

**ShipIT, An Interoffice Shipping Application**CS633 Software Quality, Testing and Security Management  
**Term Project—Version 1.0  
Final Report**Group #6  
Brian Calhoun  
Alicia Gallagher  
Steven Hoffman  
Eunjou Kim  
Carolina Torres  
February 28, 2017



# Version History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Description | Primary Author | Date |
| 0.1 | Module 1 Delivery   * Initial Draft * Project Scope * Team Composition | Steve H. | 23-Jan-2017 |
| 0.2 | Module 2 Delivery   * Added Personas * Added Requirements * Added Development Environment Information * Addressed Sunita’s comments by adding * Development Technology * Source code repository information. * A milestone plan * Links to the pivotal project and Git repository | Carolina T. | 30-Jan-2017 |
| 0.3 | Module 3 Delivery   * Added Estimation Record * Added CI Docs * Reformatted to group related sections together. * Addressed Sunita’s comments by adding * RASCI Chart | Steve H. | 06-Feb-2017 |
| 0.4 | Module 4 Delivery   * Added Use Cases * Added State Transition Diagram * Added Components Interaction Diagram * Miscellaneous Changes * Updated Track Shipments Wireframe * Added story references to the requirements. * Updated Story 1.1 to include the notes field. * Updated Test Cases. * Updated CI List. * Updated printing information in personas and requirements. * Addressed Sunita’s comments by adding * Links to the project documents in GIT. * (Note: We will update the estimation record with actual values next week.) | Steve H. | 13-Feb-2017 |
| 0.5 | Module 5 Delivery   * Miscellaneous Changes * Updated Test Cases * Final Version of Wireframes * Updated Estimation Record with actuals. * Updated CI List. * Addressed Sunita’s comments by adding * Added non-human persona as well as requirements. | Steve H. | 20-Feb-2017 |
| 0.6 | Module 6 Delivery   * Added Allpairs analysis. * Updated Test Cases with execution information. * Added Deviations discovered during the execution of the test cases. * Miscellaneous Changes * Updated estimation record. * Updated CI List | Steve H. | 27-Feb-2017 |

# Table of Contents

Version History 1

Table of Contents 2

Project scope 4

A. Overview 4

B. Roles and Responsibilities 4

C. Configuration Item List 4

D. Milestone Plan 6

E. Estimation Record 7

F. Assumptions 8

Client Information 8

A. Personas 8

1. Persona #1: SHIPPER & RECEIVER 8

2. Persona #2: Receiver 9

3. Persona #3: Interoffice Courier 10

4. Persona #4: Shipping Manager/ADMIN 10

5. Persona #5 – JSON Deserializer 11

B. Requirements 12

C. RASCI Chart 14

D. Selected Use Cases 15

1. Create a Shipment in ShipIT 15

2. Edit a Shipment in ShipIT 16

3. Track a Shipment in ShipIT 17

4. Print a Shipment in ShipIT 18

5. Mark a Shipment as Picked Up in ShipIT 19

6. Mark a Shipment as Delivered in ShipIT 20

7. Add Users to ShipIT 21

8. Remove Users from ShipIT 22

Product 22

A. Development Environment 22

B. Components Interaction diagram 22

C. State Transitions 23

D. Wireframes 24

1. Main Form/Track Shipments 24

2. Create Shipment 25

3. Edit Shipment 26

4. Update Status 27

5. User Management 28

6. Add Employee 29

7. Add Courier 30

E. Test Cases 30

1. Test Case #1: A User Can Create a Shipment 31

2. Test Case #2: Edit/Cancel a Shipment 34

3. Test Case #3: Shipper&Reciever or Receiver Can View Shipment Information 37

4. Test Case #4: Courier Can View & Update Status 38

5. Test Case #5: Admin can Add/Remove Employee Information 39

6. Test Case #6: Admin can Add/Remove Interoffice Courier Information 41

F. Deviations 43

G. Data Driven Combinations/Allpairs Results 43

References 45

# Project scope

## Overview

ShipIT is a lightweight inter-office shipping application that was developed in-house for BigCorp to alleviate problems that were experienced routing packages between departments. ShipIT provides a shipper the ability to register one or more packages for pickup. Inter-office couriers are notified of all pickups with source and destination information. Recipients use ShipIT to confirm receipt. Finally, shipping managers/administrators rely on ShipIT to provide real time information on all packages.

## Roles and Responsibilities

|  |  |  |
| --- | --- | --- |
| Individual | Role | Primary Responsibilities |
| Steven Hoffman | Team Lead | * Facilitates Meetings * Tracks Actions * Weekly Submissions |
| Brian Calhoun | Senior Developer | * Architecture * Platform Selection * Guides Resolution of Merge Conflicts |
| Alicia Gallagher | Usability Lead | * Creates Wireframes * UX Design |
| Eunjou Kim | Test Expert | * Creates and Executes test cases |
| Carolina Torres | Business Analyst | * Requirements Gathering * Process Modeling/Mapping |

## Configuration Item List

The following documents are initially edited and stored in Drive or locally on individual PCs. Upon publishing a new revision, a copy of the document is updated within the Controlled Docs section of the project’s GIT repository.

| # | Configuration Item Name | Version | Date | Owner | Repository |
| --- | --- | --- | --- | --- | --- |
| 1 | Personas | 4.0 | 19-Feb-2017 | Carolina T. | GIT |
| 2 | Requirements | 5.0 | 19-Feb-2017 | Carolina T. | GIT |
| 3 | Use cases | 1.0 | 11-Feb-2017 | Carolina T. | GIT |
| 4 | Estimation record | 2.0 | 20-Feb-2017 | Carolina T. | GIT |
| 5 | Configuration Items List | 8.0 | 27-Feb-2017 | Steven H. | GIT |
| 6 | Components Interaction Diagram | 1.0 | 13-Feb-2017 | Brian C. | GIT |
| 7 | State Transition Diagram | 1.0 | 11-Feb-2017 | Steven H. | GIT |
| 8 | Wireframes | 1.0 | 19-Feb-2017 | Alicia G. | GIT |
| 9 | Test Cases | 1.0 | 26-Feb-2017 | Eunjou K. | GIT |
| 10 | Data-driven combinations | 1.0 | 24-Feb-2017 | Steven H. | GIT |
| 11 | Report | 0.5 | 20-Feb-2017 | Steven H. | GIT |
| 12 | Project Policy | 1.0 | 24-Jan-2017 | Steven H. | GIT |
| 13 | RASCI Chart | 0.1 | 05-Feb-2017 | Steven H. | GIT |
| 14 | Presentation | 1.0 | 26-Feb-2017 | Steven H. | GIT |

The following table provides URLs for the documents in GIT. To access the file, add the value in the table to the following root: https://github.com/calhooligan/ShipIT/blob/master/ . Clicking the links in the table brings you directly to the file.

