# CRAZE
## Ignite Passion

Ingrid Chan
Anusha Goyal
Becca Zhu

# Background



**Group Contributions**

Ingrid Chan: 33%

Anusha Goyal: 33%

Becca Zhu: 33%

## Problem/Opportunity

The COVID-19 pandemic transformed consumer behavior in numerous ways. Notably, individuals are increasingly seeking personal enrichment by engaging in hobbies and building community. However, the current market is fragmented, with information scattered across platforms making it challenging and time-consuming for hobby-enthusiasts to find and book activities/classes. Inspired by an existing solution for the fitness space called ClassPass, we've improved upon their database framework to streamline the discovery and booking process for diverse hobbies ranging from carpet tufting to woodworking.

## Concept

Craze aggregates a wide range of hobby classes and experiences onto an accessible platform simplifying the user journey while also providing local studios with greater visibility. We aim to revolutionize the way people discover and engage with hobbies, addressing a significant gap in the market and fostering a vibrant community around shared interests.

## Key Features

- Class/Hobby Discovery
- Personalized Recommendations
- Integrated Booking System
- Social Integration
- Reviews and Ratings

# A Timely Solution in a Growing Market

Consumers increasingly seeking out experiences

The recreation market size is expected to see strong growth the next few years

No current solution meets consumer needs

## Revenue Streams

- **Monthly Subscriptions**
  - Users; $49.99 per month which goes towards booking
- **Booking Fees**
  - Studios pay a fixed price for each booking completed
- **Analytics & Insights**
  - For businesses to optimize their bookings and offerings
- **Enterprise Solutions**
  - For employers to offer as part of a wellness benefit

# Data Sources & Volume

## User Data

*Profiles*

User data including demographic information, preferences, and necessary geographical information.

*Interactions*

Data on user interactions in the app such as preferences friends, booking history, and reviews.

## Payment/Booking

This table will hold transaction history, booking and cancellation payments. Additionally, data on payment preferences will be stored here.

To ensure a seamless booking process, Craze will utilize a third-party integration with Stripe (more suited for digital businesses) for payment processing.

**stripe**

## Activity Lists

This table will hold lists of activities/classes to be displayed on Craze. We will pull data from the Meetup API which catalogues local groups and events.

*Additional Data Sources*

- Instagram Graph API: scrape data from posts, stories, and specific tags
- Facebook Graph API: public posts and page activity
- TripAdvisor API
- Direct Partnerships with local businesses
- New York Public Library Class Directory
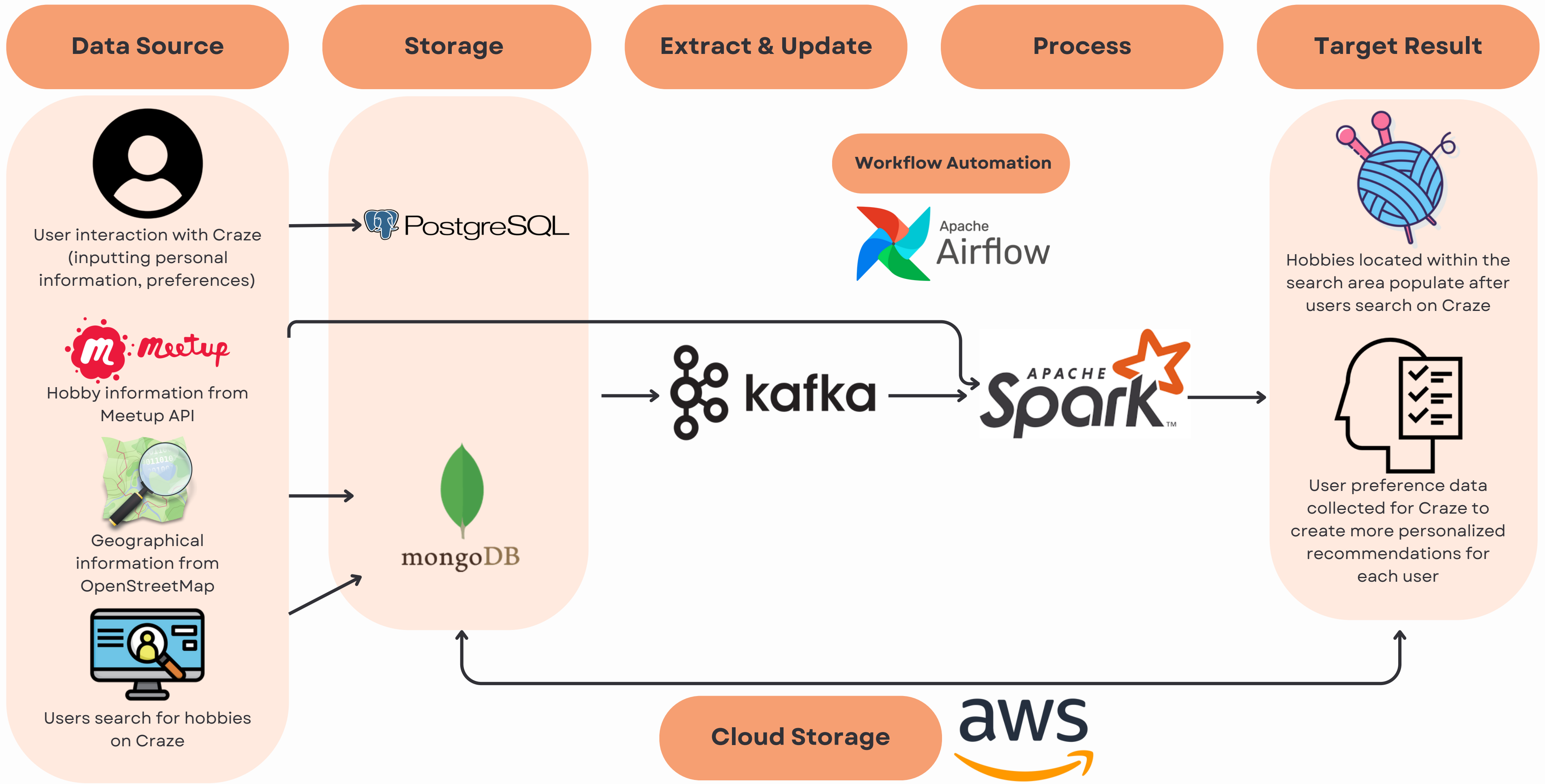
## Location Data

