### **Blueprint Document**

### 

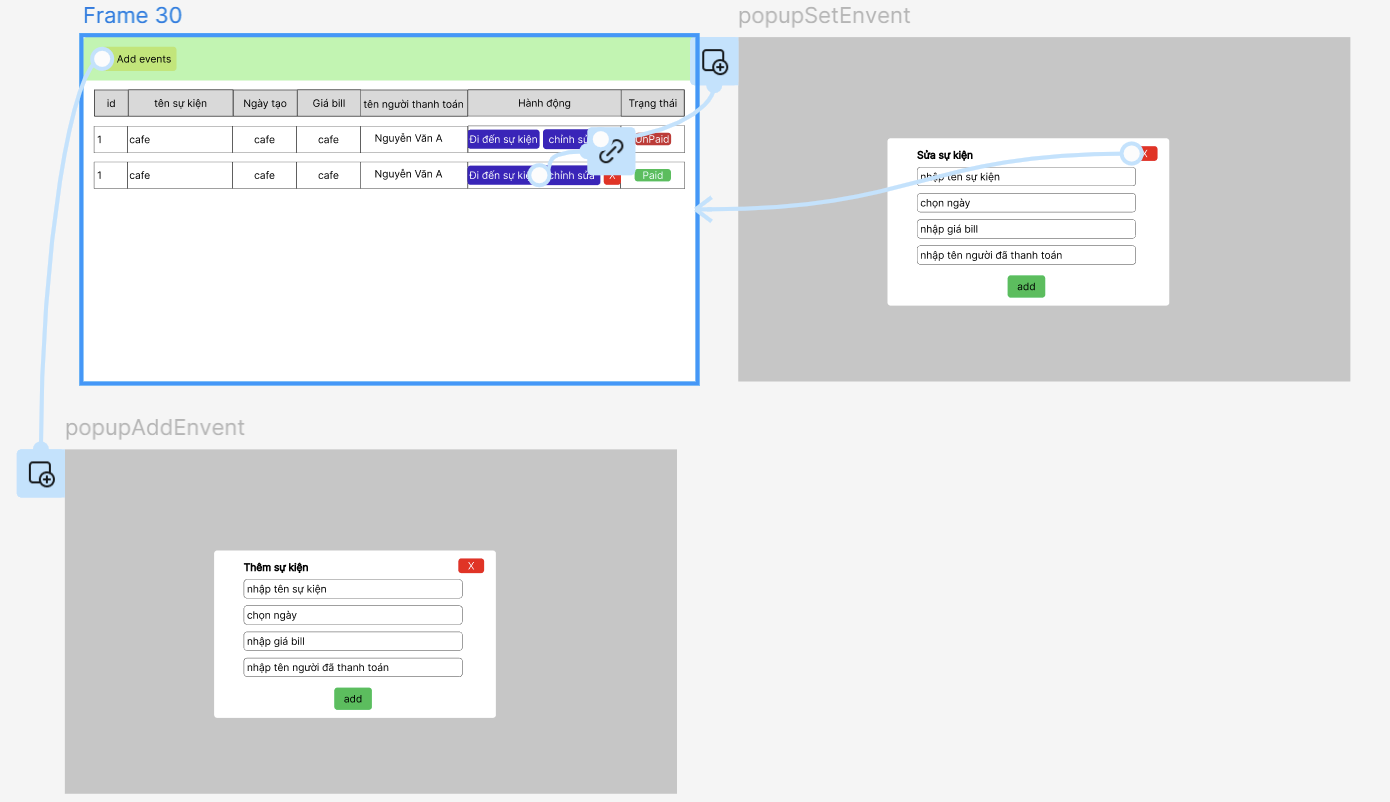
### **Tiếp nhận bài toán**

Người dùng cần một công cụ để chia tiền cho những người tham gia một sự kiện như buổi cafe, buổi nhậu... Hệ thống cần tự động tính toán số tiền mỗi người phải trả và gửi thông tin qua kênh chat của công ty.

