Recipe book

By students of group PMIm-13

Baikenych Anastasia, Hytsiv Oleksii,

Kusiak Vitalii, Liesnik Uliana

Agenda

1. Introduction
2. User roles
3. Subsystems
4. Use Case UML diagrams
5. Functional metrics
6. Funnels
7. Architecture
8. Interactions
9. Possible failures and responses to them
10. ER-diagram
11. Security Considerations
12. Retention policy by table
13. Deployment model
14. Metrics
15. Alerting
16. Functional requirements
17. Non-functional requirements
18. Security model
19. Sequence diagrams
20. Demo

Introduction

Recipe Book - is a digital platform we developed as part of our course. This project aimed to create a user-friendly tool for organizing, sharing, and managing recipes, combining technical skills with real-world problem-solving to make cooking more convenient and enjoyable.

User roles

Our web application provides guest and user app roles which refer to different levels of access.

- Guest - someone who has not yet registered or created an account. They have limited access to application’s features and content. They are able to create an account or login to the existing one.
- User is someone who signed in the app. They have full access to all application features and content.

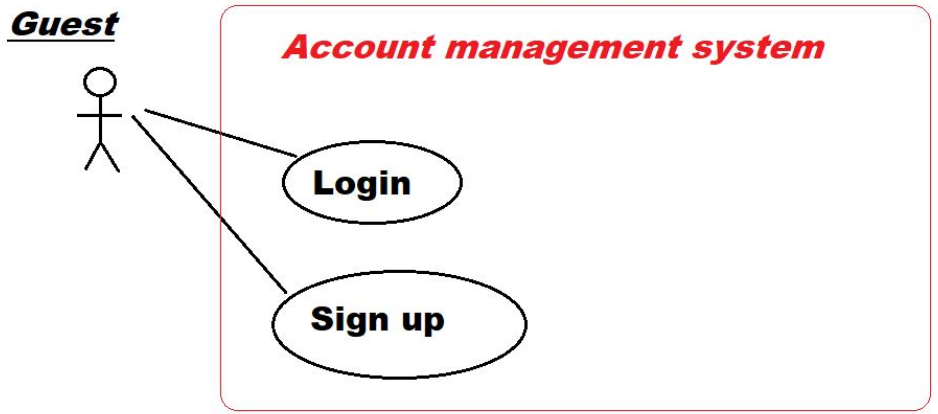
Subsystems

Two main subsystems may be highlighted in Recipe Book application:

- Account management system is a part of this application that handles operations with user's account, such as signing up or in, editing personal information, sending verification emails etc.
- Recipes management system lets user interact with their own recipe records as well as with those created by other users.

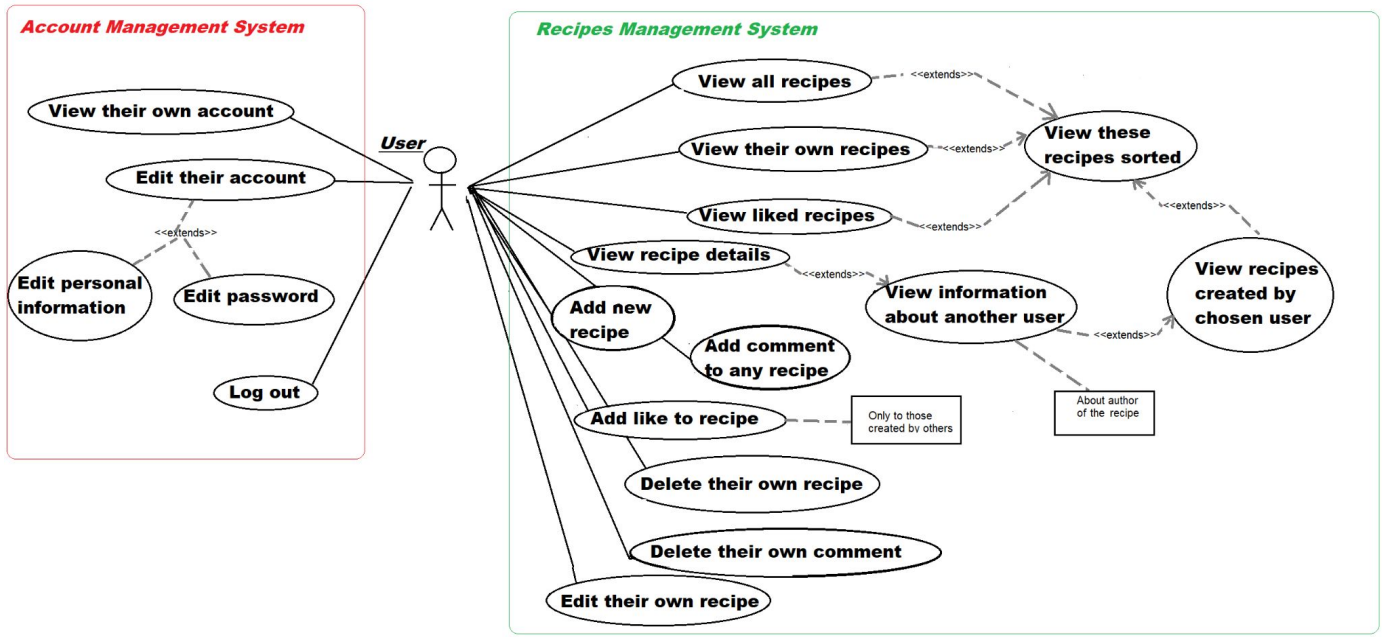
Use Case UML diagrams

For Guest:



Use Case UML diagrams

For Guest:



Functional metrics

Metric	Description	Measurement Unit	Connection to Functional Requirements
User Signups	Number of new users registered over time.	Count	Monitors the user onboarding functionality. Helps identify if signup processes are user-friendly.
Active Users	Number of users actively engaging with the platform daily/weekly/monthly.	Count	Tracks user retention and engagement. Ensures the application is effectively attracting and keeping users.
Recipe Uploads	Number of recipes created and uploaded by users.	Count	Measures the utilization of the recipe creation functionality. Helps assess the feature's relevance.
Likes per Recipe	Average number of likes received per recipe.	Count	Evaluates user interaction and engagement with uploaded recipes.
Comments per Recipe	Average number of comments left per recipe.	Count	Measures community engagement and use of the comment functionality.
Recipe Popularity	Number of views per recipe.	Count	Tracks content popularity. Helps identify trending recipes for potential recommendations.
Average Time Spent	Average time users spend on a recipe page.	Seconds	Measures user engagement with recipe content. Validates the interest in detailed recipe instructions.
Failed Logins	Number of failed login attempts.	Count	Monitors the login system's robustness and detects potential brute force attacks.

Metric	Description	Measurement Unit	Connection to Functional Requirements
Email Open Rate	Percentage of users opening emails sent from the app.	Percentage (%)	Evaluates the effectiveness of email communication and engagement through Mailtrap.
Search Queries	Number of searches performed and success rate of queries returning results.	Count / Percentage (%)	Assesses the efficiency of the search functionality and its relevance to user needs.
Server Response Time	Average response time of the application to user requests.	Milliseconds (ms)	Monitors system performance and compliance with performance requirements.
Error Rate	Percentage of failed operations (e.g., failed uploads, API errors).	Percentage (%)	Identifies potential issues in core functionalities and backend reliability.
Image Upload Success	Number of successful vs. failed image uploads.	Count / Percentage (%)	Tracks performance of image upload and validation processes.
Recipe Categories Usage	Number of recipes per category.	Count	Helps understand user preferences and guides feature enhancements like tailored recommendations.
User Retention Rate	Percentage of users returning to the app after a set time (e.g., 7 or 30 days).	Percentage (%)	Monitors long-term user engagement and satisfaction with the platform.
Abandoned Sessions	Number of users who abandon the app after starting certain actions (e.g., incomplete signups).	Count	Indicates usability issues or pain points in user workflows.

Funnels

There are a few funnels which may be created to visualize how users interact with different features and track their progression through a series of steps. For example:

Onboarding Funnel

- Reveals if the signup process or initial experience is discouraging users.
- Metrics used: Search queries, User signups, Active users, Recipe uploads.

