

Gusty's Bike Shop

Test Plan Documentation

Section 2

Chelsea Irizarry, Ian Carlyle,
R. Jamal Washington
& Quinn Ogden

Table of Contents

Title Page	1
Introduction	3
Purpose	3
Statement of scope	3
Overview of the remainder of the document	3
System Overview	3
Project Description	3
Requirements List	4
Test Plan	6
Testing strategy	6
Testing resources and staffing	6
Test work products	8
Test record keeping	9
Test schedule	9
Test Procedure	10
Appendix	13
Glossary	13
Contributions	14
Additional Information	14

1. Introduction

1.1. Purpose

The purpose of this document is to provide clarification on details of the test plan. The test plan will also present the test team with an overview of Gusty's e-commerce website, so that they have an understanding of the material they are testing. The document highlights features that must be well-tested by the implementation and testing team. The document also contains instructions for both teams on which features they responsible for, testing procedures, and recording results.

1.2. Statement of scope

This project involves constructing an e-commerce website that specializes in the selling of bicycles, cycling-related accessories, and apparel. All customer, employee, and product data will be stored in a database. A customer will be able to log in, browse, and checkout products. An employee will solely be able to update their timesheets. Employee administrators will be able to perform tasks specific to their position, such as editing product information or viewing employee hours.

1.3. Overview of the remainder of the document

Section 2 of the document will explain the project's expected behavior and functionality. This section will present an up-to-date list of the project requirements as user stories. Section 3 of the document will discuss the project's testing plan, describing in detail how to set up and run the locally-hosted application. This section will also explain how to carry out the testing procedures. Section 4 of the document will provide the test cases to be used throughout testing.

2. System Overview

2.1. Project Description

The system is intended to be suitable for multiple simultaneous users. Users will interact with the storefront using a graphical user interface (GUI), which will be compatible with different devices. The website will span multiple pages. Different user types will have different functional capabilities. The system will communicate with an online database to store user and product information. The system will not involve any actual billing of customers, nor will it involve the shipping of products. The e-commerce website will not be fully functional and will simulate different functions.

2.2. Requirements List

2.2.1. Guest

1. As a guest, I want to create an account so that I can log in as a customer.
2. As a guest, I want to be able to browse the store's product pages so that I can gauge my interest in the store.
3. As a guest, I want to be able to read reviews that customers have left on products so that I can decide if I want to buy them.

2.2.2. Customer

1. As a customer, I want to log into my account so that I can order products.
2. As a customer, I want to add items to a cart so that I can purchase those items.
3. As a customer, I want to store my billing information so that I can purchase products more quickly and easily.
4. As a customer, I want to leave reviews on products so that other customers and guests can make informed decisions.
5. As a customer, I want to delete my account so that I can remove my personal information from the site's database.
6. As a customer, I want to change my account's information (password, email, billing information, etc.) so that I can ensure that my information is up-to-date and accurate.

2.2.3. Basic Employee

1. As an employee I want to log in so that I can document my hours worked.
2. As an employee, I want to view my timesheet so that I can ensure that my billing hours are accurate.
3. As an employee, I want to update my timesheet so that I can update my hours worked.

2.2.4. Sales Admin

1. As a sales administrator, I want to add products to a page so that I can display new products for sale.
2. As a sales administrator, I want to remove products from the storefront so that I can ensure that discontinued products are no longer being sold to customers.
3. As a sales administrator, I want to edit product details so that I keep product descriptions and prices up-to-date.

2.2.5. Management Admin

1. As a management administrator, I want to view work hours of other employee accounts so that I can ensure that they are accurate.
2. As a management administrator, I want to edit the work hours of other employee accounts so that I can correct any inaccuracies in employee billing.
3. As a management administrator, I want to create new employee accounts so that staff can perform their duties.

2.2.6. Master Admin

1. As a master administrator, I want to create any type of account so that I can fill any position that I currently need.
2. As a master administrator, I want to be able to access the information of any account so that I can maintain control over the use of the site.
3. As a master administrator, I want to be able to delete any type of account so that I can remove accounts that are no longer needed or are detrimental to the site.
4. As a master administrator, I want to create and edit blog posts so that I can distribute information about the site or other topics of interest.

