints.

## **2.5 Assumptions and Dependencies**

I'm starting to run out of steam here guys.

## 3 Specific Requirements

As opposed to general requirements.

### 3.1 External Interface Requirements

aryweguiahergearpguhestrgperasguheasrgpewuhr

#### 3.1.1 System Interfaces

This product will interface with a Gibson. You know? Like, supercomputers they use to do physics and search for oil? Wouldn't you love to get one of those Gibsons?

#### 3.1.2 User Interfaces

CLI fo LYFE. Firefox is for nubs. Real haxors use lynx!

#### 3.1.3 Hardware Interfaces

awrygaeghsrethsegwearfawfewaf. I feel better now.

#### 3.1.4 Software Interfaces

gbsetghbsetgesrgfawefawefwaefawefawefawef. Much better.

#### 3.1.5 Communication Interfaces

Does the internet constitute a communication, software, or hardware interface? All of the above? All of the others? Some? Who knows.

### 3.2 Functional Requirements

Ok. Now I'm really running low on ideas.

#### 3.2.1 Create an Account

The system shall spam you. Violently.

PLEASE GIVE US YOUR EMAIL SO WE CAN SPAM YOU! I mean... OFFER YOU EXCLUSIVE PROMOTIONS ON INSIDER DEALS!

### **3.2.2 Print Postage**

The system shall...

1. First, ensure...
  - a) that there is a valid, logged-in user requesting the postage.
  - b) that the user is authorized to print the postage.
  - c) that the user has sufficient credit to print the requested quantity of postage.
  - d) that the user's printer is available and functional.
2. Then, the system shall prompt the user to confirm the printing of the postage, showing current and subsequent account balances as necessary.
3. Then, once confirmation from the user is obtained, the system shall generate the postage on a special page, devoid of any other content, that the user may print.
4. Finally, the system shall, upon detection of the completed print job, return the user to the "account status" page, where the new balance is displayed. The system shall also update the new balance in the user's account and add the job to the user's usage-history.

### **3.2.3 Buy Postage**

The system shall take your money and run. Maniacally.

### **3.2.4 List Postage Transactions**

The system shall list all the times it serviced your account. Suggestively.

### **3.2.5 Get Postage Balance**

The system shall...

1. First, ensure...
  - a) that there is a valid, logged-in user requesting the postage balance.
  - b) that the user is authorized to view the postage balance.
2. Then, the system shall find user account, and display postage balance in order for the user to see.

### **3.2.6 Calculate Postage**

The system shall...

1. First, ensure that there is a valid, logged-in user requesting the postage balance.
2. Then, the system shall provide a method for the user to enter a Name,Address,Packaging type, mail class, and weight.
3. Then, the system shall calculate estimated shipping cost for the users package.
4. Finally, the system shall, upon completion of calculation, display the estimated shipping costs.

### **3.2.7 Track/Confirm Package**

The system shall give you your tracking number. A day later.

### **3.2.8 Login**

The system shall let you in upon uttering the magic word. Please.

### **3.2.9 Refund Request**

The system shall take your request and pipe it to /dev/null. Cheerfully.

### **3.2.10 Validate Address**

The system shall throw its hands in the air and just flip a coin to determine whether the address is valid.

## **3.3 Performance Requirements**

Don't fail.

## **3.4 Logical Structure of the Data**

That would imply “logic” and “structure,” neither of which are currently present. The logic is back-ordered and the structure was never there to begin with.

In any case, this is where the main diagram goes.

## **3.5 Design Constraints**

Why would we put design constraints all the way at the bottom? You'd think the design constraints might illuminate pretty much all of the subsequent design decisions that were made and expounded upon in the entire SRS...

## 3.6 Software System Attributes

Is it at-tri-bute? Or a-trib-ute? Because one is a noun and one is a verb. Yeah, it's kinda the English language's way of saying FUCK YOU to foreigners.