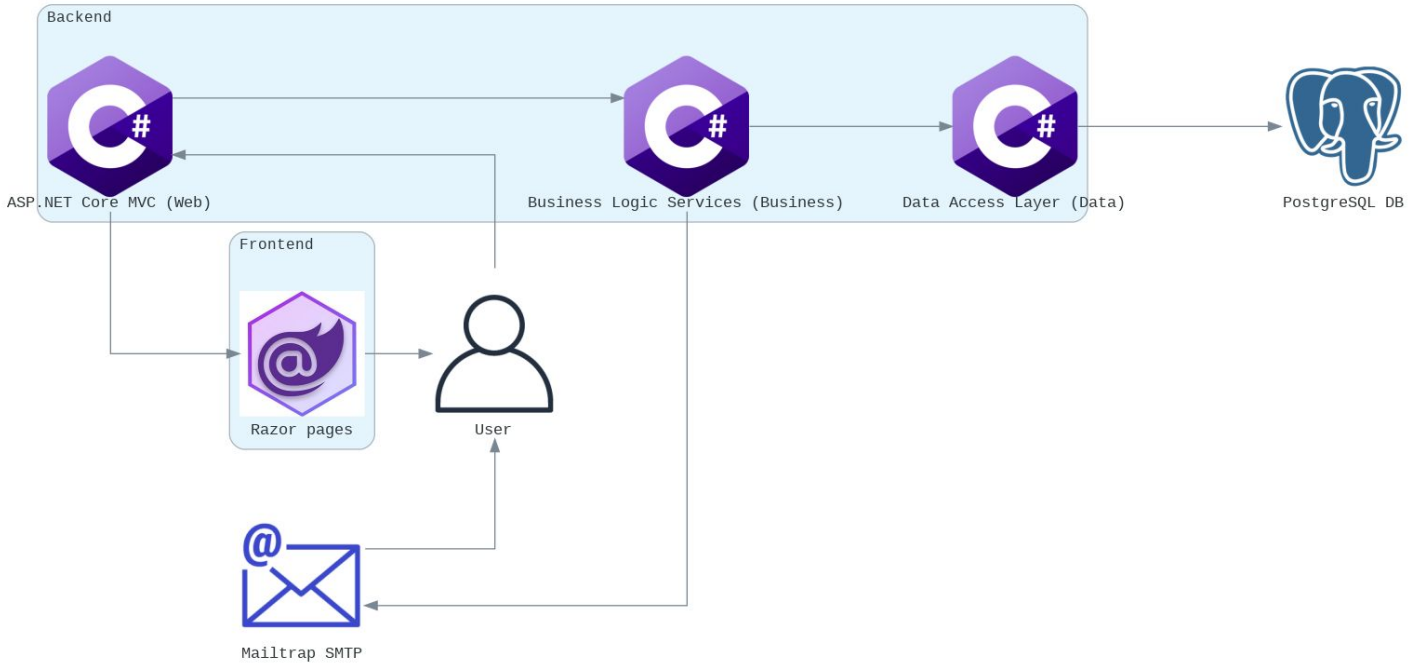
Content Engagement Funnel

- Identifies how effectively recipes drive user engagement and participation.
- Metrics used: Recipe popularity, Average time spent, Likes per recipe, Comments per recipe

Architecture

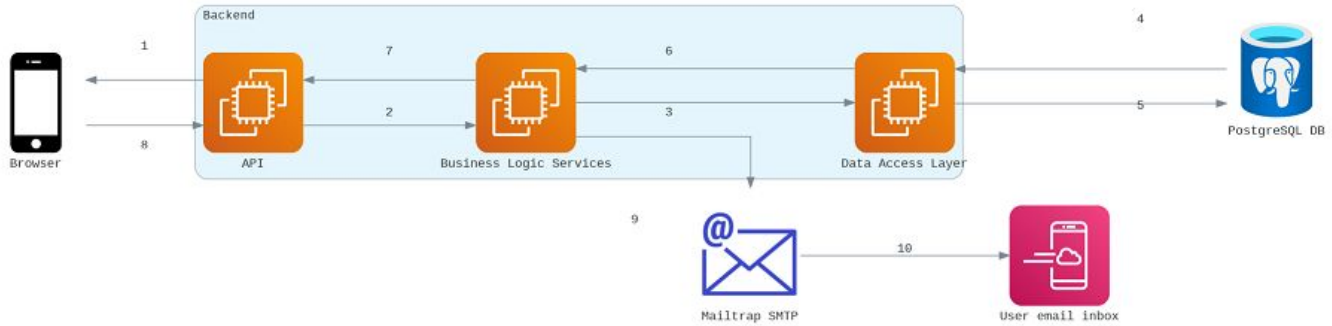
This architecture diagram describes a web application built using ASP.NET Core MVC, divided into multiple layers, utilizing Razor Pages for the frontend, and interacting with a PostgreSQL database and an external SMTP service for email delivery.

1. User – Represents the client interacting with the web application.
2. Backend Cluster:
    - ASP.NET Core MVC (Web) – The web application responsible for handling user requests, serving the interface, and facilitating communication between the frontend and backend layers.
    - Business Logic Services (Business) – The business logic layer where application rules and processing occur.
    - Data Access Layer (Data) – The data access layer that manages interactions with the database.
3. PostgreSQL Database (PostgreSQL DB) – A centralized database that stores and retrieves data through the data access layer.
4. Frontend Cluster:
    - Razor Pages– The frontend component responsible for rendering pages to be displayed in the user's browser.
5. Mailtrap SMTP – An external email service used to send emails, invoked by the business logic to deliver messages to the user.



Interactions

Interaction ID	From	To	Description
1	Browser	API	Makes HTTP requests (GET/POST/DELETE/PUT) to access or modify resources in the backend.
2	API	Business Logic Services	API forwards requests to the business logic layer for processing and validation.
3	Business Logic Services	Data Access Layer	Business logic sends queries or commands to interact with the database.
4	Data Access Layer	PostgreSQL DB	Executes SQL queries (e.g., SELECT, INSERT, UPDATE) to read/write data.
5	PostgreSQL DB	Data Access Layer	Returns results of database operations back to the data access layer.
6	Data Access Layer	Business Logic Services	Provides processed data or results from the database to the business logic layer.
7	Business Logic Services	API	Sends response or processed data back to the API layer for delivery to the client.
8	API	Browser	Sends HTTP responses (JSON, XML, HTML) back to the browser for user interaction.
9	API	Mailtrap SMTP	Sends requests to the SMTP service to deliver transactional or notification emails.
10	Mailtrap SMTP	User email inbox	Delivers emails (e.g., account verification, password reset) to the user's email inbox.

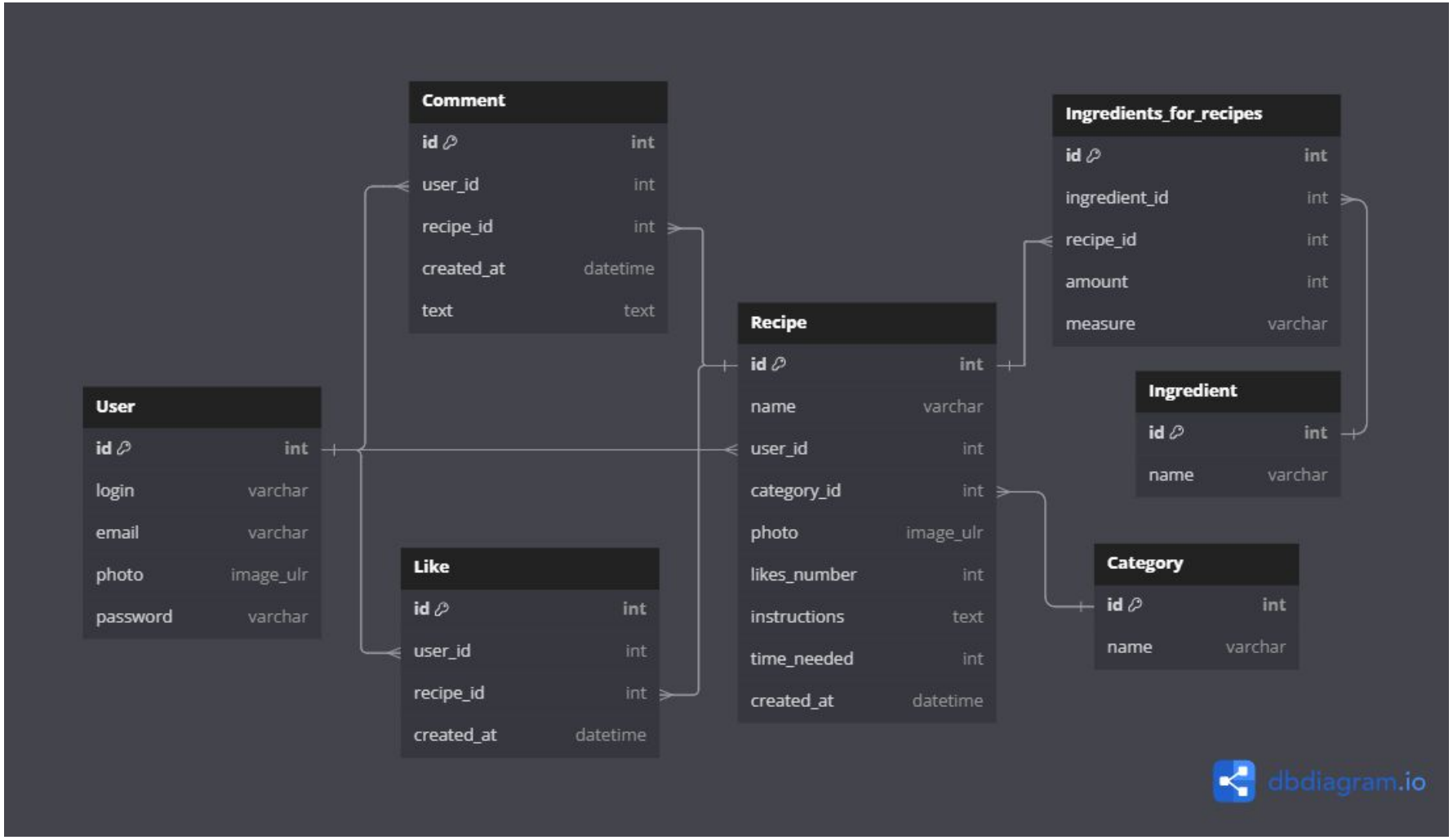


Possible failures and responses to them

Interaction ID	Possible Failure	Description	Response/Triggered Action
1	Invalid HTTP request	Browser sends malformed or unauthorized requests to the API.	Return HTTP 400/401/403 response; log the issue; notify the user of request error.
	Network connectivity issue	User's device cannot reach the API due to network failure.	Show "Network Error" on the browser; log timeout events; retry after delay.
2	API routing failure	API fails to forward the request to the correct business logic service.	Return HTTP 500 response; log detailed error for debugging.
	API authentication/authorization failure	Invalid or missing tokens during request validation.	Return HTTP 401/403; notify the user of authentication/authorization failure.
3	Missing or incorrect query parameters	Business logic receives invalid or incomplete data from the API.	Log invalid requests; return HTTP 400 response with validation error details.
	Business logic processing error	Bug or failure in processing logic, such as null pointer exceptions.	Return HTTP 500; log the error; trigger an alert to the development team.
4	Database connection timeout	Data Access Layer cannot establish a connection to PostgreSQL DB.	Log timeout error; trigger error response to API; retry with exponential backoff.
	Query execution failure	Invalid SQL query or resource deadlock during query execution.	Log query error; return HTTP 500 to API with "Internal Server Error" message.
5	Database schema mismatch	Query attempts to access nonexistent or mismatched schema/table.	Log schema mismatch error; notify database administrator.
	Insufficient database resources	Database runs out of memory or connections due to high traffic.	Trigger a scaling event; log resource exhaustion; return HTTP 503 to clients.

