**STUDENT**

SDS Report

# **Executive Summary**

In this report, we comprehensively analyze and explore different factors that contributed to the designing, development, and implementation of the project. As discussed we are developing an application that allows the customers to purchase a ticket while also allowing the organization to list their tickets on the platform. The project is all about creating an environment where all of these functions can be operated. The report consists of a brief explanation of the project, its conceptual view, interactive view, and design viewpoints. We together discuss the importance of the tools and technologies that we are using in the designing, development, and implementation of the project.

In the report, we also discuss the different design aspects of our application and comprehensively explain individual components that together made up our web application. Furthermore, we also discuss the traceability matrix and sequence diagram to share the individual feature status and program sequence respectively.

Table of Contents

[**Executive Summary** 1](#_Toc110896828)

[**1.0** **Introduction** 4](#_Toc110896829)

[**1.1** **Project Overview** 4](#_Toc110896830)

[1.1.1 Project Aim 4](#_Toc110896831)

[1.1.2 Project Objective 4](#_Toc110896832)

[1.1.3 Project Solution 5](#_Toc110896833)

[**1.2** **Document Usage** 5](#_Toc110896834)

[**1.3** **Reading Recommendations** 6](#_Toc110896835)

[**2.0** **Conceptual Model** 6](#_Toc110896836)

[**2.1** **Technologies** 6](#_Toc110896837)

[2.1.1 Backend (FLASK) 6](#_Toc110896838)

[2.1.2 PSQL 7](#_Toc110896839)

[2.1.3 Draw.io 7](#_Toc110896840)

[2.1.4 GitHub 8](#_Toc110896841)

[2.1.5 Bootstrap 8](#_Toc110896842)

[**2.2** **System Overview** 8](#_Toc110896843)

[2.2.1 Account Functionality 9](#_Toc110896844)

[2.2.2 Search Bar 9](#_Toc110896845)

[2.2.3 Category Page 9](#_Toc110896846)

[2.2.4 Organization Page 10](#_Toc110896847)

[**2.3** **Architectural Pattern** 10](#_Toc110896848)

[2.3.1 Model 10](#_Toc110896849)

[2.3.2 View 11](#_Toc110896850)

[2.3.3 Controller 11](#_Toc110896851)

[**3.0** **Design Viewpoint** 13](#_Toc110896852)

[**3.1** **Logical Viewpoint** 13](#_Toc110896853)

[**3.2** **Information Viewpoint** 14](#_Toc110896854)

[**3.3** **Interface Viewpoint** 15](#_Toc110896855)

[**4.4** **Site Map** 16](#_Toc110896856)

[**4.5** **User Interfaces** 17](#_Toc110896857)

[**4.6** **Interaction Viewpoint** 30](#_Toc110896858)

[**5.0** **Traceability Matrix** 31](#_Toc110896859)

[**6.0** **References** 32](#_Toc110896860)

# **1.0 Introduction**

## **1.1 Project Overview**

### 1.1.1 Project Aim

One of the core aims of the project is to create a platform through which users can view different tickets from different organizations. The project is also set to deliver important tools that are needed for the proper management of the tickets for the organization. The main motivation for the project is inspired by the long lines at the ticket station and the lack of collaboration between platforms and organizations relate to the ticket. The customer needs to visit a different website or web application to get their ticket sourced. We set to design and implement a solid platform that enables users to search and booked their tickets from one platform instead of browsing a different website.

### 1.1.2 Project Objective

After comprehensive research and analysis, we concluded that the company objective needs to be straightforward for anyone to understand. As the company needed to share and communicate its plans with the implementation team and the success of the project depended upon the quality of communication and the implementation team's understanding of the objectives. Following are some of the important objectives decided by the team and the stakeholders.

* The project management or project lead is responsible for enabling communication between the team and the stakeholder (executive), this person works towards locking the project scope and delivered and approves important project documents such as project charter, project SRS, and project SDS.
* The project SRS document is set to share with the executive team by 24 July 2022, and the project SDD is set to share with the implementation team by 15 August. 2022.
* All the project functional and non-functional requirements are locked by 20 July and the Product backlog is developed.
* The project development team is responsible for the designing, development, and implementation of the project's functional and non-functional requirements.
* The test is done at the end of each sprint phase and the final test is conducted at the end of the project when all the functional and non-functional requirement is complete.

### 1.1.3 Project Solution

As discussed in the previous document, the company trying to solve a ticket accessibility issue for the customers. Currently, as we know by now that customers cannot able to find a platform where they can search for any type of ticket but this is just where our solution fits right in. The company is set to design, develop and implement a web application that let different organization list their ticket without a platform that let the customer view different tickets on one platform. For this, to successfully function, the company needs to design an inventory management and order management system for the organization through which the organization can add tickets and manage their orders.

## **1.2 Document Usage**

The report is designed to list and document all the necessary things that needed to be done before the development phase is initiated. This document is developed for the project manager, project stakeholder, project implementation team, and others to review all the design components and put them through and review each functionality. These inputs are recorded and presented in a final form onto which everyone approved the project and gives a green signal to the development team for the development of the application. This process helps to ensure that the whole team is in one phase, this way the team can focus on the development of the application rather than continuously reevaluating the requirements.

## **1.3 Reading Recommendations**

The document is divided into three parts, the first is concerned with the introduction of the project, its aim, its objective, and what is the solution we proposed. After which is comprehensively discuss the importance of the conceptual models and what role different technologies make in our application, what some of the systems view, and what is the architectural plan we set to achieve. Lastly, we discussed the logical, informative, and interface design viewpoints and explain how each component worked towards helping the project.

# **2.0 Conceptual Model**

## **2.1 Technologies**

### 2.1.1 Backend (FLASK)

The Flask is the backend technology (OSADCHUK, 2021) that we are using for defining web applications' different routing and enabling the connection between the frontend and the database. Recalling our MVC architecture (Team, 2021) the backend here plays an important role as the controller which is the communicator for the view and model. The FLASK is the web framework offered under the python web solution framework. Following are some of the important features of the backed.

* Easy integration of the model (database) with the backend
* A comprehensive document is needed for each function that is developed for the backend, this includes the best practices that are used for handling database queries, routing information, and others.
* Customer scripts are designed to easily upload tickets from the backend to the database, this help to ensure that organization can efficiently upload all of their tickets at once.

### 2.1.2 PSQL

PSQL or PostgreSQL (Peterson, 2022) is the database that we are using for our application. The PSQL belongs to the relational database (IBM Cloud Education, 2019) category meaning here we create different relationships such as one-to-one, one-to-many, and many-to-many relationships. This helps us to enforce data laws through which the data is inputted into the system and according to these laws, the data get output from the system. Here are some of the important points about the PSQL database.

* We needed to develop a comprehensive document that help to define all the relationships among database different entities.
* The PSQL can easily integrate with FLASK backend only with a couple of lines of codes which helps the team to better focus on the designing rather than on the integration technicalities.

