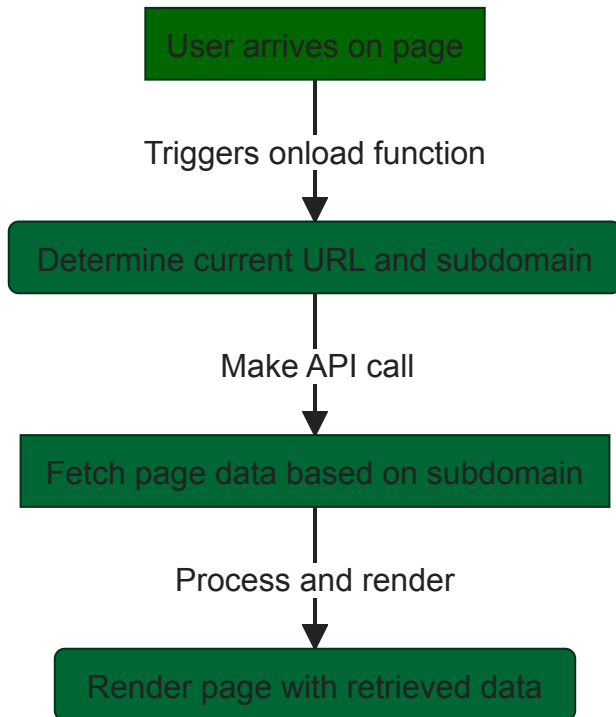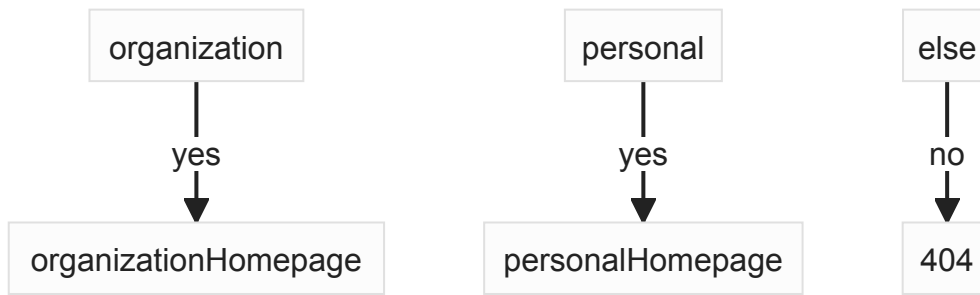Pages will undergo processing by examining search parameters, subdomains, and custom URLs. Subsequently, API calls will be made to retrieve the specified data based on these criteria.

```mermaid
graph TD
    A[User arrives on page] -->|Triggers onload function| B[Determine current URL and subdomain]
    B -->|Make API call| C[Fetch page data based on subdomain]
    C -->|Process and render| D[Render page with retrieved data]
```
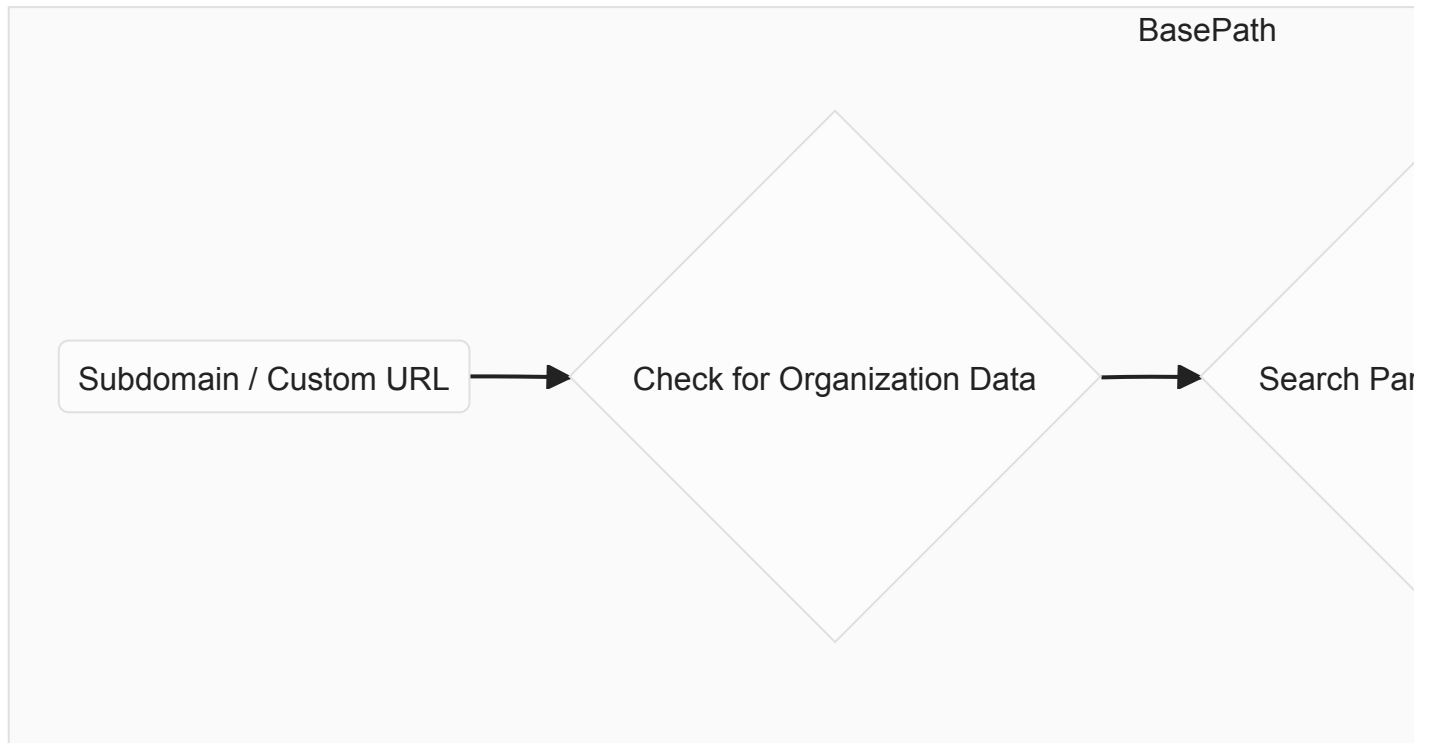
Upon a user's page arrival, an `onload` function is activated. This function examines the current URL for a subdomain, searching for page data linked to the organization's username. The objective is to enhance page content uniqueness, prioritizing data over design improvements.

