Chapter 3. Design

Introduction  
Design is the third phase that comes after analysis phase in system development methodology. It is the process which consists of different perspective which drives development based on your specific needs. All the key structures, achievement measures and structure are discussed in this phase. The design process varies between different projects and design fields.

The importance of design for our project are as follows:

* Design makes it easier to implement
* It helps users to visualize the system and its functions.
* It promotes the use of different diagram which includes classes and object that helps to further ease the programming process.

For this design process I have included the following 4 models which are listed below:

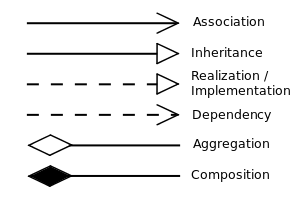
* Structural model
* Behaviour Model
* Database Modelling
* Architectural Model
* Prototyping

Structural modelling:

Structural modelling is an UML model that describes the static structure i.e. relationships and properties. It is used to analyse the structural relationship between measured variables and latent constructs. It estimates the multiple and interrelated dependence in a single analysis.

Class diagram:  
Class diagram is a static diagram that visualizes and describes the different aspects of the system. They are widely used as it is the main building blocks of an application. It describes various kinds of objects and its static relationship.   
Purpose of class diagram:

1. It is the base for deployment of the application.
2. It analyses and design the static view of an application i.e. (it contains classes and object, inheritance and interface).

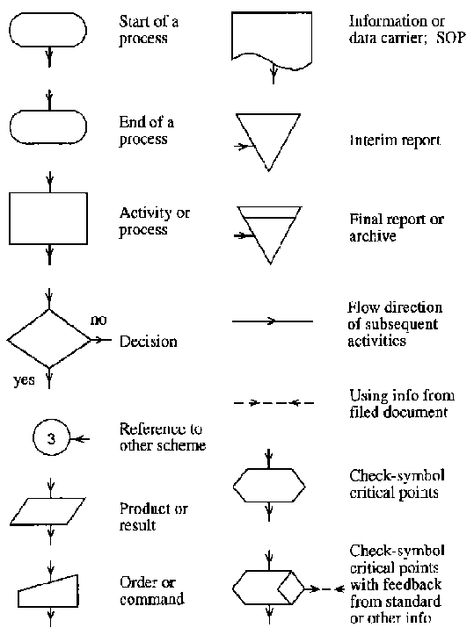


Class diagram of the project is as follows:

Data Flow Diagram

A **data-flow diagram** (DFD) is a way of representing a flow of a data of a process or a system. It is the diagram that describes the boundaries and scope of the system.

Notations



The data flow diagram for this project is as follows:

Behaviour modelling

UML behaviour diagrams is the diagram that visualizes, specifies, construct, and document the dynamic aspects of a system. The behaviour diagrams are categorized as follows:

* Activity diagram
* Sequence diagram

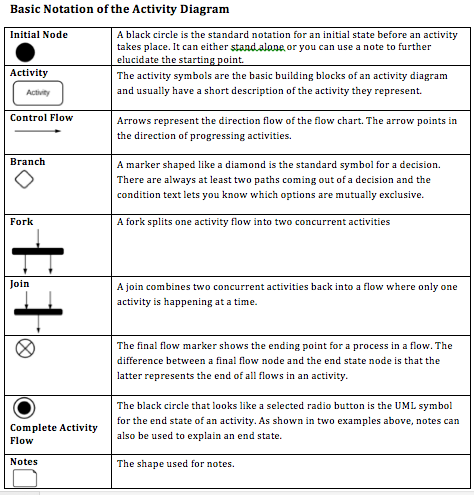
Activity diagram:

Activity Diagrams is the diagram that is used to describe how activities are coordinated to provide a service which can be at different levels of concept. It is a flowchart that represents the flow of activity from one activity to another activity. The activity can be described as an operation of the system.

The purpose of an activity diagram are as follows −

* Draw the activity flow of a system.
* Describe the sequence from one activity to another.