We will use data from Open Street Map (OSM) to ensure recommendations are geographically appropriate.
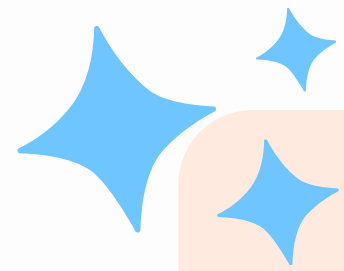
We will export POIs into a CSV and use the longitude and latitude to provide proximity-based recommendations.

We chose OSM over Google Maps because it is free, more malleable, and better suited for our purposes.

## Data Volume

*As our frameworks are similar to ClassPass we expect to handle a large amount of data eventually. Along with the classes, we will need to have the capacity to deal with subscription, user and payment information.*

**Data Source**

**Storage**

**Extract & Update**

**Process**

**Target Result**

User interaction with Craze (inputting personal information, preferences)

PostgreSQL

**Workflow Automation**

Apache Airflow

Hobbies located within the search area populate after users search on Craze

Meetup

Hobby information from Meetup API

kafka

Apache Spark

Geographical information from OpenStreetMap

mongoDB

User preference data collected for Craze to create more personalized recommendations for each user

Users search for hobbies on Craze

**Cloud Storage**

aws

5

# Pros

- Apache Kafka is horizontally scalable and has good fault tolerance.
- Apache Spark works well with Apache Kafka and has great speed and performance capabilities.
- MongoDB will provide the flexibility Crave needs as more user and reservation data grows.
- MongoDB uses sharding to ensure the database can hand large volumes of data.
- PostgreSQL has no licensing costs and if all data maintenance and management is kept in house, there will be no additional costs.
- Users can secure their Postgres databases and create different permission levels. This will be important for Crave to ensure user data is protected.
- AWS is user friendly and relatively easy to learn, and has unlimited server capacity.

# Cons

- MongoDB's default consistency is eventual consistency, which may lead to temporary data inconsistencies.
- PostgreSQL can have longer processing times for complicated SQL queries.
- Apache Kafka is complex to set up and master due to its technological complexities. Crave can hire experts as part of the data team to mitigate this issue.
- Apache Spark requires higher memory in order to support its high processing speeds, which can become costly.
- AWS servers sometimes lose power or connectivity to the cloud provider because it has millions of users. However, this is a common cloud computing issue.