### 2.1.3 Draw.io

The draw.io is the graphical open source solution that we are extensively using for the creation of the different diagrams and figures. This is the primary tool that we are using for the creation of the class diagram, entity relationship diagram, sequence diagram, and other diagrams. The application is accessible through the internet its figure can be locally downloaded which make it easy to create these diagram and to input and share in a document and among the team members.

### 2.1.4 GitHub

GitHub (kinsta, 2022) is the version control software that helps to track the different versions of the code, the software is specially designed to help the developers to save different versions of the same code during the development of the new feature. A good example is when the developer is working on a feature and the testing of the code is complete push the code to GitHub. After the changes have been staged and committed, the developer can start with developing a new feature and further developing the same feature, ensure that whatever he/she does, does not affect the code and he/she can always go back to the previous code. Here are some of the important points related to GitHub.

* All the changes are individual stages, meaning the current version of the code can always refer back to the previous stage
* Branches help to develop a new feature on a working application, which ensure the stability of the application, and only after all the feature are developed and tested are they merged with the original branch.
* GitHub also has CI/CD functionality which can be integrated to automatically perform tests whenever the new code is pushed in.

### 2.1.5 Bootstrap

Bootstrap is a design framework, which helps in building designing components. It is a framework that works on top of the CSS sheets or is quite similar to the CSS sheet. Together the bootstrap and CSS sheet help the frontend developers to create interactive designs without needing to build things from the scratch. The bootstrap is a flexible framework that can work with the existing sheet to provide a style guide or can use standalone for the different components.

## **2.2 System Overview**

It is clear by now that we are working in a module-based environment where each module is individually designed, developed, and implemented, and only after some set of predefined tasks does the team integrate all the functionality and perform integration testing. Following are some of the important functionality of the system.

### 2.2.1 Account Functionality

The first is concerned with the designing, development, and implementation of the account's functionality. This includes the development of the register function, login function, and logout functions. The accounting functionality allows the customer to log in to their account and view different options such as their recent order, the account detail, and place orders. On the other side the account functionality allows the organization to register, login, and view their products, list them and if needed remove them. They also can edit the ticket details and log out of the website.

### 2.2.2 Search Bar

The search bar plays an important role in defining one of the functionalities that are crucial for the company and the customer. As nobody is going to view 10 pages of the ticket only to find the ticket they are searching for. The company understands this and that is why they work towards the designing, development, and implementation of the search bar that does not only find the matching product but also list all the product that is close to the search result.

### 2.2.3 Category Page

The category pages allow us to group similar industry tickets into one group, a good example of which is that the baseball ticket, football ticket, and tennis ticket should belong to the sports category while the drama, concert, and play ticket should belong to the entertainment section of the category. It distinction among the ticket helps the user to view tickets according to their interest and it also allows the company to view insights about what industry or what category their customers are more interested in, so they can double down to bring more options in that industry while working to resolving a problem with other non-performing categories.

### 2.2.4 Organization Page

The organization page help to better understand and manage the individual organization to view their listed tickets, list their tickets, and also help them to edit or delete any listing that they want. This act as the main dashboard that helps the organization manage and operate their listing from this protocol. The company also later be working on the development of analytical tools and business tools that help the organization to better evaluate the performance of their different listing.

## **2.3 Architectural Pattern**

As discussed in the previous report, the implementation team conducted a comprehensive survey into defining the architectural pattern that is composed of the three important components the model, the view, and the controller. This model is commonly referred to as the MVC architecture, this architecture helps us to individual work of the different components, and only after certain pre-defined completion of features these three modules are integrated and tested.

Following is a comprehensive explanation of these components.

### 2.3.1 Model

The model defines the protocol and processing logic onto which the data and information are stored, flow, and process among different entities. The model is mainly responsible for the management of the storage solution for the company. The model we choose for the project belongs to the relational database category which helps the developers to draw relations between the entities which later become the protocol through which the data are processed and stored.

Our database referred to as the PSQL or PostgreSQL integrates with the main controller, which is responsible for querying the database with correct information and with correct syntax.

### 2.3.2 View

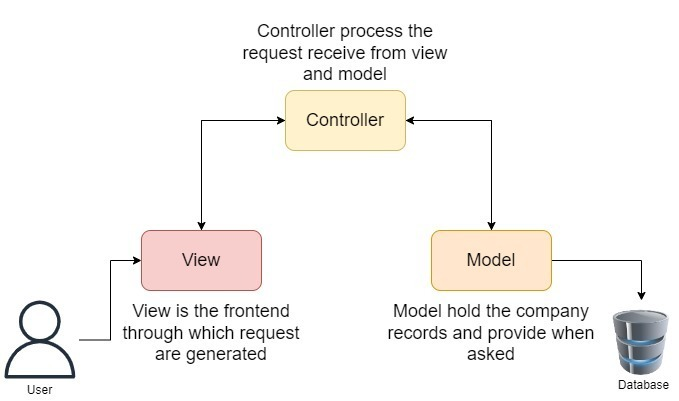
The view on the other side works for the designing, development, and implementation of the front screen that the user sees when they enter the application URL. The screen that comes after the loading of the website is the work of the view. Meaning the view is the only component in the MVC architecture that directly connect with the user and take their queries and provide them the response with the help of the controller and the model.

The technology that we are using for the view is HTML, CSS, and JavaScript with bootstrap. All of these help us to structure the basic HTML page and the other help to make the page design-wise pleasing and help the implementation team to integrate user interface and user experience into the static HTML structure.

### 2.3.3 Controller

The controller is the middle person between the two modules, which take input from the view process it, and then send the query to the model, and then the output from the model process it and send it back to the view. The controller is responsible for ensuring that all the best practices are used while the extraction of the data, processing the data, and sending queries to the model. The controller ensures the system is saved from potential SQL injection attacks, password sniffing, and other bad activities.

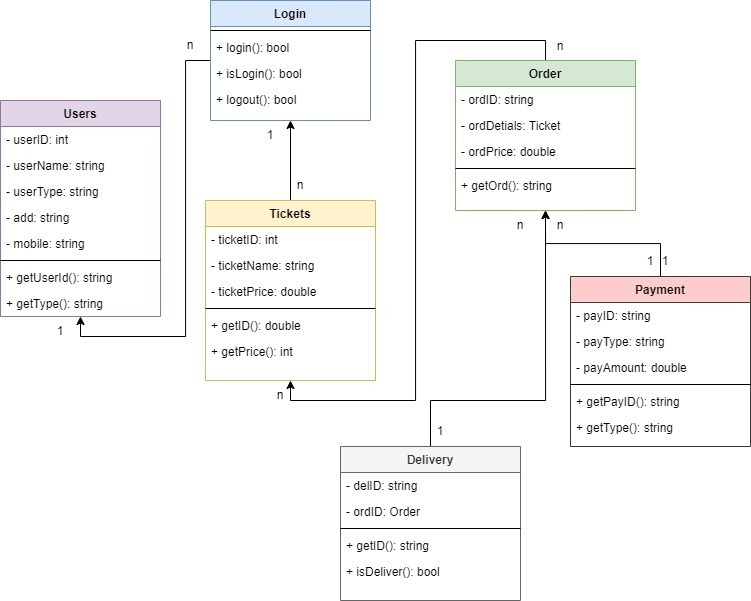
The technology we choose as our controller is the FLASK, which is a python web framework equipped with drawing simple application and also help in the designing, and development of a complex process.