Notations of Activity diagram are as follows:



Activity diagram of my project are as follows:

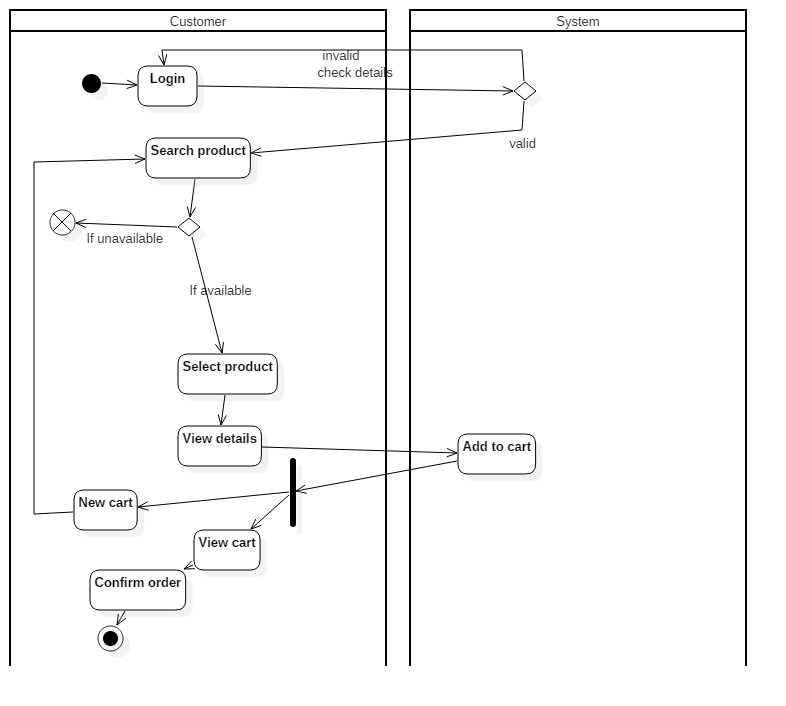


Figure 1 Activity diagram for purchase

In this activity diagram shown above, the customer first logs into the system. The system then checks if the log in information provided by the user is valid or not. If the information is valid then the user will get the access to the products list. And if the information provided is not valid then the user must try again to log into the system. The product consists of the items that are to be sold in the website. The customer then searches for the product and when they get the product, they will apply for purchasing process. The product that is to be purchased will be placed in the cart by the system. And then the user will confirm order for the product.

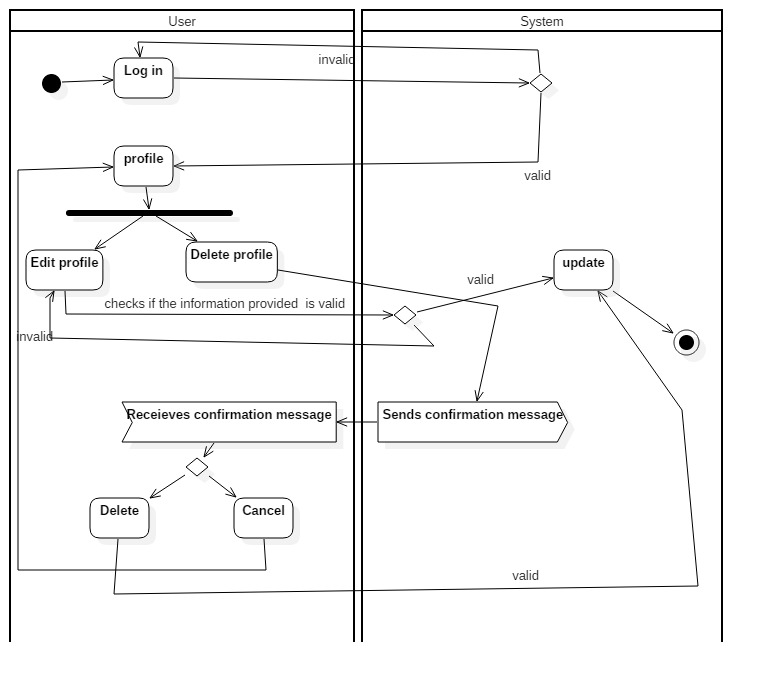


Figure 2 Activity diagram for editing profile

In this activity diagram shown above, the user first logs into the system. The system then checks if the log in information provided by the user is valid or not. If the information is valid then the user will get the access to the products list. And if the information provided is not valid then the user must try again to log into the system. Then the user goes to his/hr profile to manipulate their profile. The user can edit or delete the profile. The system checks if the information provided is validate to the system requirements. If it meets the requirement the change will be updated.