| # | CI Name | GIT URL |
| --- | --- | --- |
| 1 | Personas | [Controlled%20Docs/ShipIT%20CS633%20Term%20Project%20Group%206-Personas.docx](https://github.com/calhooligan/ShipIT/blob/master/Controlled%20Docs/ShipIT%20CS633%20Term%20Project%20Group%206-Personas.docx) |
| 2 | Requirements | [Controlled%20Docs/ShipIT%20CS633%20Term%20Project%20Group%206-Requirements.docx](https://github.com/calhooligan/ShipIT/blob/master/Controlled%20Docs/ShipIT%20CS633%20Term%20Project%20Group%206-Requirements.docx) |
| 3 | Use cases | [Controlled%20Docs/ShipIT%20CS633%20Term%20Project%20Group%206-Use%20Cases.docx](https://github.com/calhooligan/ShipIT/blob/master/Controlled%20Docs/ShipIT%20CS633%20Term%20Project%20Group%206-Use%20Cases.docx) |
| 4 | Estimation record | [Controlled%20Docs/ShipIT%20CS633%20Term%20Project%20Group%206-Estimation%20Record.docx](https://github.com/calhooligan/ShipIT/blob/master/Controlled%20Docs/ShipIT%20CS633%20Term%20Project%20Group%206-Estimation%20Record.docx) |
| 5 | Configuration Items List | [Controlled%20Docs/ShipIT%20CS633%20Term%20Project%20Group%206-CI%20List.docx](https://github.com/calhooligan/ShipIT/blob/master/Controlled%20Docs/ShipIT%20CS633%20Term%20Project%20Group%206-CI%20List.docx) |
| 6 | Components Interaction Diagram | [Controlled%20Docs/ShipIT%20CS633%20Term%20Project%20Group%206-Components%20Interaction.docx](https://github.com/calhooligan/ShipIT/blob/master/Controlled%20Docs/ShipIT%20CS633%20Term%20Project%20Group%206-Components%20Interaction.docx) |
| 7 | State Transition Diagram | [Controlled%20Docs/ShipIT%20CS633%20Term%20Project%20Group%206-State%20Transitions.docx](https://github.com/calhooligan/ShipIT/blob/master/Controlled%20Docs/ShipIT%20CS633%20Term%20Project%20Group%206-State%20Transitions.docx) |
| 8 | Wireframes | [Controlled%20Docs/ShipIT%20CS633%20Term%20Project%20Group%206-Wireframes.docx](https://github.com/calhooligan/ShipIT/blob/master/Controlled%20Docs/ShipIT%20CS633%20Term%20Project%20Group%206-Wireframes.docx) |
| 9 | Test Cases | [Controlled%20Docs/ShipIT%20CS633%20Term%20Project%20Group%206-Test%20Case.docx](https://github.com/calhooligan/ShipIT/blob/master/Controlled%20Docs/ShipIT%20CS633%20Term%20Project%20Group%206-Test%20Case.docx) |
| 10 | Data-driven combinations | [Controlled%20Docs/ShipIT%20CS633%20Term%20Project%20Group%206-Data%20Driven%20Combinations.docx](https://github.com/calhooligan/ShipIT/blob/master/Controlled%20Docs/ShipIT%20CS633%20Term%20Project%20Group%206-Data%20Driven%20Combinations.docx) |
| 11 | Report | [Controlled%20Docs/ShipIT%20CS633%20Term%20Project%20Group%206.docx](https://github.com/calhooligan/ShipIT/blob/master/Controlled%20Docs/ShipIT%20CS633%20Term%20Project%20Group%206.docx) |
| 12 | Project Policy | [Controlled%20Docs/ShipIT%20CS633%20Term%20Project%20Group%206-Policy.docx](https://github.com/calhooligan/ShipIT/blob/master/Controlled%20Docs/ShipIT%20CS633%20Term%20Project%20Group%206-Policy.docx) |
| 13 | RASCI Chart | [Controlled%20Docs/ShipIT%20CS633%20Term%20Project%20Group%206-RASCI.docx](https://github.com/calhooligan/ShipIT/blob/master/Controlled%20Docs/ShipIT%20CS633%20Term%20Project%20Group%206-RASCI.docx) |
| 14 | Presentation | [Controlled%20Docs/ShipIT%20CS633%20Term%20Project%20Group%206.pptx](https://github.com/calhooligan/ShipIT/blob/master/Controlled%20Docs/ShipIT%20CS633%20Term%20Project%20Group%206.pptx) |

## Milestone Plan

| % Complete | Task Name | Due | Responsible |
| --- | --- | --- | --- |
| 100% | Week 1 Deliverables | Mon 1/23/17 |  |
| 100% | Register with Pivotal | Tue 1/17/17 | All |
| 100% | Create GIT Account | Tue 1/17/17 | All |
| 100% | Propose Project Scope | Mon 1/23/17 | All |
| 100% | Propose Team's Composition | Mon 1/23/17 | All |
| 100% | Submit Deliverables | Mon 1/23/17 | Steve H. |
| 100% | Week 2 Deliverables | Mon 1/30/17 |  |
| 100% | Document Personas | Sun 1/29/17 | Carolina T. |
| 100% | Develop Requirements in Pivotal | Thu 1/26/17 | Carolina T. |
| 100% | Start Coding | Thu 1/26/17 | Brian C. |
| 100% | Feedback from Sunita | Mon 1/30/17 | All |
| 100% | Submit Deliverables | Mon 1/30/17 | Carolina T. |
| 100% | Week 3 Deliverables | Mon 2/6/17 |  |
| 100% | Provide Estimation Record |  | Carolina T. |
| 100% | Compile CI List |  | Steve H. |
| 100% | Submit Deliverables |  | Steve H. |
| 100% | Week 4 Deliverables | Mon 2/13/17 |  |
| 100% | Document Selected Use Cases |  | Alicia G. |
| 100% | Components Interaction Diagram |  | Brian C. |
| 100% | Document State Transitions |  | TBD |
| 100% | Week 5 Deliverables | Mon 2/20/17 |  |
| 100% | Transform Mock-Ups into Wireframes |  | Alicia G. |
| 100% | Complete Coding |  | Brian C. |
| 100% | Week 6 Deliverables | Mon 2/27/17 |  |
| 100% | Develop Test Cases in Standard Format |  | Eunjou K. |
| 100% | Reduce Data Driven Combinations using Allpairs |  | Steve H. |
| 100% | Submit Draft Report |  | Steve H. |
| 100% | Present Basic Functionality of Term Project |  | All |
| 100% | Final Report | Sat 3/4/17 |  |
| 100% | Provide Final Feedback/Comments | Thu 3/2/17 | All |
| 100% | Deliver and submit Final Report | Sat 3/4/17 | Steve H. |

Please note that all the above documents were stored for approval in Google Drive. Upon approval, finalized versions of the documents are moved to the ShipIT/Controlled Docs folder in GIT. In addition, Pivotal Tracker was used for requirements and wireframes were designed in Cacoo.

## Estimation Record

| **Phase** | **Deliverable** | **Size Measure** | **Effort Per Size Measure (person/hrs)** | **Size** | **Effort (person/hrs)** | **Fibonacci** | **Actuals** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Planning** | Definition of Scope | #Attributes | 0.50 | 4 | 2.00 | 3 | 2 |
| **Requirements** | Definition of Users (personas) | #Roles | 1.00 | 4 | 4.00 | 5 | 6 |
| Definition of Requirements | #Requirements | 0.25 | 23 | 5.75 | 8 | 5 |
| Entering Stories into Pivotal | #Stories | 0.08 | 23 | 1.84 | 2 | 1 |
| **Configuration Management** | Configuration Items List | #CI items | 0.33 | 12 | 3.96 | 5 | 5 |
| **Estimation** | Estimation Record | #Activities | 0.25 | 16 | 4.00 | 5 | 6 |
| **Design** | State Transition Diagram | #States | 0.75 | 4 | 3.00 | 5 | 3 |
| Definition of Use Cases | #Use cases | 2.00 | 6 | 12.00 | 13 | 6 |
| Components Interaction Diagram | #Components | 3.00 | 1 | 3.00 | 5 | 2 |
| Creation of Wireframes | #Wireframes | 0.50 | 4 | 2.00 | 3 | 5 |
| **Implementation** *Building ShipIT* | Creating Screens | #Screens | 2.00 | 4 | 8.00 | 13 | 6 |
| Creating Classes | #Models | 0.25 | 4 | 1.00 | 2 | 30 |
| Establish Relations\* | #Connections | 24.00 | 1 | 24.00 | 34 | 3 |
| Database\* | #Tables | 1.00 | 4 | 4.00 | 5 | 10 |
| **Testing** | Test Cases | #Test cases | 0.50 | 19 | 9.50 | 13 | 4 |
| Defects from Testing Recorded & Execution | #Defects | 0.08 | 3 | 0.25 | 1 | 2 |
| **Team Report& Presentation** | Report with all deliverables | #Documents | 4.00 | 7 | 28.00 | 34 | 12 |
| Final team's presentation of app's functionality | #Slides | 1.00 | 20 | 20.00 | 21 | 12 |
|  |  |  | ***Total*** |  | ***136.30*** | ***177*** | 120 |

## Assumptions

The application relies on single sign-on (SSO) for user authentication, so users do not have to explicitly log into the ShipIT desktop application.

A list of user accounts is also populated within the application from the organization’s directory services database (Active Directory, etc.) via lightweight directory access protocol (LDAP).

SSO and LDAP user account population is simulated via a list of personas found in one or more JavaScript Object Notation (JSON) files.

# Client Information

## Personas

### Persona #1: SHIPPER & RECEIVER

**Dan Cummings**

[](http://www.google.com/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&ved=0ahUKEwjDxIHshdnRAhWGNSYKHQUlBrMQjRwIBw&url=http://www.sevenlakes.com/customer.php&psig=AFQjCNHCsNUv4xZYFKSqFLDbmhSGRjNrbw&ust=1485287621800173)

Sr. Financial Analyst [Digital image], 2017

**Male, 40 years old**

**Background:** Dan is a Sr. Financial Analyst that works in the Finance Department of BigCorp. He is responsible for supporting the comptroller on month end financial reporting activities and assisting with tasks related to auditing other departments in the organization.

**Scenarios:** The comptroller asks Dan to complete an audit scorecard for the month end reporting. This tasks requires to send a formal physical letter to each head of the departments and have it returned signed. Dan recalls that recently a new application has been developed at BigCorp that allows him to prepare and request a shipment. Using this application Dan can request a shipment of the mail/parcel, and track the item.