2.2.7. Non-Functional

1. The system is expected to be able to run on a standard consumer-model personal computer without significant slowdown.
2. The system is expected to be able to run on the browser of a standard consumer-model mobile phone without significant slowdown.
3. Every page will have full functionality and accessibility from a mobile device.
4. The system is expected to be able to handle input from multiple simultaneous users without significant slowdown.
5. The system can only be accessed if the user's device has a working internet connection.
6. The system is not intended to be used with real credit cards and will not bill a customer.
7. The system will not make use of HTTPS encryption.
8. The system will use a simple hashing system for password encryption.
9. Usernames and password sizes will be limited to counter SQL injection.
10. The software will be hosted online, with no offline option.

11. The ability for employees in the system to provide customer service may be limited due to the small size of the client's company.

3. Test Plan

3.1. Testing strategy

To ensure that Gusty's Bike Shop e-commerce website has proper functionality, tests will be conducted on the login and signup pages. The login function must be tested for both a customer and an employee user type. The login function will be tested for multiple simultaneous users, and incorrect or invalid form input, such as password entry. The login feature will also be tested for proper information retrieval. The signup function will be tested for valid user input into the database.

3.2. Testing resources and staffing

3.2.1. Cloud9 Setup

Cloud9 (<https://c9.io/>) is required to locally host the website. The testing team must create or log into a Cloud9 account, and create a new workspace with a blank template. Once a workspace is created, the team will then clone the git repository in the Cloud9 terminal with the following command:

```
> git clone GitHub.com/qogden/Gustys-bike-shop
```

3.2.2. Database Setup

To set up the database, the team will run the following command prior to starting the server. These will install the necessary libraries to properly run the web application.

```
> sudo apt-get update
> sudo apt-get install python-psycopg2
> sudo apt-get install postgresql-contrib-9.3
> sudo easy_install flask markdown
```

The server can be started with the following commands:

```
> sudo service postgresql start
> sudo sudo -u postgres psql
```

Finally, the SQL file is loaded into PostgreSQL with the following command:

```
> \i Gustys-bike-shop/bikes.sql
```

3.2.3. Running the Application

The application can be started by opening the **server.py** file, and clicking the run button on the top of the page. Alternatively, it can be run from the command line using the following command:

```
> python server.py
```

A new tab will open with a URL link to where the website is running locally.

3.2.4. Testing the Signup Page

Two members of the testing team will navigate to the signup page and test for invalid email entries, such as those with abnormal characters. The pair will also test for null entries on required fields and report if an error was presented for either situation. They will then create a customer by entering the provided test case data into the required fields. Once a customer is created, they will attempt to create another customer using the same email in the signup page. The pair will note if an error message was provided. They will then proceed to check the database for a duplicate users in the customer and user tables. The pair will note if a duplicate customer was created in the database.

3.2.5. Testing the Login Page

Two members of the testing team will navigate to the login page and attempt to log in with an email and password that is not in the database. The pair will note if an error message was provided. They will then attempt to log in with a valid customer email, but with an incorrect password, and note if the system provided an error message. The testing pair will then log in as a valid customer and employee. The customer login should be considered successful if the customer's name appears at the top of the site. The employee login is considered successful if it redirects to a new blank page and displays the user's name on the top of the site.

3.2.6. Testing Simultaneous Users

Two members of the implementation team will log in as a customer on a new tab. The pair will then open two more tabs using the URL link and log in as a customer and employee. If successful, the name of the customer will appear at the top of the screen. The pair will then refresh each of the tabs to test simultaneous usability. The implementation pair must note if each tab remains logged in as different users, or if all of them are logged into a single user account.

3.2.7. Testing the Database

Two members of the implementation team will be responsible for testing the product, cart, timesheet, and review tables. To test the product table, the couple will create and remove an item from the table.

To test the cart table, two items must be added to the table for a customer and another item must be added for a different customer. To locate the customer ID, use the provided email to search the customers table. After adding the items into the cart table, the pair must then display the products in the cart table for each customer. The pair will then edit a single item from a cart table and display the content of the cart table for that customer. Afterwards, they will remove a single item from the cart table and then display the content of the cart for that particular customer.