### **Lên tính năng**

#### **Trang chủ(/home):**

1. **Tạo/thêm sự kiện:**
   * Tên sự kiện
   * Tổng bill
   * Ngày tạo



#### **2. API & Job xử lý**

**2.1** **Với trang HomeEvent:**

| **#** | **API** | **Method** | **Endpoint** |
| --- | --- | --- | --- |

| 1 | Thêm sự kiện | POST | /api/events |
| --- | --- | --- | --- |

| 2 | Lấy danh sách sự kiện | GET | /api/events |
| --- | --- | --- | --- |

| 3 | Lấy thông tin sự kiện | GET | /api/events/{id} |
| --- | --- | --- | --- |

| 4 | Ẩn sự kiện (Xóa mềm) | PUT | /api/events/{id}/hide |
| --- | --- | --- | --- |

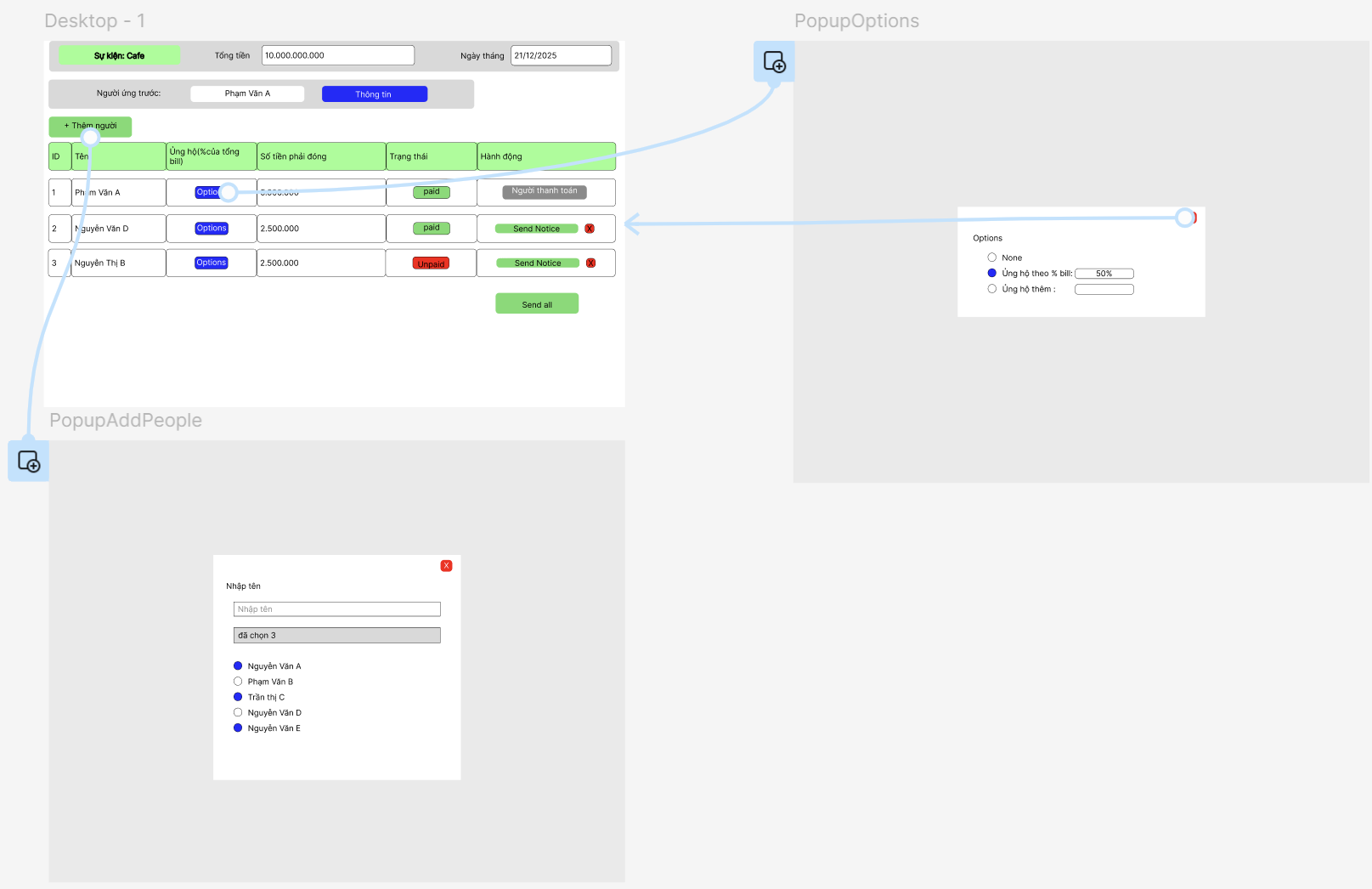
5 chính sửa sự kiện PUT /api/events/{id}/adjust

|  |  |  |  |
| --- | --- | --- | --- |

—------------------------------------------------------------------------------------------------------------------------

