|  |
| --- |
| **ShoeStars – Group 10**  <https://github.com/NathanParisot/Shoestars-Project-Group-10.git>  Shape Text divider |
|  |
| COM-430 SOFTWARE ENGINEERING PROJECT – Dr. Adams  05/03/2022  SAINT LEO UNIVERSITY  Shape Text divider  **Authored by: Nathan Parisot and Sahil Jain** |

Table of Contents

[Revision History 3](#RevisionHistory)

[Introductory Summary 4](#IntroductorySummary)

[1.1 Purpose 5](#Purpose)

[1.2. Charter 5](#Charter)

[Personas 1.3 5](#Personas)

[Initial Scenarios 1.4 6](#InitialScenarios)

[Initial Features 1.5 7](#InitialFeatures)

[User Stories 1.6 7](#UserStories)

[User Stories Tasks 1.7 8](#UserStoriesTasks)

[Configuration Management Rules 1.8 9](#ConfigurationManagementRules)

[Code Rules 1.9 9](#CodeRules)

[Testing Rules 2.0 9](#TestingRules)

[Function Requirements 2.1](#FunctionalReq)…………………..…………………………………………………………10 Non-functional 2.2 ……………………………………………………………………………………..10 [Security & Licensing 2.3 10](#NonFunctionalReq)

[Individual Learning Plan Diagram 2.4 11](#LearningPlan)

[Data Dictionary 2.5 12](#DataDict)

[Roles and Responsibilities 2.6 12](#RolesResponsibilities)

[Architecture Diagram 2.7 14](#ArchitectureDiagram)

[Azure Scrum Board 2.8 15](#AzureScrumBoard)

[Login/Registration Sequence Diagram 2.9 16](#LoginDiagram)

[Automated Testing Diagram 3.0 17](#AutomatedTestingDiagram)

[Manual Testing Diagram 3.1 18](#ManualTestingDiagram)

[Appendix 3.2 19](#Appendix)

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author(s)** |
| 3/4/2022 | 1.0 | First Draft | Nathan Parisot & Sahil Jain |
| 3/24/2022 | 1.1 | Added diagrams | Nathan Parisot & Sahil Jain |
| 04/18/2022 | 1.2 | Added licensing and revised user stories | Nathan Parisot |
| 04/20/2022 | 1.3 | Revised user tasks based on user stories | Nathan Parisot |
| 04/22/2022 | 1.4 | Wrote and revised introductory summary | Nathan Parisot |
| 04/27/2022 | 1.5 | Added manual and automated testing diagram and revised Functional and Non-functional Requirements | Nathan Parisot |
| 04/28/2022 | 1.6 | Revised based on project checklist | Nathan Parisot |
| 5/02/2022 | 1.7 | Final revision | Nathan Parisot |

Revision History

Introductory Summary

ShoeStars allows you access to luxury and international designer collections at a fraction of the retail price. At ShoeStars, we are rethinking the way people have previously approached their closets. We exist because we believe a premium product should not only be experienced through owning it. Our team is prepared to go the extra length to ensure you look marvelous at all your special events. We give anyone who uses our product access to remarkable luxury experiences, we are changing the meaning of ownership- and revolutionizing retail in the process. Our customers each get a chance to enter a free giveaway by using our service to win an NFT, valued at up to $100. Would you spend thousands of dollars on expensive attire just for a one-time event? The choice is yours, for a much lower price you can get the same product without having to own it.

1. Introduction
   1. Purpose

Here at ShoeStars, our purpose is to make luxury available to anyone. We want you to be able to change your style/attire to whatever you desire without having any limitations with pricing. Our services will not only benefit the customer but will also help change the way we perceive renting a luxurious item without having to own it.