# **3.0 Design Viewpoint**

## **3.1 Logical Viewpoint**

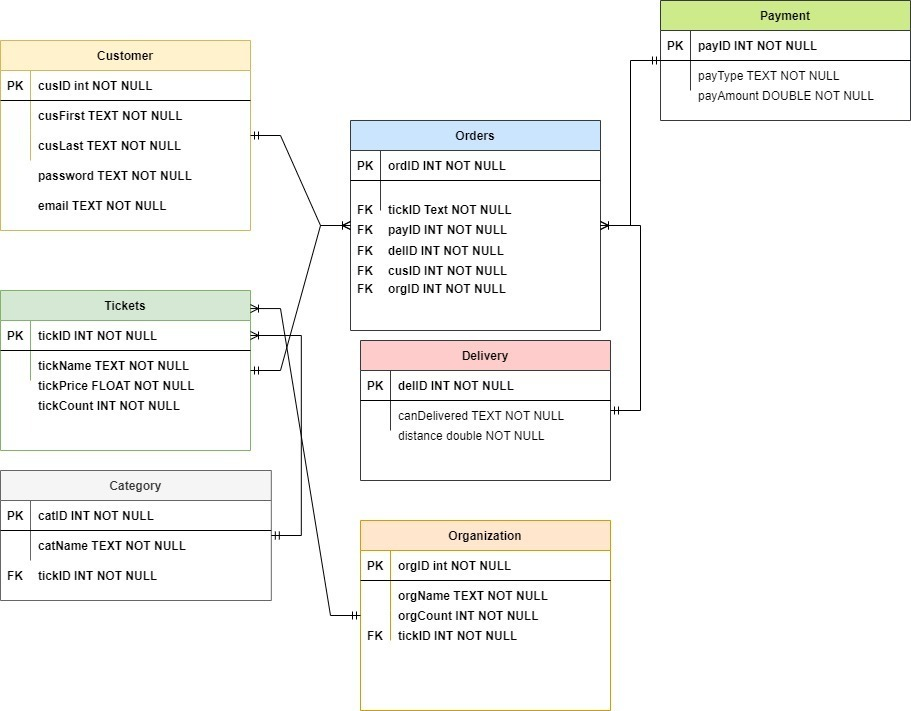
The logical view helps us to understand the complexity of the application, this helps us to define how different information flows in the system and how the data is stored. We take help from the class diagram to explain the functional dependencies of the project. This act as the guide system using which different structure is designed and developed. This helps the development team to better understand the flow of information and define functions that work according to the set business and project requirements.



## **3.2 Information Viewpoint**

The information viewpoint is the actual representation of how the information and data should flow in the working application. In the information viewpoint, we take help from the entity relationship diagram to define different tables that are responsible for defining what the components are stored in the table and how these table content are linked with the different table components. It is this relationship between the table that later help to find record depending on some filter value. This is useful in finding the record with the help of another record entry.

In the diagram, we can observe that the users' table hosts the information related to user account details such as their name, password, and other details. We also observed that the vendor does have other tables where it takes userID and further defined the vendor details and then there is the order table that is connected to the product page and the user pages defining that one order can belong to only one user but a user can have many orders.



## **3.3 Interface Viewpoint**

Interface viewpoint help to define the different interface that the development and designing work upon to make the platform more user-friendly. Following are some of the important components of the user interface viewpoint.

**Categories**

These are the set of categories that helped to group different similar tickers into different categories which help the customer browsing experience. So when the customer is not looking for a particular product then they can utilize our different categories to find the ticket that they are interested in.

**Navbar**

The continuous navigation bar across different web pages allows the customer to view different web pages of the website from anywhere around the website which help to quickly switch from the category page to the about page and to the about page to the contact page.

**Interfaces**

The primary aim of the project is to find a way not to repeat the component of the website. That is why we first carefully design different components of the web application and ensure less redundancy among the application. While we ensure that there is no redundancy among the application there are different components that are needed to copy such as the navigation bar and the footers. To save from redundancy we create a global file from where all the pages inherit the global component so that the developer does not need to copy and paste components.

