

Gusty's Bike Shop

Documentation 2: Project Plan

Section 2

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

Table of Contents

Title Page	1
Table of Contents	2
Introduction	3
Purpose	3
Scope	3
References	3
Overview of the remainder of the document	3
Project Description	4
Project Scope	4
User Characteristics	4
Functional Requirements	5
Constraints	7
Project Schedule	8
Approach	8
Milestones and Deliverables	8
Work Breakdown Structure	12
Gantt chart	15
Task Dependency Diagram	16
Appendix	17
Glossary	17
Contributions	18
References	18

1. Introduction

1.1. Purpose

The purpose of this document is to propose an implementation plan for the Gusty's e-commerce bike shop. This document will lay out a realistic schedule that both the development team and the client can view. Designing the website will demystify the tasks the team needs to accomplish in order to construct the site. Laying out a schedule will help the team determine how much work can be accomplished over the semester. Development time is very limited, with a deadline of May 1st, 2017, so it is important to focus the team's efforts on the highest priority tasks. The intended audience of this document is the development team, the client, and the project manager.

1.2. Scope

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

1.3. References

This document follows the IEEE standard format.

1.4. Overview of the remainder of the document

Section 2 of this document will outline specifics of the project. This includes the system's intended use, and the characteristics of each type of user. This section will also provide a list of functional requirements as user stories and system constraints.

Section 3 will outline the project schedule and justifications, explaining how the development team approached each task. This section will also present the project's milestones and deliverables, as well as the project's "Work Breakdown Structure." For each milestone listed, the Work Breakdown Structure will contain a time estimate and an explanation as to

why it was picked. A Gantt chart and Task Dependency Diagram will visually display the project schedule.

Section 4 of the document is the Appendix, which contains the glossary, team member contributions, and references for the document.

2. Project Description

2.1. Project Scope

- 2.1.1. The system is intended to be used by multiple users simultaneously. 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 be able to operate different features. 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 merely be a model to simulate different functions.

2.2. User Characteristics

- 2.2.1. **Guest:** A guest describes a user of the system that is simply browsing the website. The guest is able to view the store's catalog, read reviews of products, and perform other passive, observatory functions related to the website. While guests are unable to make purchases or leave product reviews, they are able to register an account and log into that account. A logged in guest will be considered a customer.
- 2.2.2. **Customer:** A customer describes a user of the system that is currently logged into their account with intentions of purchasing a product. A customer is able to perform any of the actions available to a guest. In addition, a customer may add items to a cart, and purchase the items in that cart at checkout. Customers can also leave product reviews for other users to read. They may also manage personal information related to their account, such as their billing information and associated email address.

- 2.2.3. **Employee:** An employee describes a user of the system that is logged into an account designated for documenting hours worked. Employee accounts are created by the master administrator or a management administrator. Employees will have access to a timesheet to record their current number of worked hours on a biweekly basis.
- 2.2.4. **Sales Administrator:** Sales administrators are a type of employee focused primarily on updating the storefront. In addition to normal employee functions, sales administrators are able to add or remove products from the store. They can also change existing product information.
- 2.2.5. **Management Administrator:** Management administrators are a type of employee focused on managing employee accounts. A management administrator is able to create any type of employee account except for a master administrator account. They are also able to view other employees' work hours, edit those hours if necessary, and are responsible for processing the billing of all employees.
- 2.2.6. **Master Administrator:** The master administrator is a singular user who has control over the system's functions. He is able to create, edit, and delete all types of employee accounts. He is also able to view and edit data related to other accounts, such as an employee's hours worked. Finally, he is able to create blog posts on the website detailing information about the store, or simply topics he has an interest in.

2.3. Functional Requirements

2.3.1. Guest User Stories

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.

2.3.2. **Customer User Stories**

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.

2.3.3. **Employee User Stories**

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.

2.3.4. **Sales Administrator User Stories**

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.

2.3.5. **Management Administrator User Stories**

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.

2.3.6. **Master Administrator User Stories**

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.

2.4. **Constraints**

2.4.1. **Performance**

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. The system is expected to be able to handle input from multiple simultaneous users without significant slowdown.