1.2 Charter

* Statement of Needs
* Major project requirements
* Major Milestones and Key Dates
* Roles of team members
* Stakeholders and Team
  1. Personas

1. Customer has an unexpected wedding and needs dress shoes urgently

2. Customer has a business meeting and is required by their boss to wear something formal.

* 1. Initial Scenarios

1. Customer has an unexpected wedding and needs dress shoes urgently. The customer does not want to buy shoes and only wants the dress shoes for the occasion. Therefore, they will use our system. Customer goes onto website and is required to make an account on our website. This is because if they need to return the shoe, they will have an account with an email, phone number, and other basic contact information. Once the customer has registered, they will be able to browse through various categories where they will be able to find the exact shoes they desire. Customer proceeds to checkout and rents the shoe.

2.  Customer has a business meeting and is required by their boss to wear something formal. They go to our website and checkout with the shoes they want. They accidentally order the wrong size after checking out. Customer then cancels order and submits a reason ticket. They get a confirmation email about the cancelled order from customer service. The user can then place another order for the right pair.

* 1. Initial Features

1. Subscriptions

2. Randomly placed QR codes in shoes boxes for a chance to enter a giveaway for a NFT owned by our company.

3. Recommendations for customers based on shoes they have bought.

4. Homepage for featured/available shoes

* 1. User Stories

User Stories:

Marcus makes a new account with email, phone number, address, etc. to be able to rent out shoes on our website. Marcus gets an order of 2 pairs of shoes. Lastly, Marcus logs in to the website and is immediately introduced to shoes that are like what they have ordered in the past.

Chantelle gets her product from us and finds an interesting QR code on the inside of the box. She then scans the QR code with her smartphone and follows the prompts to login into her account and enter our free NFT monthly giveaway valued up to ~$100. She then proceeds to tell her friends and family.

One of our best customers, Dr. Jones, has decided to use our subscription. He goes on with his day and gets notified on his email that we offer advantages like, 20% off every purchase, and getting informed on all new upcoming products when using our subscription.

* 1. User Stories Tasks

User Story 1

1) Log in using credentials

2) Order 2 pairs of shoes

3) Redirected to homepage and is introduced shoes that are like what they have ordered in the past.

User Story 2

1. Open shoe box

2) Scan QR code using smartphone

3) Enter login credentials to enter free giveaway

User Story 3

1) Click add new subscription

2) Enter payment method

3) Confirm subscription selection and payment

* 1. Configuration Management Rules

GitHub will be used to help inspire/assist with coding, and layouts of the website. If any coding or layout is edited/changed, use commits. Only use branches when necessary and properly. Commit messages in general should be made together and will be enforced with profession. Workflow will be split using branches from GitHub with certain sections.

* 1. Code Rules

The Technology Stack will be Windows’ Operating System, Microsoft PowerPoint, HTML, CSS, and GitHub. New team members will meet with current team members. After that, the current team members will go over all coding and other aspects of the projects. New roles will be sorted out and the workload will be divided from there. <https://github.com/philipwalton/html-inspector>

1. Testing Rules

We will run the code thoroughly at each stage and make sure that it runs smoothly. If not, we will for sure run into errors and fix them right then and there, instead of finding out later that there are a lot of errors that should have been fixed earlier. Before the code is even committed, all tests must be successful overall.

2.1 Functional Requirements

1. Customer wants to go to account to see previously worn shoes. A section labeled Order History is required. Customer can reorder what they have purchased before.

2. Customer can accept or refuse shoe suggestions. A similarity tool is required. A yes or no button may also be required for the user to accept or decline the shoe.

3. Customer uses search bar to find specific shoe. A search bar tool is required to be embedded within the website. Search filters may also be required to filter through the searches

4. Customer types in billing information to place an order or to purchase Shoe Star subscription. A verification code may be required to verify user before placing order.

5. Customer scans QR code inside shoebox for an NFT giveaway. Understanding the concept behind NFT’s and crypto currency is required. A functioning QR code is also required.

2.2 Non-Functional Requirements

1. A Third-party API is required to protect customer card information. A save option to save the user card information may also be required. Not to mention security is also a big requirement to ensure to safety of the user.

2. User can browse through inventory. A discovery or browse page is required to allow casual shopping through the website. Listing filters may also be required

3. Customer can keep track of new NFT giveaways. An event page is required to list all previous, current, and upcoming giveaway

4. Customer can refer a friend to get a discount on next purchase. A sharable link is required along with individual codes to identify which account is being referred.