PageTable

+ id: random
+ pageId: int
+ Images: []string
+ Iframe: []string
+ H1: []string
+ H2: []string
+ P: []string
+ Template: int

This management approach enables efficient and diverse saving of website data. Individual pages will be rendered conditionally based on the specified route, allowing for flexibility in presenting content as per the given conditions.

```
   organization          personal              else
        |                    |                   |
       yes                  yes                  no
        |                    |                   |
        v                    v                   v
organizationHomepage   personalHomepage        404
```

use cases

```
                                              BasePath

  ┌──────────────────────┐        ╱╲
  │ Subdomain / Custom URL │──▶  ╱    ╲                      ╱
  └──────────────────────┘    ╱ Check for ╲──▶  Search Pa
                              ╲ Organization ╱
                               ╲   Data    ╱
                                ╲        ╱
                                  ╲    ╱
                                   ╲╱
```
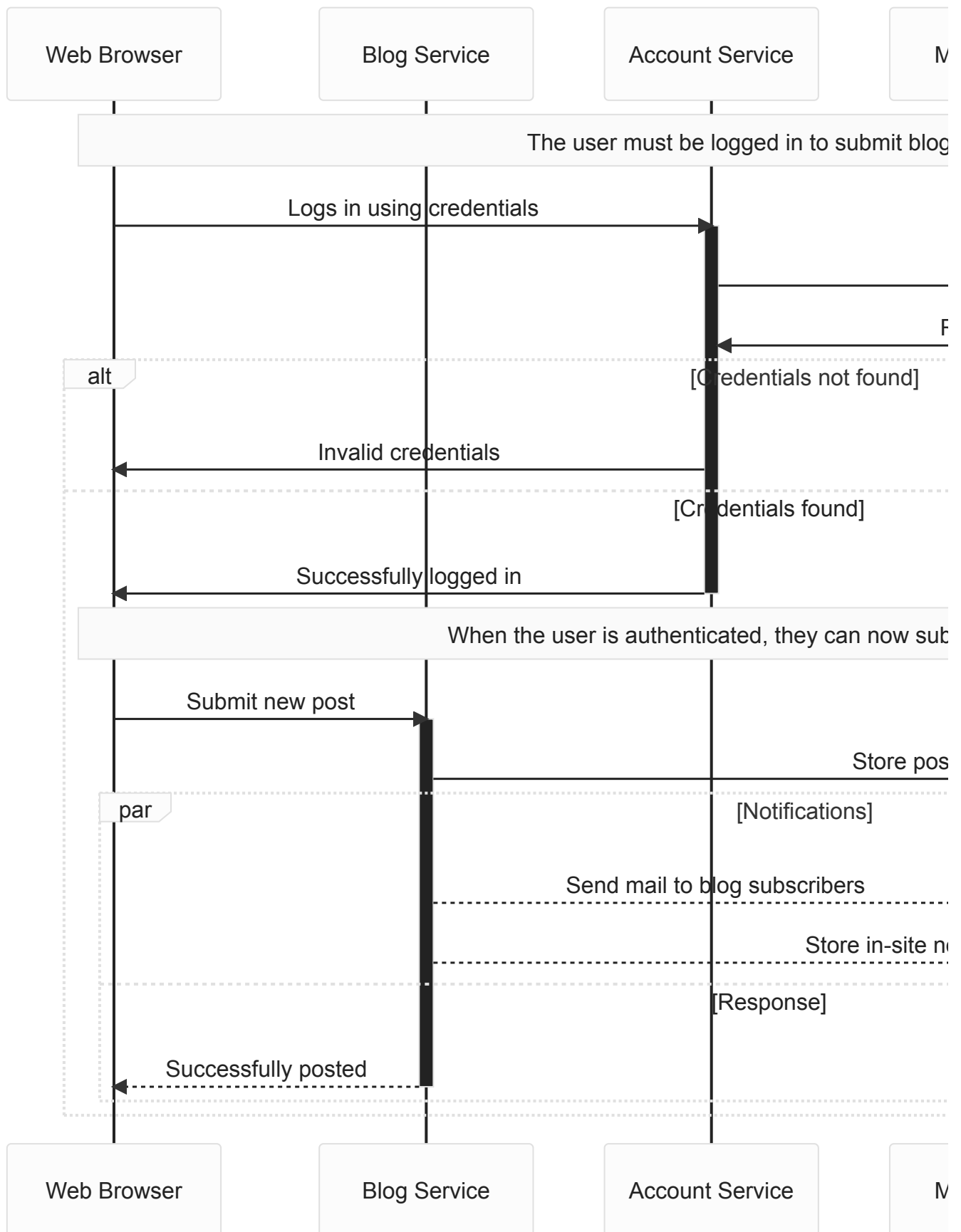
**Home

Organization

- ☑ ~~backdrop photo~~
- ☑ ~~services overview~~
- ☐ donation
- ☐ videos
- ☐ mission & objective
- ☐ mailing list
- ☐