**Goals:**

To create/cancel/edit a shipment request

To track shipment

**Ideal Features:**

Ability to receive email notification once the status of shipment has changed

Ability to see in one screen a list of shipments created and their statuses

Ability to print information for selected shipment(s).

### Persona #2: Receiver

**Teresa Portman**

[](http://www.google.com/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=0ahUKEwjgx47QjdnRAhXEKiYKHVt3BTwQjRwIBw&url=http://setxfoodbank.org/staff-and-board/&bvm=bv.144686652,d.eWE&psig=AFQjCNEnaSVJoXuWT7K1XXAi8DfSOSrLnQ&ust=1485289728809987)

Data Entry Clerk [Digital image], 2017

**Female, 50 years old**

**Background:** Teresa was hired temporarily by BigCorp to assist the Finance Department with data entry tasks. She has been assigned to a project that requires data entry into excel for further statistical analysis.

**Scenarios:** The controller explains to Teresa her functions on how she will be recording data into specific format in an Excel spreadsheet. The data does not currently exist in an electronic format and therefore Teresa will receive physical invoices from the Accounts Payable Department recurrently. Teresa is instructed that using the ShipIT Application she will be able to check the status of the mail to be sent from the Accounts Payable Department and print information for selected shipment(s).

**Goals:**

To track shipment

**Ideal Features:**

Ability to receive email notification once the status of shipment has changed

Ability to see in one screen a list of shipments created and their statuses

Ability to print information for selected shipment(s).

### Persona #3: Interoffice Courier

**John Armstrong**

[](http://www.google.com/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&ved=0ahUKEwj0mJ_ajtnRAhUKSSYKHQBhCB4QjRwIBw&url=http://www.expeditedcouriergroup.com/mail--interoffice---bank-deliveries.html&bvm=bv.144686652,d.eWE&psig=AFQjCNHzPveRmL8a17a1JrJAneiR1g_M3g&ust=1485290006495608)

Interoffice Courier [Digital image], 2017

**Male, 22 years old**

**Background:** John works for the supplier courier that has the current contract with BigCorp to provide the interoffice shipping. John is responsible for picking up the mail/parcel and delivering the items to the desired internal destination.

**Scenarios:** John daily picks-up and delivers internal mail/parcel within BigCorp. He reviews daily the ShipIT application to organize his route based on the location of the departments. Using the application John reviews a list of items that currently have a tracking number assigned and the appropriate status for pick up. Once John picks up the mail/parcel he can mark the item as “Picked-up” and in the same way once he completes the delivery he can mark it as “Delivered”.

**Goals:**

To obtain information to organize shipment routes

To track shipment

To confirm pick-up and delivery of shipments

**Ideal Features:**

Ability to receive email notification once the status of shipment is ready for pick-up

Ability to see in one screen a list of shipments created, their statuses and locations

Ability to confirm shipment has been picked-up

Ability to confirm shipment has been delivered

### Persona #4: Shipping Manager/ADMIN

**Dayne Bratsman**

[](http://www.google.com/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=0ahUKEwimuo3WldnRAhWKTCYKHcXQCksQjRwIBw&url=http://abcotechnology.edu/programs/network-administrator-systems-engineer/&bvm=bv.144686652,d.eWE&psig=AFQjCNHdZKHuW6Fv1LxLWbVtcZMwc7KZkw&ust=1485291843277955)

Admin [Digital image], 2017

**Male, 35 years old.**

**Background:** Dayne is the admin of ShipIT the new software application developed by BigCorp to provide the interoffice shipping. Dayne is also the admin for other tools in the company like SharePoint sites and he works directly with HR to obtain the data of employees.

**Scenarios:** Dayne receives communication that several employees have been recently hired by BigCorp and that these individuals need to be granted access to the ShipIT application. Using the new application Dayne can add the employees, give them a ShipIT id number, identify their department and define their roles in the system.

**Goals:**

To add/remove/edit employees and their roles

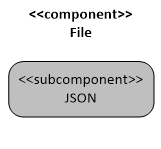
To add/remove/edit couriers

**Ideal Features:**

Ability to have integration with the HR database to obtain employee information

Ability to allow employees to sign-in to application using the company email and same password utilized for company network.

### Persona #5 – JSON Deserializer



**JSON Deserializer Service**

**Non-human**

**Background:** The ShipIT application uses the JavaScript Object Notation (JSON) Deserializer service to transform JSON data from permanent storage into instantiated objects of the Shipment class. Each shipment object has a tracking number, date created, status, delivery date, destination, name, department, and sender.

**Scenarios:** A user starts the ShipIT application. Upon application start, a method is called that searches for application data. If application data is found, the JSON Deserializer parses the application file and creates a Shipment object for each corresponding JSON object. Information is displayed in a new row for each Shipment object in the Track Shipments table on the main screen.

**Goals:**

* To retrieve shipment details to display in the Track Shipments table.

**Ideal Features:**

* Ability to convert JSON objects in persistent storage to Shipment objects in memory.

## Requirements

| **ID** | **Epic/Story** | **Description** | **Story Points\*[[1]](#footnote-1)** |
| --- | --- | --- | --- |
| 1 | **Epic** | **As a Shipper&Receiver, I can create a shipment of one or more packages for pickup using ShipIT, so that the interoffice courier is informed and completes my shipment.** |  |
| 1.1 | Story | As a Shipper&Receiver, I can create shipment event that includes Date, Destination Name, Destination Department, Originator Name, Originator Department and Notes, so that the interoffice courier has the information necessary to complete the shipment. |  |
| 1.2 | Story | As a Shipper&Receiver, I can edit/cancel a shipment event, so that corrections can be completed for the interoffice courier to have the most accurate information necessary to complete the shipment. |  |
| 2 | **Epic** | **As a Shipper&Receiver, I can track a shipment of one or more packages, so that I have information of the status of my shipment.** |  |
| 2.1 | Story | As a Shipper&Receiver, I can see a list of items in transit that includes tracking #, status, Date shipment was created, Date Delivered, Destination Name, Destination Department, Originator Name and Originator Department so that I have in one place all the information of the status of my shipment. |  |
| 2.2 | Story | As a Shipper&Receiver, I can select a shipment(s) from a list to print information for selected shipment(s), so that I can have a physical receipt for record keeping or in case a proof is required. |  |
| 3 | **Epic** | **As a Receiver, I can track a shipment of one or more packages, so that I have information of the status of my shipments.** |  |
| 3.1 | Story | As a Receiver, I can see a list of items in transit that includes tracking #, status, Date shipment was created, Date Delivered, Destination Name, Destination Department, Originator Name and Originator Department so that I have in one place all the information of the status of my shipment. |  |
| 3.2 | Story | As a Receiver, I can select a shipment(s) from a list to print information for selected shipment(s), so that I can have a physical receipt for record keeping or in case a proof is required. |  |
| 4 | Epic | **As an Interoffice Courier, I can track the information of one or more shipments, so that I have the latest information to complete the shipments.** |  |
| 4.1 | Story | As an Interoffice Courier, I can see a list of items in transit that includes tracking #, status, Date shipment was created, Date Delivered, Destination Name, Destination Department, Originator Name and Originator Department, so that I have in one place all the information necessary to complete the shipment. |  |
| 4.2 | Story | As an Interoffice Courier, I can mark a shipment as picked up, so that a status and tracking of the shipment can be updated. |  |
| 4.3 | Story | As an Interoffice Courier, I can mark a shipment as delivered, so that a status and tracking of the shipment can be updated. |  |
| **5** | **Epic** | **As an Admin, I can add/ remove employees to/from the ShipIT application, so that employees have the appropriate access to the application** |  |
| 5.1 | Story | As an Admin, I can add a BigCorp Employee to the ShipIT application populating his/her name, assigning an ID, populating corresponding Department and role, so that new authorized users have the appropriate access to the application |  |
| 5.2 | Story | As an Admin, I can remove a BigCorp Employee from the ShipIT by marking the employee as removed, so that users can be removed from the application if necessary. |  |
| **6** | **Epic** | **As an Admin, I can add/ remove interoffice couriers to/from the ShipIT application, so that couriers have the appropriate access to the application** |  |
| 6.1 | Story | As an Admin, I can add a courier to the ShipIT application populating the company name, courier ID, address, telephone and assigning it the role of courier, so that authorized couriers have the appropriate access to the application |  |
| 6.2 | Story | As an Admin, I can remove a courier from the ShipIT by marking the courier as removed, so that couriers can be removed from the application if necessary. |  |
| 7.0 | Story | As a JSON Deserializer, I need the ability to load shipment details from a JSON file so that the ShipIT application can display it in the Track Shipments table. |  |

