Software Requirements Specifications

Introduction

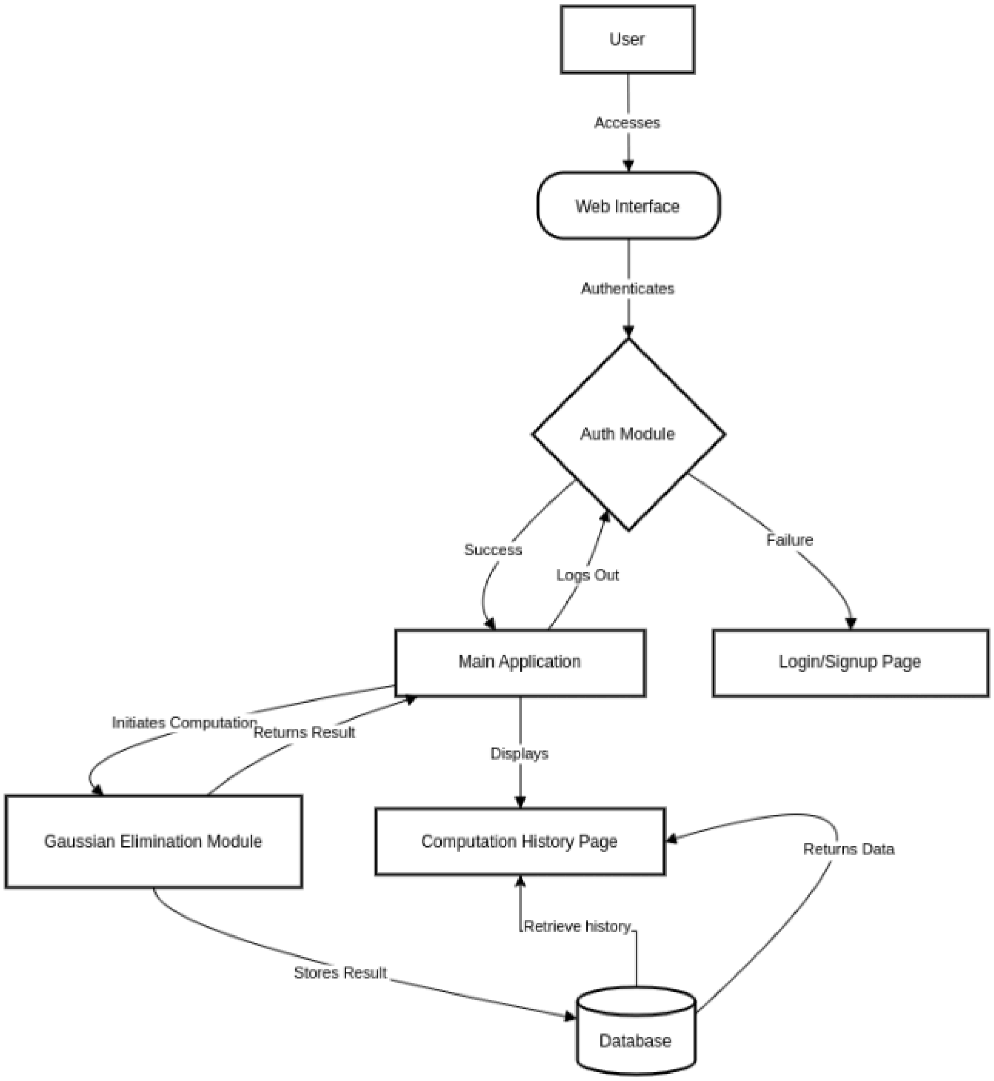
Scope

This web application enables users to perform random Gaussian elimination computations through a simple, accessible interface. Users can register for an account, log in securely, and input parameters such as the number of unknowns and coefficient range to initiate calculations. The system then quickly processes these inputs, performing complex Gaussian elimination operations. Users can view their computation history, allowing them to track and review past calculations, including details like matrix size and processing time. This tool will enable user to solve systems of linear equations of different sizes from very small 4 - 1000 unknowns. The application offers a user-friendly experience for both initiating new calculations and reviewing previous work, making it a valuable resource for students, researchers, or professionals dealing with linear algebra problems.

Abbreviations

UML: Unified Modeling Language. Standardized visual notation for system design.
SQL: Structured Query Language. Language for managing relational databases.
ORM: Object-Relational Mapping. Technique for converting data between databases and objects.
API: Application Programming Interface. Set of protocols for building software.
HTML: Hypertext Markup Language. Standard language for creating web pages.
CSS: Cascading Style Sheets. Language for styling web pages.
HTTP: Hypertext Transfer Protocol. Protocol for web communication.
HTTPS: Hypertext Transfer Protocol Secure. Secure version of HTTP.
UI: User Interface. Point of human-computer interaction.
SHA: Secure Hash Algorithm. Family of cryptographic hash functions.
JSON: JavaScript Object Notation. Lightweight data interchange format.
CRUD: Create, Read, Update, Delete. Basic database operations.

