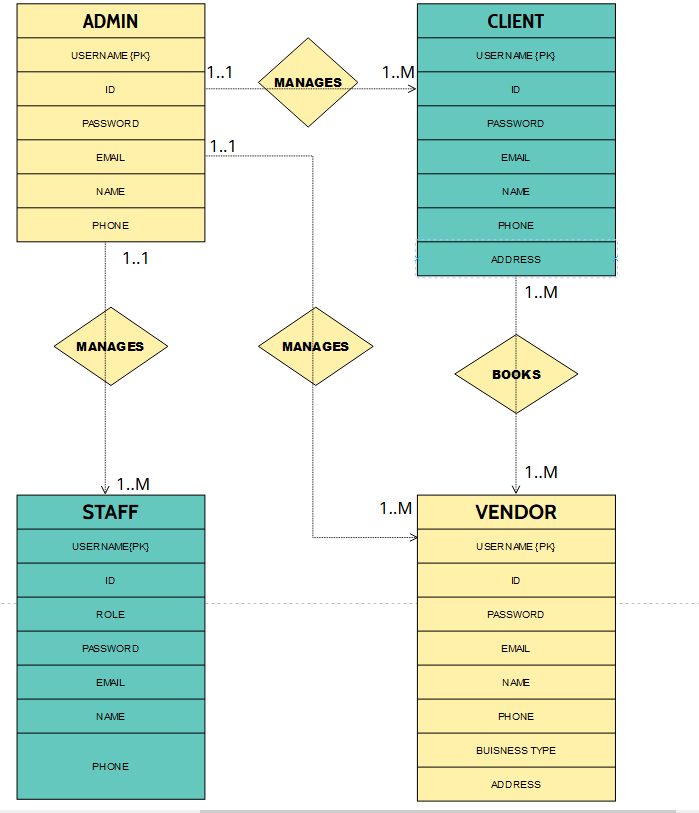
**Problem Statement:** Inefficient Wedding Event Planning Process for User  
  
**User Requirements:**

1. **User Registration:** Users will be able to create an account to access wedding event planning platform. They should be able to provide event details such as guest list and can see total budget expenditure, rest all the required information will be shared with Admin which will update accordingly.
2. **Vendor Management:** Users will be able to select vendor of their choice based on their ratings.
3. **Budget Management:** Users should be able to track their wedding budget.
4. **Guest Management:** Users should be able to manage the guest list for their wedding events. They should be able to track RSVPs.
5. **Communication:** Users will be able to communicate with the admin through the platform.