The Epics and Stories for the ShipIT project are also located in Pivotal Tracker. For access to the ShipIT pivotal project please refer to the following link: <https://www.pivotaltracker.com/n/projects/1957935>

## RASCI Chart

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Activity | Responsible | Accountable | Supporting | Consulted | Informed |
| Create a shipment of one or more packages for pickup. | Shipper | Shipping Manager | Courier |  | Receiver |
| Track a shipment of one or more packages. | ShipIT | Shipping Manager | Courier |  | Shipper  Receiver |
| Track a shipment of one or more packages. | ShipIT | Shipping Manager |  |  | Shipper  Receiver  Courier |
| Track the information of one or more shipments. | ShipIT | Shipping Manager | Courier |  | Shipper  Receiver |
| Add/ remove employees. | Shipping Manager | ShipIT |  |  | Shipper  Receiver  Courier |
| Add/ remove interoffice couriers. | Shipping Manager | ShipIT |  |  | Shipper  Receiver  Courier |

## Selected Use Cases

### Create a Shipment in ShipIT

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case | | | |
| Use Case Name | Create a Shipment in ShipIT | | |
| Use Case ID | 1 | | |
| Description | This use case describes the event of a Shipper&Receiver creating a Shipment by entering the destination and the sender information. | | |
| Actor | Shipper&Receiver | | |
| Preconditions | Shipper&Receiver is Logged into the system | | |
| Cross Reference | Story 1.1 | | |
| Typical Course of Events | **Actor Actions** | | **System Response (Optional)** |
|  | | 1. System displays the Track Shipments Screen |
| 2. Shipper&Receiver Clicks the *Create* New Shipment Link | | 3. System displays the Create Shipment Screen |
| 4. Shipper&Receiver enters Destination Name | | 5. System displays entry on Destination Name field |
| 6. Shipper&Receiver enters Destination Department | | 7. System displays entry on Destination Department field |
| 8. Shipper&Receiver enters Sender Name | | 9. System displays entry on Sender Name field |
| 10. Shipper&Receiver enters Sender Department | | 11.System displays entry on Sender Department field |
| 12. Shipper&Receiver enters Notes | | 13. System displays entry on Notes field |
| 14. Shipper&Receiver clicks the Submit button | | 15. System closes Create Shipment Screen and displays the Track Shipments Screen |
| Alternate Courses | | | |
| Alt-Step 14 | Shipper&Receiver clicks the Cancel button | System displays the Track Shipments Screen | |
| Wrap Up | | | |
| Assumptions | Shipper&Receiver has access to the ShipIT application. | | |
| Post Conditions | The shipment has been created. | | |
| Remarks |  | | |

### Edit a Shipment in ShipIT

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case | | | |
| Use Case Name | Edit a Shipment in ShipIT | | |
| Use Case ID | 2 | | |
| Description | This use case describes the event of a Shipper&Receiver editing a shipment by updating the destination and or the sender information. | | |
| Actor | Shipper&Receiver | | |
| Preconditions | Shipper&Receiver is Logged into the system | | |
| Cross Reference | Story 1.2 | | |
| Typical Course of Events | **Actor Actions** | | **System Response (Optional)** |
|  | | 1. System displays the Track Shipments Screen |
| 2. Shipper&Receiver selects a row or shipment | | 3. System highlights selection |
| 4. Shipper&Receiver clicks Edit Selected button | | 5. System displays Edit Shipment Screen, displaying existing information |
| 6. Shipper&Receiver updates Destination Name | | 7. Systems displays updated entry in Destination Name field |
| 8. Shipper&Receiver clicks the Save | | 9. System closes Edit Shipment Screen and displays the Track Shipments Screen |
| Alternate Courses | | | |
| Alt-Step 6 | Shipper&Receiver updates Destination Department | Systems displays updated entry in Destination Department field | |
| Alt-Step 6 | Shipper&Receiver updates Sender Name | Systems displays updated entry in Sender Name field | |
| Alt-Step 6 | Shipper&Receiver updates Sender Department | Systems displays updated entry in Sender Department field | |
| Alt-Step 6 | Shipper&Receiver updates Notes | Systems displays updated entry in Notes field | |
| Alt-Step 6 | Shipper&Receiver clicks the Go Back Button | System displays the Track Shipment Screen | |
| Wrap Up | | | |
| Assumptions | Shipper&Receiver has access to the ShipIT application. | | |
| Post Conditions | The shipment has been edited. | | |
| Remarks |  | | |

### Track a Shipment in ShipIT

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case | | | |
| Use Case Name | Track a Shipment in ShipIT | | |
| Use Case ID | 3 | | |
| Description | This use case describes the event of a Shipper&Receiver, Receiver or Interoffice Courier tracking a shipment by viewing shipment information. | | |
| Actor(s) | Shipper&Receiver, Receiver or Interoffice Courier | | |
| Preconditions | Shipper&Receiver, Receiver or Interoffice Courier is Logged into the system | | |
| Cross Reference | Stories 2.1 and 3.1 | | |
| Typical Course of Events | **Actor Actions** | | **System Response (Optional)** |
|  | | 1. System displays the Track Shipments Screen |
| 2. Shipper&Receiver, Receiver or Interoffice Courier views a list of shipments and their tracking information | |  |
| Alternate Courses | | | |
| Alt-Step |  |  | |
| Wrap Up | | | |
| Assumptions | Shipper&Receiver, Receiver or Courier have access to the ShipIT application. | | |
| Post Conditions | The shipment has been tracked. | | |
| Remarks |  | | |

### Print a Shipment in ShipIT

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case | | | |
| Use Case Name | Print a Shipment in ShipIT | | |
| Use Case ID | 4 | | |
| Description | This use case describes the event of a Shipper&Receiver or Receiver selecting and item from the tracking shipment list and requesting the details of the shipment to be printed. | | |
| Actor(s) | Shipper&Receiver or Receiver | | |
| Preconditions | Shipper&Receiver or Receiver is Logged into the system | | |
| Cross Reference | Stories 2.2, 3.2, and 4.1 | | |
| Typical Course of Events | **Actor Actions** | | **System Response (Optional)** |
|  | | 1. System displays the Track Shipments Screen |
| 2. Shipper&Receiver or Receiver selects a row or shipment | | 3. System highlights selection |
| 4. Shipper&Receiver or Receiver selects Print Selected button | | 5.System displays the Track Shipments Screen |
| Alternate Courses | | | |
| Alt-Step |  |  | |
| Wrap Up | | | |
| Assumptions | Shipper&Receiver or Receiver have access to the ShipIT application. | | |
| Post Conditions | The shipment confirmation for shipment selected has been printed. | | |
| Remarks |  | | |

### Mark a Shipment as Picked Up in ShipIT

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case | | | |
| Use Case Name | Mark a Shipment as Picked Up in ShipIT | | |
| Use Case ID | 5 | | |
| Description | This use case describes the event of an Interoffice Courier marking a shipment as picked up and updating its status. | | |
| Actor | Interoffice Courier | | |
| Preconditions | Interoffice Courier is Logged into the system | | |
| Cross Reference | Story 4.2 | | |
| Typical Course of Events | **Actor Actions** | | **System Response (Optional)** |
|  | | 1. System displays the Track Shipments Screen |
| 2. Interoffice Courier selects a row or shipment | | 3. System highlights selection |
| 4. Interoffice Courier selects Update Status button | | 5. System displays the Update Shipment Status screen |
|  | 6. Interoffice Courier selects Picked Up from the Update Shipment Status drop down list | |  |
|  | 7. Interoffice Courier clicks the Save button. | | 8. System displays the Track Shipments Screen and displays the Status as Picked Up for the updated shipment. |
| Alternate Courses | | | |
| Alt-Step | 1. In the Update Shipment Status screen, the Interoffice Courier clicks the Go Back button. | 2. System displays the Track Shipments Screen with no changes made to the shipment status. | |
|  |  |  | |
| Wrap Up | | | |
| Assumptions | Interoffice Courier has access to the ShipIT application. | | |
| Post Conditions | The shipment status has been updated to Picked Up. | | |
| Remarks |  | | |