2.4.2. **Hardware**

1. The system can only be accessed if the user's device has a working internet connection.

2.4.3. **Security**

1. The system is not intended to be used with real credit cards and will not bill a customer.
2. The system will not make use of HTTPS encryption.
3. The system will use a simple hashing system for password encryption.

4. Usernames and password sizes will be limited to counter SQL injection.

2.4.4. Software

1. The software will be hosted online, with no offline option.

2.4.5. Employee

1. 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. Project Schedule

3.1. Approach

The project interface will be constructed using HTML and CSS. The functionality of the site will be constructed using Flask and Python. The database language will be PostgreSQL. The client would like the site to be hosted locally, though he would also like the code located on GitHub for him to use on a later date. The team has decided to use Cloud9 as a development environment, Git for version control, and Taiga for the team storyboard.

3.2. Milestones and Deliverables

The following is a task list for each sprint of deliverables. Each task has a description of the subtasks involved.

- **Sprint 1: Setup**
 - Template: The template task will focus on the setup of the website template. A template will be selected, downloaded, and Flask-ified.
 - Navigation Bar: The navigation bar task will focus on the setup of the different pages on the main menu of the website. These include the home, products, sales, blog, login, signup, and cart pages.

- Database: The database task will focus on the initial setup of the database and tables within the database. Included in this section are the users, customers, employees, products, products types, and the cart tables. The users table will include the usernames and passwords of all types of users. The customers table will include the user's name, email, phone, billing/shipping information, and credit card information. The employees table will include the user's name, email, phone, permissions, employee type, and hours worked for the pay period. The product table will contain different products and their attributes such as the item's name, description, picture, product type, product attributes, and quantity in stock. The product types table will contain the different products types such as apparel, replacement parts, bikes, and tools. The cart table will include user's ID and the product ID. Test users or products must be added into each table.
- **Sprint 2: Basic Functionality**
 - Login: The login function will be a pop-up window that requests a user's username and password. A user will be able to enter their username and password into a form, and then hit the login button at the bottom of the window. The system will then verify that the username exists in the users table. If the username exists, the system will then verify that the password associated with the username matches the inputted password. If the password matches, the user successfully logged in and will be directed to the appropriate page.
 - Signup: The signup page will request a user's name, email, phone, billing/shipping information, and credit card information via a form. When the user hits the signup button at the bottom of the page, the system will verify that the email and username is not in the database. If both the username and email are not in use, the user information will be inputted into the database and the account will be created.

- Checkout: The checkout task will involve adding an item to the cart. While viewing the cart, the customer must be able to adjust the quantity of a product or remove an item entirely. When the customer clicks the checkout button at the bottom of the page, the system must check that the user is logged in. Once the customer is verified by the system, the customer may then begin finalizing the checkout process. The customer will be presented with an order summary. Once the customer hits the confirm checkout button, the order will be finalized. A mock email with the order confirmation will be sent to the customer.
- Timesheet: In the timesheet task, an employee user type must be logged into the system. The employee will be able to enter their total hours for the day and submit them on a biweekly basis. The save button will store the employee hourly data and display it onto the screen. It will then update the total hours worked during the pay period for each employee.
- **Sprint 3: Displays**
 - Homepage: The homepage display will show the navigation bar, and each link must redirect to the appropriate pages.
 - Item Page: The item page must display the product's name, picture, description, and quantity left in stock. The customer must be able to select a quantity to add to the cart. When the add to cart button is clicked, the item will be displayed in the cart page.
 - Product Page: The product page will display all available products in the products table. Subpages will be apparel, bicycles, replacement parts, and tools. Each subpage will display the appropriate products based on their product category. The apparel subpage may allow for additional sorting categories such as shirts, shorts, and shoes.