To test the timesheet table, two different members of the implementation team will find the employee's ID from the employee table, using the provided email. They will input the test cases for the two employees, and display all entries for each employee entered. Finally, to test the review table, the pair will input two reviews for one product, and a single review for a different product. The pair will use the provided customer email to determine the customer ID, and display all entries for each product reviewed. For each database test, the implementation team will document the SQL statement used and the output entries for each product.

3.3. Test work products

The testing team will be documented each phase in in a table format indicating the test case, tester, and result description. When a bug is encountered the results must describe the issue in detail. While indicating the identity of the tester is not critically important, it is beneficial information to have if the error cannot be reproduced. Upon completing testing they will combine their data and submit a single document to the implementation team. Each test case will be tested at least by two testers. There will be approximately 20 responses from the testing team.

Test Case	Tester	Results
1	sample-tester1	Signup successfully created account.
2	sample-tester2	System created account

		will invalid email.
--	--	---------------------

The implementation team will produce a different table document for the database and simultaneous users tests. Along with the test case, tester identity, and results description, SQL commands will be recorded. SQL commands will be recorded to ensure that errors do not occur when entering database commands. When a bug is encountered a detailed description will be provided. The team will produce approximately 16 test result entries.

Test Case	Tester	SQL Command	Results
11	sample-tester3		The system did not support multiple simultaneous users.
12	sample-tester4	INSERT into product (name, description, price, stock) VALUES ('Bicycle', 'Test bike.', 300, 200);	The entry was successfully entered into the product database.

3.4. Test record keeping

Both teams will use the following link to enter their testing results.

https://docs.google.com/spreadsheets/d/1fj57UDgLyml5srJAAow6s5_cGC1Ldwse0OWq-79ag2Q/edit?usp=sharing

The testing team will document their results on the sheet labeled testing team, while the implementation team will document their results on sheet labelled implementation team.

3.5. Test schedule

The testing team will be assigned to tasks 1-11, in order to test the functionality of the signup and login page. The implementation team will address every task, though during the testing period, particular emphasis will be placed on tasks 12-18 to ensure a reliable database. During the testing period, each task will be tested by at least two members. This test plan document will be reviewed by the testing team on Monday, April 6th. Testing will take approximately an hour and be conducted on Wednesday,

April 5th and Friday, April 7th. The implementation team will do periodic integration testing and validation throughout the building process.

4. Test Procedure

Our test procedure heavily involves unit testing, which tests each component on our website. Both the implementation and testing team will record their observation and bugs in the excel document discussed in section 3.5.

4.1. Unit Tests:

Test Number	Related to Requirement	Subsystem	Purpose	Test Case Data	Expected Results
1	2.2.1.1	Signup	Test account creation	Username: John@email Password: pass	Account created, and added to database.
2	2.2.1.1	Signup	Test email for specific characters	Username: Bill'maze@email.com Password: 1234	Account should be created and added to database.
3	2.2.1.1	Signup	Test null email input	Username: '' Password: 123	Account should not be created.
4	2.2.1.1	Signup	Test null password input	Username: gran@email.com Password: ,	Account should not be created.
5	2.2.1.1	Signup	Check for duplicate email	Username: John@email.com Password: pass	Account should not be created.
6	2.2.2.1	Login	Bad email, bad password	Username: false@email.com Password: egg	Should stay on login page and display message that login failed.
7	2.2.2.1	Login	Good email, bad password	Username: John@email.com Password: egg	Should stay on login page and display message that login failed.

8	2.2.2.1	Login	Bad email, good password	Username: flase@email.com Password pass	Should stay on login page and display message that login failed.
9	2.2.2.1	Login	Good email, good password	Username: John@email.com Password: pass	Redirect to homepage. Displayed account name in corner.
10	2.2.3.1	Login	Employee login	Username: Gusty@gustybikeshop.com Password: 1234	Redirect to new blank page.
11	2.2.3.1	Login	Simultaneous users	Username: employee@gustybikeshop.com Password: Simon Username: user1@email.com Password: Simon1 Username: user2@email.com Password: simon2	The employee session will redirect to new blank page and have their account name displayed at the top of the site. The employee names remain the same when each page is refreshed. The two customer sessions will redirect to a homepage and have their account name displayed at the top of the site. The customer names remain the same when each page is refreshed.
12	2.2.4.1	Database	Creating a product	Name: Bicycle Description: Test bike. Price: 300 Stock: 200	The item will be added to the product table.
13	2.2.4.2	Database	Removing a product	Product Id: 3	The item will be deleted from the product table.
14	2.2.2.2	Database	Adding items to cart	Customer email: user1@email.com Product id: 1 Quantity:2	Three items will be added to the cart table in the database.