### Mark a Shipment as Delivered in ShipIT

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case | | | |
| Use Case Name | Mark a Shipment as Delivered in ShipIT | | |
| Use Case ID | 6 | | |
| Description | This use case describes the event of an Interoffice Courier marking a shipment as Delivered and updating its status. | | |
| Actor | Interoffice Courier | | |
| Preconditions | Interoffice Courier is Logged into the system | | |
| Cross Reference | Story 4.3 | | |
| Typical Course of Events | **Actor Actions** | | **System Response (Optional)** |
|  | | 1. System displays the Track Shipments Screen |
| 2. Interoffice Courier selects a row or shipment | | 3. System highlights selection |
| 4. Interoffice Courier selects Update Status button | | 5. System displays the Update Shipment Status screen |
|  | 6. Interoffice Courier selects Delivered from the Update Shipment Status drop down list | |  |
|  | 7. Interoffice Courier clicks the Save button. | | 8. System displays the Track Shipments Screen and displays the Status as Delivered for the updated shipment. |
| Alternate Courses | | | |
| Alt-Step | 1. In the Update Shipment Status screen, the Interoffice Courier clicks the Go Back button. | 2. System displays the Track Shipments Screen with no changes made to the shipment status. | |
|  |  |  | |
| Wrap Up | | | |
| Assumptions | Interoffice Courier has access to the ShipIT application. | | |
| Post Conditions | The shipment status has been updated to Delivered. | | |
| Remarks |  | | |

### Add Users to ShipIT

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case | | | |
| Use Case Name | Add Users to ShipIT | | |
| Use Case ID | 7 | | |
| Description | This use case describes the event of user management by admin. | | |
| Actor | Shipping Manager | | |
| Preconditions | Shipping Manager is Logged into the system | | |
| Cross Reference | Stories 5.1 and 6.1 | | |
| Typical Course of Events | **Actor Actions** | | **System Response (Optional)** |
|  | | 1. System displays user management screen |
| 2. Admin views a list of employees and clicks add employee or add user. | | 3. System displays input fields for user information |
| 4. Admin inputs user information and clicks save button | | 5. System saves the data and loads refreshed screen |
| Alternate Courses | | | |
| Alt-Step |  |  | |
| Wrap Up | | | |
| Assumptions | Shipping manger has access to the ShipIT application. | | |
| Post Conditions | New employee information has been added to the user list. | | |
| Remarks | In the above use case, the term user may refer to either an employee or to an interoffice courier. | | |

### Remove Users from ShipIT

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case | | | |
| Use Case Name | Remove Employees to ShipIT | | |
| Use Case ID | 8 | | |
| Description | This use case describes the event of user management by admin. | | |
| Actor | Shipping Manager | | |
| Preconditions | Shipping Manager is Logged into the system | | |
| Cross Reference | Stories 5.2 and 6.2 | | |
| Typical Course of Events | **Actor Actions** | | **System Response (Optional)** |
|  | | 1. System displays user management screen |
| 2. Admin views a list of employees and select a record to remove | | 3. System highlights selection |
| 4. Admin clicks remove button | | 5. System removes the data and loads refreshed screen |
| Alternate Courses | | | |
| Alt-Step |  |  | |
| Wrap Up | | | |
| Assumptions | Shipping manger has access to the ShipIT application. | | |
| Post Conditions | Selected employee information has been removed from the user list. | | |
| Remarks |  | | |

# Product

## Development Environment

The ShipIT application will be a desktop application created with Windows Presentation Foundation in Visual Studio 2015. A Model-View-ViewModel (MVVM) architecture will be attempted to build the application. The C# programming language will be utilized for building classes and constructing business logic while Extensible Application Markup Language (XAML) will be leveraged to develop the user interface, or views. GitHub is currently in use for source control and collaboration.

For access to the ShipIT Git repository please refer to the following link: <https://github.com/calhooligan/ShipIT>

## Components Interaction diagram

ShipIT will be a Windows desktop application using Windows Presentation Foundation technology. A Model-View-View Model architecture is utilized to separate the user interface (UI) from the logic and data. The view model instance collects shipment objects created from the JSON data source. The main view (i.e. UI window) is presented with the collection of shipment objects via the view model object instance. The shipment collection can be modified through three other views- create new shipment, edit shipment, and update status.



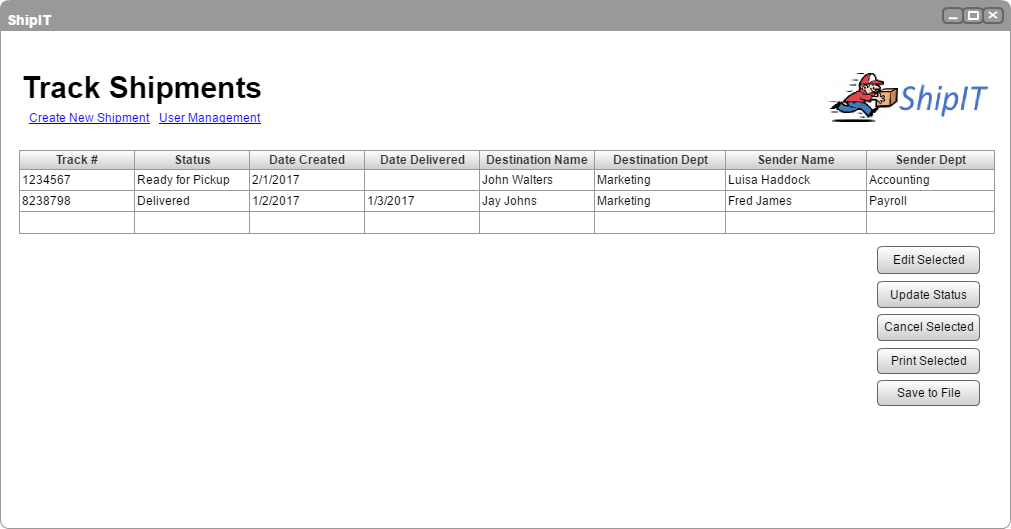
## State Transitions

The following diagram documents the state transitions that occur within ShipIT. The diagram consists of four large boxes that represent the ShipIT forms. Smaller boxes, representing individual states, are contained within the larger form boxes. Arrows are used to represent activities that result in state transitions. Arrows leading from one state to another represent transitions that can only occur between those two states. An arrow leading directly from a form box indicates that the transition may occur no matter which state the form is in. Finally, diamonds are used to represent conditionals and circles with numbers are used to jump to different parts of the diagram, eliminating the need for crossed lines.