- Blog Page: The blog page will be powered by WordPress. More research on how to implement the connection for WordPress must be conducted.
- **Sprint 4: Stretch Functions**: In the event that the above sprints are completed ahead of schedule, the following tasks will be attempted.
 - Product Variations: Each item page will have different variation options. For example, apparel will have a size option. Different products will also have the option to choose from a variety of colors. The different product variations must be stored in the product table. The cart table must accommodate these variations.
 - Database Interactions: A new page must be created to allow for the master administrator to interact with a user-friendly interface rather than directly interacting with the database. The master administrator should be able to create, edit, and delete posts and accounts. The master administrator will also be able to view employee hours, and update product information from this screen. The sales administrator will be able to update product information. The management administrator will be able to create employee accounts and view employee hours.
 - Reviews: A review form must be created. A new table will be created to hold all review information such as customer ID, review ID, product ID, comments, and ratings. A customer must be logged into their account to submit a review for an item.
 - Threshold Alert: When a product's stock quantity drops below a specified number, an alert will be sent to the master administrator. A new column must be added to the products table to hold the threshold numbers.

3.3. Work Breakdown Structure

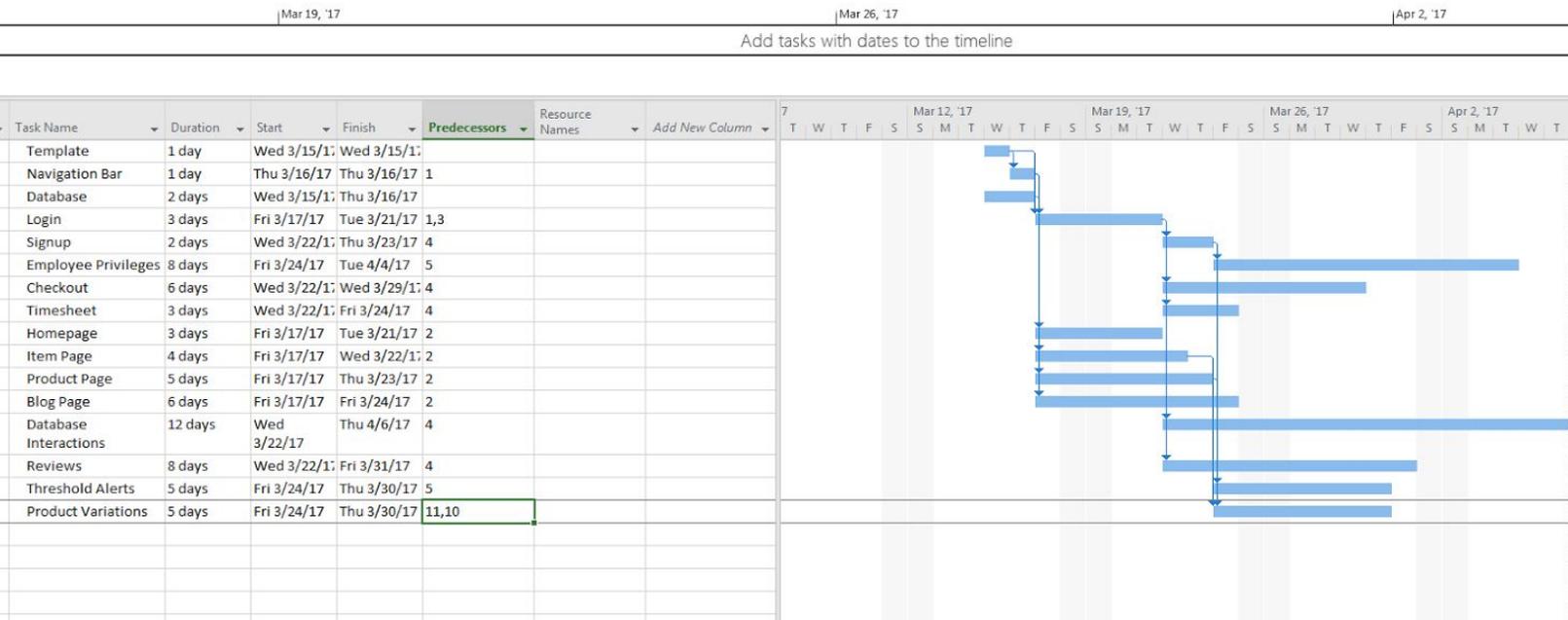
The following list of milestones and deliverables are the team's estimated time schedule and justifications for each time.