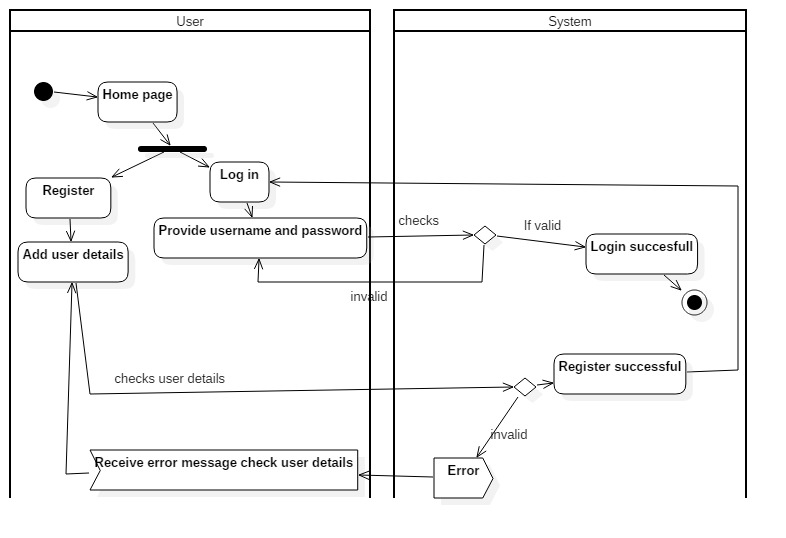


Figure 3 Activity diagram for registration and log in

In the activity diagram shown above, the user must choose between registering or logging into the system. When the logs into the system he must verify the information that they have provided to the system. If the information provided by the user meets the requirement of the system, then it will be validated and then the user will get access to the system. If the user wants to register their account into the system, they must provide the necessary information asked by the system. After getting accepted by the system the registration process is complete and now the user can successfully log in to the system.

Sequence diagram:

Sequence Diagrams are interaction diagrams that shows how the operations of the system are carried out. It captures the interaction between objects in the context of a collaboration. They are time focused and shows the order of the interaction visually by using the vertical axis of the diagram to represent time what messages are sent and when.

The justification of using sequence diagram in my project are as follows.

* It creates a common ground to help the developers and business analysts understand the system.
* It helps to visualize what will happen during the execution of each use cases created in the project.

**Notations**

|  |  |  |  |
| --- | --- | --- | --- |
| S.no | Notations Used | Notation Name | Description |
|  |  | Life Line | Represents the objects that participate in an interaction. |
|  |  | Message | Represents the flow or function or message between different objects. |
|  |  | Self-Message | Represents a message sent by an object within itself. |
|  |  | Alt. Combined Fragment | Represent multiple condition scenarios of an action |
|  |  | Actor | The entity that initiate the functionality of the system. |
|  |  | Focus of control | Represents the period of time during which an object performs an action |

The sequence diagram for my project are as follows:

