

Let's FLOCK:

A Social Dining App to Connect Over Shared Interests"

Let's FLOCK is a unique, non-dating app designed to bring people together for lunch based on mutual interests and preferences. With no focus on photos or dating, this app fosters a community where individuals can make new connections and enjoy meaningful conversations over a meal, simply by matching with others nearby who share similar tastes and hobbies.

Repository: <https://github.com/charleshuurman/Let-s-FLOCK>

User Story:

As a user of "Let's FLOCK," I want to be able to log in, select topics of interest, choose headlines, schedule meetups, chat with other users, manage my profile, and ensure that my data is secure and my privacy is protected.

Acceptance Criteria:

Authentication:

- Users can register for an account or log in using their credentials.
- Passwords are securely hashed and stored in the database.
- User sessions are managed using express-session and cookies.

Homepage:

- After logging in, users are directed to the home page.
- Users can select topics of interest from the provided categories: WORLD, NATIONAL, BUSINESS, TECHNOLOGY, ENTERTAINMENT, SPORTS, SCIENCE, HEALTH.

Headlines Selection:

- For each selected topic, the app displays five headlines.
- Users can click checkboxes below each headline to select the articles they want to discuss.
- The app keeps track of how many users have selected each article in the user's geolocation.

Schedule Meetups:

- Below the selected headlines, users can see the headlines they've chosen and available timeslots.
- Users can click checkboxes next to timeslots that work for them.
- Users can see other users within 20 miles who have also selected the same timeslots for the selected articles.

Chat Functionality:

- Below each timeslot, there is a comment section where users can chat.
- Comments are displayed in reverse chronological order (latest first).
- Users can send and receive messages in real-time.
- Chats and selected headlines automatically disappear after 9 pm and are reset.

Profile Management:

- Users can access their profile to:
 - Change their bio.
 - Choose an avatar from a selection of images.
 - View and update their personal information.
 - Change their password.

Data Security:

- API keys and sensitive information are protected using environment variables.
- Personal information and passwords are securely stored.

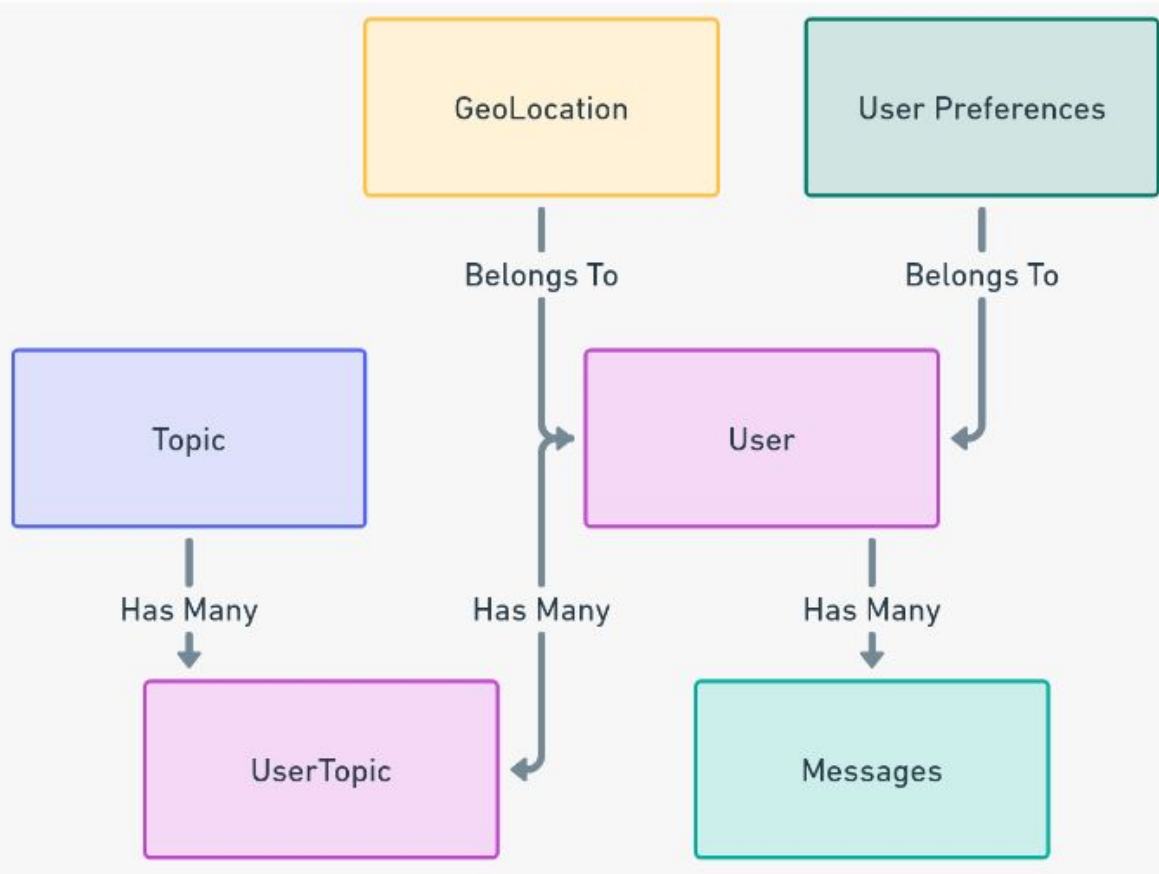
Polished UI:

- The app has a visually appealing and user-friendly interface.
- Navigation is intuitive, and elements are well-organized.
- The design follows a responsive web design approach for different screen sizes.

Deployment:

- The app is deployed on Heroku, ensuring it is accessible to users.

Schema of Let's Flock DB



This diagram illustrates the relationships between different entities in the database:

User: Central entity representing the users of the app.

UserTopic: A junction table for the many-to-many relationship between users and topics.

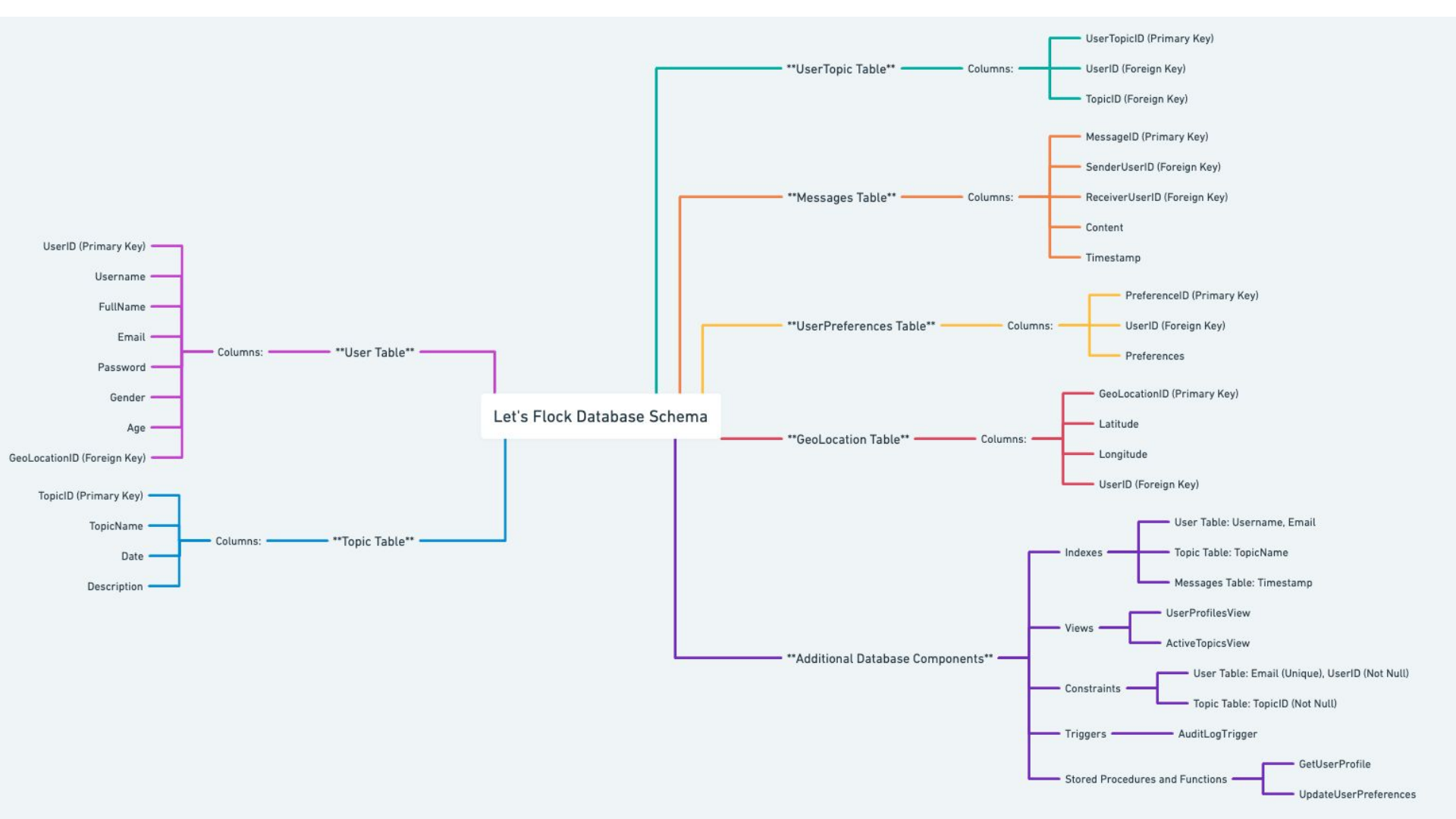
Topic: Represents the topics of the day or other discussion topics.

Messages: Stores messages or interactions between users.

User Preferences: Stores individual user preferences, linked to the User.

GeoLocation: Contains geolocation data associated with users.

User Table	Topic Table	UserTopic Association Table
<ul style="list-style-type: none">- Columns:<ul style="list-style-type: none">• username: String (Unique, Not Null)• fullName: String• gender: Enum ('Male', 'Female', 'Other')• overEighteen: Boolean• password: String (Encrypted)• email: String (Unique, Not Null)• geoLocation: String- Primary Key: username- Foreign Keys: None- Indexes: On email for faster lookup- Constraints: Unique constraint on email and username	<ul style="list-style-type: none">- Columns:<ul style="list-style-type: none">• topicName: String• date: Date• description: Text- Primary Key: topicName- Foreign Keys: None- Indexes: On date for querying topics by date- Constraints: None	<ul style="list-style-type: none">- Columns:<ul style="list-style-type: none">• userId: Integer (References User Table)• topicId: Integer (References Topic Table)- Primary Key: Composite key (userId, topicId)- Foreign Keys: userId references User, topicId references Topic- Indexes: On both userId and topicId- Constraints: None



Start: Log In

Username, Password

Start: Sign Up page

User Details Input: Username, Name, Age, Gender
and Accept Geo-Location, Password

API: Trending Topics

API: Trending Topics

Trending Topics Display

User Selects Interests
User selects Location Radius

Adds your Topics to the Database,
Checks the database for other users that choose the same topics in the same geolocation

Interest-Based Post Wall

Post Interaction: Comment on Posts

Every Day Posts Auto-Delete

Users Database

- Username id
- Username
- Full Name
- Over 18 selected
- Gender
- Accept
- Geo-Location

Trending topics DB

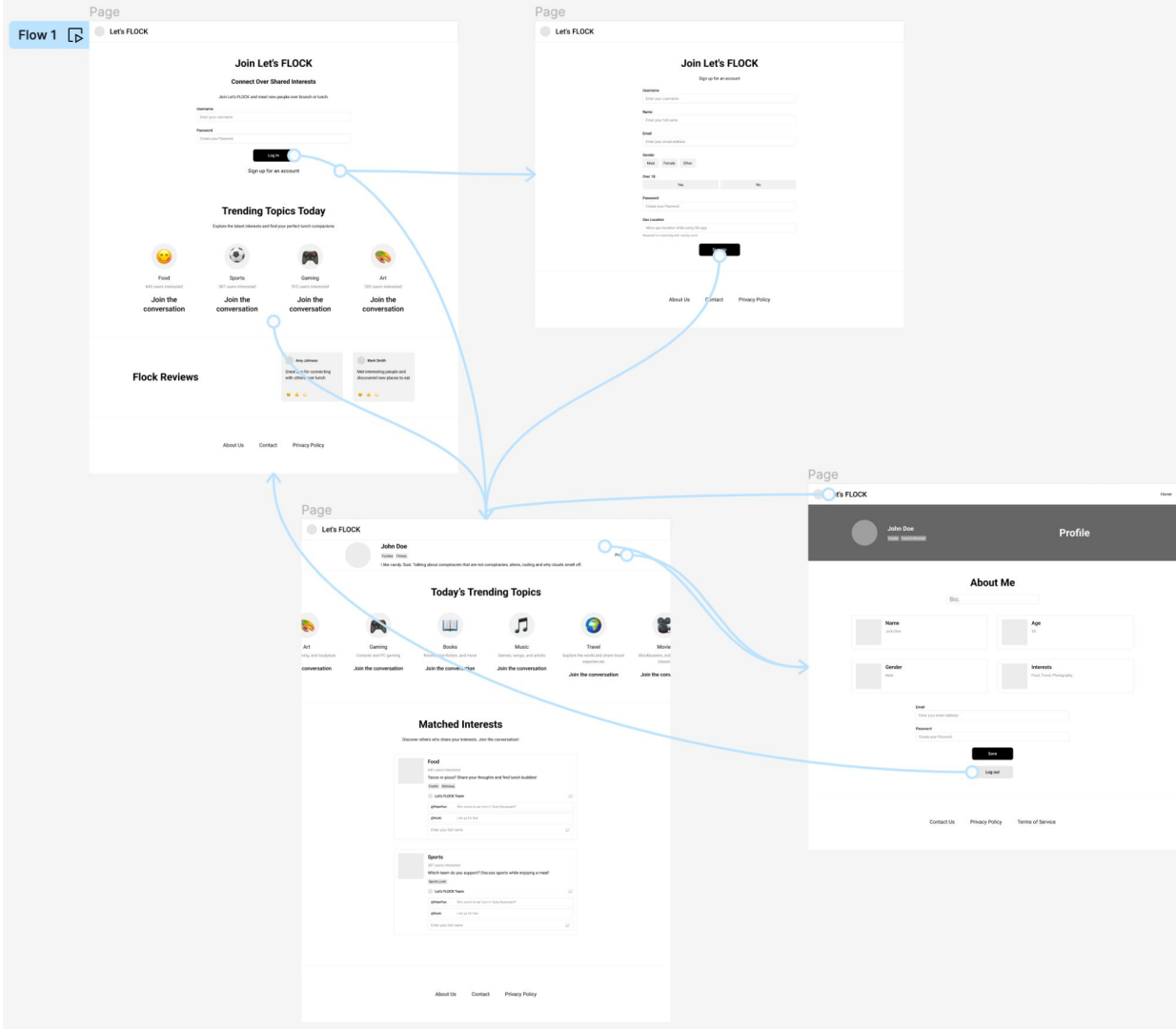
- Topic id
- Topic title
- Date

Selected Trending topics DB

- Selected-Topic id
- Topic id
- users
 - user id
 - user id
 - ...

UI (User Interface):

Done with Figma: [Figma Prototype Link](#)



API

API Real-Time News Data: <https://rapidapi.com/letsrcape-6bRBa3QguO5/api/real-time-news-data>

Get top news globally, per topic or search local news on Google News in real-time.

This API will require that BEFORE using it, you select one of these topics:

WORLD

NATIONAL

BUSINESS

TECHNOLOGY

ENTERTAINMENT

SPORTS

SCIENCE

HEALTH

THEN it will GET top headlines of the day from this topic.

Roles divided in 5

- **ALFA: Backend Developer (Node.js/Express.js/Sequelize)** **ALEXANDER, DAVON CHARLES**

Initial Setup: Set up the Node.js and Express.js server.

API Development: Develop RESTful API routes (GET, POST) and complex database queries.

Database Integration: Integrate Sequelize with MySQL for database operations.

API Integration Assistance: Collaborate with other team members for API integration.

Backend Support: Provide backend support for frontend integration and security implementation.

- **OMEGA: Frontend Developer (Frontend Developer & Coordinator)** **CHARLES, CALEB**

Wireframe Design: Create initial wireframes and design the app layout.

UI/UX Development: Develop the frontend using Handlebars.js, focusing on a responsive and user-friendly UI.

Project Coordination: Act as the project coordinator, organizing tasks and ensuring smooth collaboration.

Presentation Preparation: Lead the preparation for the project presentation.

Integration: Work closely with backend developers to integrate the frontend with the API.

- **BETA: (Authentication and Security Specialist)** **DAVON, ALEXANDER**

User Authentication: Implement user authentication using express-session and cookies.

Security Implementation: Ensure the security of the app, particularly in handling user data and sessions.

Research and Integration: Assist in researching and integrating suitable APIs for the app's features.

Security Enhancements: Implement additional security measures and best practices.

- **DELTA: (DevOps/Deployment Manager)** **GILDARDO, DAVON,**

Deployment Setup: Set up and manage the initial deployment process on Heroku.

Environment Management: Manage environment variables and deployment configurations.

CI/CD Pipeline: Begin setting up a CI/CD pipeline for streamlined development.

Deployment Troubleshooting: Address any deployment-related issues and ensure smooth operation on Heroku.

Technical Presentation Assistance: Assist in finalizing the technical aspects of the project presentation.

- **GAMA (Quality Assurance/Documentation Specialist)** **ALEXANDER, GILDARDO, CALEB**

Documentation: Write comprehensive documentation, including README.md and code comments.

Testing: Conduct thorough testing of the application, focusing on both functionality and user experience.

Presentation Development: Prepare and design the PowerPoint presentation for the project.

Quality Assurance: Ensure code quality and adherence to best practices.

Final Presentation Finalization: Finalize the PowerPoint presentation, ensuring it's comprehensive and visually appealing.

Day 1 and 2: Initial Setup and Planning

Player ALFA (Backend Developer)	Player Omega (Frontend Developer & Coordinator)	Player Beta (Authentication and Security Specialist)	Player Delta (DevOps/Deployment Manager)	Player Gama (QA/Documentation Specialist)
<ul style="list-style-type: none">• - Set up the Node.js and Express.js server.• - Begin basic RESTful API routes (GET, POST).	<ul style="list-style-type: none">• - Create initial wireframes for the app's layout and user journey.• - Design the basic layout using Handlebars.js.	<ul style="list-style-type: none">• - Start implementing basic user authentication (express-session, cookies).• - Begin setting up security measures for user data.	<ul style="list-style-type: none">• - Set up initial deployment on Heroku.• - Prepare environment variables and deployment configurations.• - Assist Player Beta in researching suitable APIs for the app's features.	<ul style="list-style-type: none">• - Begin documentation (README.md, code comments).• - Set up a basic testing framework.• - Start creating a basic PowerPoint presentation layout for the project.

Day 3: Development and Intermediate Progress

Player ALFA (Backend Developer)	Player Omega (Frontend Developer & Coordinator)	Player Beta (Authentication and Security Specialist)	Player Delta (DevOps/Deployment Manager)	Player Gama (QA/Documentation Specialist)
<ul style="list-style-type: none">• Develop complex database queries and associations.• Assist in API integration.	<ul style="list-style-type: none">• Refine wireframes and start converting them into actual UI components.• Continue frontend development, focusing on integrating the API data.	<ul style="list-style-type: none">• Enhance authentication mechanisms.• Implement additional security measures.	<ul style="list-style-type: none">• Address any deployment-related issues.• Begin setting up a continuous integration/continuous deployment (CI/CD) pipeline to streamline development.	<ul style="list-style-type: none">• Conduct testing of new features.• Update documentation with new developments.• Start drafting the final presentation content based on the project's progress.

Day 4: Finalization, Testing, and Presentation Preparation

All Players	ALFA & Beta	Player Omega (Frontend Developer & Coordinator)	Player Delta (DevOps/Deployment Manager)	Player Gama (QA/Documentation Specialist)
<ul style="list-style-type: none">• Collaborate to finalize the app.• Prepare for the project presentation, dividing roles and content.	<ul style="list-style-type: none">• Finalize backend and API integrations.• Ensure smooth functionality of API features.	<ul style="list-style-type: none">• Finalize UI/UX aspects.• Lead the preparation for the project presentation.	<ul style="list-style-type: none">• Final deployment checks.• Ensure full functionality on Heroku and manage any last-minute deployment issues.• Assist in finalizing the presentation, focusing on technical aspects.	<ul style="list-style-type: none">• Final round of testing, including user experience aspects.• Complete all documentation.• Finalize the PowerPoint presentation, ensuring it's comprehensive and visually appealing.