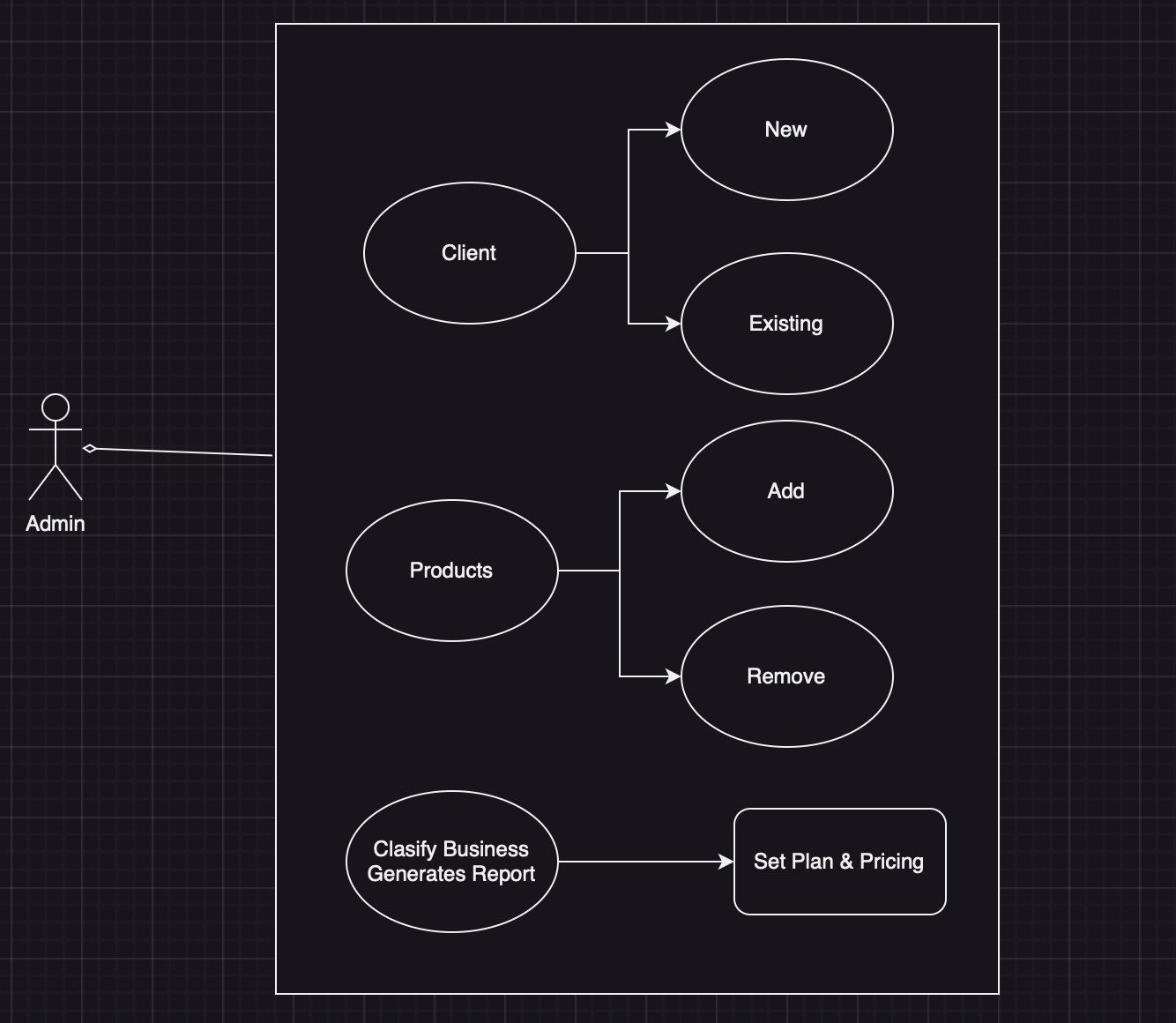
**Background**

In order to design and develop the admin side of the advanced web application for ByteFlow Technologies, it is necessary to follow a systematic approach that involves several key steps. These steps include requirement analysis, identification of use cases, project planning, and database design. Each step in the process is important for ensuring the high quality and robust functionality of the final product. In the following paragraphs, we will examine each of these steps in more detail and explain why they are critical to the success of this project.

**ERD**

The management area of the web application should incorporate a range of features, such as adding new clients, handling client information, introducing new products, modifying product details, establishing plans, pricing, and discounts, categorizing businesses, and producing reports.

****

**Requirement Analysis**

In order to create the best possible admin web application for ByteFlow Technologies, it is crucial to understand the needs and expectations of stakeholders. This can be achieved through meetings and discussions where stakeholders can express their requirements in detail. In addition to general functionalities such as client and product management, it is important to gather requirements for specific aspects such as pricing and plans, business classifications, and report generation. Through this process of gathering requirements and understanding stakeholder needs, the admin web application can be tailored to meet the needs of ByteFlow Technologies**.**

**Use Case**

As we begin to develop the admin web application, it is important to understand the various use cases that the application must satisfy. One of the primary use cases is allowing administrators to create and manage user accounts. Additionally, the application must allow administrators to view and edit user information as well as manage user permissions. Finally, the admin web application should be able to generate reports on system usage and user activity.

**Use Case Description**

Use Case Name: - Project analysis.

Use Case Description: - This use case refers to a scenario where a system or application performs a flexible set of tasks based on a predefined sequence of events.

Flow of Events: -

1. The user logs in the application
2. Then the credentials are checked by the system.
3. Once the user is verified the uses gets access to the system.
4. When a trigger event or request is received, the system will assess the entry condition to determine if it meets the requirements to continue.
5. The system carries out a sequence of tasks or actions, depending on the use case requirements and the supplied input or parameters.
6. The system may communicate with users, retrieve, or modify data, or connect with external systems as required during the process.
7. The system employs a variety of statistical and analytical methods on the dataset, including descriptive statistics, regression analysis, and data visualization techniques.
8. The system produces understanding, trends, and patterns derived from the analytical findings.
9. The user could interact with the system to examine the data, modify the analysis settings, and delve into segments of the dataset.
10. At last, the system terminates the user session and logs him out.

Entry Conditions: - The administrator needs to have a valid user id and password.

Exit Conditions: - The administrator needs to successfully log out from the web application.

**Project Planning**

Developing a software project requires careful planning and management to ensure success. One key aspect of this process is breaking down the project tasks into smaller, manageable units. By estimating the time required for each task and determining dependencies between tasks, a project plan can be created, including a timeline, milestones, and resource allocation. It is important to consider factors such as development methodology to ensure a successful project.

**Database Design**

Once the project tasks have been broken down and a plan has been created, the next step is to analyse the data requirements of the project based on the gathered requirements and use cases. This is critical in ensuring that the project meets the needs of the client and any stakeholders. A logical data model must then be designed that represents the necessary entities, relationships, and attributes required for the admin web application. This step is essential in ensuring the smooth functioning and success of the software project.

In database management, designing a logical data model is a crucial step towards ensuring data accuracy and consistency. Normalizing the model helps eliminate data redundancy and maintain data integrity by reducing data manipulation anomalies. However, the logical data model is just a representation of the system's data requirements and does not consider performance or scalability. To develop a physical database schema, we need to translate the logical model into an executable data structure that meets the requirements of the system and can scale to handle increasing amounts of data.