5. Customer can keep up to date with the latest shoes. An email notification tool is required to send reminders about new shoes being released.

2.3 Security & Licensing

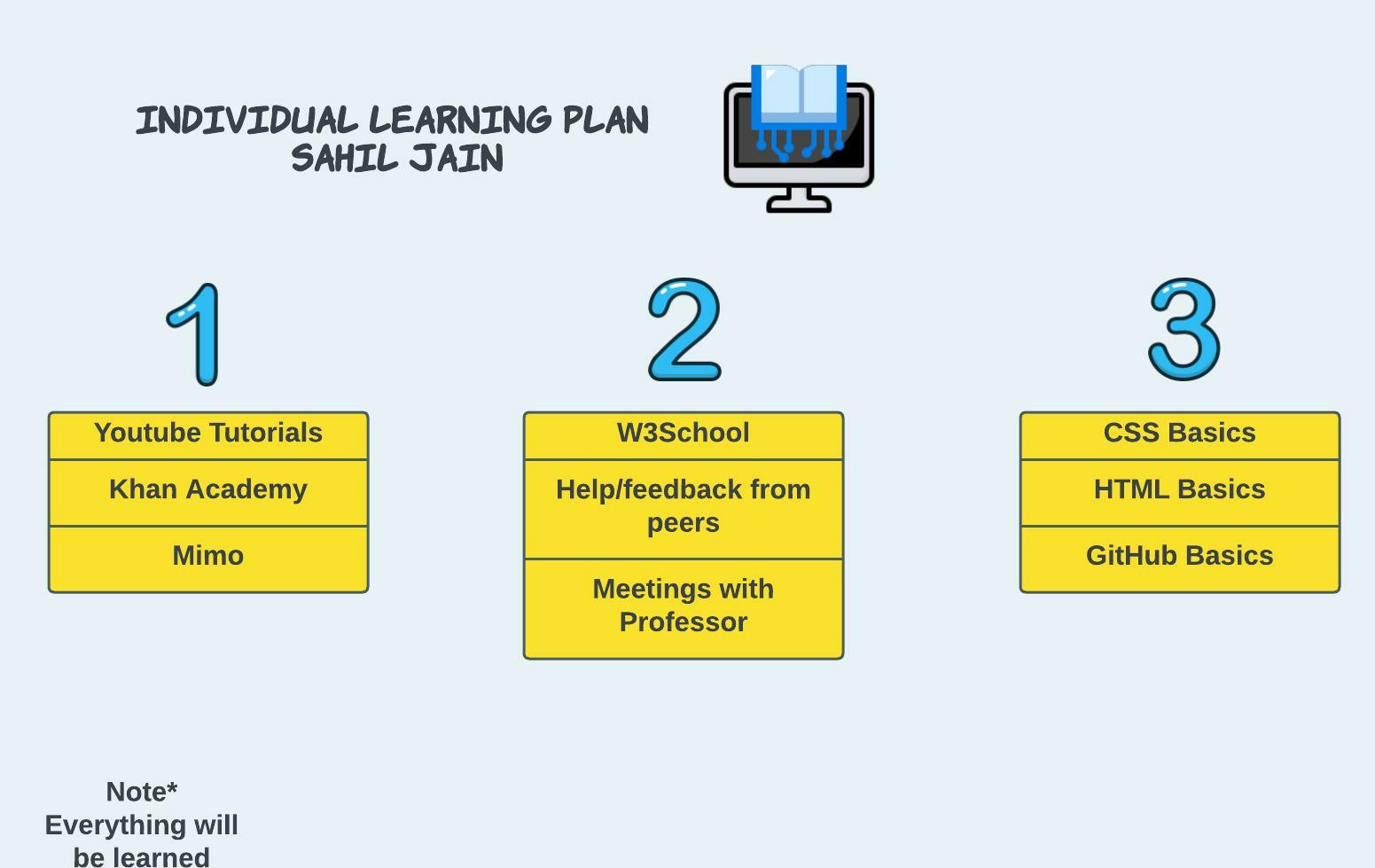
User log in or user making a new subscription/account. Security should be factored in every part of the development. We will be using a 3rd party API to take the factor of transactions being in danger off our hands. Stripe’s advanced API has predictable resource-oriented URLs, accepts form-encoded request bodies, returns JSON-encoded responses, and uses standard HTTP response codes, authentication, and verbs. <https://api.stripe.com> is the base URL.