**Trang sự kiện (home/sự kiện name/)**

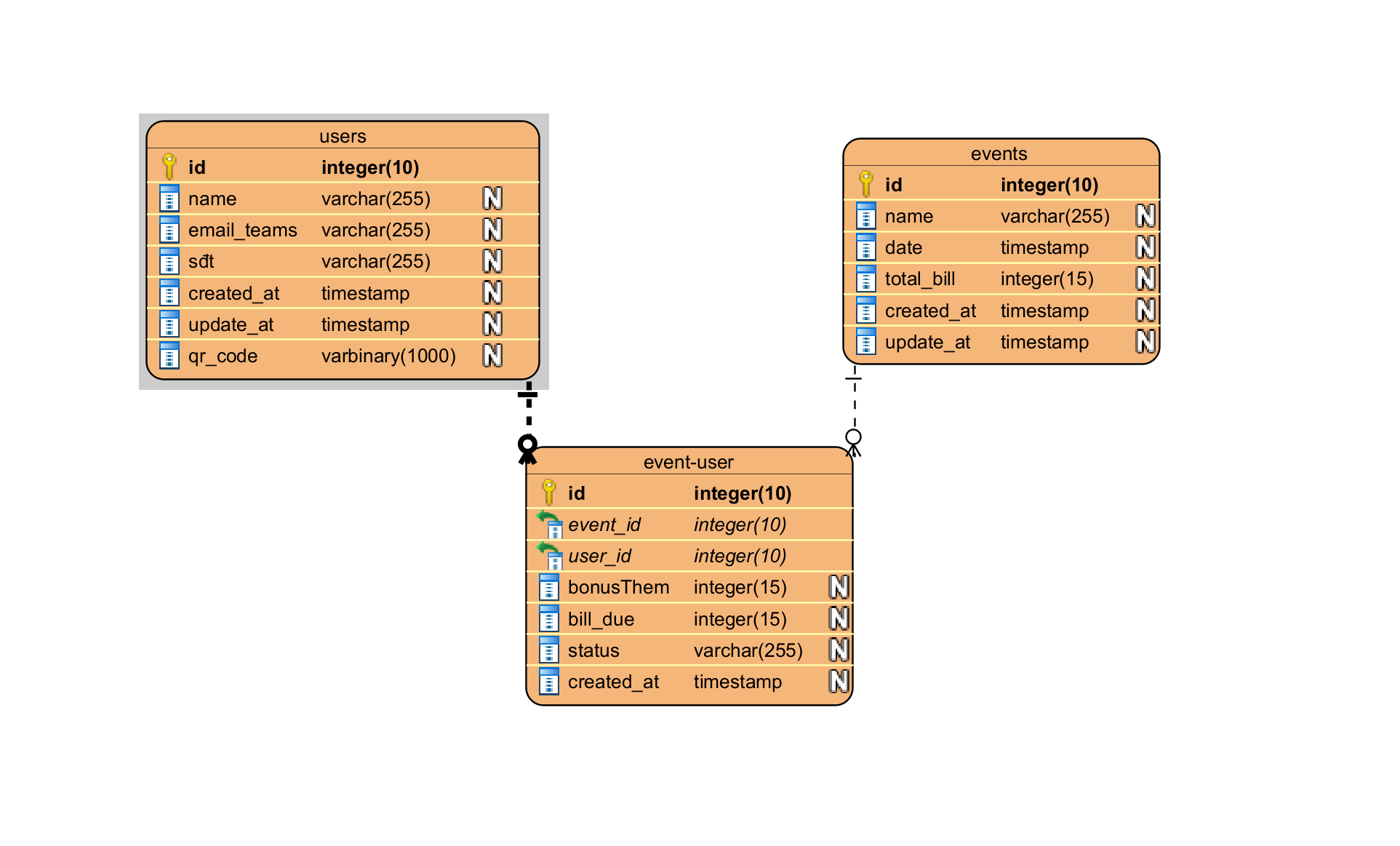
* + Tạo thêm người đã trả + số tiền lấy từ giá bill trang home
  + Tạo/thêm người:
    - Tên
    - Sđt
    - microsoft teams
    - Lựa chọn Bonus( theo % hoặc số tiền)
    - số tiền phải đóng
  + Tính tiền
    - cập nhập số tiền phải đóng(người bonus tính thì giá tiền theo %bonus, người bình thường thì trừ người bonus ra còn lại chia đều).
  + Gửi thông báo
    - gửi đến microsoft team của mỗi người



http://localhost:5000/events/1/users/2

| **#** | **API** | **Method** | **Endpoint** |
| --- | --- | --- | --- |
|  |  |  |  |
| 2 | Lấy danh sách người | GET | /api/people |
| 3 | Thêm người vào sự kiện | POST | /api/event/{event\_id}/add\_people |
| 4 | Cập nhật Bonus | PUT | /api/event/{event\_id}/people/{person\_id}/bonus/options |
|  |  |  |  |
| 6 | Gửi thông báo cá nhân | POST | /api/event/{event\_id}/notify/{person\_id} |
| 7 | Gửi thông báo tất cả | POST | /api/event/{event\_id}/notify\_all |
| 8 | Xóa mềm người khỏi sự kiện | DELETE | /api/event/{event\_id}/people/{person\_id} |
| 9 | Cập nhật trạng thái thanh toán | PUT | /api/event/{event\_id}/people/{person\_id}/status |
|  |  |  |  |