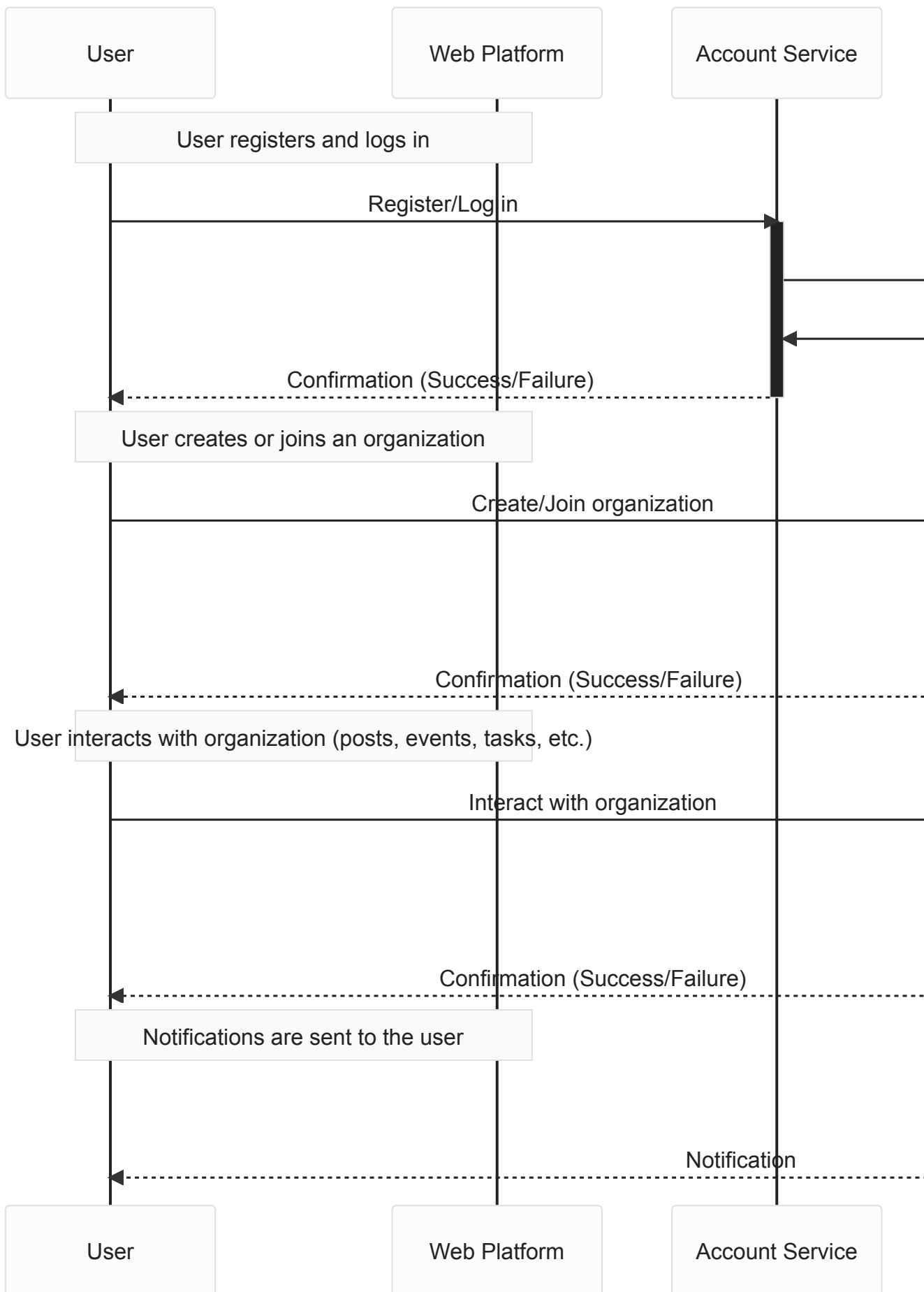Authentication Middleware

Web Browser | Blog Service | Account Service | M

The user must be logged in to submit blog

Logs in using credentials

[Credentials not found]

alt
Invalid credentials

[Credentials found]

Successfully logged in

When the user is authenticated, they can now sub

Submit new post

Store pos

par
[Notifications]

Send mail to blog subscribers

Store in-site n

[Response]

Successfully posted

Web Browser | Blog Service | Account Service | M

This platform offers users the ability to create an account, unlocking a myriad of features. Upon registration, users can either establish their own organization or explore existing ones housed in nearby establishments. The process is streamlined, complete with confirmation emails to ensure

security. they can craft a comprehensive CV file by seamlessly integrating in-house and external data. The platform facilitates communication through acceptance notifications or invitations, accessible both within the app and through email links.