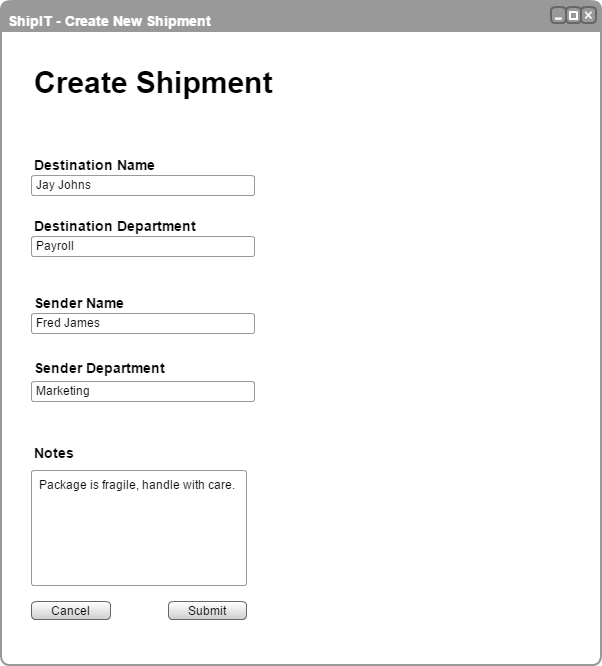


## Wireframes

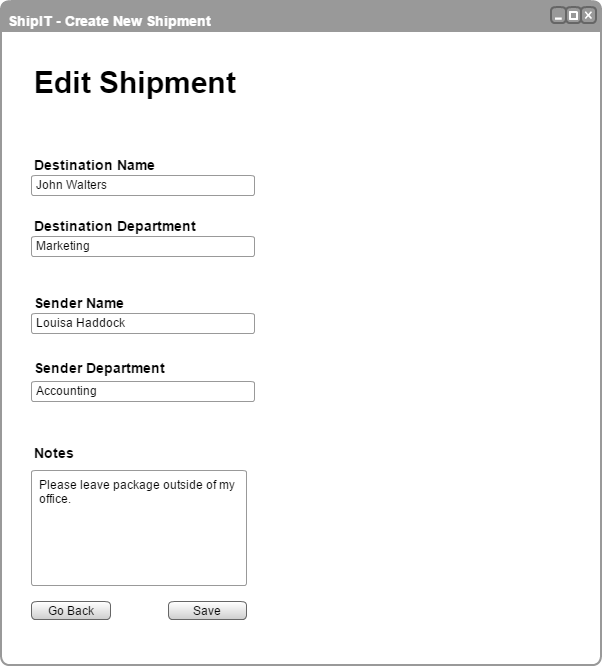
### Main Form/Track Shipments



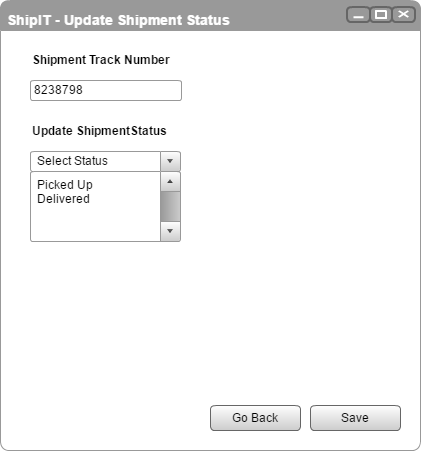
### Create Shipment



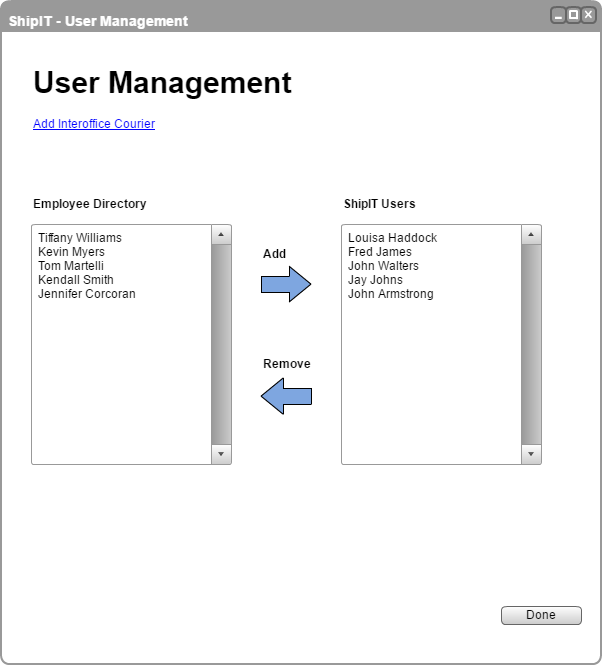
### Edit Shipment



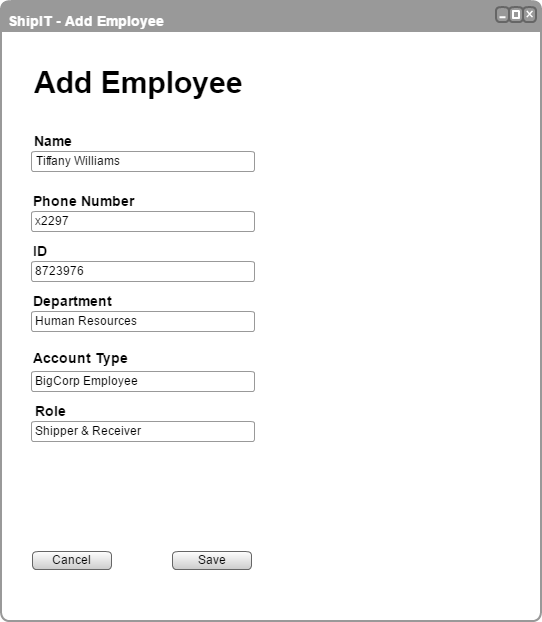
### Update Status



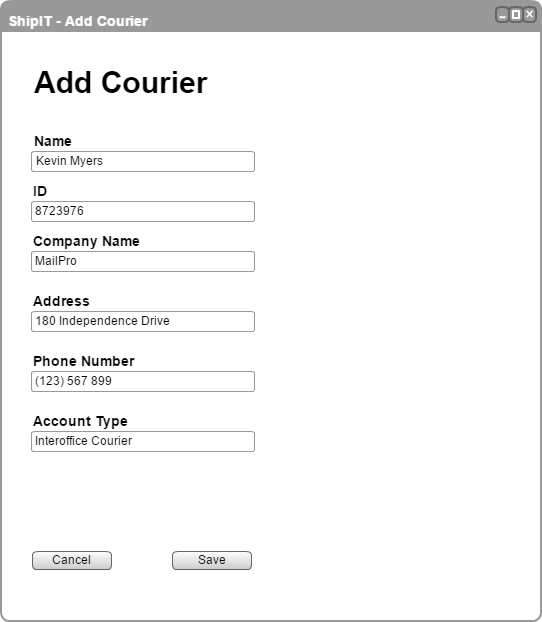
### User Management



### Add Employee



### Add Courier



## Test Cases

The following is a summary of the execution of the following 6 test cases.

3 of the 6 test cases could not be run because functionality had not been completed.

Of the 3 test cases that were run, two of them passed and one failed.

Three separate deviations were recorded during testing. See the following section for more information.

### Test Case #1: A User Can Create a Shipment

|  |  |  |  |
| --- | --- | --- | --- |
| **General Information** | | | |
| Test Type | ⮽ Functionality □ Positive □ Negative □ Performance  □ Regression □ Overload | | |
| Test Number | 1 | Test Date | 2/24/2017 |
| Test Case  Description | Verify that a shipper&receiver user can create a shipment of one or more packages for pickup.  Relevant data is mapped and the user has access to the created shipment data. | | |
| Results | ⮽ Pass □ Fail | | |
| **Introduction** | | | |
| Requirement(s) to be tested | All relevant data is successfully captured by the shipment system and correctly transferred between the different screens of the application.  Related to requirement number 1. | | |
| Set Up and Constrains | To complete this test, the user must be setup and have access to the shipment system. The proper permissions must be in place so that they can create a shipment. | | |
| **Test** | | | |
| Input | The user will need to enter entries for the following fields while completing the test on the create shipment screen of the application.  **User Input :**   * Destination Name – Enter receiver’s name * Destination Department – Enter receiver’s department * Sender Name – Enter sender’s name * Sender Department – Enter sender’s department * Notes – Enter description   **System Input :**   * Status – Default value (Ready for Pickup) * Track Number – Sequence number * Submitted Date – Data creation date | | |
| Procedural Steps | 1. Open the shipment application. 2. Navigate to the create shipment screen. 3. Enter in values for the above mentioned columns, outlined in the input section of the test. 4. Save the change with submit button and close the screen. 5. Open up the track shipments screen. 6. Verify all the data is entered and mapped to the proper locations from the create shipment screen to the track shipments screen. 7. Repeat step 1-6 to verify that multiple shipments data is successfully transferred. | | |
| Expected Results | The test is a success if the values entered in the create shipment screen per the input and procedural steps listed above successfully map to the proper track shipments screen columns. The values entered should map as follows:   * Track Number → Track # * Status → Status * Submitted Date → Date Created * Destination Name → Destination Name * Destination Department → Destination Dept * Sender Name → Sender Name * Sender Department → Sender Dept | | |
| **Actual Results** | | | |
| Actual Results | Relevant outputs from the procedural steps:  Step 1: The shipment application opens as expected.  Step 2: Application opens to the create shipment screen.  Step 3: All column headers exist and values are entered.    Step 4: User is able to navigate to the track shipments screen.  Step 5: The corresponding data appears as entered on the track shipments screen from the create shipment screen.    The test passed. All steps were able to be completed as expected with the outcomes that were listed in the test case.  Screenshots for step 7: | | |
| Comments | Shipment data of one or more packages is created and stored in the JSON file. The most recently created item is populated in the track shipments list. | | |