**Class Description**

1. Customer: -

Overview: Denotes a customer with attributes such as ID, name, email, and contact number. –

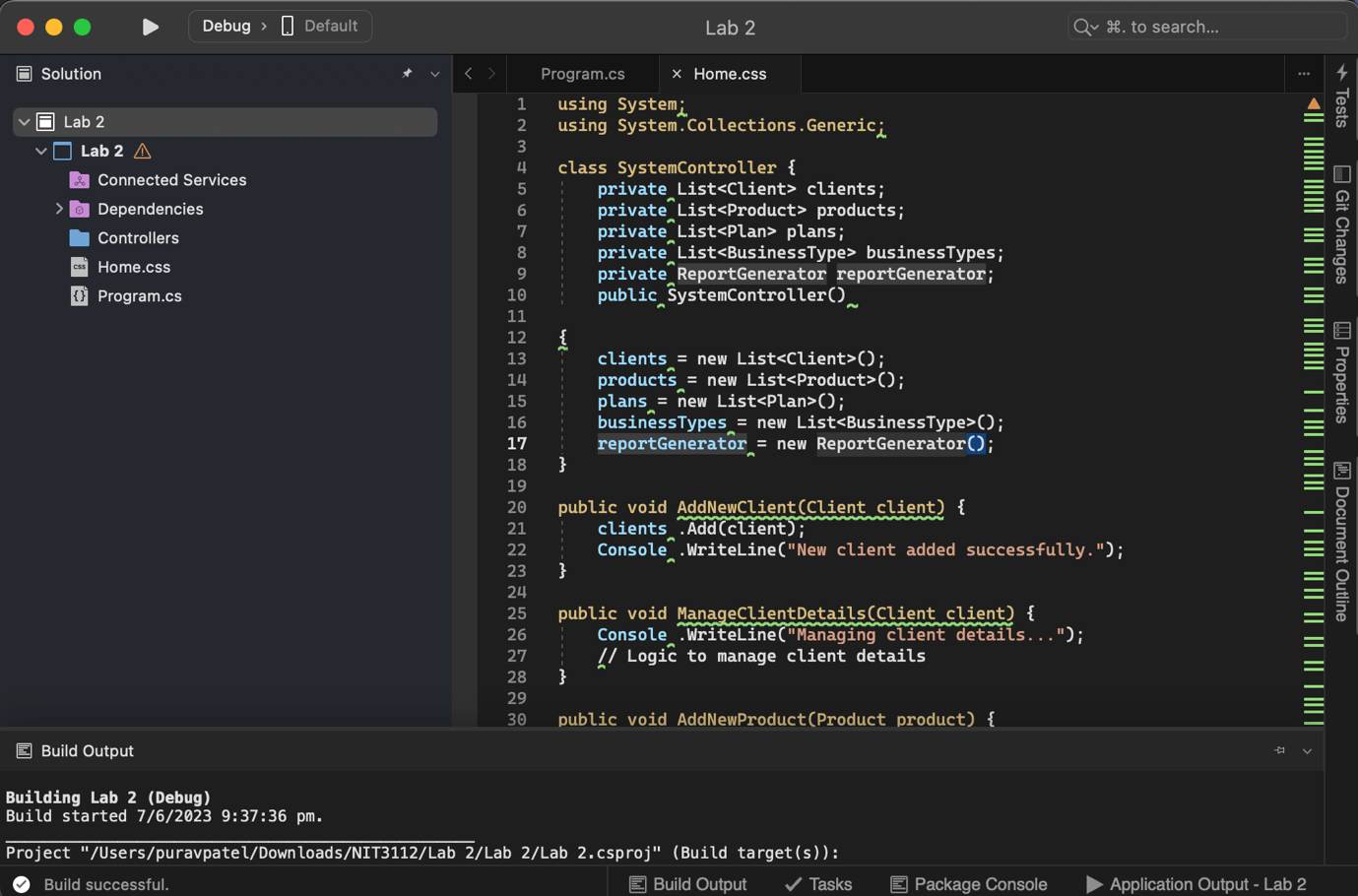
Attributes:

* Id: An integer indicating the customer's ID.
* Name: A string denoting the customer's name.
* Email: A string signifying the customer's email address.
* Phone: A string illustrating the customer's phone number.

1. Product: - Description: Signifies a product with features like ID, name, and price.

Properties:

* Id: An integer symbolizing the product's ID.
* Name: A string denoting the product's name.
* Price: A decimal value indicating the product's price.

****