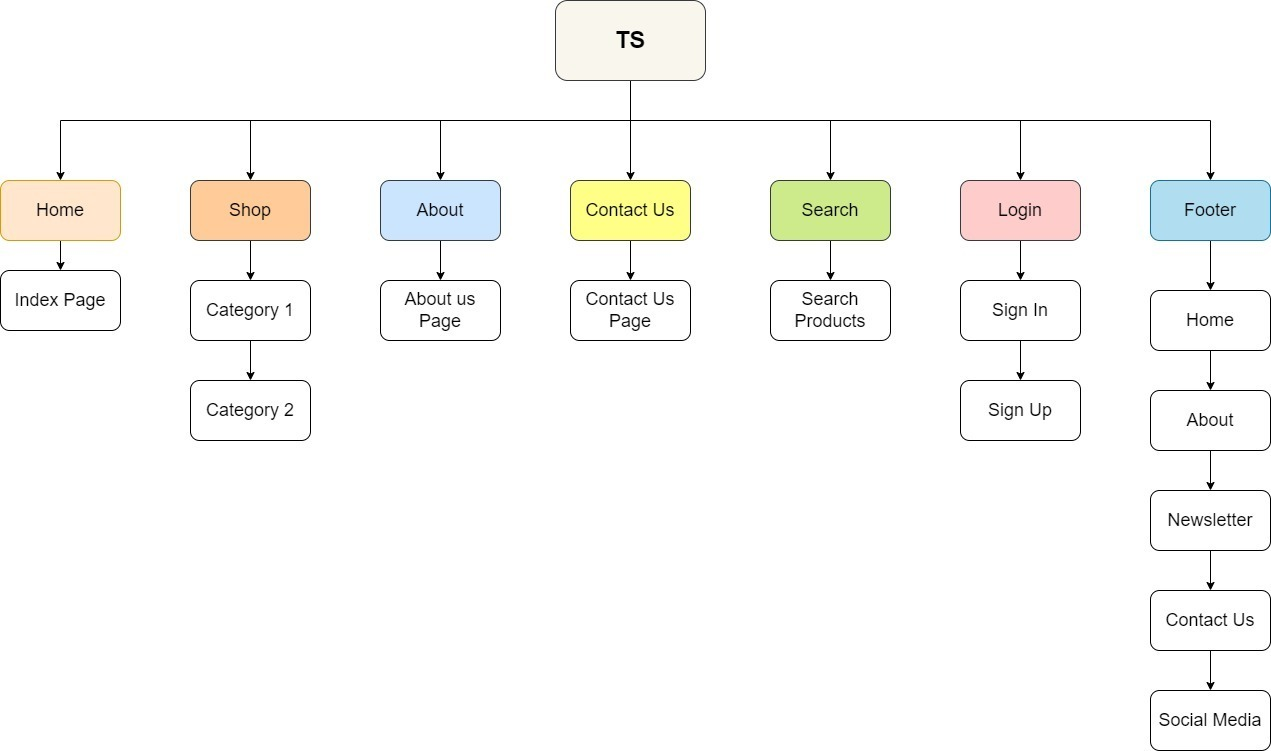
**Layout**

The company wanted to go for the web solution because they are searching for a way that helps them to reach different screen sizes, and customers, without creating a specific code base from them. We design our layout in a way that runs over all the screens and uses different browsers.

**Color and Other**

The primary aim with the color and layout of the application is to use simple and intuitive styling that matches the modern design theme and to achieve that we use a combination of attractive coloring schemes all across the web application.