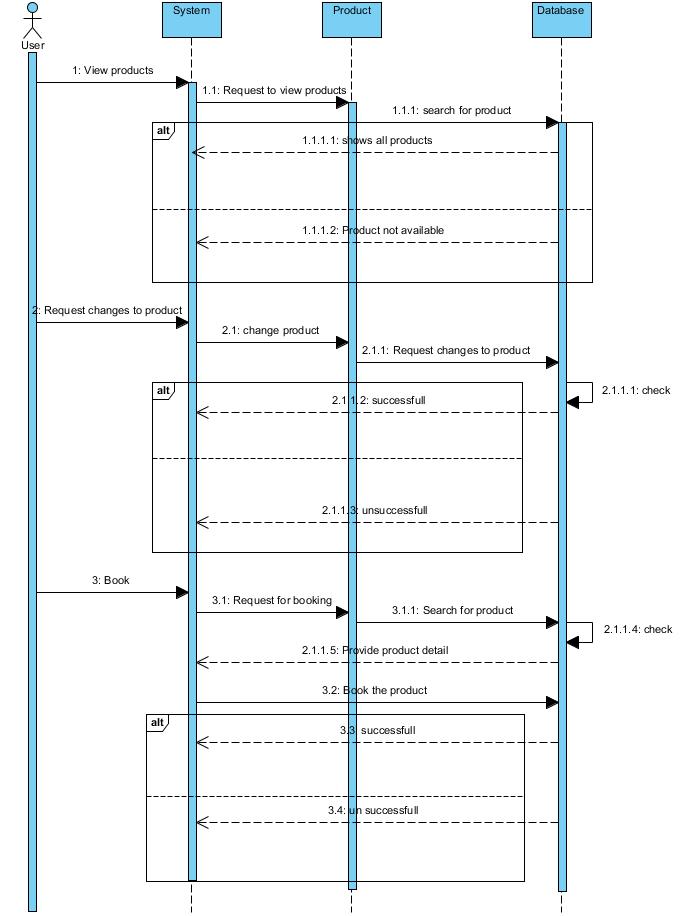


Figure 4 Purchase sequence diagram

In the above sequence diagram, the user will request for purchasing the item to the system. The system will then send the request to the database and will look for the product asked by the user. If it is successful to find out the product then they will send the successful message back to the user. After that the user can place the order for that product and the purchase process will be completed.

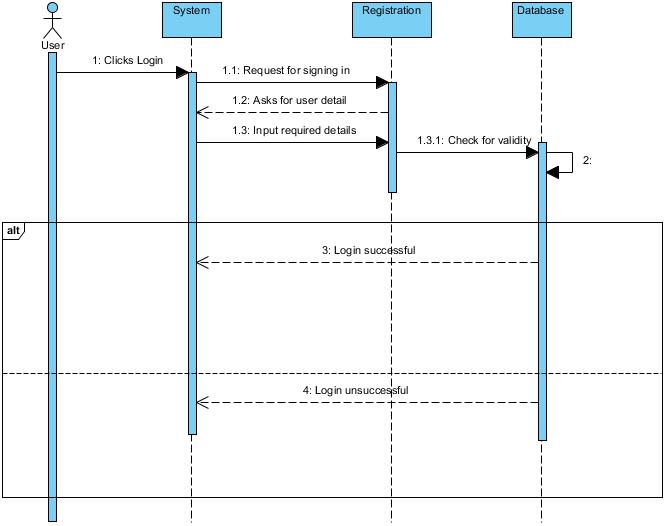


Figure 5 Login Sequence diagram

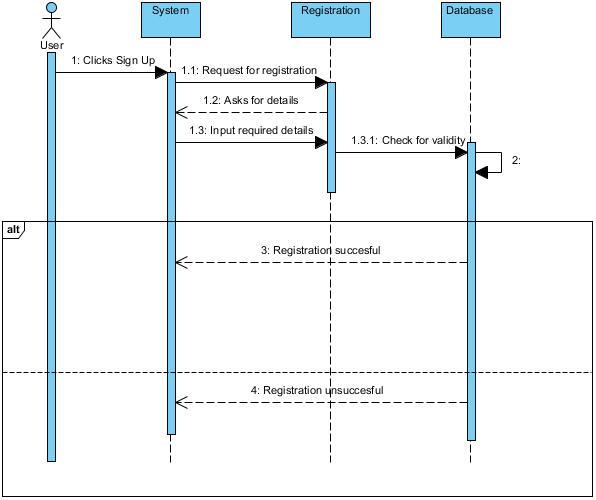


Figure 6 Sign up Sequence diagram

The user can login or sign into the system by requesting for it to the system. The page will send the request to the system to fill the details. The user will then fill the form provided by the system and will review it. When the review is completed the user will send the user details to the database and will save the user. Now the user will be allowed to sing in to the system when they want to.