### Test Case #2: Edit/Cancel a Shipment

|  |  |  |  |
| --- | --- | --- | --- |
| **General Information** | | | |
| Test Type | ⮽ Functionality □ Positive □ Negative □ Performance  □ Regression □ Overload | | |
| Test Number | 2 | Test Date | 2/24/2017 |
| Test Case  Description | Verify that a shipper&receiver user can edit and cancel a shipment of one or more packages.  Relevant data is updated and the user has access to the recent shipment data. | | |
| Results | ⮽ Pass □ Fail | | |
| **Introduction** | | | |
| Requirement(s) to be tested | All relevant data is successfully captured by the shipment system and correctly transferred between the different screens of the application.  Related to requirement number 1. | | |
| Set Up and Constrains | To complete this test, the shipment data must be created. The user defined as a sihpper&receiver must view all shipment data. | | |
| **Test** | | | |
| Input | The user will need to edit entries for the following fields while completing the test on the edit shipment screen of the application.  **User Input :**   * Destination Name – Enter receiver’s name * Destination Department – Enter receiver’s department * Sender Name – Enter sender’s name * Sender Department – Enter sender’s department * Notes – Enter description | | |
| Procedural Steps | 1. Open the shipment application. 2. Navigate to the track shipments screen. 3. Select a row and click edit selected button. 4. Edit values for the above mentioned columns. 5. Save the change with submit button and close the screen. 6. Open up the track shipments screen. 7. Verify all the data is entered and mapped to the proper locations from the edit shipment screen to the track shipments screen. 8. Repeat step 1-3 and click cancel selected. 9. Verify that the selected items are canceled on the track shipments screen. | | |
| Expected Results | The test is a success 1) if the values entered in the edit shipment screen per the input and procedural steps listed above successfully map to the proper track shipments screen columns. The values entered should map as follows:   * Destination Name → Destination Name * Destination Department → Destination Dept * Sender Name → Sender Name * Sender Department → Sender Dept   2) if selected items were canceled on the track shipments screen with cancel selected button. | | |
| **Actual Results** | | | |
| Actual Results | Relevant outputs from the procedural steps:  Step 1: The shipment application opens as expected.  Step 2: Application opens to the track shipments screen.  Step 3: All column headers exist and values are entered.  Step 4: User is able to select a row to edit.  Step 5: The edit shipment screen opens with edit selected button.    Step 6: User is able to change values in the field.    Step 7: User is able to select multiple rows on the track shipments screen.  Step 8: Cancel selected button removes selected item and track shipments screen is refreshed.  (Note: Shipment 3001 was canceled)    The test passed. All steps were able to be completed as expected with the outcomes that were listed in the test case. | | |
| Comments | Shipment data of one or more packages is updated and stored in the JSON file. | | |

### Test Case #3: Shipper&Reciever or Receiver Can View Shipment Information

|  |  |  |  |
| --- | --- | --- | --- |
| **General Information** | | | |
| Test Type | ⮽ Functionality □ Positive □ Negative □ Performance  □ Regression □ Overload | | |
| Test Number | 3 | Test Date | \*\*\*NOT RUN\*\*\*See Comments\*\*\* |
| Test Case  Description | Verify that a shipper&receiver or receiver user can see the latest information of shipments.  Selected items are printable as physical receipts. | | |
| Results | □ Pass □ Fail | | |
| **Introduction** | | | |
| Requirement(s) to be tested | All relevant data is successfully captured by the shipment system and correctly printed.  Related to requirement number 2 and 3. | | |
| Set Up and Constrains | To complete this test, the shipment data must be created. The user defined as a sihpper&receiver or receiver must view all shipment data. | | |
| **Test** | | | |
| Input |  | | |
| Procedural Steps | 1. Open the shipment application. 2. Navigate to the track shipments screen. 3. Select one or more rows and click print selected button. 4. Verify all the printed data is matching with the displayed data on the tracking shipments list. | | |
| Expected Results | The test is a success if the values printed are map to the proper track shipments screen columns. | | |
| **Actual Results** | | | |
| Actual Results | Relevant outputs from the procedural steps:  Step 1: The shipment application opens as expected.  Step 2: Application opens to the track shipments screen.  Step 3: All column headers exist and values are entered.  (Screenshot to be added)  Step 4: User is able to click one or more rows.  Step 5: The corresponding data is printed when the print selected button is clicked. | | |
| Comments | This test was not able to be run because printing functionality has not been enabled yet. | | |

### Test Case #4: Courier Can View & Update Status

|  |  |  |  |
| --- | --- | --- | --- |
| **General Information** | | | |
| Test Type | ⮽ Functionality □ Positive □ Negative □ Performance  □ Regression □ Overload | | |
| Test Number | 4 | Test Date | 2/24/2017 |
| Test Case  Description | Verify that a courier user has access to the list of shipments and update status screen.  User completes the shipment by updating its status. | | |
| Results | □ Pass ⮽ Fail | | |
| **Introduction** | | | |
| Requirement(s) to be tested | Updated status is successfully captured by the shipment system and correctly transferred between the different screens of the application.  Related to requirement number 4. | | |
| Set Up and Constrains | To complete this test, the shipment data must be created. The user defined as a interoffice courier must view all shipment data. | | |
| **Test** | | | |
| Input | The user will need to edit status field while completing the test on the update status screen of the application.  **User Input :**   * Status – Selected per dropdown menu (Picked Up / Delivered) | | |
| Procedural Steps | 1. Open the shipment application. 2. Navigate to the track shipments screen. 3. Select one row and click update status button. 4. Edit status to ‘picked up’. 5. Save the change with save button and close the screen. 6. Open up the track shipments screen. 7. Verify that the selected item’s status is changed. 8. Repeat step 1-7 for ‘delivered’ status. | | |
| Expected Results | The test is a success if the status is successfully updated to the status column on the track shipments screen.   * Status→ Status | | |
| **Actual Results** | | | |
| Actual Results | Relevant outputs from the procedural steps:  Step 1: The shipment application opens as expected.  Step 2: Application opens to the track shipments screen.  Step 3: All column headers exist and values are entered.  Step 4: User is able to select a row to edit status.  Step 5: The updated shipment screen opens with update status button. NOTE: This window was not updated with the current status.    Step 6: Clicking the Save button has no effect. | | |
| Comments | This feature is restricted to the interoffice courier user group. | | |

### Test Case #5: Admin can Add/Remove Employee Information

|  |  |  |  |
| --- | --- | --- | --- |
| **General Information** | | | |
| Test Type | ⮽ Functionality □ Positive □ Negative □ Performance  □ Regression □ Overload | | |
| Test Number | 5 | Test Date | \*\*\*NOT RUN\*\*\*See Comments\*\*\* |
| Test Case  Description | Verify that an admin user can add or remove employees’ information and the data is saved.  Newly created users should have access to ShipIT. | | |
| Results | □ Pass □ Fail | | |
| **Introduction** | | | |
| Requirement(s) to be tested | All relevant data is successfully saved or removed by the shipment system and newly created users grant login access to the application.  Related to requirement number 5. | | |
| Set Up and Constrains | To complete this test the admin user must be setup. | | |
| **Test** | | | |
| Input | The user will need to enter entries for the following fields while completing the test on the add employee of the application.  **User Input :**   * Name – Enter employee’s name * Phone Number – Enter employee’s telephone number * ID – Enter employee’s ID * Department – Enter employee’s department * Role – Enter employee’s role | | |
| Procedural Steps | 1. Open the shipment application. 2. Navigate to the user management screen. 3. Select user creation button. 4. Open the add employee screen and enter new employee’s data. 5. Save the data with save button and close the screen. 6. Verify that the data has been entered on the employee list. 7. Login with the newly created user account. 8. Verify that the user has access to the screens and authority to create a shipment. 9. Login with admin account. 10. Select a user to remove. 11. Remove the employee’s data with remove button. 12. Verify that the user data has been removed on the ShipIT user list. | | |
| Expected Results | The test is a success 1) if the values entered in the user creation screen per the input and procedural steps listed above successfully map to the proper user list screen columns. The values entered should map as follows:   * Name → Name * Phone Number → Phone Number * ID → ID * Department → Department * Role → Role * 2) if selected employee was deleted on the ShipIT user list with remove button. | | |
| **Actual Results** | | | |
| Actual Results | Relevant outputs from the procedural steps:  Step 1: The shipment application opens as expected.  Step 2: Application opens to the user management screen.  Step 3: Application opens to the add employee screen.  Step 4: All column headers exist and values are entered.  Step 5: User is able to see the row with new employee information.  (Screenshot to be added)  Step 6: Created user has access to the shipment system.  (Screenshot to be added)  Step 7: The updated user management screen opens.  Step 8: Admin user is able to select a user and remove.  Step 9: Selected row is deleted on the ShipIT user list.  The test passed. All steps were able to be completed as expected with the outcomes that were listed in the test case. | | |
| Comments | This test was not able to be run because this functionality has not been enabled yet. | | |