Organization Middleware

```
User          Web Platform          Account Service

        User registers and logs in

    Register/Log in
  ───────────────────────────────────►
                                        ◄──────
                                        ◄──────
    Confirmation (Success/Failure)
  ◄╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌

    User creates or joins an organization

    Create/Join organization
  ──────────────────────────────────────────

    Confirmation (Success/Failure)
  ◄╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌

User interacts with organization (posts, events, tasks, etc.)

    Interact with organization
  ──────────────────────────────────────────

    Confirmation (Success/Failure)
  ◄╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌

    Notifications are sent to the user

                            Notification
  ◄╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌

User          Web Platform          Account Service
```
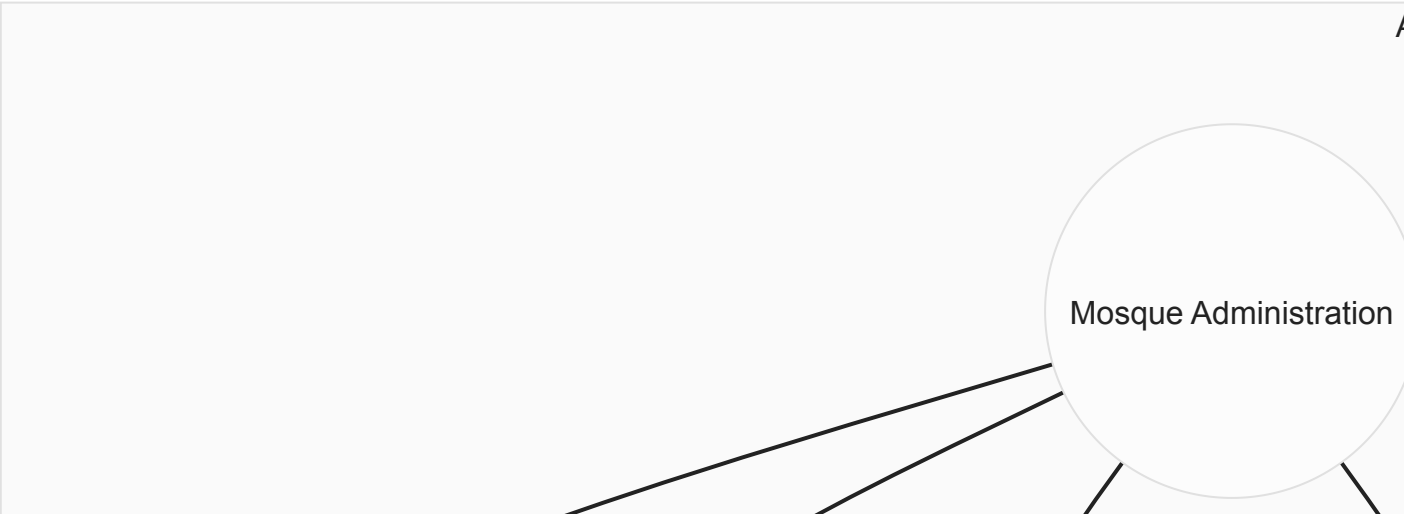
Once users are part of an organization, members gain access to a suite of tools, empowering them to create and view articles, peruse event timetables, request stewardship of prayers, contribute tickets, and even be assigned maintenance tasks. All these activities contribute to the development of a personal website for each member, enhancing their online presence.
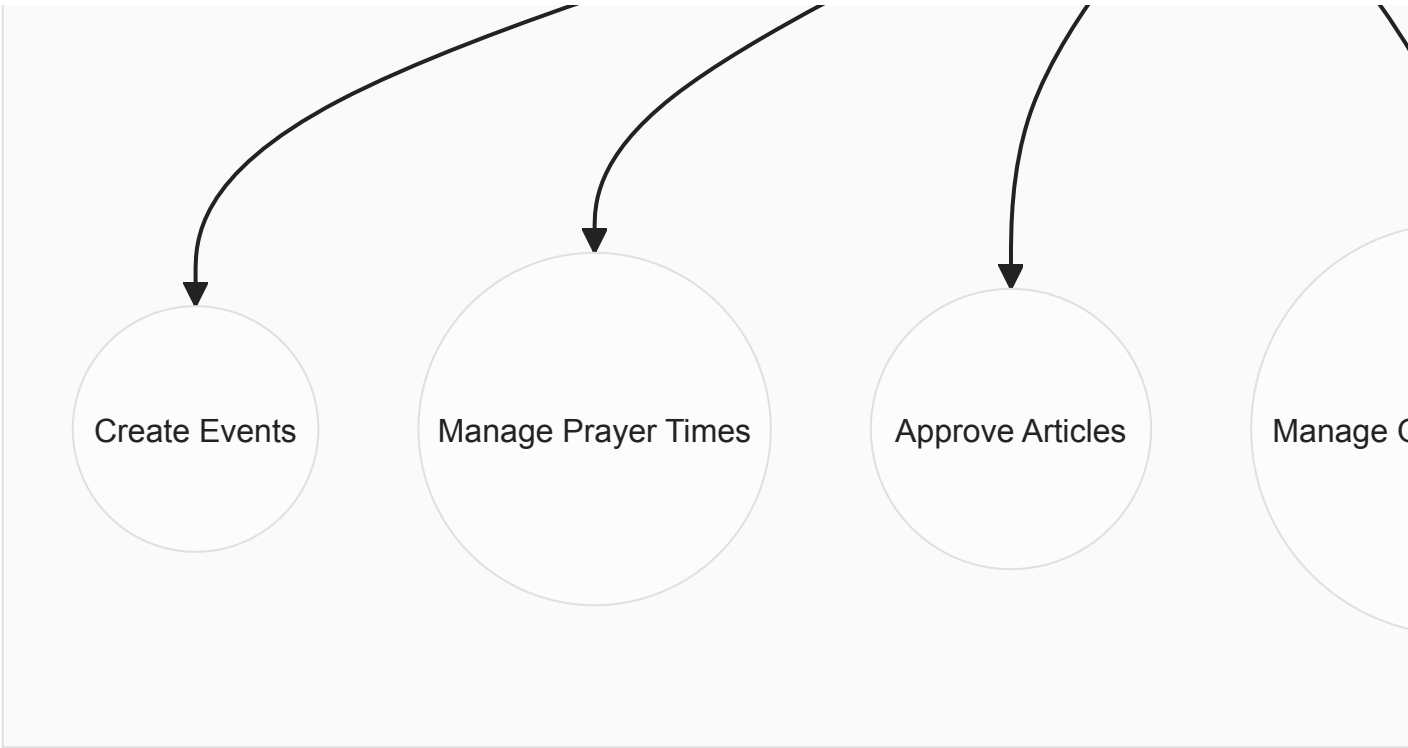
For users operating in organization owner mode, they can navigate to their organization website, mirroring the organization's information onto a website users and organisations alike have the option to add custom domains, tailoring their online presence.