The license: we will be using the MIT license which grants us the rights to use, change, and share for free.

Graphical user interface, application

Description automatically generated STRIPE API

Tools used: Windows Operating System, GitHub, HTML, CSS, Stripe (3rd party API)

Diagram

Description automatically generated with low confidence2.4 Individual Learning Plan Diagram

2.5 Data Dictionary

**API** - An application programming interface is a connection between computers or between computer programs. It is a type of software interface, offering a service to other pieces of software. A document or standard that describes how to build or use such a connection or interface is called an API specification.

**Structured Database** - Structured data is when data is in a standardized format, has a well-defined structure, complies with a data model, follows a persistent order, and is easily accessed by humans and programs. This data type is generally stored in a database.

**Unstructured Database** - Unstructured data is information that either does not have a pre-defined data model or is not organized in a pre-defined manner. Unstructured information is typically text-heavy, but may contain data such as dates, numbers, and facts as well.

2.6 Roles and Responsibilities

|  |  |  |
| --- | --- | --- |
| Name | Role | Responsibilities |
| Nathan Parisot | Team Lead | Oversees team project and ensures tasks are completed |
| Nathan Parisot | Documentation Tester | Ensures documents explain how the system works and are correctly filed |
| Sahil Jain | Code Developer | Develops the software for the application |
| Nathan Parisot & Sahil Jain | Tester | Tests the software |

Diagram

Description automatically generated2.7 Architecture Diagram

An architectural diagram is a visual representation that maps out the physical implementation for components of a software system. It shows the general structure of the software system and the associations, limitations, and boundaries between each element.

Graphical user interface, text, application, email

Description automatically generated2.8 Azure Scrum Board

Shown is a scrum board which helps team determine which tasks need to be completed

2.9 Login/Registration Sequence Diagram

Diagram

Description automatically generated

Shown is a diagram explaining the process of creating/logging into our website

A picture containing diagram

Description automatically generated3.0 Automated Testing Diagram

A diagram describing the relationship between automated testing, continuous integration, and continuous delivery

Diagram

Description automatically generated3.1 Manual Testing Diagram

Manual Testing. Manual testing is a software testing process in which test cases are executed manually without using any automated tool.

3.2 Appendix

Individual Contributions Breakdown

Nathan Parisot and Sahil Jain started this software engineering project and the creation of the ShoeStars web application as a team. Portions of the project in which they worked together include a small section of 1.6 User Stories, the configuration management rules 1.8, code rules 1.9, testing rules 2.0, and scrum board 2.6.

Portions of the project report that were created and developed by Nathan Parisot include the Revision History Table, Introductory Summary, 1.1 Purpose, 1.2 Charter, 1.3 Personas, 1.4 Initial Scenarios, 1.5 Initial Features, majority of 1.6 User stories, 1.7 User Story Tasks, 2.1 Functional Requirements, 2.2 Non-Functional Requirements, 2.3 Security & Licensing, 2.4 Individual learning Plan Diagram, 2.5 Data Dictionary, 2.6 Roles & Responsibilities, 2.7 Architecture Diagram, 2.9 Login/Registration Sequence Diagram, 3.0 Automated Testing Diagram and 3.2 Appendix. Both members contributed to giving other teams feedback on their project items and Nathan Parisot edited the ShoeStars project items based on other team’s feedback.

As for the sections that Sahil Jain worked on or created himself include 3.1 Manual testing Diagram, 2.4 self-learning diagram, the ShoeStars website, and the ShoeStars project presentation.

As this was a team project, it is important to note that majority of the project was created by Nathan Parisot.