### Test Case #6: Admin can Add/Remove Interoffice Courier Information

|  |  |  |  |
| --- | --- | --- | --- |
| **General Information** | | | |
| Test Type | ⮽ Functionality □ Positive □ Negative □ Performance □ Regression □ Overload | | |
| Test Number | 6 | Test Date | \*\*\*NOT RUN\*\*\*See Comments\*\*\* |
| Test Case  Description | Verify that an admin user can add or remove interoffice courier information.  Newly created interoffice account users should have access to ShipIT. | | |
| Results | □ Pass □ Fail | | |
| **Introduction** | | | |
| Requirement(s) to be tested | All relevant data is successfully saved or removed by the shipment system and newly created users grant login access to the application.  Related to requirement number 6. | | |
| Set Up and Constrains | To complete this test the admin user must be setup. | | |
| **Test** | | | |
| Input | The user will need to enter entries for the following fields while completing the test on the courier management screen of the application.  **User Input :**   * Name – Enter contact person of courier * ID – Enter courier’s ID * Company Name – Enter courier company’s name * Address – Enter courier’s address * Phone Number – Enter courier’s telephone number   **System Input :**   * Account Type – Default value (Interoffice Courier) | | |
| Procedural Steps | 1. Open the shipment application. 2. Navigate to the user management screen. 3. Select add interoffice courier button. 4. Enter new courier’s data. 5. Save the data with save button and close the screen. 6. Verify that the data has been entered on the courier list. 7. Login with the newly created courier user account. 8. Verify that the user has access to the screens and authority to view shipment items. 9. Login with admin account. 10. Select a courier to remove. 11. Remove the courier’s data with remove button. 12. Verify that the courier data has been removed on the list. | | |
| Expected Results | The test is a success 1) if the values entered in the courier creation screen per the input and procedural steps listed above successfully map to the proper courier list screen columns. The values entered should map as follows:   * Name → Name * ID → ID * Company Name → Company Name * Address → Address * Phone Number → Phone Number * Account Type → Account Type   2) if selected courier was deleted on the courier list with remove button. | | |
| **Actual Results** | | | |
| Actual Results | Relevant outputs from the procedural steps:  Step 1: The shipment application opens as expected.  Step 2: Application opens to the user management screen.  Step 3: Application opens to the add courier screen.  Step 4: All column headers exist and values are entered.  Step 5: User is able to see the row with new courier information.  (Screenshot to be added)  Step 6: Created courier user has access to the shipment system.  (Screenshot to be added)  Step 7: The updated courier management screen opens.  Step 8: Admin user is able to select a courier and remove.  Step 9: Selected row is deleted on the courier list.  The test passed. All steps were able to be completed as expected with the outcomes that were listed in the test case. | | |
| Comments | This test was not able to be run because this functionality has not been enabled yet. | | |

## Deviations

The following deviations were noted while testing the application.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Description | Found | Status | Notes |
| 1 | Notes Entry not stored. | Test case 2. | Resolved | This deviation didn’t fail the execution of the test case but was noticed when opening a shipment that had just been created. |
| 2 | Update Shipment Status field not set. | Test Case 4 | Pending | When opening the Update Status screen, the initital value of Shipment Status should match what is shown in the Track Shipments screen. |
| 3 | Unable to Update Shipment Status | Test Case 4 | Pending | Clicking the Save button in the Update Status window has no effect. |

## Data Driven Combinations/Allpairs Results

There aren’t very many different combinations of data/actions contained within ShipIT. While the following list may not be a very real world example, due to the number of invalid combinations, the information demonstrates the overall process.

For this analysis, the following combinations were used.

| Role | Activity | Entries |
| --- | --- | --- |
| Shipper/Receiver | Track Shipments | Valid |
| Courier | Create a Shipment | Invalid |
| Receiver | Edit a Shipment |  |
| Admin | Update Status |  |
|  | Cancel a Shipment |  |
|  | Print a Shipment |  |
|  | Save Data |  |
|  | Employee Management |  |
|  | Courier Management |  |

A full matrix of all combinations would generate 4 x 9 x 2 = 72 different pairings. Running this information through AllPairs, reduced the number of combinations by 50% to 36.

| Role | Activity | Entries |
| --- | --- | --- |
| Shipper/Receiver | Track Shipments | Valid |
| Courier | Track Shipments | Invalid |
| Shipper/Receiver | Create a Shipment | Invalid |
| Courier | Create a Shipment | Valid |
| Receiver | Edit a Shipment | Valid |
| Admin | Edit a Shipment | Invalid |
| Receiver | Update Status | Invalid |
| Admin | Update Status | Valid |
| Shipper/Receiver | Cancel a Shipment | Valid |
| Courier | Cancel a Shipment | Invalid |
| Shipper/Receiver | Print a Shipment | Invalid |
| Courier | Print a Shipment | Valid |
| Receiver | Save Data | Valid |
| Admin | Save Data | Invalid |
| Receiver | Employee Management | Invalid |
| Admin | Employee Management | Valid |
| Shipper/Receiver | Courier Management | Valid |
| Courier | Courier Management | Invalid |
| Receiver | Track Shipments | ~Valid |
| Admin | Track Shipments | ~Invalid |
| Receiver | Create a Shipment | ~Invalid |
| Admin | Create a Shipment | ~Valid |
| Shipper/Receiver | Edit a Shipment | ~Invalid |
| Courier | Edit a Shipment | ~Valid |
| Shipper/Receiver | Update Status | ~Valid |
| Courier | Update Status | ~Invalid |
| Receiver | Cancel a Shipment | ~Valid |
| Admin | Cancel a Shipment | ~Invalid |
| Receiver | Print a Shipment | ~Invalid |
| Admin | Print a Shipment | ~Valid |
| Shipper/Receiver | Save Data | ~Invalid |
| Courier | Save Data | ~Valid |
| Shipper/Receiver | Employee Management | ~Valid |
| Courier | Employee Management | ~Invalid |
| Receiver | Courier Management | ~Valid |
| Admin | Courier Management | ~Invalid |

Note, the values in the Entries column above that are preceded with at tilde (~) are entries that do not matter because all of the pairings needed to satisfy the test criteria have already been done. Additionally, invalid combinations (either due to access constraints or invalid operations) are highlighted in orange. By removing these invalid combinations, the test set is further reduced to 23 combinations, a 32% reduction from the full matrix.

| Role | Activity | Entries |
| --- | --- | --- |
| Shipper/Receiver | Track Shipments | Valid |
| Shipper/Receiver | Create a Shipment | Invalid |
| Admin | Edit a Shipment | Invalid |
| Receiver | Update Status | Invalid |
| Admin | Update Status | Valid |
| Shipper/Receiver | Cancel a Shipment | Valid |
| Courier | Cancel a Shipment | Invalid |
| Shipper/Receiver | Print a Shipment | Invalid |
| Courier | Print a Shipment | Valid |
| Receiver | Save Data | Valid |
| Admin | Save Data | Invalid |
| Admin | Employee Management | Valid |
| Receiver | Track Shipments | ~Valid |
| Admin | Create a Shipment | ~Valid |
| Shipper/Receiver | Edit a Shipment | ~Invalid |
| Courier | Edit a Shipment | ~Valid |
| Courier | Update Status | ~Invalid |
| Admin | Cancel a Shipment | ~Invalid |
| Receiver | Print a Shipment | ~Invalid |
| Admin | Print a Shipment | ~Valid |
| Shipper/Receiver | Save Data | ~Invalid |
| Courier | Save Data | ~Valid |
| Admin | Courier Management | ~Invalid |

# References

Sr. Financial Analyst [Digital image]. (2017, January). Retrieved from <http://www.google.com/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&ved=0ahUKEwjDxIHshdnRAhWGNSYKHQUlBrMQjRwIBw&url=http%3A%2F%2Fwww.sevenlakes.com%2Fcustomer.php&psig=AFQjCNHCsNUv4xZYFKSqFLDbmhSGRjNrbw&ust=1485287621800173>

Data Entry Clerk [Digital image]. (2017, January). Retrieved from <http://www.google.com/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=0ahUKEwjgx47QjdnRAhXEKiYKHVt3BTwQjRwIBw&url=http%3A%2F%2Fsetxfoodbank.org%2Fstaff-and-board%2F&bvm=bv.144686652,d.eWE&psig=AFQjCNEnaSVJoXuWT7K1XXAi8DfSOSrLnQ&ust=1485289728809987>

Interoffice courier [Digital image]. (2017, January). Retrieved from <http://www.google.com/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&ved=0ahUKEwj0mJ_ajtnRAhUKSSYKHQBhCB4QjRwIBw&url=http%3A%2F%2Fwww.expeditedcouriergroup.com%2Fmail--interoffice---bank-deliveries.html&bvm=bv.144686652,d.eWE&psig=AFQjCNHzPveRmL8a17a1JrJAneiR1g_M3g&ust=1485290006495608>

Admin [Digital image]. (2017, January). Retrieved from <http://www.google.com/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=0ahUKEwimuo3WldnRAhWKTCYKHcXQCksQjRwIBw&url=http%3A%2F%2Fabcotechnology.edu%2Fprograms%2Fnetwork-administrator-systems-engineer%2F&bvm=bv.144686652,d.eWE&psig=AFQjCNHdZKHuW6Fv1LxLWbVtcZMwc7KZkw&ust=1485291843277955>

1. The Story Point column in this table has been included only in preparation for future deliverables. It shall be populated when estimations are completed. [↑](#footnote-ref-1)