Administrators of mosques hold a distinct set of capabilities, enabling them to create events, manage prayer times, approve articles (this allows the article writers to gain a approved tick on their articles as their group agrees, oversee organization information, handle uploads (iframes) + tag members within those uploads hence them too gaining the upload on their personal site, and add buildings. The building addition process involves submission for approval, with data stored in a MongoDB database and spatially queried to enhance local organization searches.

A unique feature is the perspective incorporated into the mobile app, which extracts relevant data such as local buildings that would gender-specific perspectives. For instance, if there are four nearby buildings, the app could focus on the closest one, housing two organizations—one catering to males and the other to females.

The spatial query system further refines the user experience by displaying only the most pertinent local organizations at retreiving their data from the sql database. This dynamic approach enhances user engagement with a map and ensures that the platform remains user-centric but affordable.

A

Mosque Administration

Create Events

Manage Prayer Times

Approve Articles

Manage C

User Activities:

- Create/view articles
- Peruse event timetables
- Request stewardship of prayers
- Contribute tickets
- Be assigned maintenance tasks
- Develop personal accolade website

Organization Owner Mode:

- Navigate organization website
- Mirror organization information
- Add custom domains
- Add buildings (submission for approval) Spatially queried MongoDB for local organization searches

Mosque Administrators:

- Create events
- Manage prayer times
- Approve articles
- Oversee organization info
- Handle uploads (iframes)
- Tag members in uploads

Unique Features:

- Mobile app's gender-specific perspective

- Extracts relevant data (local buildings)
- Focus on closest building
- Displays most pertinent local organisations
- Affordable, list-based user engagement

```
                                    ┌─────────┐
                                    │  Start  │
                                    └─────────┘
                                         │
                                Extract relevant data
                                         │
                                         ▼
                            ┌──────────────────────────┐
                            │  Focus on closest building │
                            └──────────────────────────┘
                             │                        │
              Interactions with nearby buildings    Display gender-specific perspectives
                             │                        │
                             ▼                        ▼
                     ┌────────────┐           ┌──────────────────────┐
                     │ Building 1 │           │ Enhance user experience │
                     └────────────┘           └──────────────────────┘
                     │          │
   Engage with male and female organizations   Connect with female organizations
                     │          │                        │
                     ▼          ▼                        ▼
              ┌───────────────────┐              ┌─────────────────────┐
              │ Male Organization │              │ Female Organization │
              └───────────────────┘              └─────────────────────┘
```