# Cost Implications

| Technology Stack | Pricing | Initial Monthly Estimate |
|---|---|---|
| AWS S3 (Data Storage) | $0.23/GB (Standard S3) | 1,000 GB x $0.23/GB = $230 per month |
| AWS Athena (Queries) | $5/TB of data scanned | $5 |
| MongoDB | $0.12/hour (M2) | 730 hours x $0.12/hour = $87.60 per month |
| PostgresSQL | free tier available (Amazon EC2) | $0 unless limit exceeded |
| Apache Kafka | $0.10/GB (broker storage) | 500 GB x $0.10/GB = $50 per month |
| Apache Spark | $0.23/hour | 300 hours x $0.23/hour = $69 per month |
| Additional Costs | Data Transfer and Backup Costs | $9/100GB transfer |

| Expenditures | Description |
|---|---|
| Operational | Staff Salaries, Co-working Space, Insurance |
| Customer Acquisition | Digital Marketing, Promotions, PR |
| Legal & Compliance | Business Setup, Permits |
| Administrative | Software Licenses, Payroll |

The above is just a very initial estimate. As Craze scales, these associated costs are expected to increase particularly with data storage and processing needs. We intend to continuously monitor and optimize AWS usage to manage costs.

# Scalability

As outlined throughout the proposal, scalability is integral to our business so our data architecture design and choice of tools reflect this focus. Tools such as MongoDB, Apache Kafka, Apache Spark, and AWS will enable Craze to effectively manage and scale as the volume of data increases. In the future we also will integrate AI to bolster our Recommender Systems.
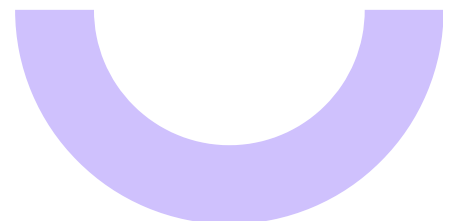
# Data Governance

We will make every effort to inform users on the data collected and their intended use. We will also take care to comply with all relevant privacy laws especially with our work with social media APIs. Our own API will be available to lisence through **CC BY-SA** to encourage further innovation and integration. Personal and sensitive information will be removed.
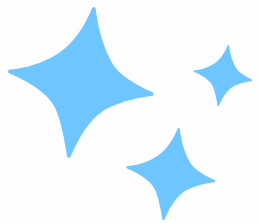
As the information on hobbies and activities is fragmented, data quality assurance and maintenance is key to providing accurate and timely results to users.

Finally, Craze will implement robust security features to protect sensitive personal and payment data collected through the platform.