Interaction ID	Possible Failure	Description	Response/Triggered Action
6	Data serialization/deserialization failure	Errors in converting database results into usable objects.	Log serialization error; return HTTP 500 to API; fix data formatting issue.
	Inconsistent data integrity	Data Access Layer returns corrupted or incomplete data to Business Logic Services.	Validate data integrity; log issue; trigger alert to review database consistency.
7	API response timeout	Business Logic Services fail to provide a timely response to the API.	Return HTTP 504 to browser; log timeout event; investigate performance bottlenecks.
	Unhandled exception in business logic	Business Logic Services encounter unexpected runtime errors.	Return HTTP 500 to API; log stack trace; notify development team for debugging.
8	Browser API communication timeout	Slow server response causes the browser to time out.	Show "Request Timed Out" message; retry the request; log performance issue.
	Invalid API response format	API returns data in an unexpected or unreadable format.	Show "Error Processing Response" in browser; log response validation error.
9	SMTP service unavailability	Mailtrap SMTP service is down or unreachable.	Queue email for retry; log outage and notify infrastructure team of SMTP issue.
	Email formatting failure	API sends malformed email content or headers to SMTP.	Log formatting error; retry with corrected email structure; notify developers.
10	Email delivery failure	User email inbox rejects or blocks the email (e.g., invalid address, spam filter).	Log delivery failure; retry delivery; notify user to update email address if needed.
	Slow SMTP processing	Email gets delayed due to high SMTP service load.	Log delayed delivery; monitor SMTP performance; escalate if delay persists.

ER-diagram


Comment
id int
user_id int
recipe_id int
created_at datetime
text text
Ingredients_for_recipes
id int
ingredient_id int
recipe_id int
amount int
measure varchar
Recipe
id int
name varchar
user_id int
category_id int
photo image_ulr
likes_number int
instructions text
time_needed int
created_at datetime
Ingredient
id int
name varchar
User
id int
login varchar
email varchar
photo image_ulr
password varchar
Like
id int
user_id int
recipe_id int
created_at datetime
Category
id int
name varchar
dbdiagram.io

Security Considerations

1. Sensitive Fields:
   - Ensure the password field in the User table is hashed and not stored in plaintext.
   - Protect email and user_id fields to maintain privacy.
2. Access Control:
   - Implement role-based access control (RBAC) for admin, user, and public access levels.
   - Restrict access to modify or delete records to authorized users.
3. Data Moderation:
   - Implement moderation for text fields in Comment to avoid inappropriate content.

Retention policy by table

User table

- Active Users: Retain data indefinitely while the user account is active.
- Inactive Users: Delete accounts and associated data after 2–3 years of inactivity unless the user requests deletion sooner.
- Deleted Accounts: Anonymize or delete personal data immediately or within 30–90 days after account deletion. Retain transaction logs (e.g., comments, recipes) without personal identifiers.

Recipe table

- Active Recipes: Retain indefinitely unless the creator deletes them.
- Deleted Recipes: Retain for a grace period (e.g., 30 days) to allow recovery in case of accidental deletion. Afterward, permanently delete.

Category table

- Categories can be retained indefinitely as they are non-sensitive and reusa

Ingredient table

- Ingredients can be retained indefinitely since they are non-sensitive.

Comment table

- Active Comments: Retain indefinitely unless flagged for moderation or deleted by the user.
- Deleted Comments: Anonymize user information and retain only for statistical or moderation purposes (e.g., 30 days before permanent deletion).

Like table

- Retain indefinitely or until the associated recipe/user account is deleted.
- For deleted accounts, anonymize user data.

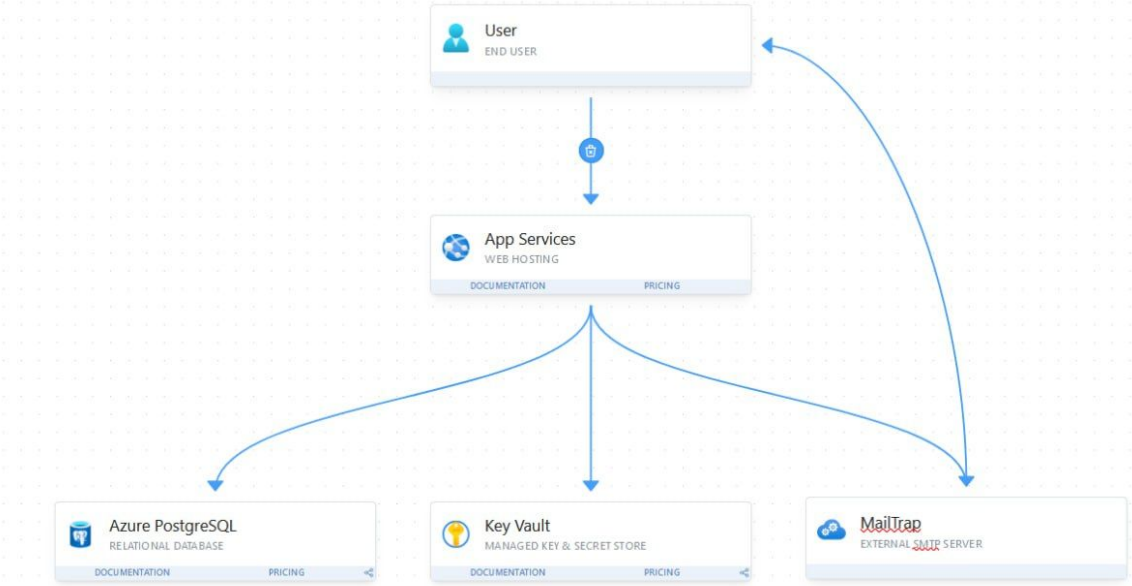
Ingredients_for_recipes table

- Retain as long as the associated recipe exists.
- Delete when the recipe is deleted.

Deployment model

Components:

- User: Represents the person or entity interacting with the system through a web application or service.
- App Services (Web Hosting): A managed hosting service where the application logic is deployed. The core of the system that connects with other Azure services and external components.
- Azure PostgreSQL (Relational Database): A managed relational database service for storing structured application data. Likely used to store persistent data for the application.
- Key Vault (Managed Key & Secret Store): A secure service for managing application secrets, encryption keys, and certificates. Ensures sensitive information is stored securely and accessed only by authorized entities.
- MailTrap (External SMTP Server): An external service for sending and testing emails. Useful for email functionality in development and production environments.



Workflow:

- User Interaction: The end user sends requests to the application hosted on App Services.
- App Services Integration: Processes user requests and interacts with: * Azure PostgreSQL to retrieve or store application data. * Key Vault to securely access secrets (e.g., database credentials, API keys). * MailTrap to send emails (e.g., notifications, verification emails).

Metrics

Metric	Description	Measurement Unit	Collection Method	Connection to Infrastructural Resources
CPU Usage	Percentage of CPU resources utilized by servers.	Percentage (%)	- Use Azure Monitor, top commands, or system logs.	Relates to Azure Servers hosting data, business, and web layers.
Memory Usage	Amount of RAM used by the application or services.	MB / GB	- Use system monitoring tools (e.g., Azure Monitor, Grafana).	Indicates load on Azure Servers.
Disk I/O	Read/write operations performed on the server's disk.	Operations per second	- Monitor with Azure Diagnostics or tools like iostat.	Tied to database and backup storage.
Database Connections	Number of active connections to the database.	Count	- Use PostgreSQL monitoring tools (e.g., pg_stat_activity).	Relates to PostgreSQL Database hosted on Azure.
Query Latency	Time taken to execute database queries.	Milliseconds (ms)	- Use PostgreSQL EXPLAIN or Azure Database Insights.	Key performance metric for the data layer.
Network Latency	Time taken for a data packet to travel from source to destination.	Milliseconds (ms)	- Use Azure Network Watcher or ping tests.	Connected to Azure networking services.
Network Throughput	Volume of data transmitted over the network per second.	Mbps / Gbps	- Use Azure Monitor or tools like Wireshark.	Impacts communication across layers and SMTP/email services.

