Club Aware System Documentation

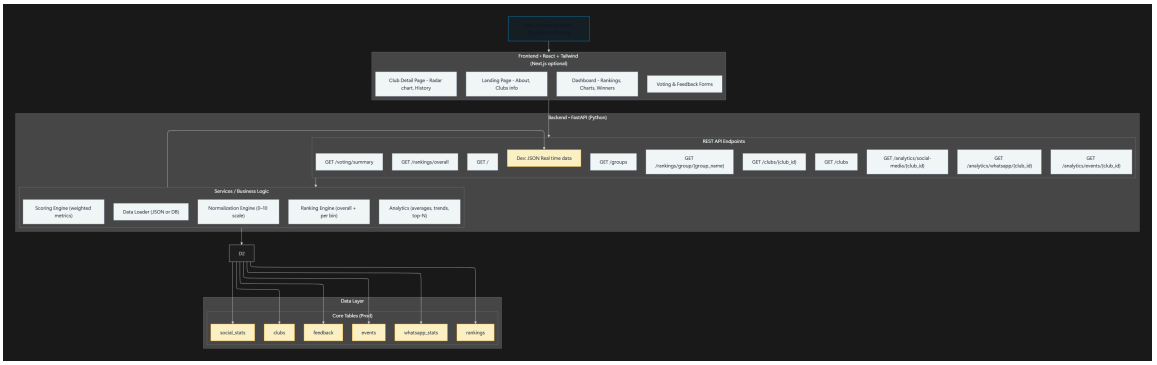
📌 Overview

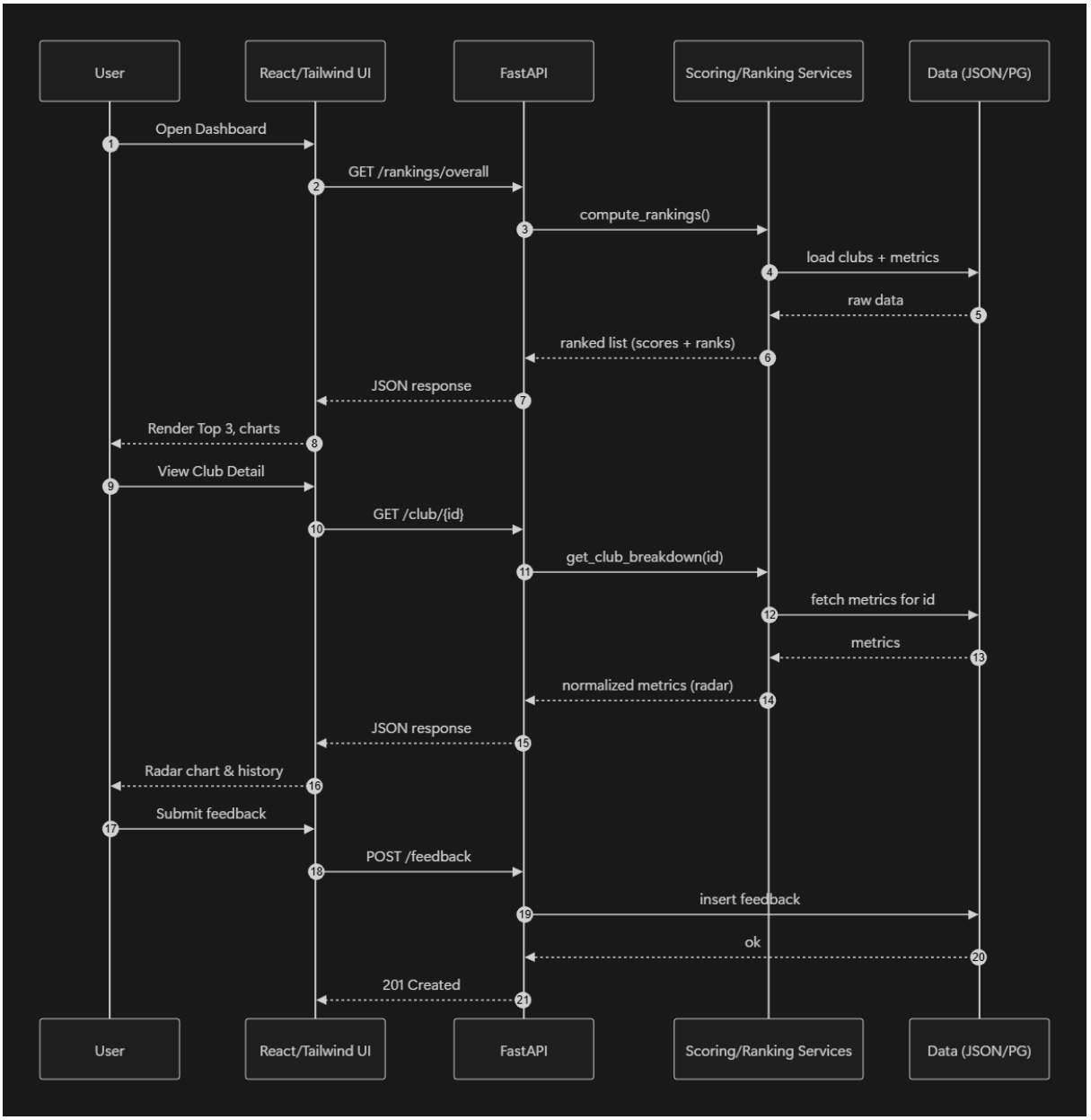
The **Club Aware System** is a data-driven evaluation framework designed to measure and compare university clubs based on **social media impact, internal engagement, event activity, and community growth**.