## **4.4 Site Map**

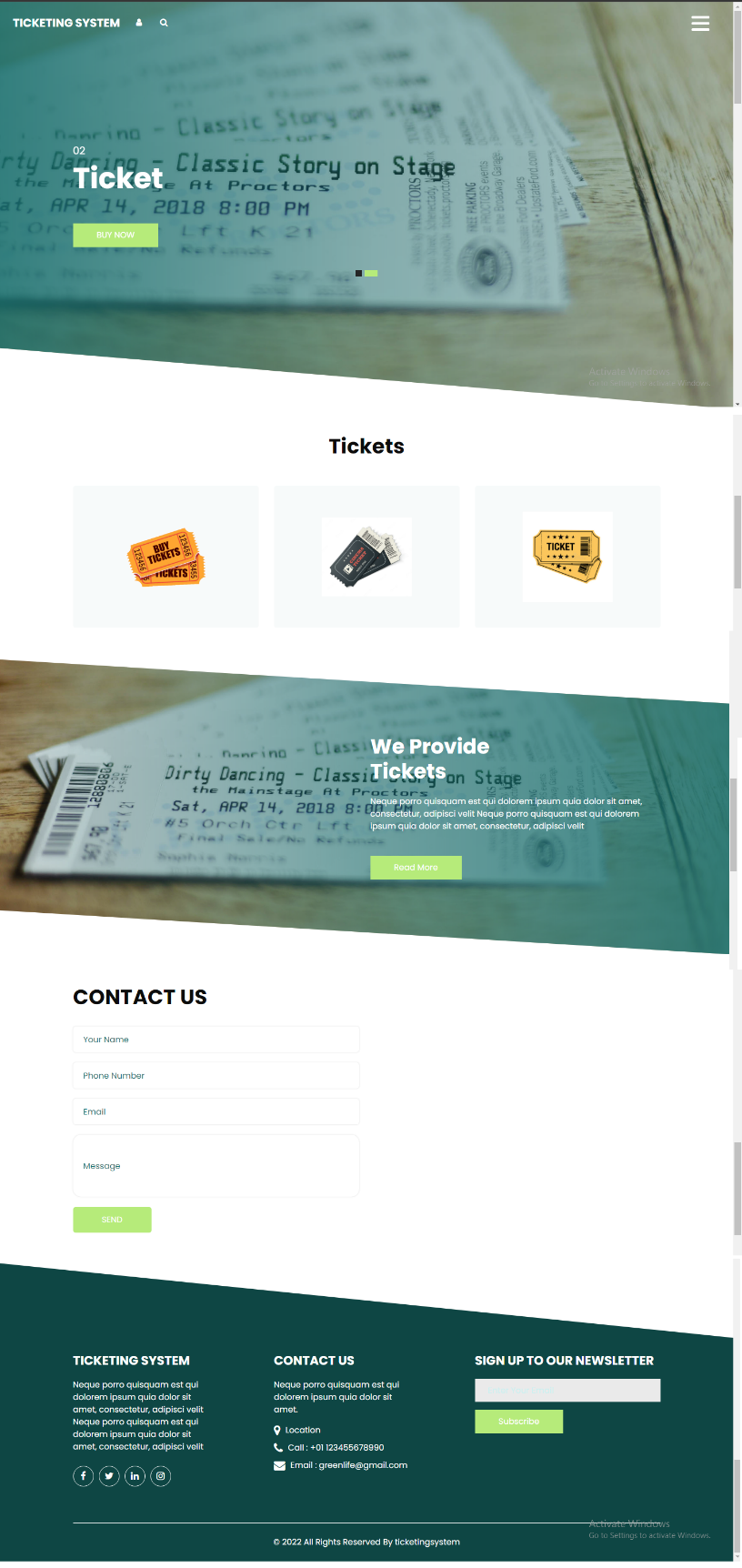


## **4.5 User Interfaces**

**Index Page**

2

0



7

6

5

4

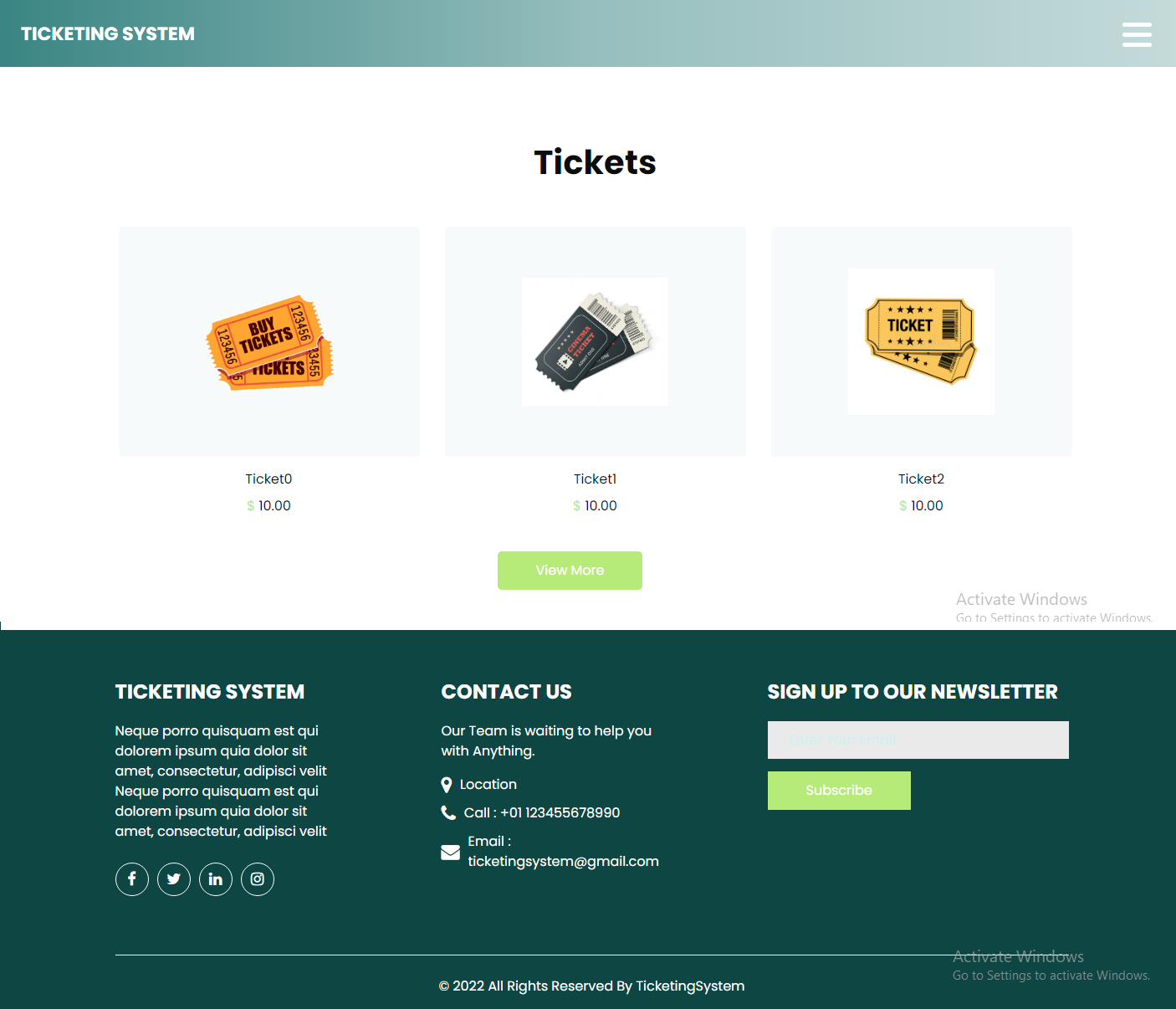
3

1

|  |  |
| --- | --- |
| **Footnote** | **Interactions** |
| 0 | On Click -> Case 0:  Open the Ticketing System homepage |
| 1 | On Click -> Case 1:  Open the TS Product Page |
| 2 | On Click -> Case 2:  Open the TS Navigation Panel |
| 3 | On Click -> Case 3:  Open the TS Featured Page |
| 4 | On Click -> Case 4:  Open the TS About US Page |
| 5 | On Click -> Case 5:  Open the TS Contact US Page |
| 6 | On Click -> Case 6:  Open the TS Footer Page |
| 7 | On Click -> Case 0:  Open the TS Email Subscription |

**Product Page**

0

****

1

3

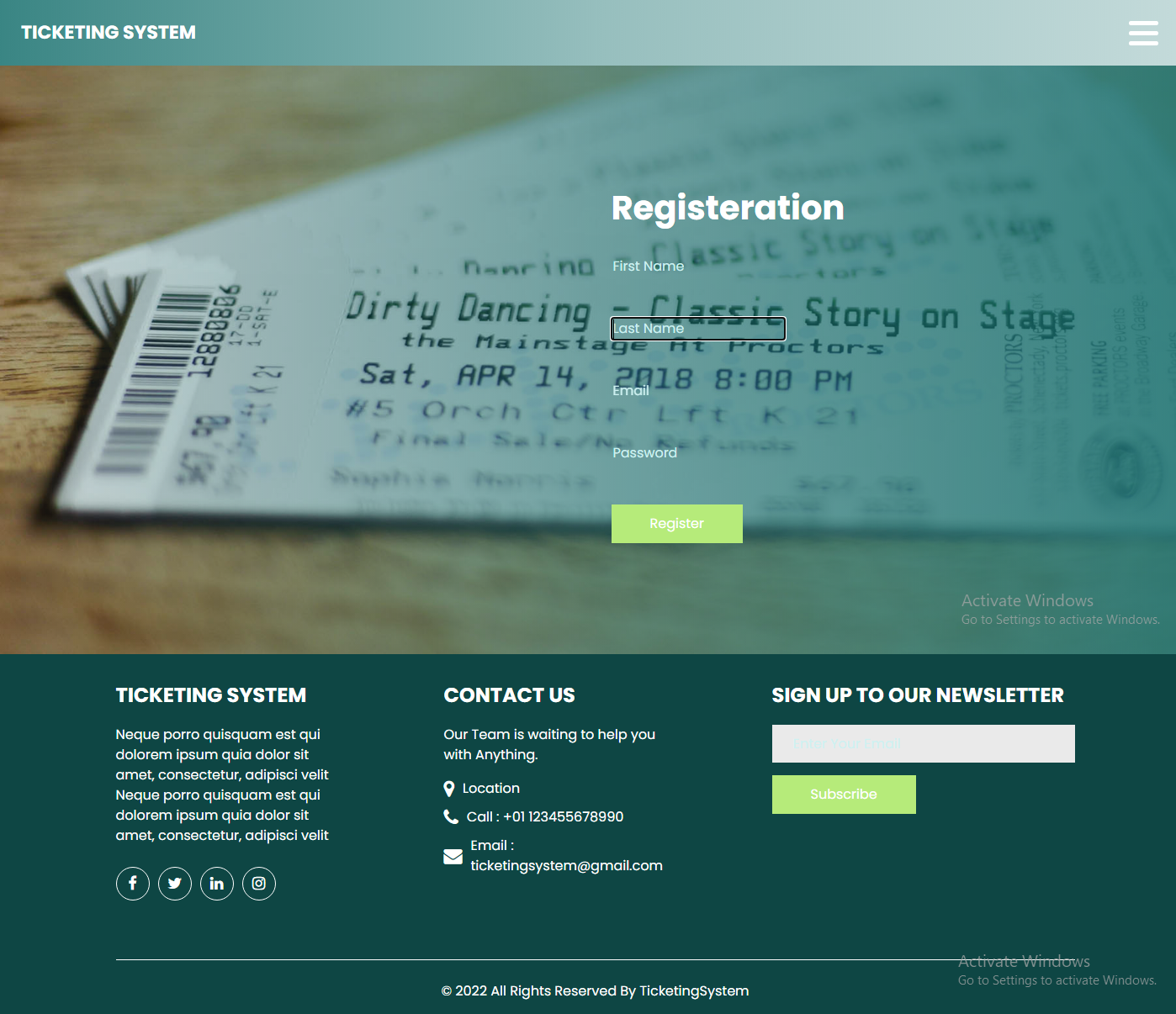
2

5

4

|  |  |
| --- | --- |
| **Footnote** | **Interactions** |
| 0 | On Click -> Case 0:  Open the Ticketing System homepage |
| 1 | On Click -> Case 1:  Open the TS Navigation Panel |
| 2 | On Click -> Case 2:  Open the TS Products |
| 3 | On Click -> Case 3:  Open the TS More Products |
| 4 | On Click -> Case 4:  Open the TS Footer Page |
| 5 | On Click -> Case 5:  Open the TS Email Subscription |