- **Sprint 1: Setup**
 - Template: 1 day
 - This is a simple task to find, download, and Flask-ify the template. This is not expected to take more than a few hours to ensure that the template is up and running. The team selected 1 day times to ensure that there was enough time to browse different templates and select one that works well with the project at hand.
 - Navigation Bar: 1 day
 - The navigation bar task is also fairly simple; the team will create new HTML pages and link each to the navigation bar. While this is expected to take only a few hours, padding was included to account for distractions and testing.
 - Database: 2 days
 - The database setup involves creating many tables with the proper columns. The team settled on 2 days to set up the database in order to ensure that the proper tables are created with the correct permission levels and the correct attributes.
- **Sprint 2: Basic Functionality**
 - Login: 3 days
 - The login task involves checking the database for an existing username and verifying the password. This feature must support multiple simultaneous users. The team determined that the login feature will take approximately 3 days to implement. This estimate includes padding for testing.
 - Signup: 2 days

- The signup task involves adding new database entries for new customer accounts. There is a guideline for account creations and the feature needs to be tested. This feature should take about 2 days to ensure this component is implemented correctly.
- Checkout: 6 days
 - The checkout task includes adding an item to a cart, as well as the finalization of the checkout process. The team's estimate for implementing this feature will be 6 days due to numerous potential issues that are likely to arise during implementation.
- Timesheet: 3 days
 - This feature will be implemented using a simple form, though it must be formatted in a way that it is cleared on a biweekly basis. The team determined that 3 days would be enough time to design and implement this feature.
- **Sprint 3: Displays**
 - Homepage: 3 days
 - The homepage is fairly simple, it includes an image slider and navigation bar. The majority of time spent on this page will be getting the image slider formatted correctly.
 - Item Page: 4 days
 - The item page will need to show any type of product's details. Since many products are different, it may take around 4 days to make this work for all products. This estimate is padded for testing time.
 - Product Page: 5 days
 - The product pages need to show different products based on the product type. This will be generated from the database. This feature is padded for formatting and testing.

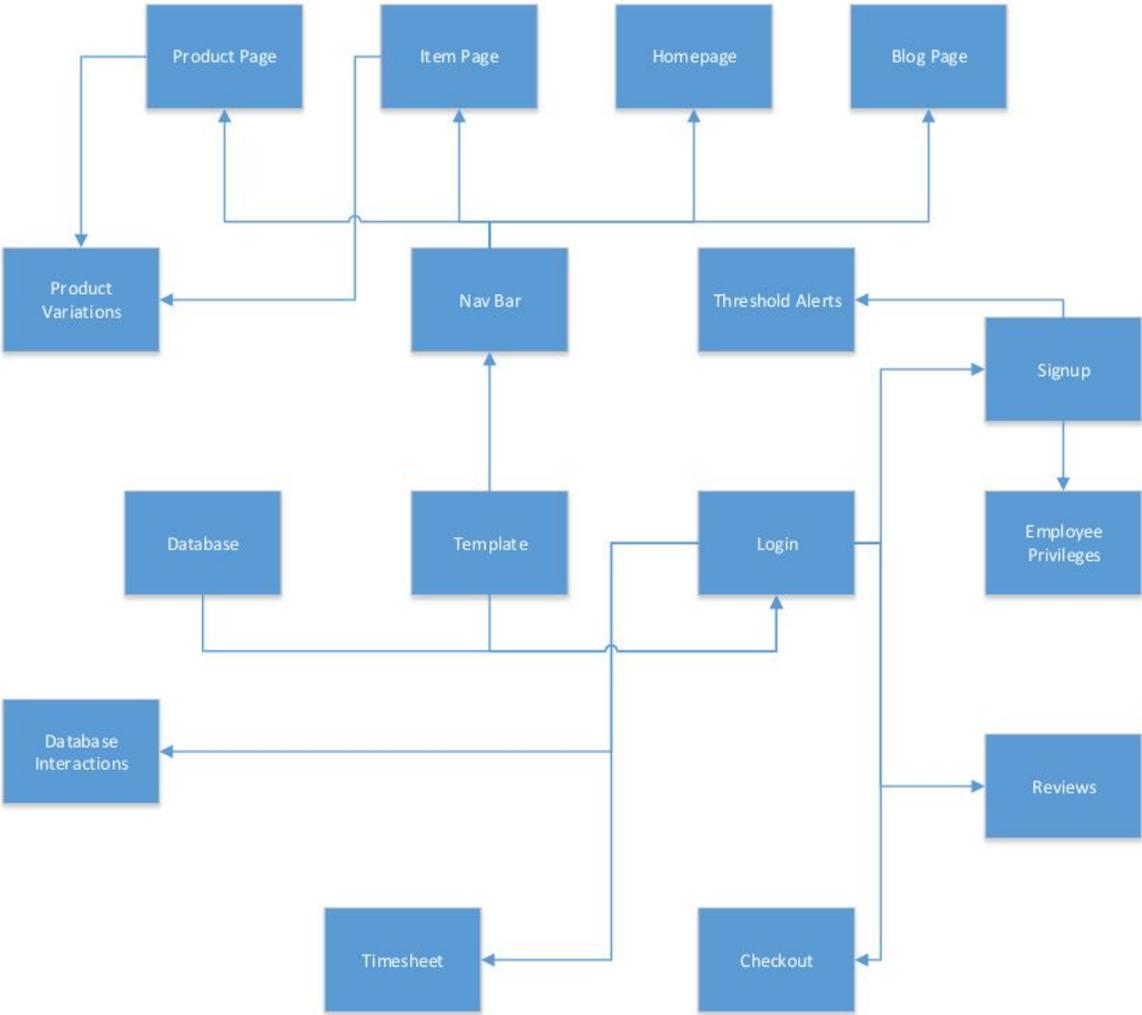
- Blog Page: 6 days
 - This page will display a limited number of blog posts posted by our client. As the team does not have much experience with WordPress, the team allowed for a 6 day period to continue research on how to implement this feature.
- **Sprint 4: Stretch Functions**
 - Product Variations: 5 days
 - This task requires additions to existing tables. This will allow additional item details to be present for the customer's benefit. This estimate is padded for testing on different products.
 - Database Interactions: 12 days
 - The team believes that this implementation will require a large amount of time due to the sizeable number of subtasks involved. The database needs to accommodate the different permissions of multiple different types of users.
 - Reviews: 8 days
 - The team determined that the review task may take approximately 8 days to implement due to the team's inexperience with creating such a function. The team must discuss this feature in depth prior to implementation.
 - Threshold Alert: 5 days
 - The threshold alert task has an estimate of 5 days due to the inexperience of the team with automated email generation.