Metric	Description	Measurement Unit	Collection Method	Connection to Infrastructural Resources
Error Rate	Percentage of requests resulting in errors (HTTP 4xx/5xx).	Percentage (%)	- Track HTTP status codes via Azure Application Insights or logs.	Relates to Web API and server communication.
Request Throughput	Number of requests processed per second.	Requests per second	- Monitor using load balancers or Azure Application Insights.	Affects scalability of the web and business layers.
Session Count	Number of active user sessions.	Count	- Track session state in web application logs or using Redis/Azure Cache metrics.	Connected to session management resources.
SMTP Queue Size	Number of emails queued for sending.	Count	- Monitor Mailtrap service metrics.	Tied to SMTP email service.
Uptime	Percentage of time the application is available.	Percentage (%)	- Use Azure Service Health.	Key metric for overall application reliability.
Azure Resource Utilization	Total usage of allocated CPU, memory, and bandwidth for Azure resources.	Percentage (%)	- Track via Azure Portal and Azure Monitor.	Indicates resource efficiency for all Azure-hosted services.
Backup Frequency and Size	Frequency and size of database backups.	Count (per time period) / GB	- Use Azure Backup reports or database logs.	Tied to PostgreSQL Database and backup storage.

Alerting

Metric	Critical Value	Alert Type	Impact Level	Mitigation Plan
CPU Usage	> 80% for 5+ minutes	Warning	Medium	Scale up VM size or enable auto-scaling in Azure.
	> 90% for 2+ minutes	Error	High	Investigate processes causing high usage and optimize resource-intensive operations.
Memory Usage	> 80% of total RAM	Warning	Medium	Optimize memory-intensive queries or increase allocated memory.
	> 90% of total RAM	Error	High	Restart services to free memory or consider auto-scaling options.
Disk I/O	> 500ms latency	Warning	Medium	Optimize disk-intensive queries or consider upgrading to SSD storage.
	> 1000ms latency	Error	High	Investigate slow queries or I/O-intensive operations causing high disk usage.
Database Connections	> 80% of connection limit	Warning	Medium	Increase max connection limits or optimize connection pooling.
	Exceeds connection limit	Error	High	Investigate connection leaks, add retries, or scale up database resources.
Query Latency	> 500ms	Warning	Medium	Add indexing, optimize queries, or cache frequent queries.
	> 1000ms	Error	High	Investigate long-running queries and adjust database resources if needed.

Metric	Critical Value	Alert Type	Impact Level	Mitigation Plan
Network Latency	> 100ms	Warning	Medium	Check for network congestion or slowdowns; optimize communication paths.
	> 500ms	Error	High	Investigate connectivity issues or increase network bandwidth.
Network Throughput	> 90% of allocated bandwidth	Warning	Medium	Scale up bandwidth or optimize data transfer.
	Saturation of bandwidth	Error	High	Throttle non-critical traffic and expand network resources.
Error Rate	> 2%	Warning	Medium	Monitor logs for recurring issues; fix errors in API or backend processes.
	> 5%	Error	High	Escalate investigation into core functionality or server health.
Request Throughput	> 90% of server capacity	Warning	Medium	Scale horizontally (add more instances) to handle additional requests.
	Exceeds capacity	Error	High	Implement rate limiting or throttling policies to manage spikes.
SMTP Queue Size	> 100 emails queued	Warning	Low	Investigate slow email delivery and review SMTP service configuration.
	> 500 emails queued	Error	Medium	Retry failed emails and review email sending rate limits.

Functional requirements

- The system should authenticate users before granting access to specific functionalities or data.
- The system must allow guests to sign up or login when trying to open any page they don’t have access to.
- The system must send verification email once a new user is registered.
- The system should validate user inputs to ensure they meet specific criteria, such as data type, length, or format.
- Users must be able to view their own account.
- Users should be able to update their account details, such as username or password.
- The system must send a verification email when resetting the password is initialized by the user.
- Users must be able to view and sort recipes made by other users.
- Users must be able to create new recipes records.
- Users must be able to update their recipes records.
- Users must be able to comment on any recipe.
- Users must be able to like any recipe but their own.
- Users must be able to look through recipes they liked.
- The system must provide the ability to log out the account.

Non-functional requirements

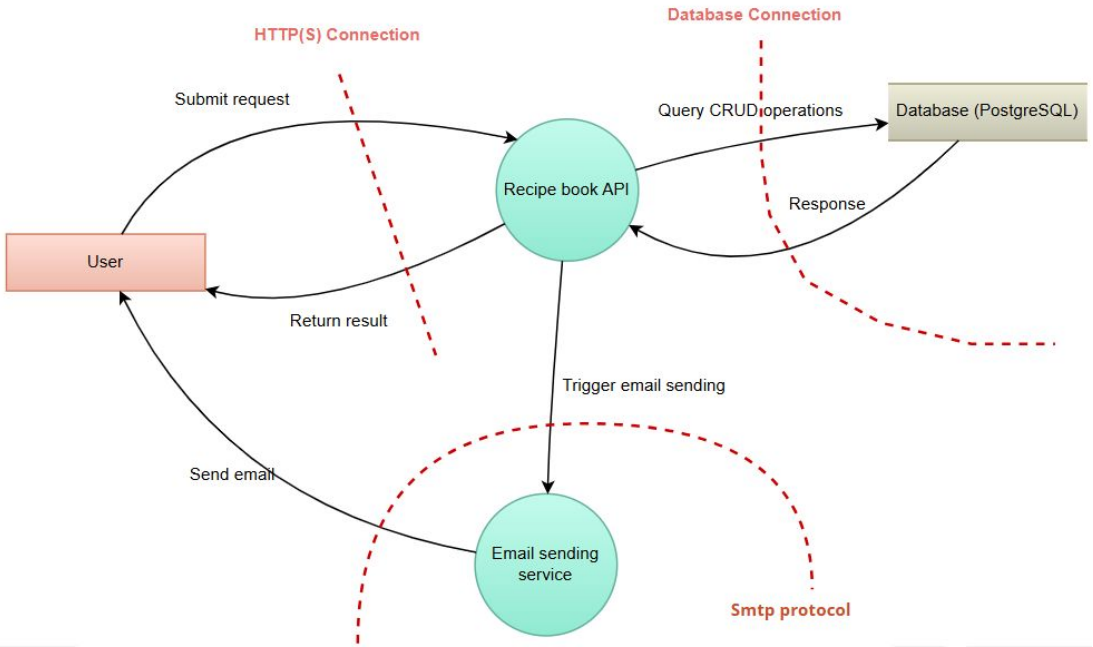
- The system must allow the user to navigate the interface easily.
- The most frequently used features must be easily accessible and reached with minimal navigation.
- Application usage must be provided for both desktop and phone users.
- All passwords must be hashed before saving them to the database.
- The system should be able to handle and recover from errors without data loss or incorrect data processing.
- The system must send email notifications within 5 minutes of the triggering event.

Security model

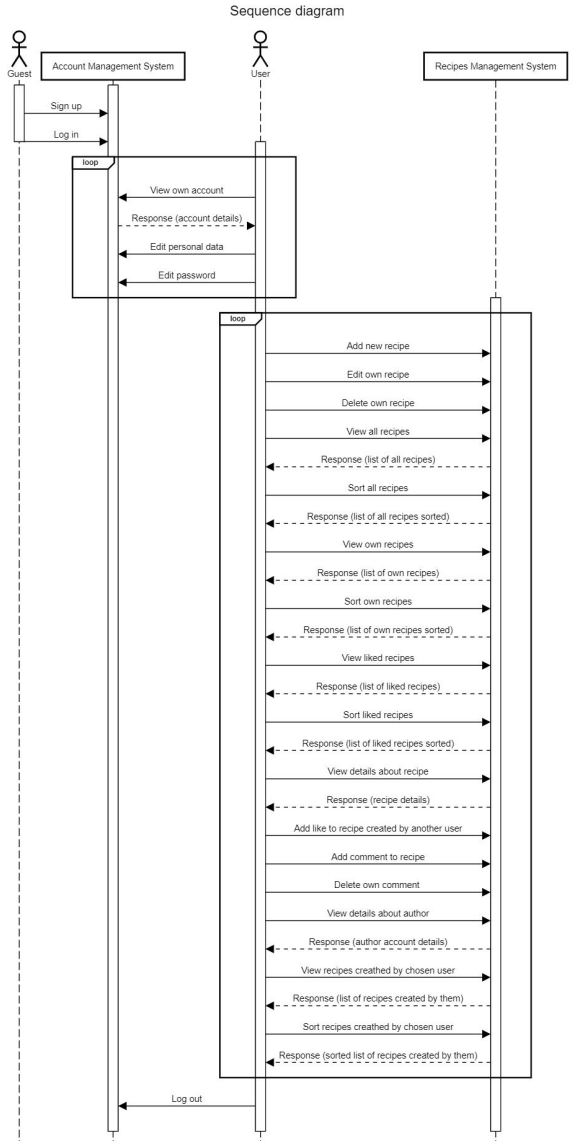
Threat	STRIDE Category	Likelihood	Impact	Mitigation Strategy
Brute force attacks to guess passwords.	Spoofing	High	High	- Enforce strong passwords. - Use rate limiting. - Implement MFA (Multi-Factor Authentication).
SQL injection via user inputs (e.g., comments, recipes).	Tampering, Information Disclosure	High	High	- Use parameterized queries and ORM frameworks. - Validate and sanitize all inputs.
Email spoofing or phishing via email system.	Spoofing, Tampering	Medium	High	- Add clear disclaimers in emails to warn users.
Excessive API calls causing denial of service (DoS).	Denial of Service	Medium	High	- Implement rate limiting and request throttling. - Monitor API usage and block suspicious IPs.
Elevation of privilege via insecure role management.	Elevation of Privilege	Medium	High	- Use RBAC (Role-Based Access Control) with least privilege principles. - Log and audit role changes.
Insecure error messages leaking sensitive system information.	Information Disclosure	Medium	Medium	- Avoid exposing stack traces or sensitive details in error responses. - Use generic error messages for end-users.
Privilege escalation via insecure database roles or configurations.	Elevation of Privilege	Low	High	- Enforce strict RBAC within the database. - Regularly review and audit role permissions.
Insider threat or misconfiguration leading to data leakage.	Information Disclosure	Low	High	- Enable logging and alerting for configuration changes.
Vulnerabilities in third-party libraries or SDKs.	Tampering, Denial of Service	Medium	High	- Conduct security reviews of dependencies.
Session fixation attacks where old session tokens are reused.	Spoofing	Low	High	- Regenerate session tokens upon user authentication. - Use strict expiration and idle timeout policies.
Man-in-the-middle (MITM) attacks intercepting data in transit.	Information Disclosure, Tampering	Medium	High	- Enforce HTTPS with a strong TLS configuration. - Use HSTS to prevent protocol downgrades.