**Registration Page**

****

0

8

7

6

5

4

3

2

1a

|  |  |
| --- | --- |
| **Footnote** | **Interactions** |
| 0 | On Click -> Case 0:  Open the Ticketing System homepage |
| 1 | On Click -> Case 1:  Open the TS Navigation Panel |
| 2 | On Click -> Case 2:  Enter the First Name |
| 3 | On Click -> Case 3:  Enter the Last Name |
| 4 | On Click -> Case 4:  Enter the Email Address |
| 5 | On Click -> Case 5:  Enter the User Password |
| 6 | On Click -> Case 6:  Submit the Form |
| 5 | On Click -> Case 5:  Enter the User Password |
| 6 | On Click -> Case 6:  Submit the Form |

|  |  |
| --- | --- |
| 7 | On Click -> Case 7:  Open the TS Footer Page |
| 8 | On Click -> Case 8:  Open the TS Email Subscription |

**Login Page**

****

0

1

7

6

5

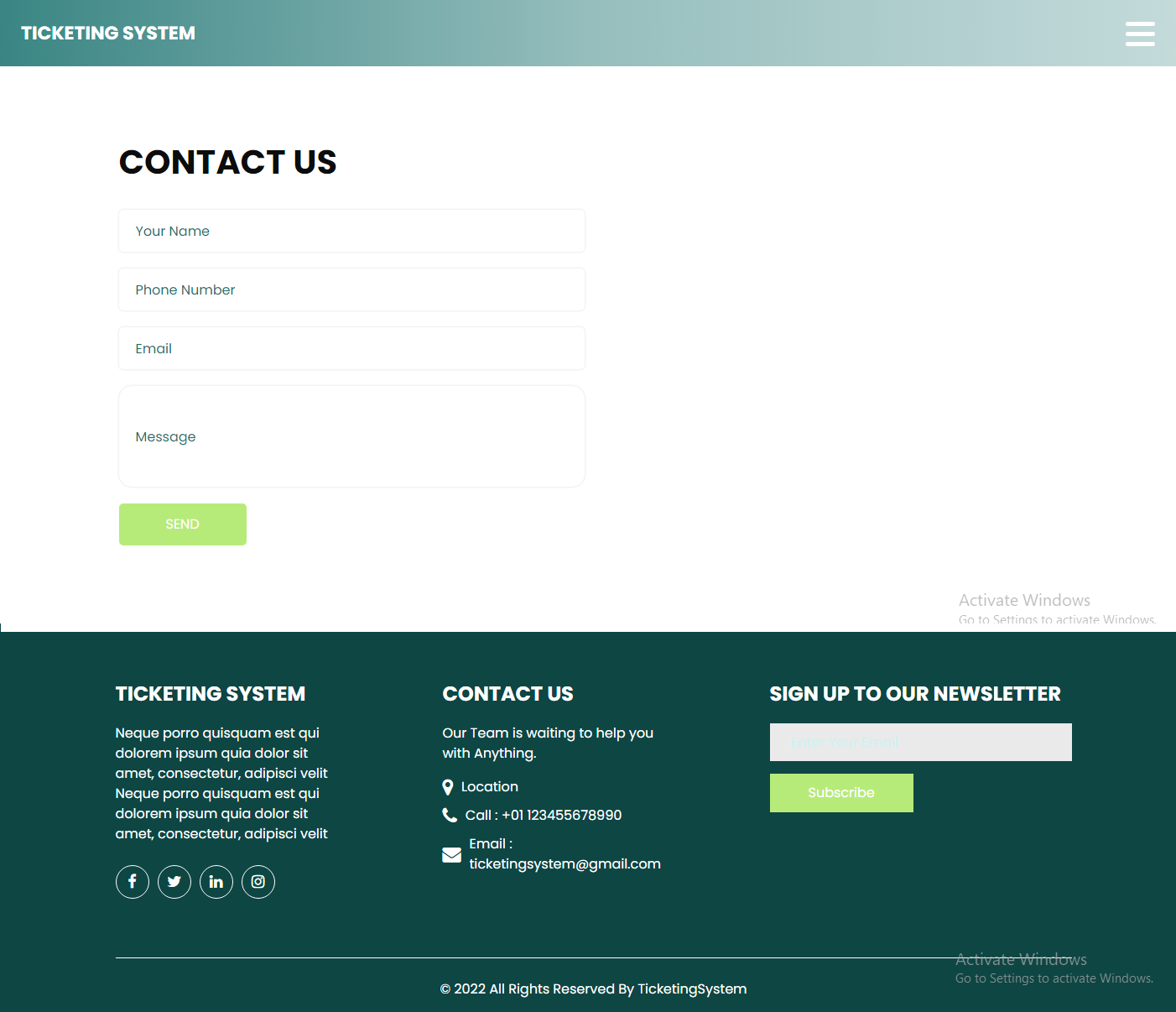
4

3

2

|  |  |
| --- | --- |
| **Footnote** | **Interactions** |
| 0 | On Click -> Case 0:  Open the Ticketing System homepage |
| 1 | On Click -> Case 1:  Open the TS Navigation Panel |
| 2 | On Click -> Case 2:  Enter the Enter Email |
| 3 | On Click -> Case 3:  Enter the Password |
| 4 | On Click -> Case 4:  Submit the Form |
| 5 | On Click -> Case 5:  Open the Register Page |
| 6 | On Click -> Case 6:  Open the TS Footer Page |
| 7 | On Click -> Case 7:  Open the TS Email Subscription |