It uses real-world data sources such as **WhatsApp chats, Instagram posts, LinkedIn followers, and event records** to generate **fair, normalized, and weighted scores**, ultimately ranking clubs according to their overall performance.

🔄 Data Collection Pipeline

1. **WhatsApp Chats** → Chat Analysis → Engagement Scores
2. **Instagram Scraping** → Event Extraction → Social Media Metrics
3. **LinkedIn Data** → Follower Counts → Member Estimates
4. **Events & Voting** → Additional Metrics → Final Evaluation





⚖️ Weighted Scoring Framework

Each club is evaluated using **weighted components**:

- **Social Media Impact** → 30%
- **WhatsApp Engagement** → 25%
- **Event Activity** → 25%
- **Community Growth** → 20%

Formula

```
final_score = (
    social_score   * 0.30 +   # Public visibility
    whatsapp_score * 0.25 +   # Internal activity
    event_score    * 0.25 +   # Community impact
    growth_score   * 0.20     # Sustainability
)
```

All sub-scores are normalized to a **0–10 scale** before weighting.

📊 Evaluation Components

1. Social Media Scoring (0–10 scale)

Metrics from Instagram/LinkedIn:

- Follower count (normalized)
- Engagement rate (likes/comments ratio)
- Post frequency
- Story views
- Collaboration posts

Example (SNUC Coding Club):

- Instagram followers = **612**
- LinkedIn followers = **226**
- Engagement rate = **2.0%**
- Posts last month = **6**
- Avg likes = **61**

Scoring Algorithm:

```
follower_score   = min(followers / 1000, 10)
engagement_score = min(engagement_rate * 2, 10)
activity_score   = min(posts_last_month / 3, 10)
```

```
social_media_score = (follower_score + engagement_score + activity_scor
e) / 3
```

---

2. WhatsApp Engagement Scoring (0–10 scale)

Metrics from group chats:

- Message volume (total_messages)
- Sender diversity (unique_senders)
- Response patterns (avg_response_time)
- Positive engagement indicators (via NLP)

Example Data:

- **Montage**: 6,692 messages, 92 senders → High engagement
- **SNUC Coding**: 78 messages, 18 senders → Low engagement

Scoring Algorithm:

```
message_score    = min(total_messages / 1000, 10)
diversity_score  = min(unique_senders / 50, 10)
engagement_score = positive_engagement / 10

whatsapp_score = (message_score + diversity_score + engagement_score)
/ 3
```

---

3. Event Impact Scoring (0–10 scale)

Based on 169 real events extracted from Instagram.