Data modelling:

Data modelling is the process of creating a data model for the data to be stored in a Database. This data model is a conceptual representation of

* Data objects
* The associations between different data objects

Importance of data modelling are as follows:

* A data model helps design the database at the conceptual, physical and logical levels.
* It helps to identify missing and redundant data.

Data dictionary

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data type | Constraints | Nullable |
| UserID | int | PK | No |
| Username | varchar |  | Yes |
| Email | Varchar |  | Yes |
| Address | Varchar |  | Yes |
| Phone | Int |  | Yes |
| Password | Varchar |  | Yes |

Product

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data type | Constraints | Nullable |
| ProductID | Int | PK | No |
| ProductName | Varchar |  | Yes |
| Product\_Price | Integer |  | Yes |
| Product\_Quantity | Integer |  | Yes |

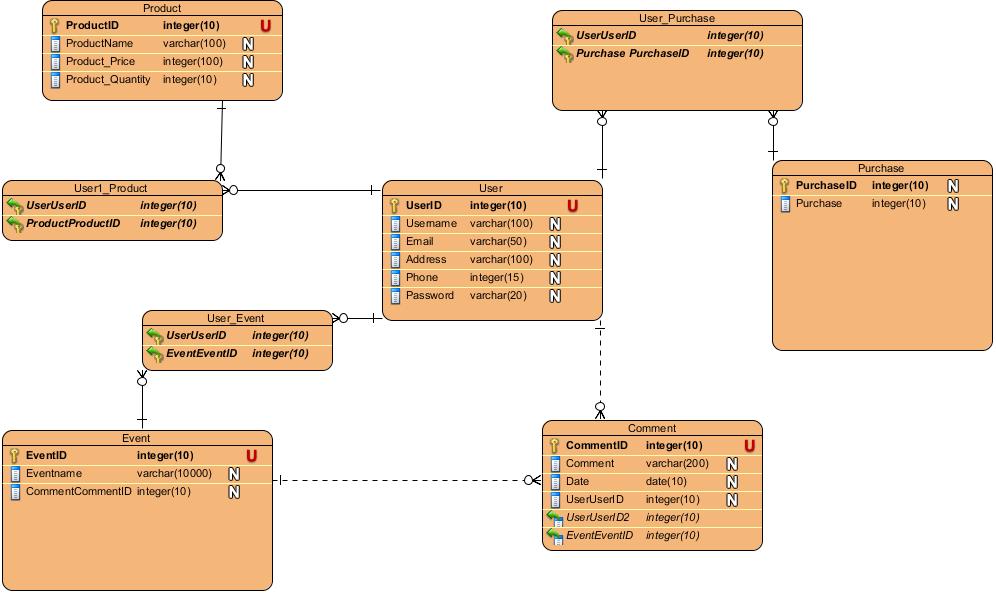
Purchase

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data type | Constraints | Nullable |
| PurchaseID |  |  |  |
|  |  |  |  |
|  |  |  |  |