**Contact Us Page**

****

8

7

0

6

5

4

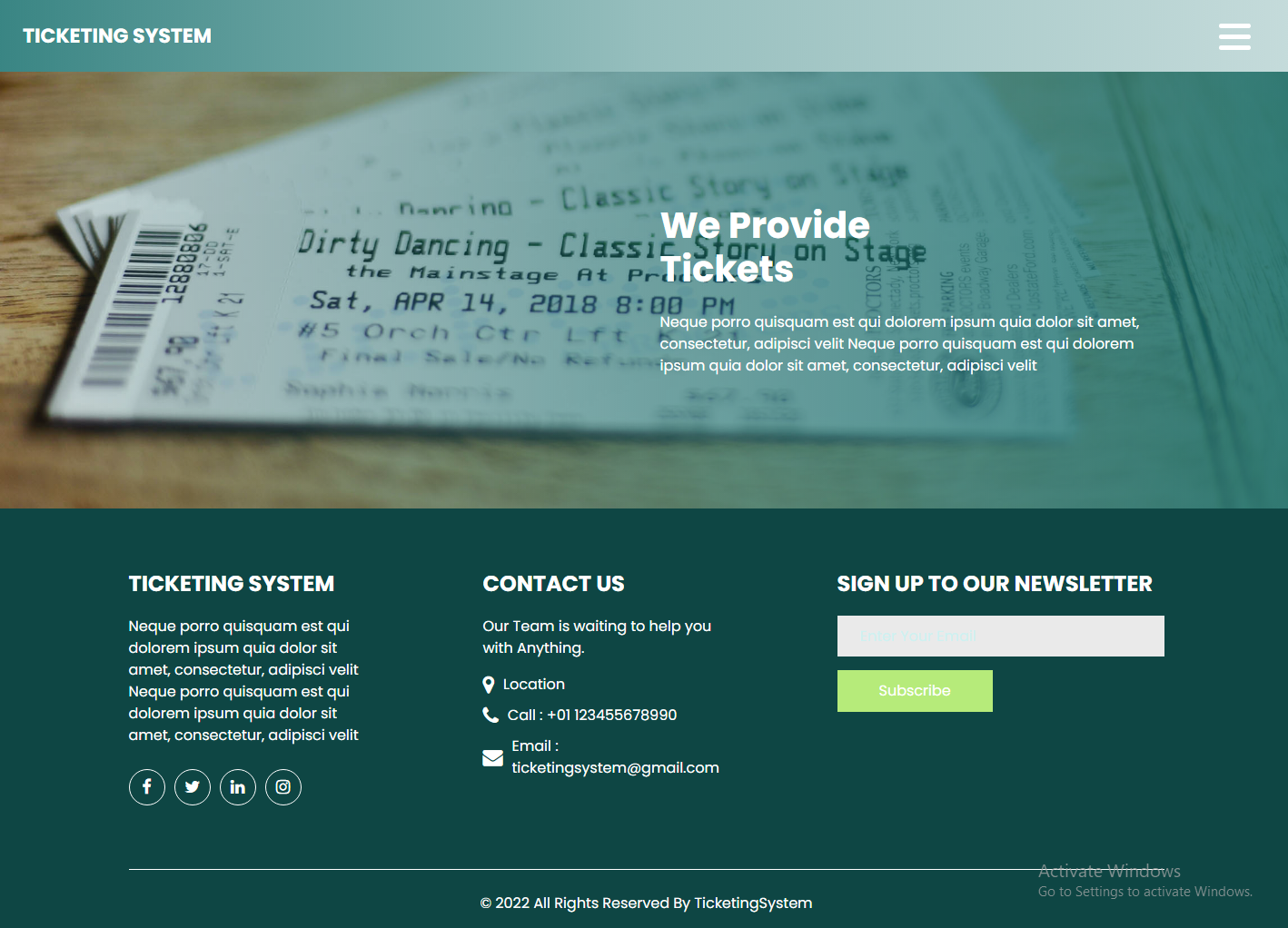
3

2

1

|  |  |
| --- | --- |
| **Footnote** | **Interactions** |
| 0 | On Click -> Case 0:  Open the Ticketing System homepage |
| 1 | On Click -> Case 1:  Open the TS Navigation Panel |
| 2 | On Click -> Case 2:  Enter the Your Name |
| 3 | On Click -> Case 3:  Enter the Phone Number |
| 4 | On Click -> Case 4:  Enter the Email |
| 5 | On Click -> Case 5:  Enter the Message |
| 6 | On Click -> Case 6:  Submit the Form |
| 7 | On Click -> Case 7:  Open the TS Footer Page |
| 8 | On Click -> Case 8:  Open the TS Email Subscription |

**About Us**

****

2

4

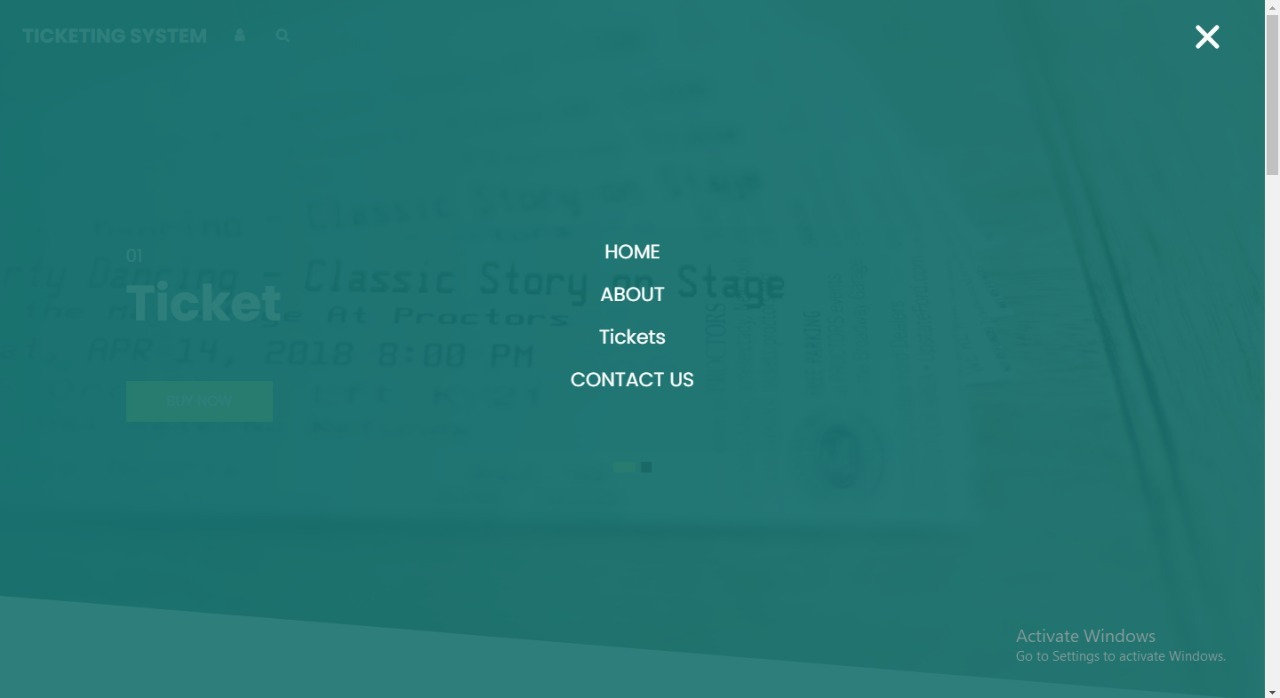
3

1

0

|  |  |
| --- | --- |
| **Footnote** | **Interactions** |
| 0 | On Click -> Case 0:  Open the Ticketing System homepage |
| 1 | On Click -> Case 1:  Open the TS Navigation Panel |
| 2 | On Click -> Case 2:  Open the TS About Us |
| 3 | On Click -> Case 3:  Open the TS Footer |
| 4 | On Click -> Case 4:  Open the TS Email Subscription |

**Navigation Bar**



3

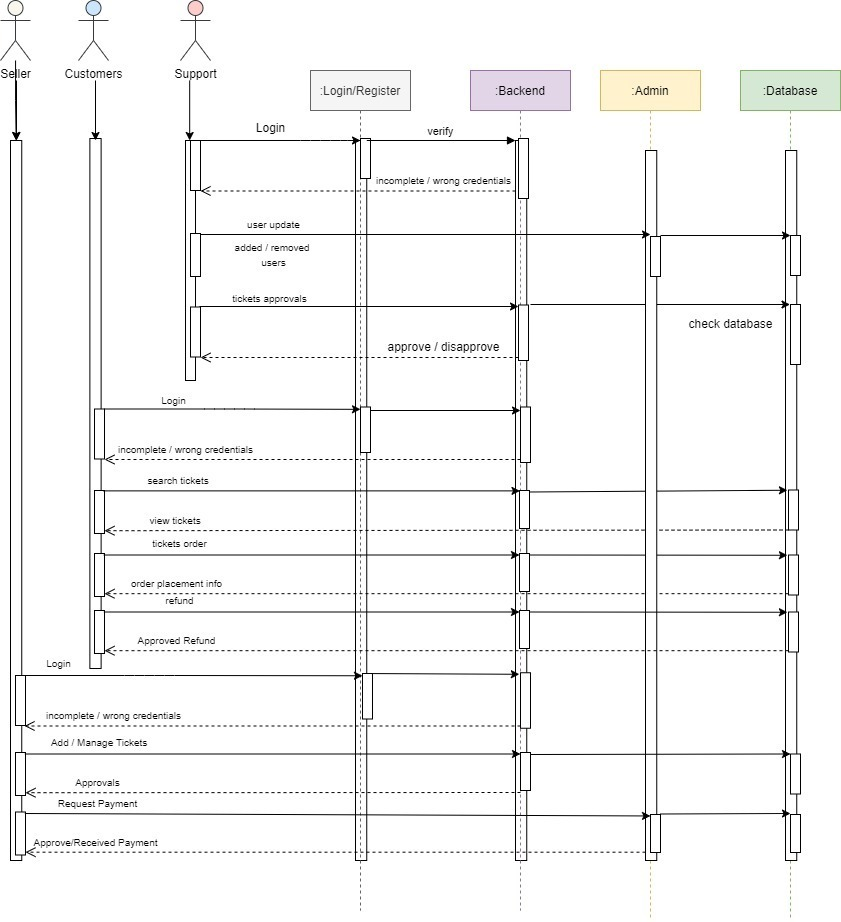
2

1

0

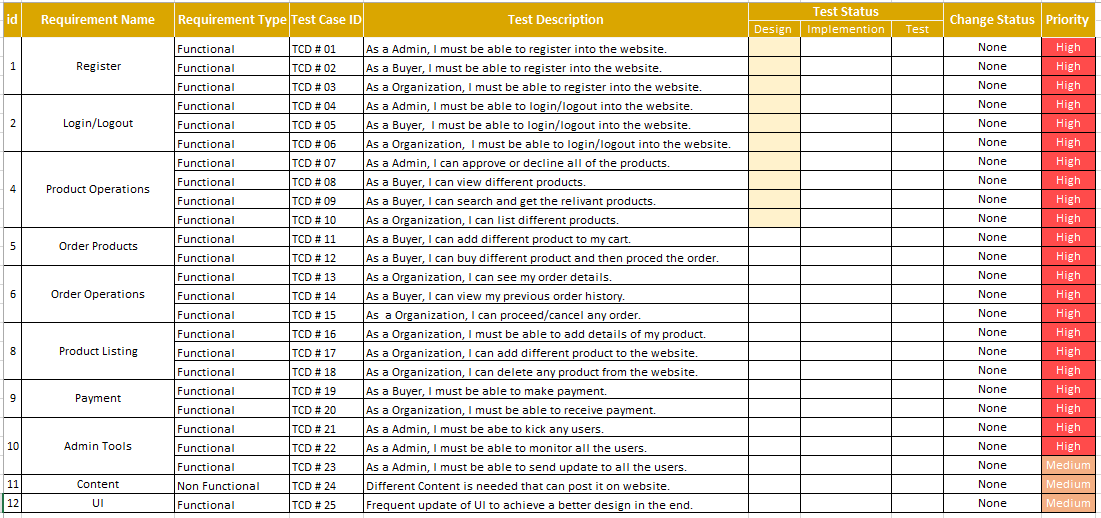
|  |  |
| --- | --- |
| **Footnote** | **Interactions** |
| 0 | On Click -> Case 0:  Open the Ticketing System homepage |
| 1 | On Click -> Case 1:  Open the TS About Us Page |
| 2 | On Click -> Case 2:  Open the TS Ticket Page |
| 3 | On Click -> Case 3:  Open the TS Contact Us Page |

## **4.6 Interaction Viewpoint**

The interaction viewpoint helps us to understand the flow of interaction between the users and the system. The interaction viewpoint helps the developer and stakeholder to better understand the involvement of the different components and how they function together for the completion of certain tasks. Here when the user register with the website and try to log in the request is sent to the login/register system and to the backend and the database from where all the components are verified and that return the result to the backend and then the backend return to the frontend or the user. A similar kind of functionality is true for the organization. The order process also send goes to the admin to the database and the result is set back to the users.

# **5.0 Traceability Matrix**

The traceability matrix helps the implementation team to list all the functional and non-functional requirements of the product on a table that has the different categories, requirement no# and test description, and the test status which is further divided into the design phase, development phase, and the testing phase. The table also contains the changing status which defines whether the project tasks have any changes or not and the priority column the urgencies of the project.



# **6.0 References**

IBM Cloud Education, 2019. *Relational Databases.* [Online]   
Available at: https://www.ibm.com/cloud/learn/relational-databases  
[Accessed August 2022].

kinsta, 2022. *What Is GitHub? A Beginner’s Introduction to GitHub.* [Online]   
Available at: https://kinsta.com/knowledgebase/what-is-github/  
[Accessed August 2022].

OSADCHUK, S., 2021. *Top Backend Technologies in 2022: choose the right one for your solution.* [Online]   
Available at: https://doit.software/blog/backend-technologies  
[Accessed August 2022].

Peterson, R., 2022. *What is PostgreSQL? Introduction, Advantages & Disadvantages.* [Online]   
Available at: https://www.guru99.com/introduction-postgresql.html  
[Accessed August 2022].

Team, I. E., 2021. *What Is Web Architecture? (With Definition and Tips).* [Online]   
Available at: https://www.indeed.com/career-advice/finding-a-job/what-is-web-architecture  
[Accessed August 2022].