3.4. Gantt chart



3.4.1. Above is the Gantt chart that was created to display the project schedule. This chart lists out all of the current tasks and dependencies for the project. The first task is scheduled to start on March 15th. The tasks were linked together depending on the estimated days to work on them.

3.5. Task Dependency Diagram



3.5.1. The task dependency diagram more clearly shows the dependency between different tasks otherwise obscured in the Gantt chart. Many tasks share the same dependencies, which allows the team to more easily determine which dependencies are mutual between tasks.

4. Appendix

4.1. Glossary

4.1.1. Blog Page

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

4.1.2. Checkout Page/Cart Page

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

4.1.3. 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.

4.1.4. Login/Sign in Page

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

4.1.5. 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.

4.1.6. 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.

4.1.7. Navigation bar/Navbar

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

4.1.8. Hashing

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

4.1.9. HTTP

Hypertext Transfer Protocol

4.1.10. HTTPS

Hypertext Transfer Protocol Secure

4.1.11. PostgreSQL

Open source relational database management system

4.1.12. SQL(Structured Query Language)

Special programming language designed to manage data in a database.

4.1.13. SQL Injection

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

4.1.14. Denial of service attack

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

4.1.15. 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.

4.1.16. Task Dependency Diagram

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

4.2. Contributions

- Ian Carlyle: Introduction, 2.3 user stories, Appendix, Project Approach Section 3.3, and Documentation Editing
- Chelsea Irizarry: Project Approach Sections 3.1, 3.2, 3.3, and Documentation Editing
- Quinn Ogden: Project Approach Sections 3.4, 3.5
- R. Jamal Washington: Project Description, Proofreading, and Language Consistency

4.3. References

- 4.3.1. <http://gustycoper.github.io/> This is our client's website. He showed this website to our team as an example of using Django.
- 4.3.2. Gusty's Bike Shop will be hosted locally. The source code can be found at <https://github.com/qogden/Gustys-bike-shop>.