- Event frequency
- Participation estimates (likes/comments as proxy)
- Event diversity (workshops, competitions, cultural, technical)
- Community impact

Example Events:

- **PCB Design Workshop** → 40 likes → Technical

- **MUN Conference** → 101 likes → Large participation
- **Coding Bootcamp** → 61 likes → Skill development

Scoring Algorithm:

```
frequency_score     = min(event_count / 10, 10)
participation_score = min(avg_participation / 50, 10)
diversity_score     = calculate_event_diversity(event_types)

event_score = (frequency_score + participation_score + diversity_score) / 3
```

---

4. WhatsApp NLP Analysis (For Engagement Quality)

- **Sentiment Analysis** → Detect positive engagement
- **Keyword Extraction** → Identify trending topics
- **Response Patterns** → Average reply times
- **Activity Mapping** → Timeline analysis

Example:

- Positive keywords → `['great', 'awesome', 'excellent', 'thanks']`
- Engagement indicators → `['question', 'help', 'participate']`

Real analysis:

- Club 1 → 7 positive engagements
- Club 3 → 33 positive engagements
- Club 4 → 14 positive engagements

---

5. Statistical Normalization

Since different clubs vary in size and scale, **normalization ensures fairness**.

Ranges observed:

- Followers → 226 to 3,095
- Messages → 78 to 6,692
- Events → 0 to 30+

Formula:

```
normalized_score = min((value / max_value) * scale, scale)
```

🏆 Final Weighted Scoring Algorithm

The **multi-criteria decision model** prevents dominance by any single metric and balances visibility, engagement, and sustainability.

```
weights = {
    'social_media': 0.30,
    'whatsapp':     0.25,
    'events':       0.25,
    'growth':       0.20
}
```

🔎 Evaluation Workflow

1. Load all data sources
2. Apply evaluation algorithms
3. Normalize and calculate weighted scores
4. Rank clubs based on final scores
5. Return formatted results

📌 Final Results (Example)

Rankings

1. **SNUC Rhythm** → **5.21** (Strong social + events)
2. **Montage** → **5.10** (Highest WhatsApp engagement)
3. **Isai** → **4.96** (Balanced performance)

Detailed Metrics (Sample Output)

```
{
  "club_rankings": [
```

```
    {
      "name": "SNUC Rhythm",
      "score": 5.21,
      "strengths": ["743 Instagram followers", "10% engagement rate", "High
event activity"],
      "metrics": {
        "social_media": 8.5,
        "whatsapp": 7.4,
        "events": 6.2,
        "growth": 7.8
      }
    },
    {
      "name": "Montage",
      "score": 5.10,
      "strengths": ["6,692 WhatsApp messages", "92 active members", "High
engagement"],
      "metrics": {
        "social_media": 6.8,
        "whatsapp": 9.2,
        "events": 5.5,
        "growth": 7.1
      }
    }
  ]
}
```


Tech stack used

Backend (FastAPI)

- **RESTful API** with comprehensive endpoints
- **Club Management**: CRUD operations for clubs
- **Ranking System**: Overall and group-based rankings
- **Analytics**: Social media, events, and WhatsApp analytics
- **Grouping**: Automatic club categorization

- **Voting System**: Integration with voting data
- **Data Models**: Pydantic models for type safety

Frontend (Streamlit)

- **Dashboard**: Key metrics and statistics overview
- **Club Rankings**: Interactive rankings with visualizations
- **Club Groups**: Browse clubs by categories
- **Club Details**: Comprehensive club information
- **Analytics Dashboard**: Visual analytics for individual clubs
- **Voting Results**: Summary of voting data
- **Responsive Design**: Clean and modern UI

Common Issues faced

1. **Backend not starting**: Check if port 8000 is available
2. **Frontend connection error**: Ensure backend is running first
3. **Data not loading**: Verify JSON files in `backend/data/`
4. **Package conflicts**: Use virtual environment
5. **issue is i have scrapped the data and stored** BEcause in the json as using the selenium to scrap insta and linkedin for the every request it will block the ip
6. **assigning the weight** weight assigning is the issue i have faced