UML Diagram



Assumptions and Dependencies

1 User Mathematical Background

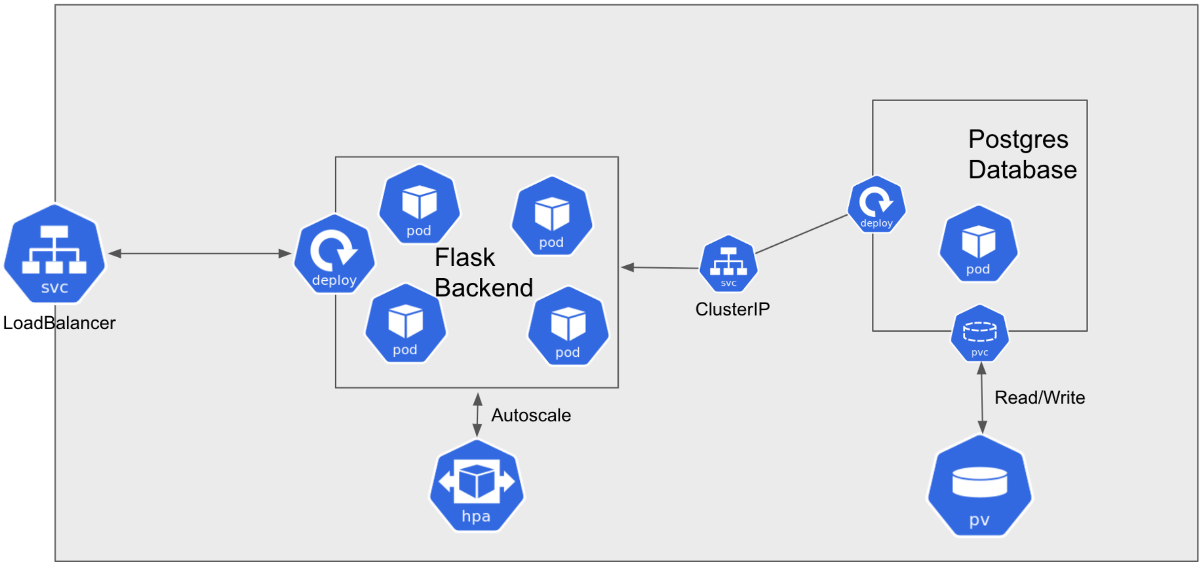
Users are assumed to have a basic understanding of linear algebra and Gaussian elimination concepts.

2 Web Browser Compatibility

The application is designed to work with modern web browsers (Chrome, Firefox, Safari, Edge) released within the last two years.

3	Server Environment	The server environment is assumed to support Python 3.7+ and PostgreSQL 12+.
4	Network Connectivity	Users are expected to have a stable internet connection with sufficient bandwidth to perform calculations and view results.
5	Maximum Matrix Size	The system is designed to handle matrices up to 1000x1000 in size. Larger matrices may result in longer computation times or system limitations.

High Level System Architecture



Functional Requirements

1. User Registration: Users can sign up with email, name, and password.
2. User Authentication: Users can log in with email and password.
3. User Logout: Authenticated users can log out.

4. Gaussian Elimination Computation: The system performs random Gaussian elimination with user-specified parameters.
5. Input Parameter Specification: Users can input the number of unknowns and range for random coefficients.
6. Computation History: The system stores and displays computation history for each user.
7. Error Handling: Appropriate error messages for invalid inputs or failed computations.
8. Home Page: A home page with a form for new computations.
9. History Page: A dedicated page for viewing computation history.
10. Remember Me Functionality: "Remember Me" option during login.
11. Flash Messages: Display notifications for important events.
12. Redirect Unauthenticated Users: Redirect to login page when accessing protected routes.
13. Prevent Duplicate Registrations: Check for existing email addresses during signup.

Non-Functional Requirements:

1. Scalability: Implement autoscaling for handling varying computation loads.
2. Performance: Optimize Gaussian elimination for matrices up to 1000x1000.
3. Security: Use SHA-256 for password hashing.
4. Database Integration: Utilize PostgreSQL for data storage.
5. Containerization: Use Docker for application containerization.
6. Orchestration: Employ Kubernetes for container management.
7. Cloud: use Azure for cloud deployment.
8. Monitoring and logging: use Prometheus, Promtail, Loki and Azure Storage Account, Grafana for visualization.
9. DNS and SSL: use external-DNS and cert-manager for DNS and SSL
10. Database: use Azure SQL for managed PostgreSQL
11. CI/CD: use Github Actions for CI/CD
12. Load Balancing: Distribute computation requests across available resources. Use Traefik reverse proxy for load balancing.
13. Responsiveness: Maintain a responsive web interface during heavy loads.
14. Reliability: Implement error handling and logging for system stability and debugging.

Actors

Unauthenticated User :An individual who has not logged into the system. They can access the login and signup pages, and attempt to register or authenticate. Unauthenticated users are redirected to the login page if they try to access protected routes.
Authenticated User: A user who has successfully logged into the system. They can access the home page to initiate new Gaussian elimination computations, view their computation history, and log out of the system.