				<p>Customer email: user1@email.com Product id: 2 Quantity:1</p> <p>Customer email: user2@email.com Product id: 4 Quantity: 2</p>	
15	2.2.2.2	Database	Edit quantity of product in cart	<p>Customer email: user2@email.com Product id: 4 New Quantity: 3</p>	The customer's item will have an updated quantity in the cart table.
16	2.2.2.2	Database	Removing an item from the cart	<p>Customer email: user1@email.com Product id: 2</p>	The item should be deleted from the cart table for that user.
17	2.2.3.1	Database	Employee clock in	<p>Employee email: employee@gustybikeshop.com Date: 3/30/17 Hours: 5</p> <p>Employee email: employee@gustybikeshop.com Date: 3/31/17 Hours: 8</p> <p>Employee email: employee1@gustybikeshop.com Date: 3/28/17 Hours: 10</p>	The three items will be entered into the timesheet table.
18	2.2.2.4	Database	Adding a review	<p>Customer email: user1@email.com Product Id: 1 Date: 3/1/17 Rating: 5 Comment: "I love this"</p> <p>Customer email: user1@email.com Product Id: 2 Date: 3/5/17 Rating: 1</p>	<p>An entry is created for a specified product id in the review table.</p> <p>The entry contains a productid and the userid of the customer who added it to cart.</p>

				<p>Comment: "I hate this"</p> <p>Customer email: user2@email.com</p> <p>Product Id: 1</p> <p>Date: 3/20/17</p> <p>Rating: 3</p> <p>Comment: "Eh"</p>	
--	--	--	--	--	--

5. Appendix

5.1. Glossary

5.1.1. Blog Page

The client will use the blog page to display blog posts. The page will be visible to all users.

5.1.2. Checkout Page/Cart Page

This page will displays items that are in the customer's cart.

5.1.3. Homepage

The main page of the website. Any user of the site will be directed to the homepage when they enter the website's URL, login into the system, or click the home button on the navigation bar.

5.1.4. Login/Sign in Page

The page used by a guest user to log into the system.

5.1.5. Products Page

The page that displays products available, image, and name. The user will be redirected to the item page when a product's picture or name is clicked.

5.1.6. Item Page

This page will display the item's name, picture, description, quantity left in stock, and will allow the customer to add products to their cart.

5.1.7. Navigation bar/Navbar

The menu bar on top of the webpage that contains buttons which link to other webpages.

5.1.8. Hashing

Passing a word, usually passwords, through a function that changes the password to random nonsensical characters.

5.1.9. HTTP

Hypertext Transfer Protocol

5.1.10. HTTPS

Hypertext Transfer Protocol Secure

5.1.11. PostgreSQL

Open source relational database management system.

5.1.12. SQL(Structured Query Language)

Special programming language designed to manage data in a database.

5.1.13. SQL Injection

Malicious code entered into an entry field that grants unauthorized access to database information.

5.1.14. Denial of service attack

An attack that sends many requests in rapid succession to a website to slow or stop the website.

5.1.15. Gantt Chart

This is a project management tool that shows tasks to complete in the format of a timeline. It shows dependencies between tasks and can show which tasks to work on to stay on a realistic course.

5.1.16. Task Dependency Diagram

This is a project management tool that specializes in showing what order tasks must be done in.

5.2. Contributions

Ian: Sections 1, 2.2, 4.1 Unit tests 1-18, 5

Chelsea: Section 3.1-3.5, 4.1 Unit tests 11-18

Quinn: Sections - 2.1, 2.2, Unit test clarification, Editor

5.3. Additional Information

Testing Record Document:

https://docs.google.com/spreadsheets/d/1fj57UDgLyml5srJAAow6s5_cGC1Ldws_e0OWq-79ag2Q/edit?usp=sharing