STRIDE is a threat modeling method from Microsoft, which classifies threats as:

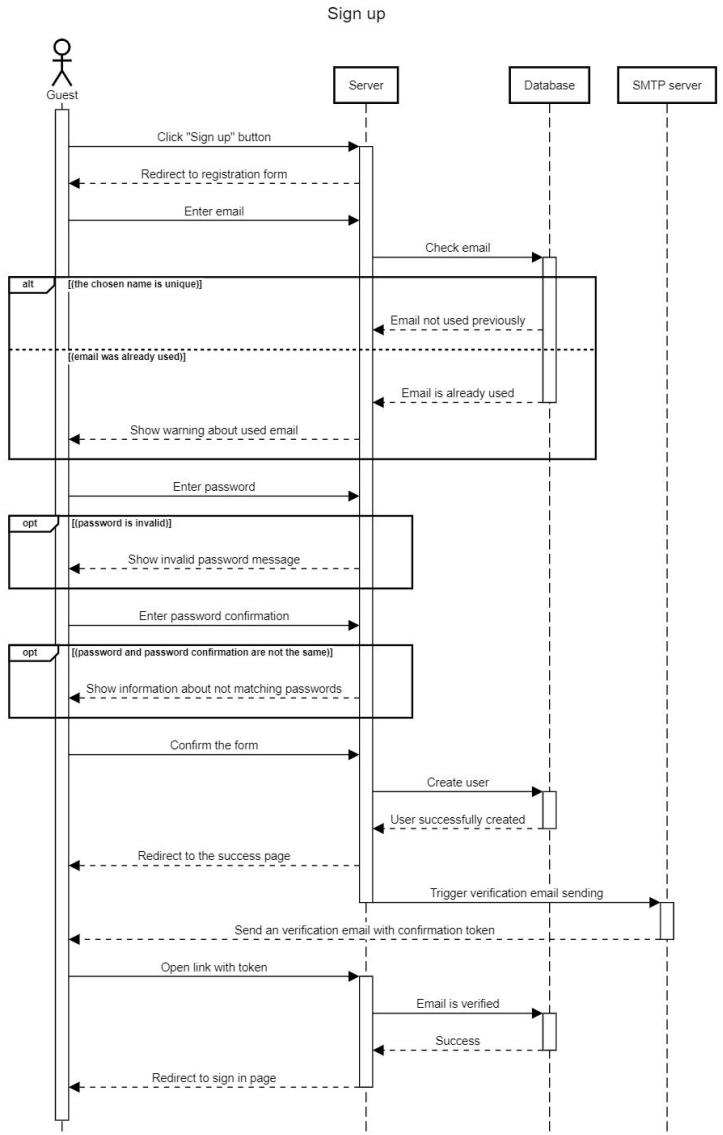
- Spoofing: Attackers impersonating users via weak authentication.
- Tampering: Unauthorized modification of recipes or user data.
- Repudiation: Users claiming they didn't perform certain actions (e.g., commenting).
- Information Disclosure: Unauthorized access to sensitive fields (e.g., password, email).
- Denial of Service: Overloading database queries or connections.
- Elevation of Privilege: Unauthorized escalation to admin privileges.