Event

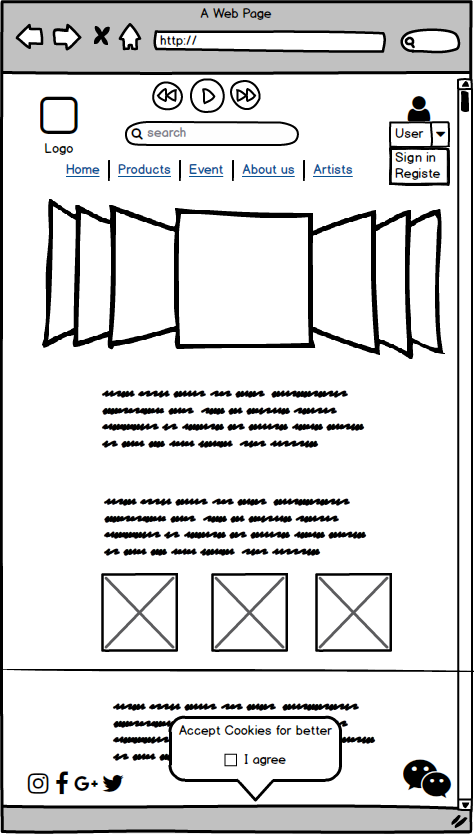
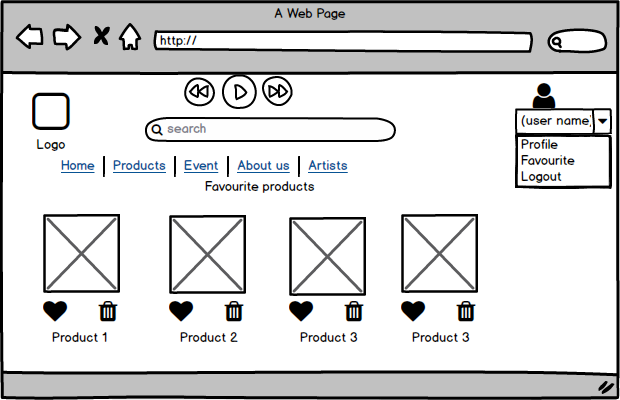
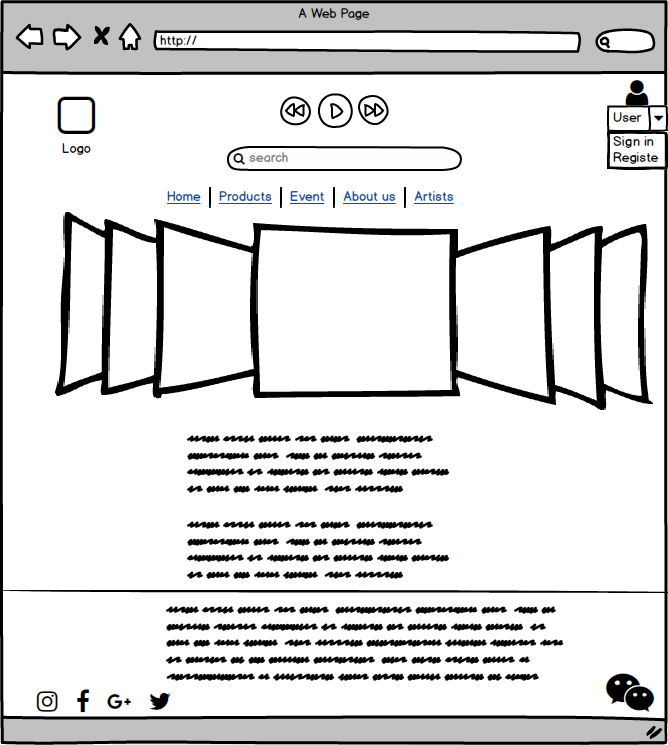
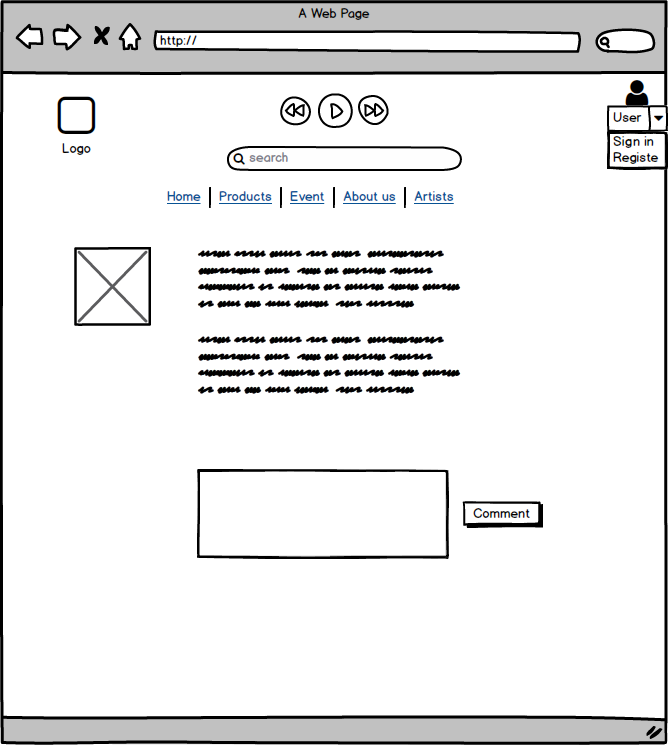
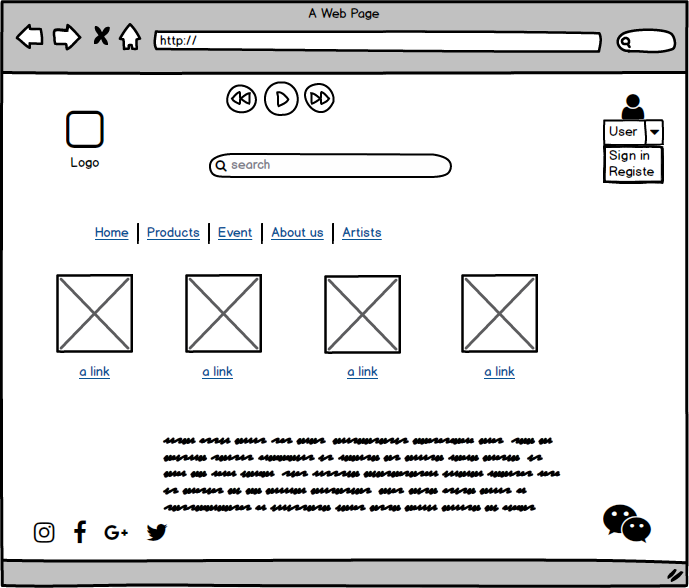
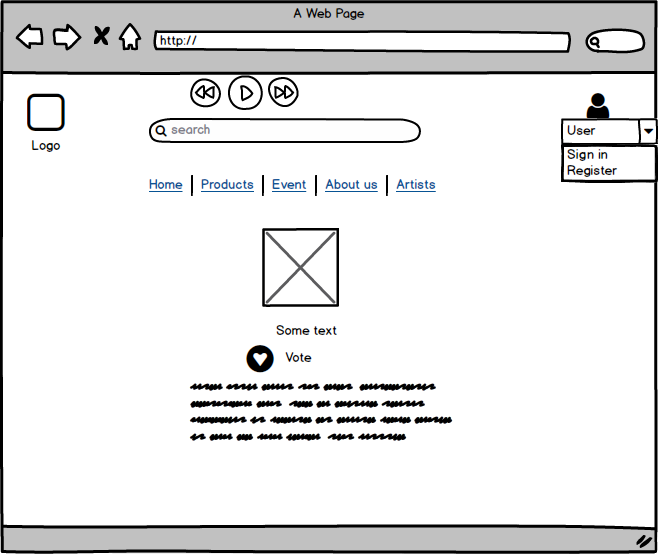
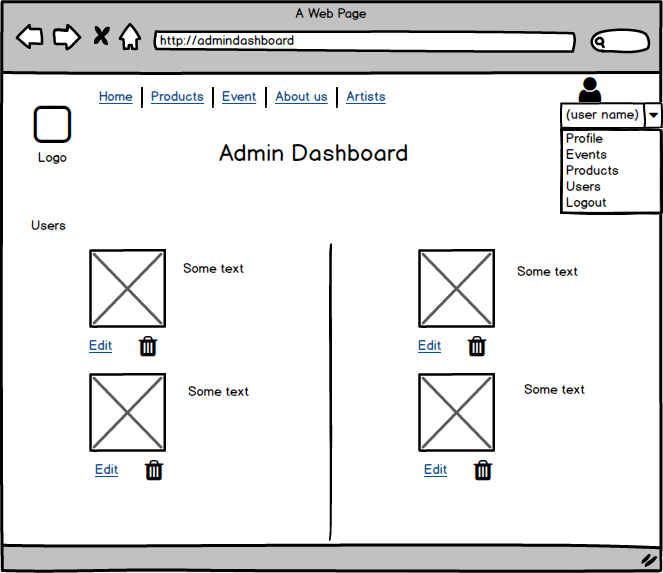