**SYSTEM SPECIFICATION (UPDATED)**

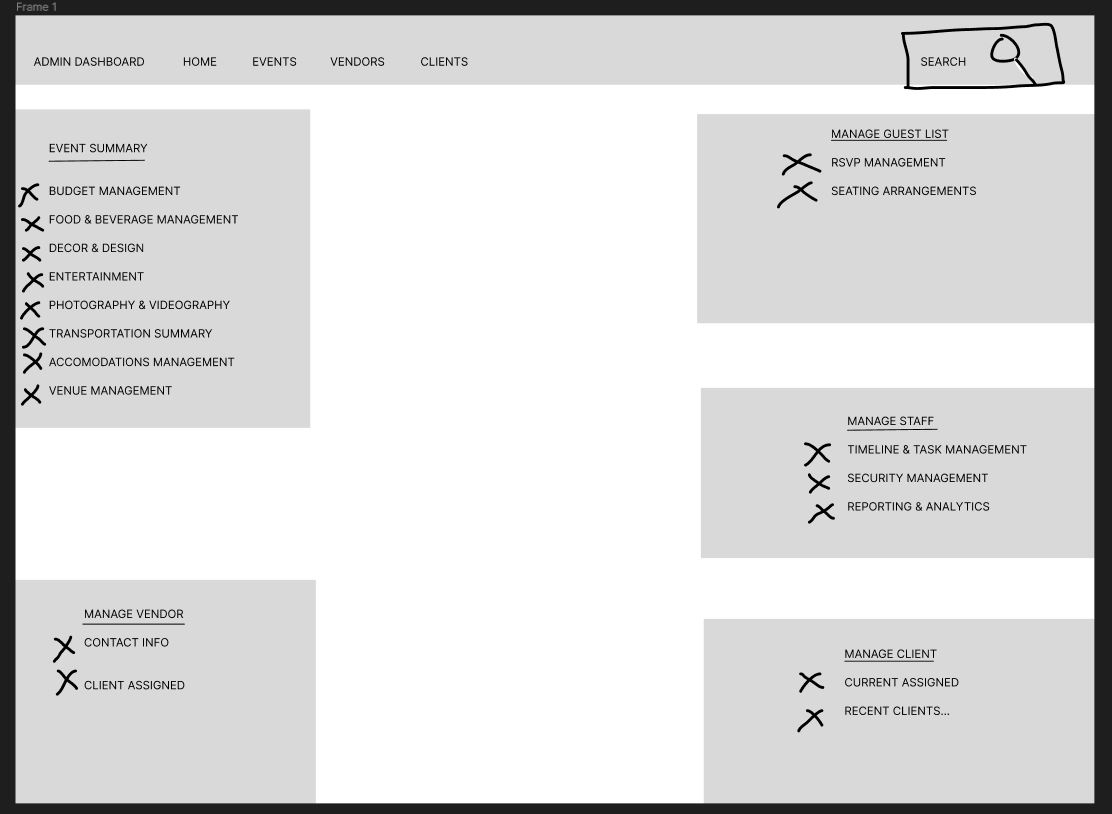
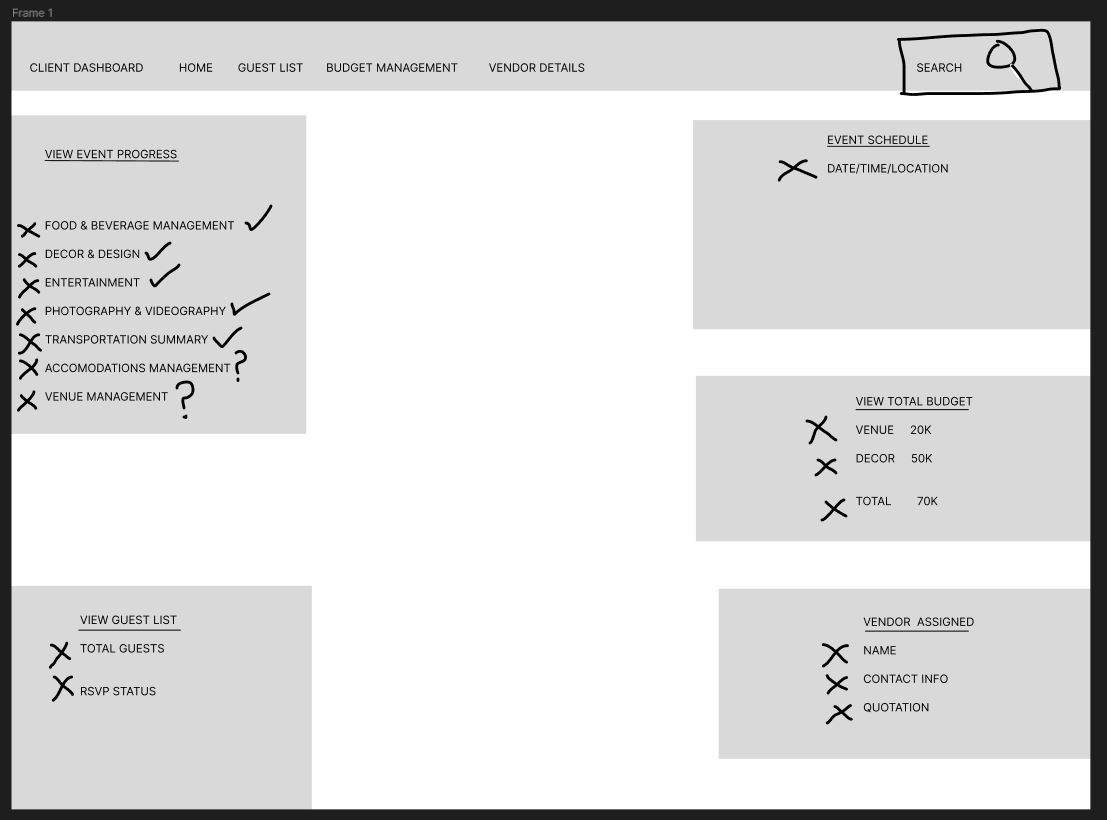
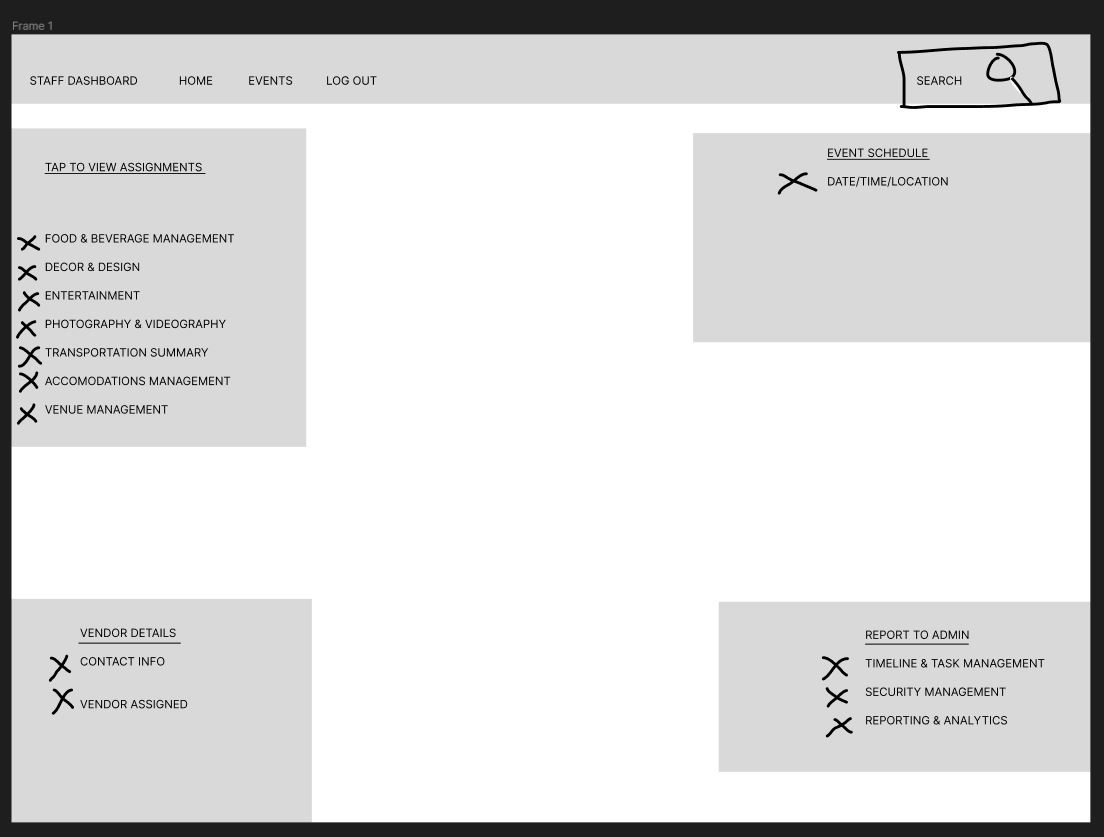
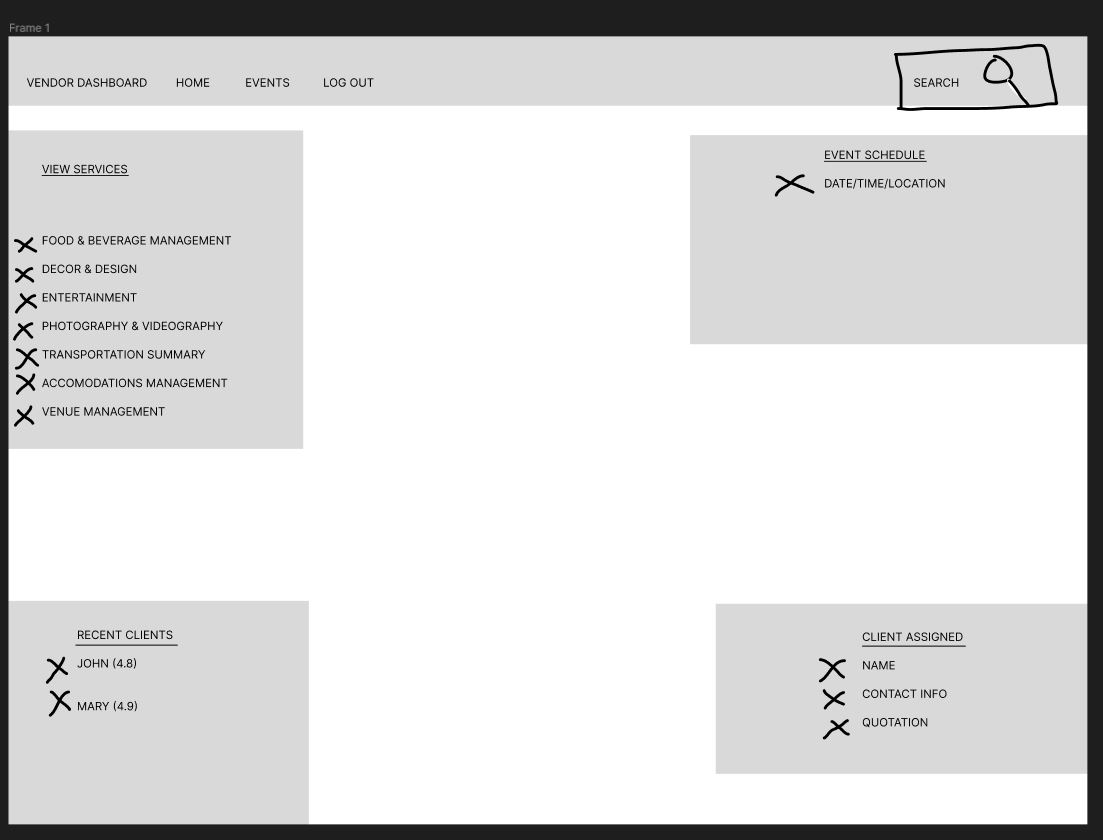
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data | Admin | Client | Vendor | Staff |
| Event Planning | ✔️ | ✔️ |  |  |
| Budget Management | ✔️ | ✔️ |  |  |
| Vendor Management | ✔️ |  |  |  |
| Guest List Management | ✔️ | ✔️ |  |  |
| RSVP Management | ✔️ | ✔️ |  |  |
| Seating Arrangements | ✔️ |  |  |  |
| Venue Management | ✔️ |  | ✔️ | ✔️ |
| Food and Beverage Management | ✔️ |  | ✔️ | ✔️ |
| Decor and Design Management | ✔️ |  | ✔️ | ✔️ |
| Entertainment Management | ✔️ |  | ✔️ | ✔️ |
| Photography and Videography Management | ✔️ |  | ✔️ | ✔️ |
| Transportation Management | ✔️ |  | ✔️ | ✔️ |
| Accommodations Management | ✔️ |  |  |  |
| Timeline and Task Management | ✔️ |  | ✔️ | ✔️ |
| Reporting and Analytics | ✔️ |  | ✔️ |  |
| User Management | ✔️ |  |  |  |
| Security Management | ✔️ |  | ✔️ | ✔️ |