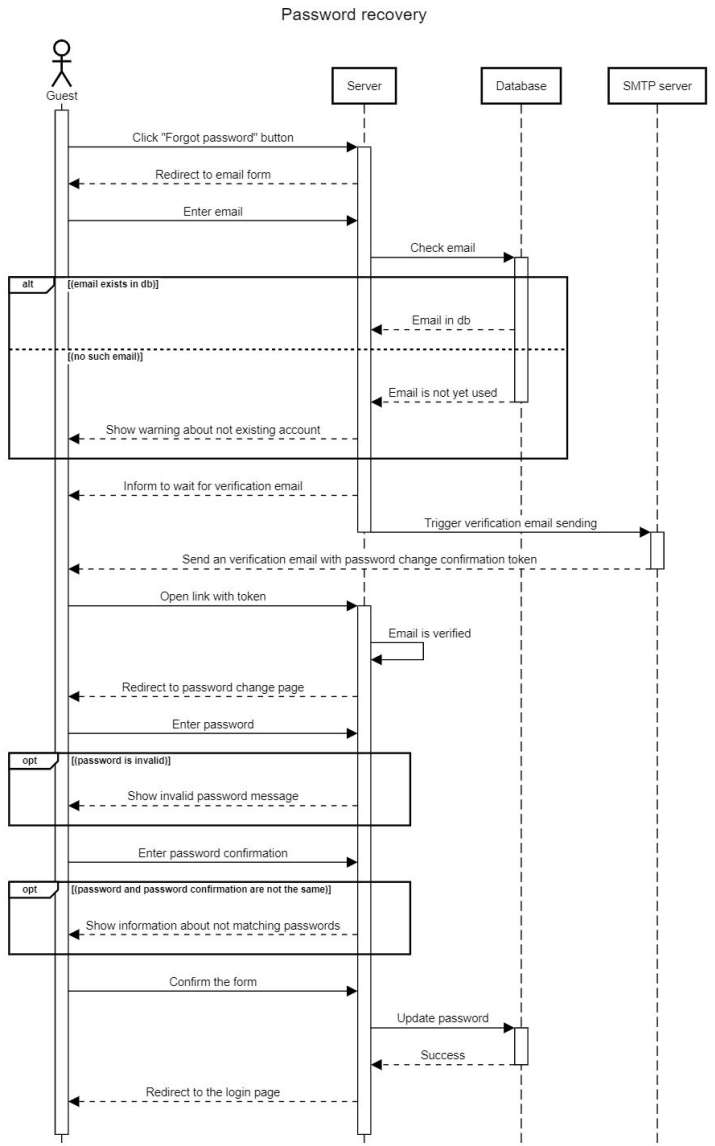
Application sequence diagram



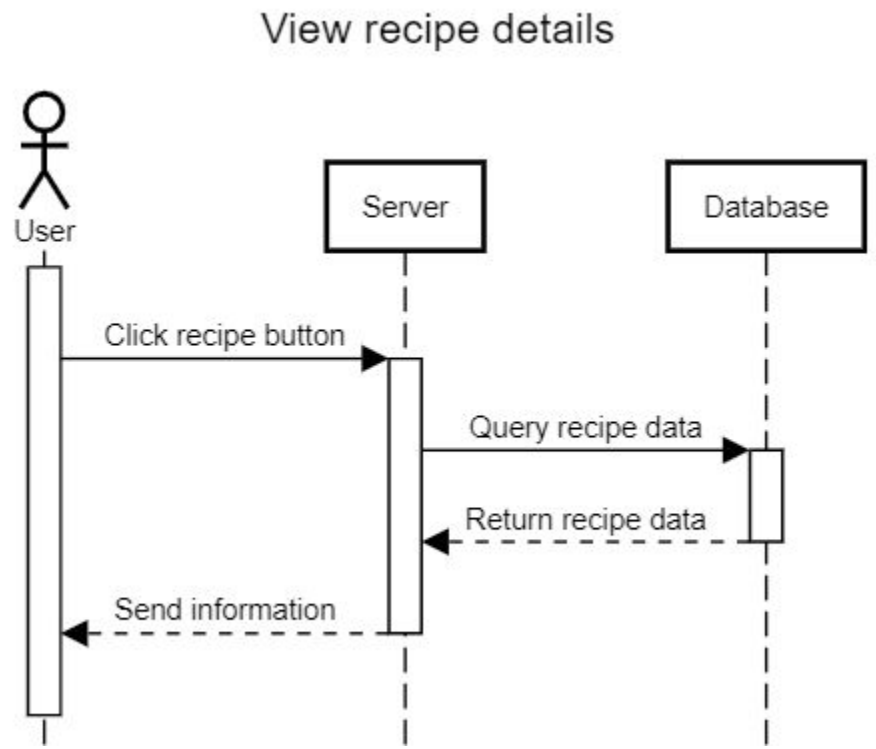
Sign up process sequence diagram



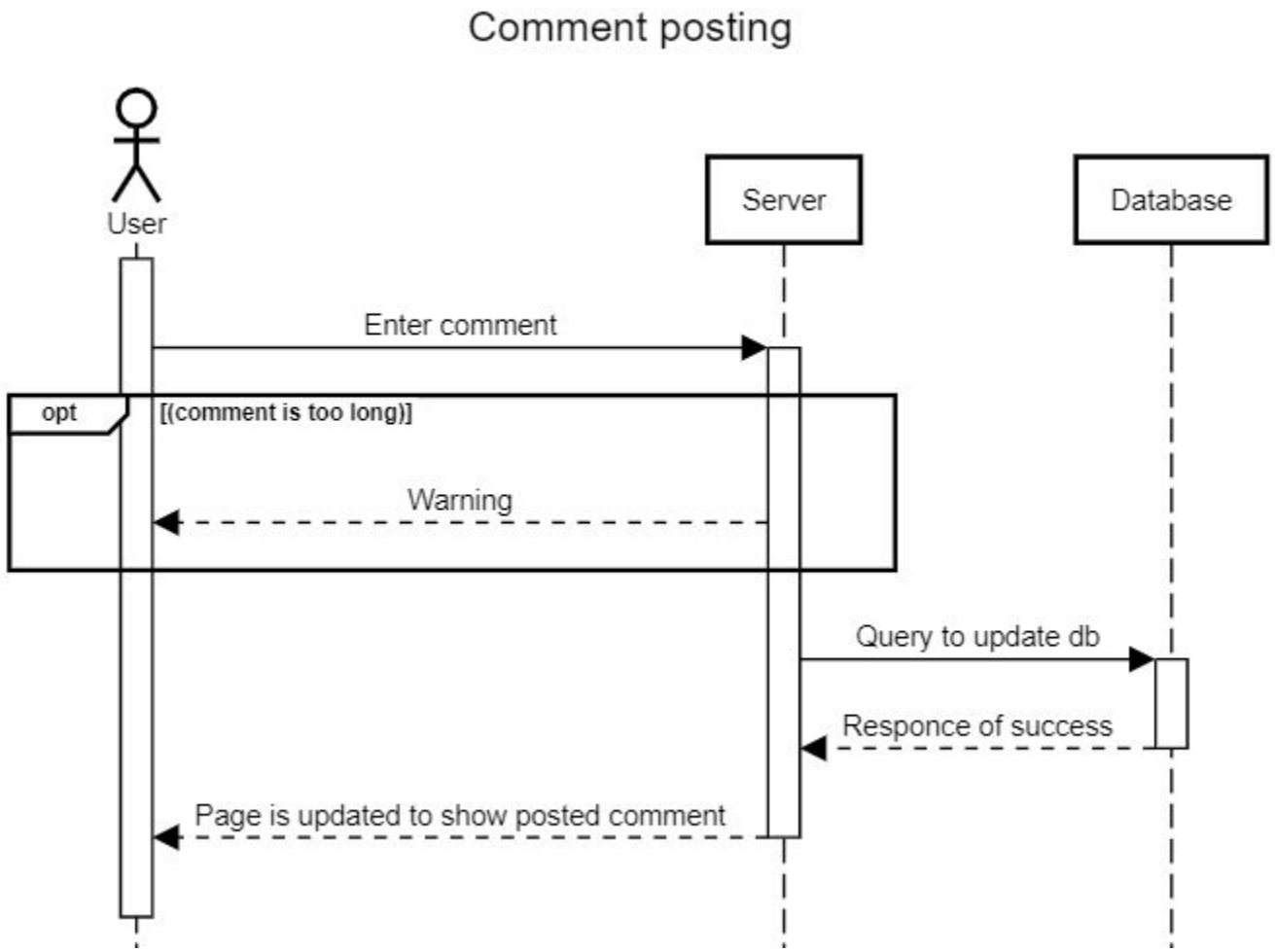
Password recovery sequence diagram



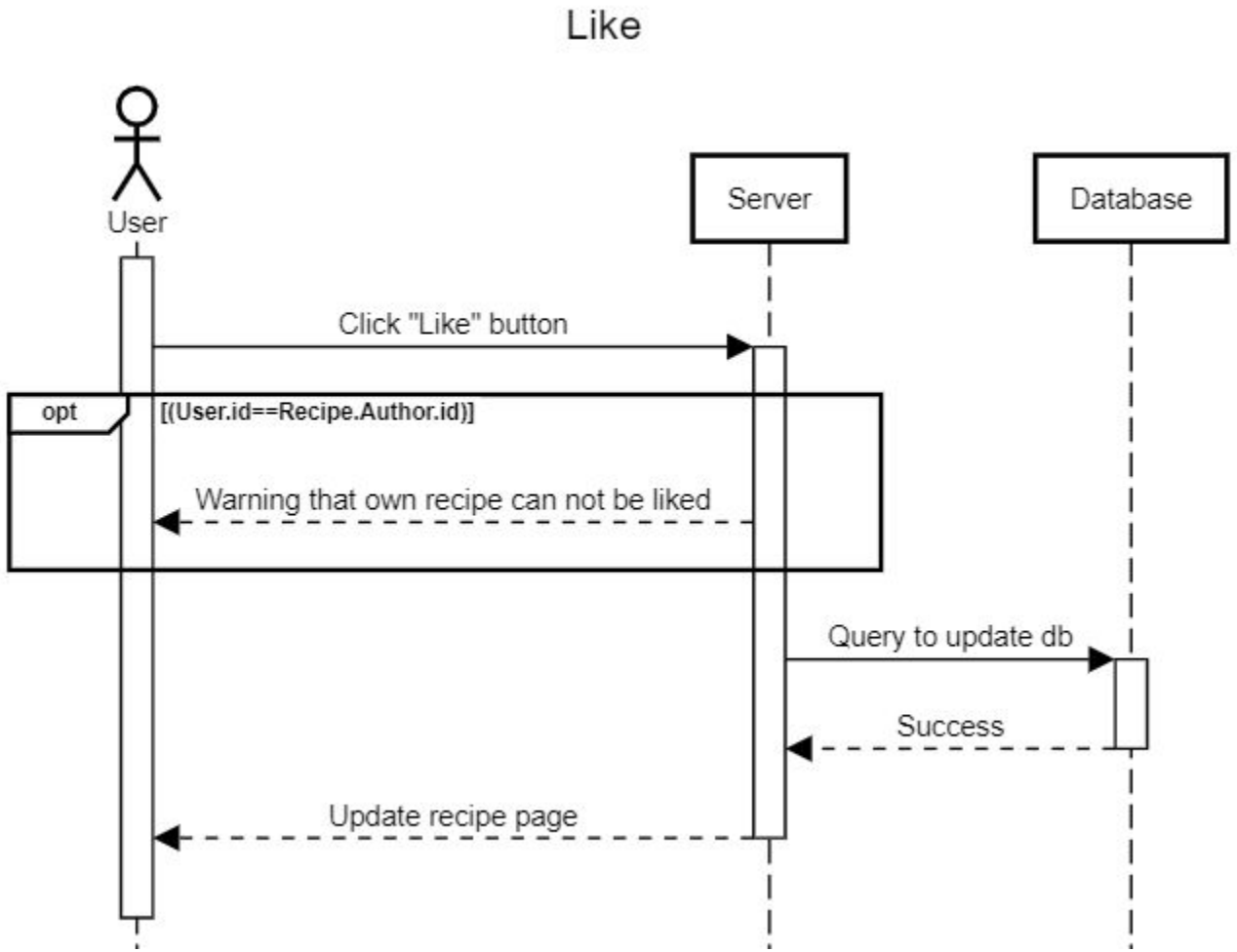
Recipe view sequence diagram



Comment adding sequence diagram

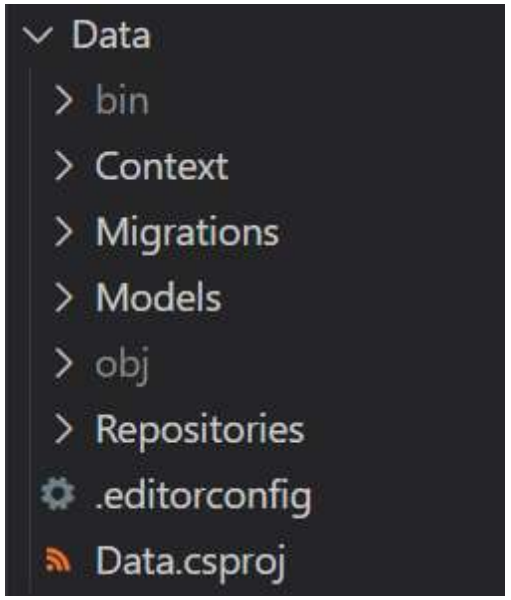


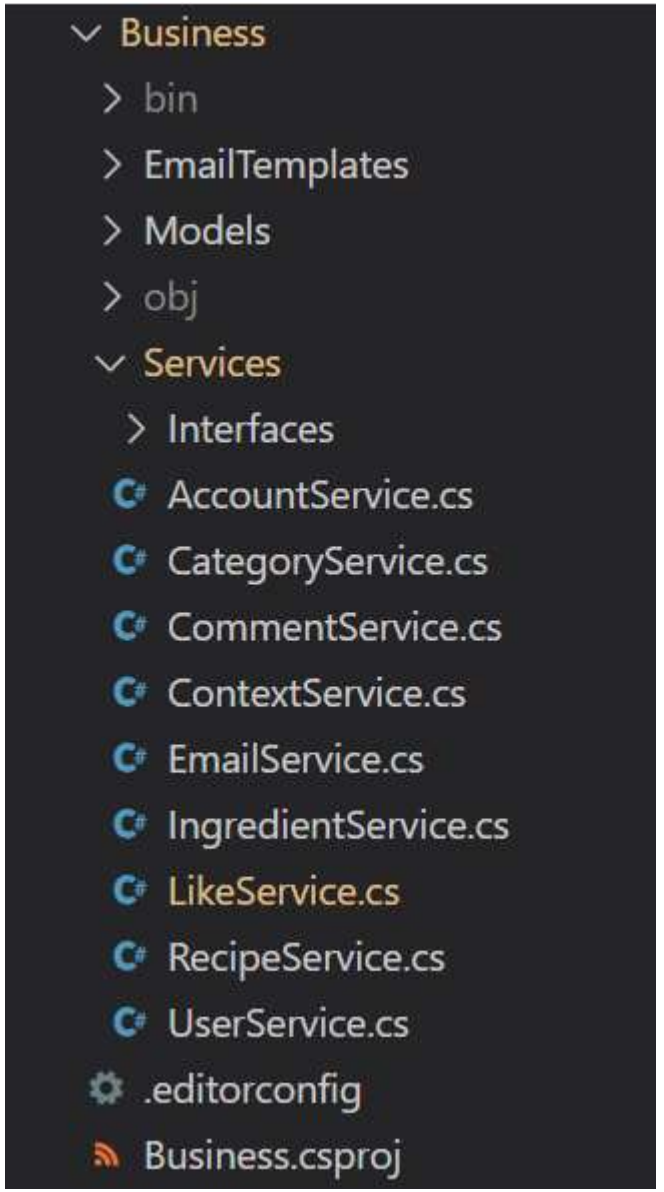
Like adding sequence diagram

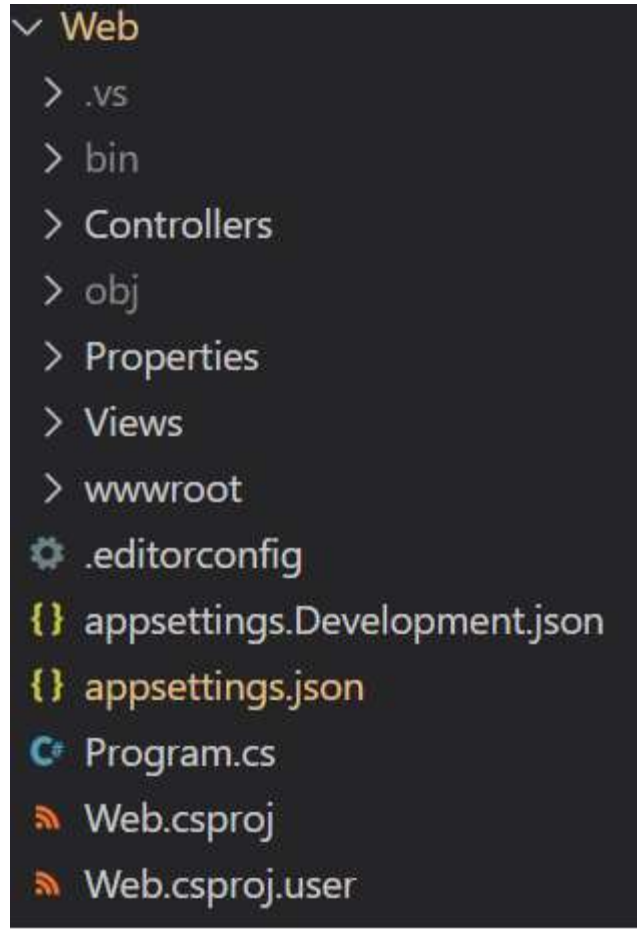


Demo – code structure

According to chosen architecture, we implemented 3 layers for our project using .Net.

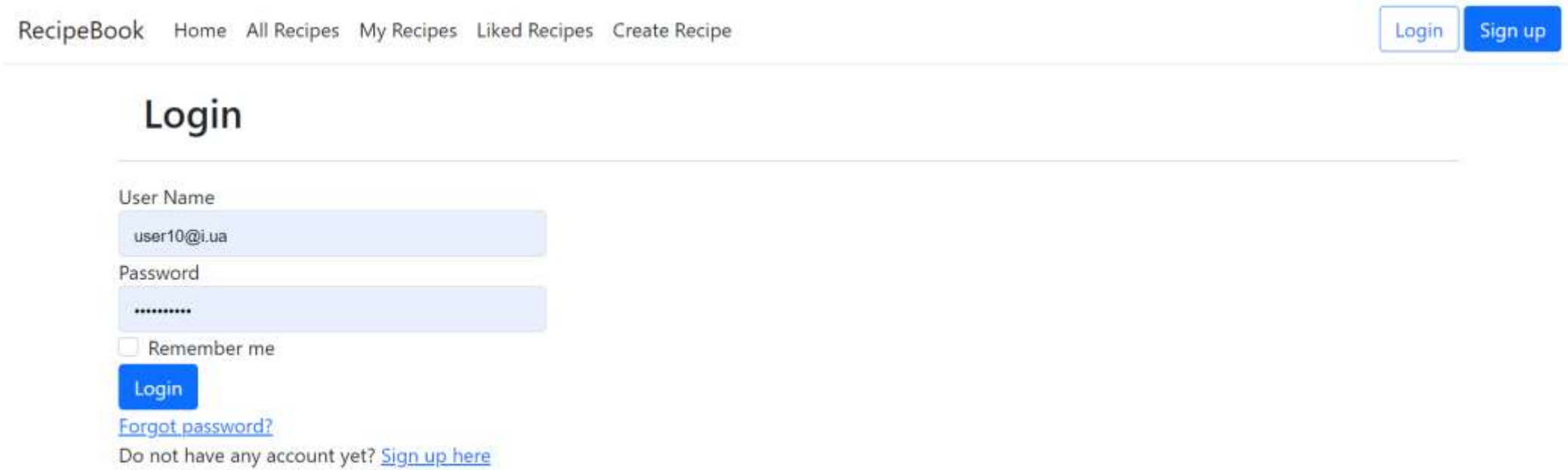






Demo – views

Login page:

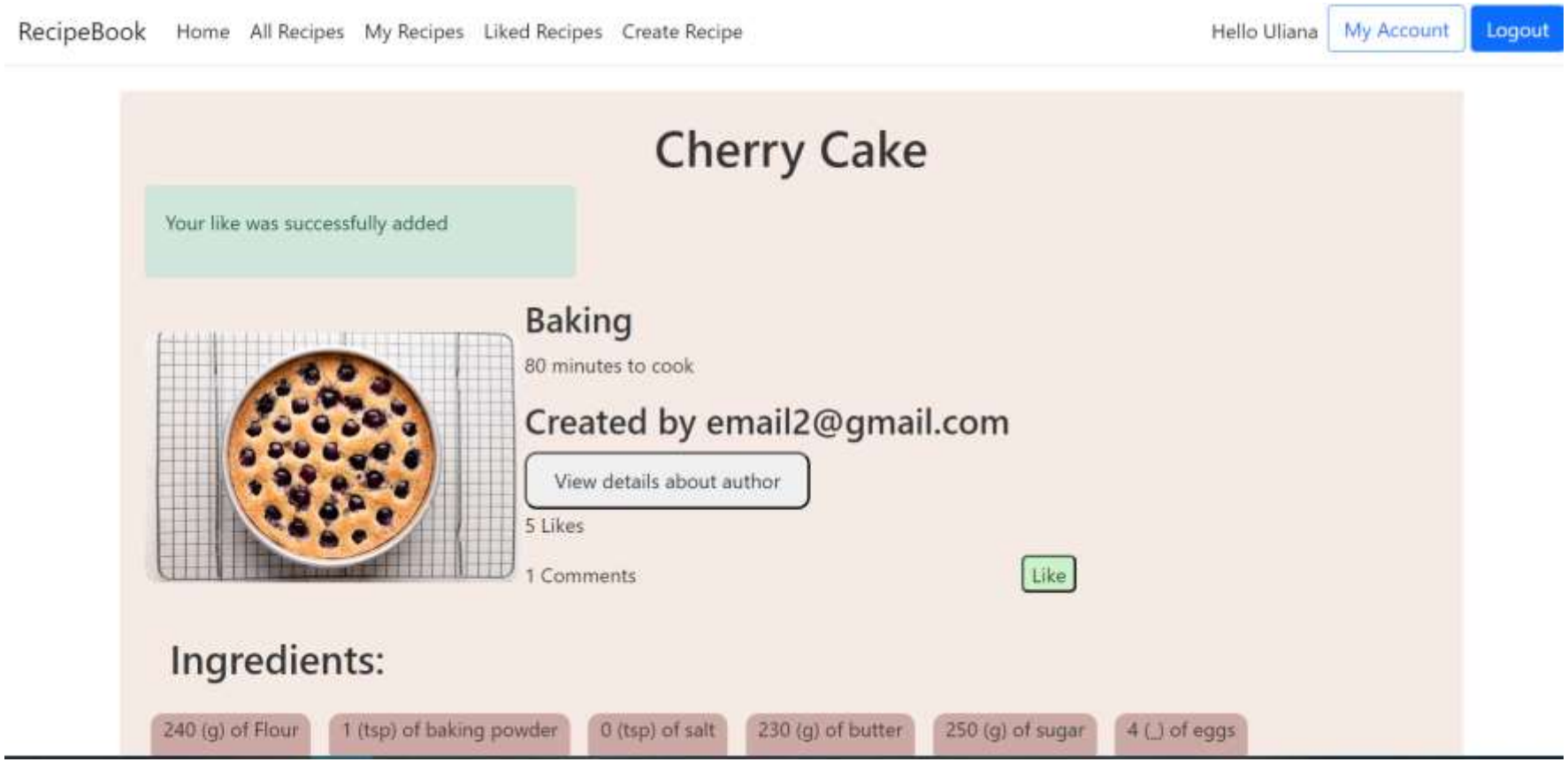


Demo – views

List of recipes:

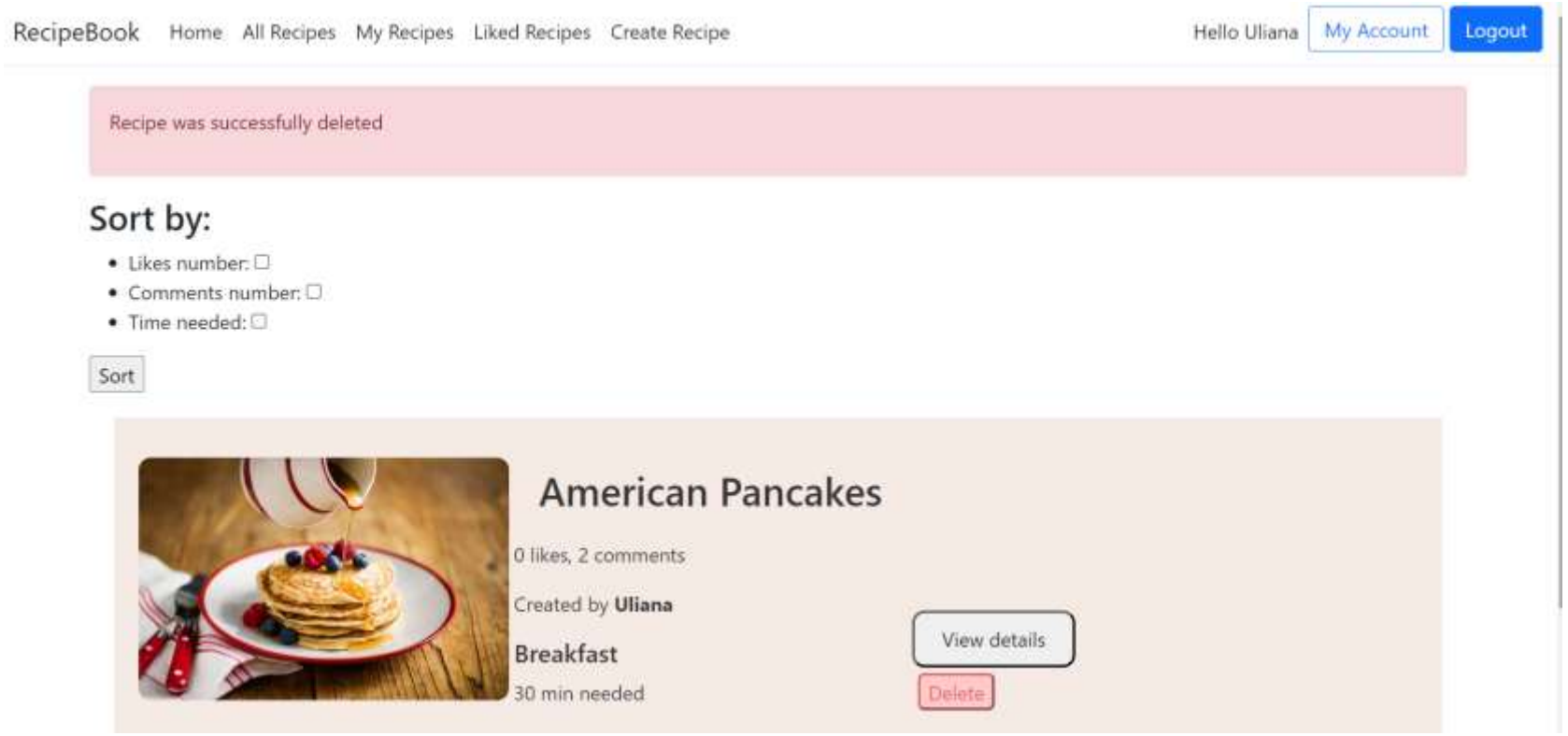
Demo – views

Recipes details (user can add likes only to posts, created by others):



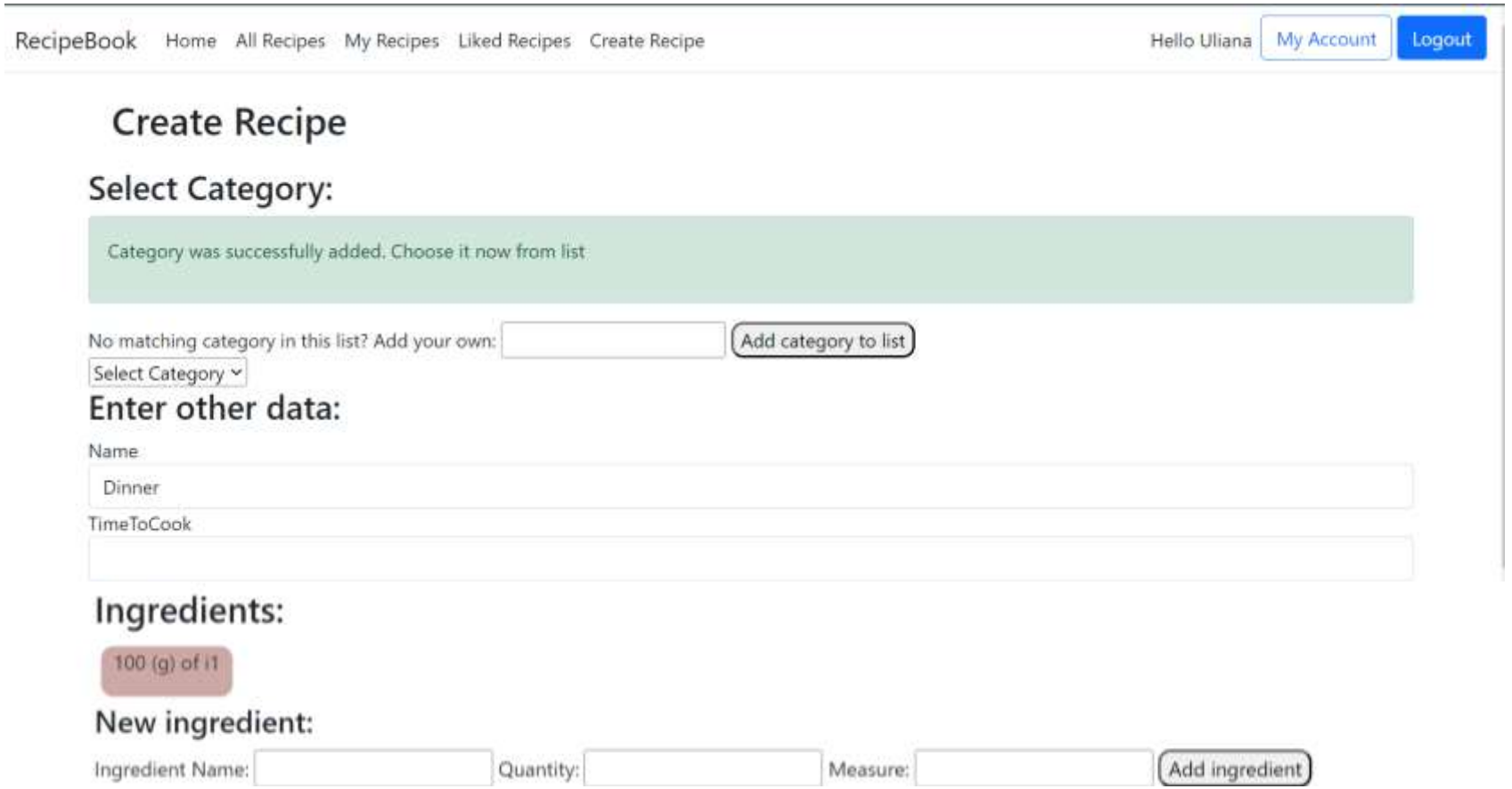
Demo – views

Users can delete only their own recipes:



Demo – views

Recipe creation:



Conclusion

We created a digital platform to simplify and enhance the cooking experience. It allows users to organize their favorite recipes, add personal creations, and share them with others. With detailed ingredient lists, step-by-step instructions, and meal planning tools, Recipe Book offers the convenience of accessing recipes anytime, anywhere. It’s a valuable tool for anyone who loves cooking and wants to streamline the process of discovering, saving, and organizing recipes.

Q&A