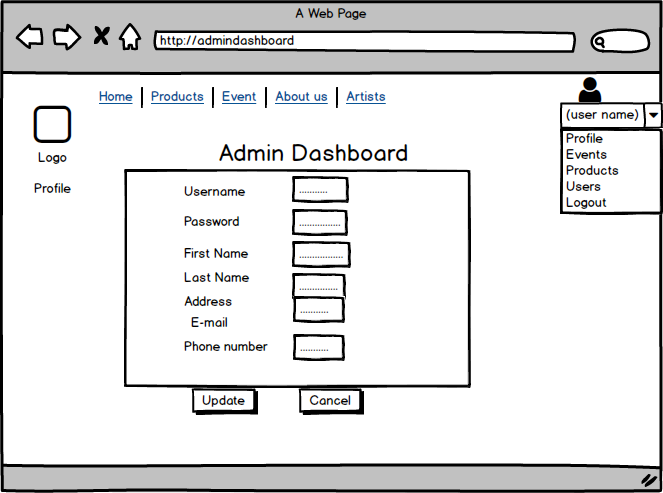
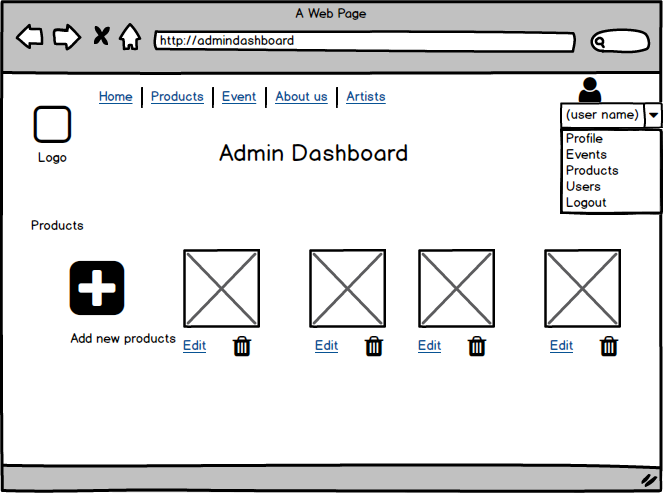
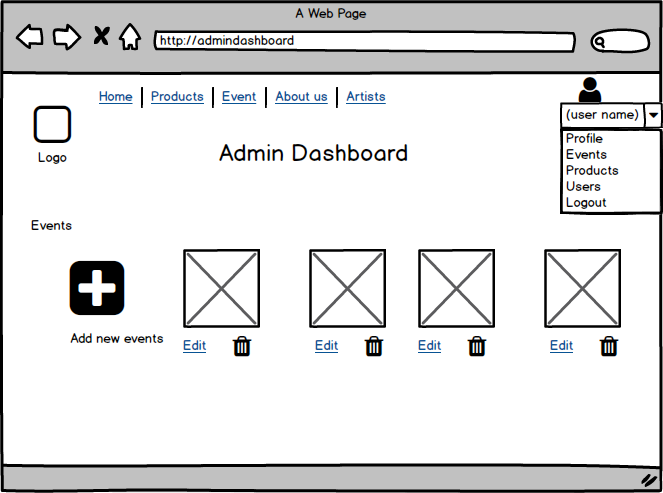
|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data type | Constraints | Nullable |
| EventID | int | PK | No |
| EventName | varchar |  | Yes |
|  |  |  |  |

E.R. Diagram:



Prototype Design

A prototype is an original model, form or an instance that serves as a basis for other processes. In software technology, the term prototype is a working example through which a new model or a new version of an existing product can be derived.

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated