LoveAtFirstBite

Group project: Lunch voting app (team edition)

A web application that allows users to vote once per day on where the team should go for lunch. All users share the same pool of restaurants, the app resets votes automatically.

Entities

User: Id, UserName, UserEmail, Password (hash)  
Restaurant: Id, Name, Address, CreatedByUserId (FK to User)  
Vote: Id, RestaurantId (fk), UserId (fk), Date

Functional Requirements

* Register/login as a user
* Add new restaurants to shared list
* Cast one vote per user per day
* Display current vote results
* Change vote
* Automatic reset (midnight)

Non-Functional Requirements

* Login/authentication must be secure
* API response time should be < 500ms
* The app should support at least 50 users
* The website should be responsive design

MoSCoW prioritation

Must have: Login/register user, Add restaurant, One vote per user per day, reset votes daily.

Should have: Change vote

Could have: Change vote shortcut, forgot password reset, users can change automatic reset time.

Won’t have: Multiple teams, admin dashboard, notifications/sms.

User Stories

As a user, I want to register and log in so I can participate in voting.  
As a user, I want to add a restaurant to the shared list.  
As a user, I want to vote once per day so my vote counts toward the lunch decision.  
As a user, I want to see which restaurant is winning today.  
As a user, I want to be able to change my vote.  
As a user, I want the vote count to reset daily.  
As a user, I want to be able to change the reset time.

Use cases

Use Case 1: Register and Log In

**Title**: Register/Login  
**Actor**: User  
**Description**: A user creates an account or logs in to access voting features.  
**Preconditions**: User is not currently logged in.  
**Postconditions**: User is authenticated and session/token is stored.

**Basic Flow**:

1. User navigates to login/register page.
2. User enters credentials.
3. System validates input and logs user in (or creates account).
4. User is redirected to the main dashboard.

Use Case 2: Add a Restaurant

**Title**: Add Restaurant  
**Actor**: Authenticated User  
**Description**: A logged-in user submits a new restaurant to the shared pool.  
**Preconditions**: User is logged in.  
**Postconditions**: Restaurant is saved and visible to all users.

**Basic Flow**:

1. User opens “Add Restaurant” form.
2. User enters name (and optional description).
3. System saves the restaurant with the user’s ID as AddedByUserId.

Use Case 3: Cast a Vote

**Title**: Vote for a Restaurant  
**Actor**: Authenticated User  
**Description**: A user votes once per day for a restaurant.  
**Preconditions**: User is logged in and has not voted today.  
**Postconditions**: Vote is saved and linked to the user and the restaurant.

**Basic Flow**:

1. User clicks “Vote” on a restaurant.
2. System checks if the user has already voted today.
3. If not, system saves the vote and updates live count.

**Alternative Flow**:  
3a. If user already voted → system shows error or prompt to change vote.

Use Case 4: View Today’s Top Restaurant

**Title**: View Current Leader  
**Actor**: Any User  
**Description**: The user sees the restaurant with the most votes for today.  
**Preconditions**: There are restaurants and at least one vote.  
**Postconditions**: UI displays today’s vote results.

**Basic Flow**:

1. User opens the vote summary or home page.
2. System fetches and displays current vote tally.
3. Restaurant with most votes is highlighted as the leader.

Use Case 5: Change My Vote

**Title**: Change Vote  
**Actor**: Authenticated User  
**Description**: A user can change their vote once before reset.  
**Preconditions**: User is logged in and has already voted today.  
**Postconditions**: Old vote is removed or updated; new vote is saved.

**Basic Flow**:

1. User clicks “Change Vote.”
2. System deletes the existing vote for today.
3. User selects a new restaurant and system saves new vote.

Use Case 6: Daily Reset

**Title**: Reset Votes  
**Actor**: System (automated)  
**Description**: System clears all votes at a scheduled time each day.  
**Preconditions**: Current time equals the configured reset time.  
**Postconditions**: All votes are deleted.

**Basic Flow**:

1. Background task (cron) runs at configured reset time.
2. System deletes all Vote entries from database.
3. Vote leaderboard resets to 0.

Use Case 7: Change Vote Reset Time

**Title**: Update Reset Time  
**Actor**: Admin  
**Description**: Admin configures the daily reset time for vote clearing.  
**Preconditions**: Admin is logged in.  
**Postconditions**: New reset time is stored in config.

**Basic Flow**:

1. Admin navigates to system settings.
2. Admin selects a new reset time (e.g., 2 PM instead of midnight).
3. System updates background scheduler accordingly.

UML diagram

A diagram of a server

AI-generated content may be incorrect.