**—------------------------------------------------------------------------------------------------------------------------**

**Mô hình cơ sở dữ liệu:**

### 

#### **3. Công nghệ sử dụng**

* **Backend:** Flask
* **Frontend:** React
* **Database:** Postgres
* **Payment API:** None
* Notification (Team)

—------------------------------------------------------------------------------------------------------------------------

**Danh sách công việc:**

+ Code api :

Bảo mật

| **Danh sách công việc** | **Trạng thái** |
| --- | --- |
| Tiếp nhận bài toán | processing testing done |
| Lên tính năng | processing testing done |
| Draf giao diện | processing testing done |
| List API (doc) | processing testing done |
| Code api cho events: (exel)   1. **Lấy thông tin toàn bộ sự kiện**: GET /events 2. **Thêm sự kiện**: POST /events 3. **Lấy thông tin sự kiện cụ thể**: GET /events/<int:event\_id> 4. **Xóa sự kiện**: DELETE /events/<int:event\_id> 5. **Chỉnh sửa sự kiện**: PUT /events/<int:event\_id> 6. **Lấy danh sách người dùng của một sự kiện**: GET /events/<int:event\_id>/users 7. **Thêm người dùng vào sự kiện**: POST /events/<int:event\_id>/users 8. **Xóa người dùng khỏi sự kiện**: DELETE /events/<int:event\_id>/users/<int:user\_id> | processing testing done  processing testing done  processing testing done  processing testing done  processing testing done  processing testing done  processing testing done  processing testing done  processing testing done |
| Code api cho users:   1. **Lấy thông tin toàn bộ người dùng**: GET /users 2. **Lấy thông tin chi tiết của một người dùng**: GET /users/<int:user\_id> 3. **Thêm người dùng**: POST /users 4. **Xóa người dùng**: DELETE /users/<int:user\_id> 5. **Chỉnh sửa người dùng**: PUT /users/<int:user\_id> | processing testing done  processing testing done  processing testing done  processing testing done  processing testing done  processing testing done |

Link git: <https://github.com/DANGPP/Thuc-tap.git>

chuyển qua exel danh sách task functions - sub task(đầu api)-   
có ô trnajg thái

tìm hiểu về Celery + bot teams:

Bot Framework Service (BFS):  
**Bot Framework Service (BFS)**

* + Đây là một dịch vụ trung gian của Microsoft giúp kết nối bot với nhiều kênh khác nhau.
  + Nó đảm nhận việc xử lý tin nhắn, gửi cập nhật hội thoại, chuẩn hóa định dạng tin nhắn giữa các nền tảng khác nhau.
  + Trong sơ đồ, BFS nhận sự kiện "User Joins", "Bot Joins" và tin nhắn từ người dùng, sau đó chuyển tiếp đến **Bot**.

1. **Bot**
   * Là phần xử lý chính của chatbot, được lập trình để phản hồi người dùng.
   * Sau khi nhận tin nhắn từ BFS, nó xử lý dữ liệu (ví dụ: hiểu nội dung, gọi API, xử lý AI) rồi tạo phản hồi.
   * Sau đó, nó gửi phản hồi lại BFS để BFS tiếp tục chuyển tin nhắn đến kênh và hiển thị cho người dùng.

### **Tại sao cần cả Bot Framework Service và Bot?**

* **Bot Framework Service** giúp **giảm tải** cho bot bằng cách quản lý các kênh giao tiếp và xử lý các tin nhắn cơ bản trước khi chuyển cho bot.
* **Bot** chỉ tập trung vào xử lý logic và trả lời tin nhắn, không cần lo về giao tiếp với nhiều nền tảng khác nhau.
* Nếu không có **Bot Framework Service**, bot sẽ phải tự kết nối với từng kênh riêng biệt (Skype, Teams, Messenger...), làm tăng độ phức tạp.

👉 **Tóm lại:** BFS hoạt động như một cầu nối giữa kênh và bot, giúp bot dễ dàng mở rộng ra nhiều nền tảng khác nhau mà không cần viết mã riêng cho từng nền tảng.

swagger