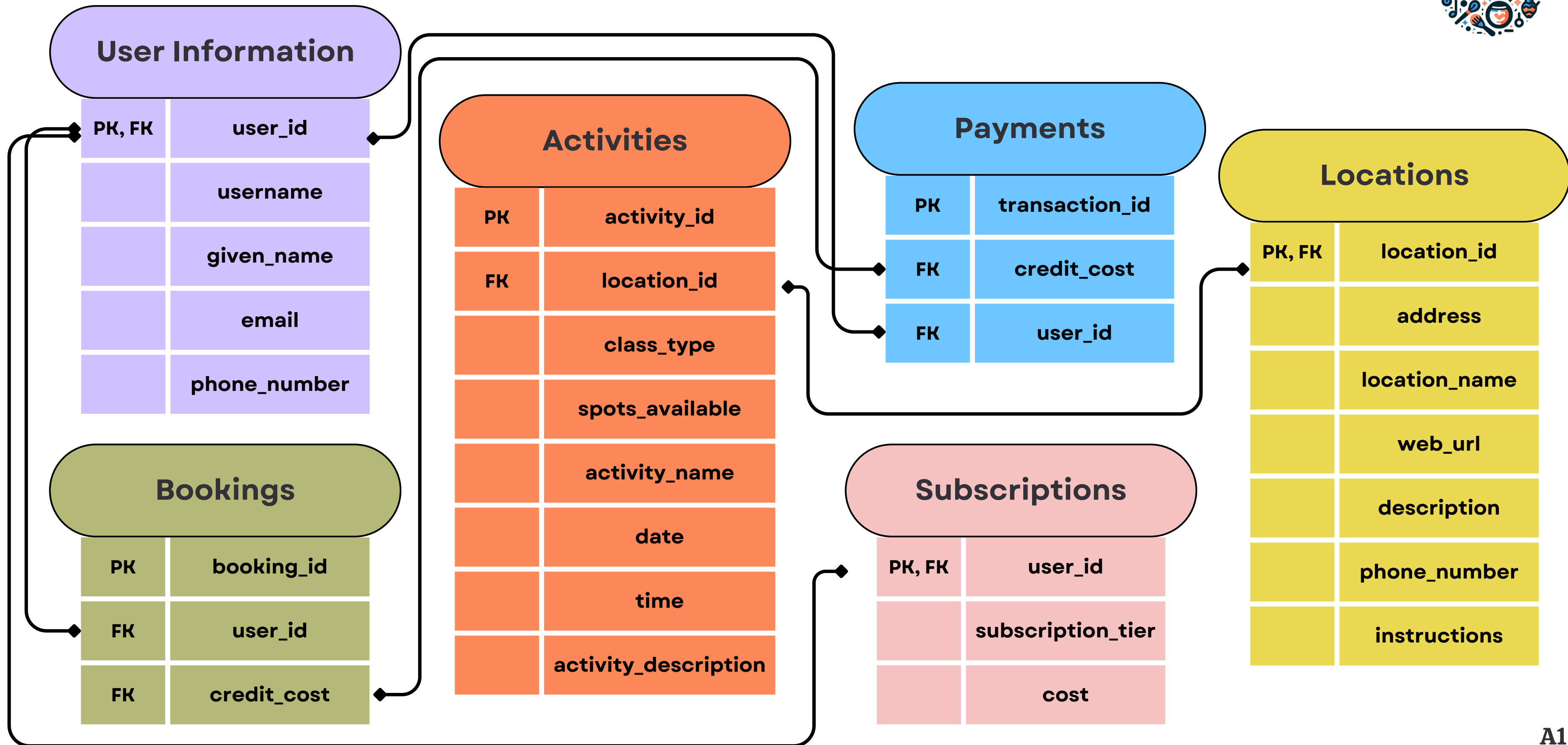# Appendix: Proposed Database Design



**User Information**

| PK, FK | user_id |
| --- | --- |
| | username |
| | given_name |
| | email |
| | phone_number |

**Activities**

| PK | activity_id |
| --- | --- |
| FK | location_id |
| | class_type |
| | spots_available |
| | activity_name |
| | date |
| | time |
| | activity_description |

**Payments**

| PK | transaction_id |
| --- | --- |
| FK | credit_cost |
| FK | user_id |

**Locations**

| PK, FK | location_id |
| --- | --- |
| | address |
| | location_name |
| | web_url |
| | description |
| | phone_number |
| | instructions |

**Bookings**

| PK | booking_id |
| --- | --- |
| FK | user_id |
| FK | credit_cost |

**Subscriptions**

| PK, FK | user_id |
| --- | --- |
| | subscription_tier |
| | cost |

A1

# Appendix: Data Architecture

| Data Storage | Real-Time Data Streaming | Data Processing and Analytics | Cloud Platform |
|---|---|---|---|
| MongoDB & PostgreSQL | Apache Kafka | Apache Spark | AWS (S3 for storing and Athena for quick querying) |
| Craze is expecting to intake data about activities and locations from a variety of resources like online APIs, social media and self-reported datasets from companies. This will result in messy and unstructured data. In order to account for this MongoDB will be the Database we use as the NoSQL format provides the necessary flexibility.<br><br>The SQL framework Craze will use is PostgreSQL. It is a open-source, easy to use data storage system that will be effective for more rigid structures of data like the input user data that Craze will be able to control. Using both SQL and NoSQL will be the most effect database management for this use-case. | Craze will need to update in real time as bookings and searches will consistently happen by users throughout the day and likely at the same time due to trends in daily routines. Apache Kafka is an event streaming platform that will be perfect for ensuring efficient data intake. It specializes in real time updating and works well for processing a continuous flow of data. Scalability is very important to our business as the volume of classes we expect from just NYC is very large. Horizontal scaling is something we expect and Apache Kafka will accommodate this. | Apache Spark is very good at handling a lot of data quickly which will be used to construct effective recommendations, analytics and further insights necessary. Spark is able to work with large amounts of data in real time and handle batch processing. We can use Spark Stream and Spark SQL to process data, and Spark Core for resource management. This will be useful since Craze would need to be able to adapt quickly to booking activity whilst also giving high level overviews. Apache Spark also works very well with Apache Kafka making it a good choice for the database design. | Amazon Web Services is able to scale up and provide cloud storage. It has a lot of integrations like AmazonDB and Athena but these segments aren't as flexible and common-place as the ones chose for Craze's design. Craze will mainly use S3 for cloud storage and EC2 for computing resources. Athena will be useful for quick and easy querying that can be very useful for debugging or script building. |