The **administrator** has complete control over the website and can modify any information.   
  
**Users** have limited access and can only view event progress, manage guests, manage the budget, and select their preferred vendor. For any other updates, users must contact the administrator.   
  
**Vendors** can only access and manage their own relevant information, such as planning their performance for the event.   
  
**Staff** members will be assigned to assist with preparations, either individually or in collaboration with vendors.  
**ERD DIAGRAM:**  
  
  
  
1) Admin and Client: One-to-Many Relationship (Admin manages 1 or many clients, but a client is managed by 1 and only one admin)

2) Client and Vendor: One-to-Many Relationship (Client can book 1 or many vendors, and a vendor can be booked by 1 or many clients)

3) Admin and Vendor: One-to-Many Relationship (Admin manages 1 or many vendors, but a vendor is managed by one and only one admin)

4) Admin and Staff: One-to-Many Relationship (Admin manages 1 or many staff members, and staff members are managed by one and only one admin)

**WIREFRAMES (UPDATED)**Wireframes were created according to the system specification table.  
  
**ADMIN DASHBOARD**  
  
  
  
  
  
  
  
  
  
  
  
  
**USER WIREFRAME**  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
**STAFF WIREFRAME**  
 **VENDOR WIREFRAME**

**DDL  
  
ADMIN TABLE**CREATE TABLE ADMIN (

A\_USERNAME VARCHAR(64),

A\_ID VARCHAR(64),

A\_PASSWORD VARCHAR(64),

A\_EMAIL VARCHAR(64),

A\_FIRST\_NAME VARCHAR(64),

A\_LAST\_NAME VARCHAR(64),

PHONE VARCHAR(64),

PRIMARY KEY (A\_USERNAME)

);

**USER TABLE**

CREATE TABLE CLIENT (

C\_USERNAME VARCHAR(64),

C\_ID VARCHAR(64),

C\_PASSWORD VARCHAR(64),

C\_EMAIL VARCHAR(64),

C\_FIRST\_NAME VARCHAR(64),

C\_LAST\_NAME VARCHAR(64),

PHONE VARCHAR(64),

ADDRESS VARCHAR(128),

ADMIN\_USERNAME VARCHAR(64),

PRIMARY KEY (C\_USERNAME),

FOREIGN KEY (ADMIN\_USERNAME) REFERENCES ADMIN(A\_USERNAME)

);

**VENDOR TABLE**

CREATE TABLE VENDOR (

V\_USERNAME VARCHAR(64),

V\_ID VARCHAR(64),

V\_PASSWORD VARCHAR(64),

V\_EMAIL VARCHAR(64),

V\_FIRST\_NAME VARCHAR(64),

V\_LAST\_NAME VARCHAR(64),

PHONE VARCHAR(64),

ADDRESS VARCHAR(128),

BUSINESS\_TYPE VARCHAR(64),

ADMIN\_USERNAME VARCHAR(64),

PRIMARY KEY (V\_USERNAME),

FOREIGN KEY (ADMIN\_USERNAME) REFERENCES ADMIN(A\_USERNAME)

);

**STAFF TABLE**

CREATE TABLE STAFF (

S\_USERNAME VARCHAR(64),

S\_ID VARCHAR(64),

S\_PASSWORD VARCHAR(64),

S\_EMAIL VARCHAR(64),

S\_FIRST\_NAME VARCHAR(64),

S\_LAST\_NAME VARCHAR(64),

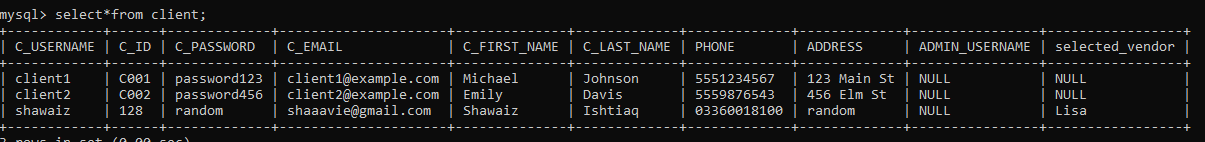
PHONE VARCHAR(64),

ROLE VARCHAR(32),

ADMIN\_USERNAME VARCHAR(64),

PRIMARY KEY (S\_USERNAME),

FOREIGN KEY (ADMIN\_USERNAME) REFERENCES ADMIN(A\_USERNAME)

);  
  
 **NORMALIZATION**:   
  
LETS TAKE EXAMPLE OF CLIENT TABLE  
  
  
IT FOLLOWS **1NF** AS THERE ARE NO MULTI-VALUED ATTRIBUTES.  
  
The CLIENT table satisfies **2NF** as it is in 1NF, and all non-key attributes (C\_ID, C\_PASSWORD, C\_EMAIL, C\_FIRST\_NAME, C\_LAST\_NAME, PHONE, ADDRESS) depend fully on the primary key (C\_USERNAME). Additionally, the ADMIN\_USERNAME attribute is a foreign key referencing the A\_USERNAME column in the ADMIN table, which establishes a relationship between the two tables.  
  
FOR 3NF: The CLIENT table adheres to **3NF** as it is already in 2NF, and there are no non-key attributes that transitively depend on the primary key (C\_USERNAME). The ADMIN\_USERNAME attribute, although a foreign key, does not introduce any transitive dependencies as it directly depends on the primary key.​  