Use Cases

Register and test one computation

The user navigates to the signup page to create a new account.
The user provides their registration details (email, name, password).
The system creates the user account and stores it in the database.
The user is redirected to the login page and enters their credentials.
After successful authentication, the user is taken to the home page.
The user enters parameters for their first Gaussian elimination computation.
The system performs the computation and stores the result in the database.
The result is displayed to the user.
Returning User Login and Multiple Computations

Login and make multiple computation

The returning user navigates to the login page and enters their credentials.
Upon successful authentication, the user is redirected to the home page.
The user enters parameters for their first computation of the session.
The system performs the computation and stores the result.
The user then initiates a second computation with different parameters.
Again, the system performs the computation and stores the result.
Finally, the user requests to view their computation history.
The system retrieves the history from the database and displays it to the user.
User Logout and Subsequent Protected Page Access

Logout and protected pages access

The user, currently logged in, clicks the logout button.
The system ends the user's session and redirects them to the login page.

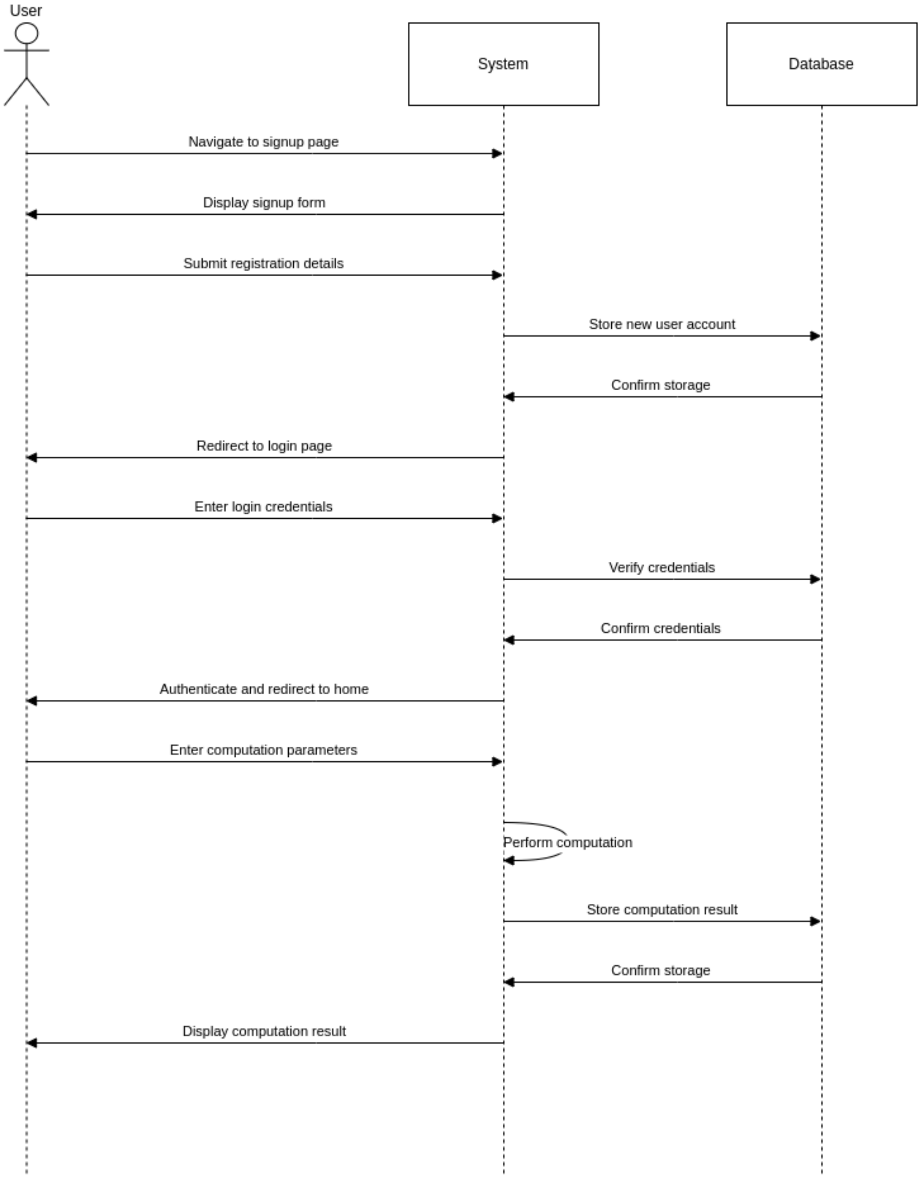
The user then attempts to access a protected page (e.g., computation history).
The system recognizes that the user is not authenticated and redirects them to the login page.
The user enters their login credentials.
The system verifies the credentials against the database.
Upon successful authentication, the system retrieves the necessary user data for the protected page.
The user is then redirected to the originally requested protected page, now with the appropriate data displayed.

Viewing History

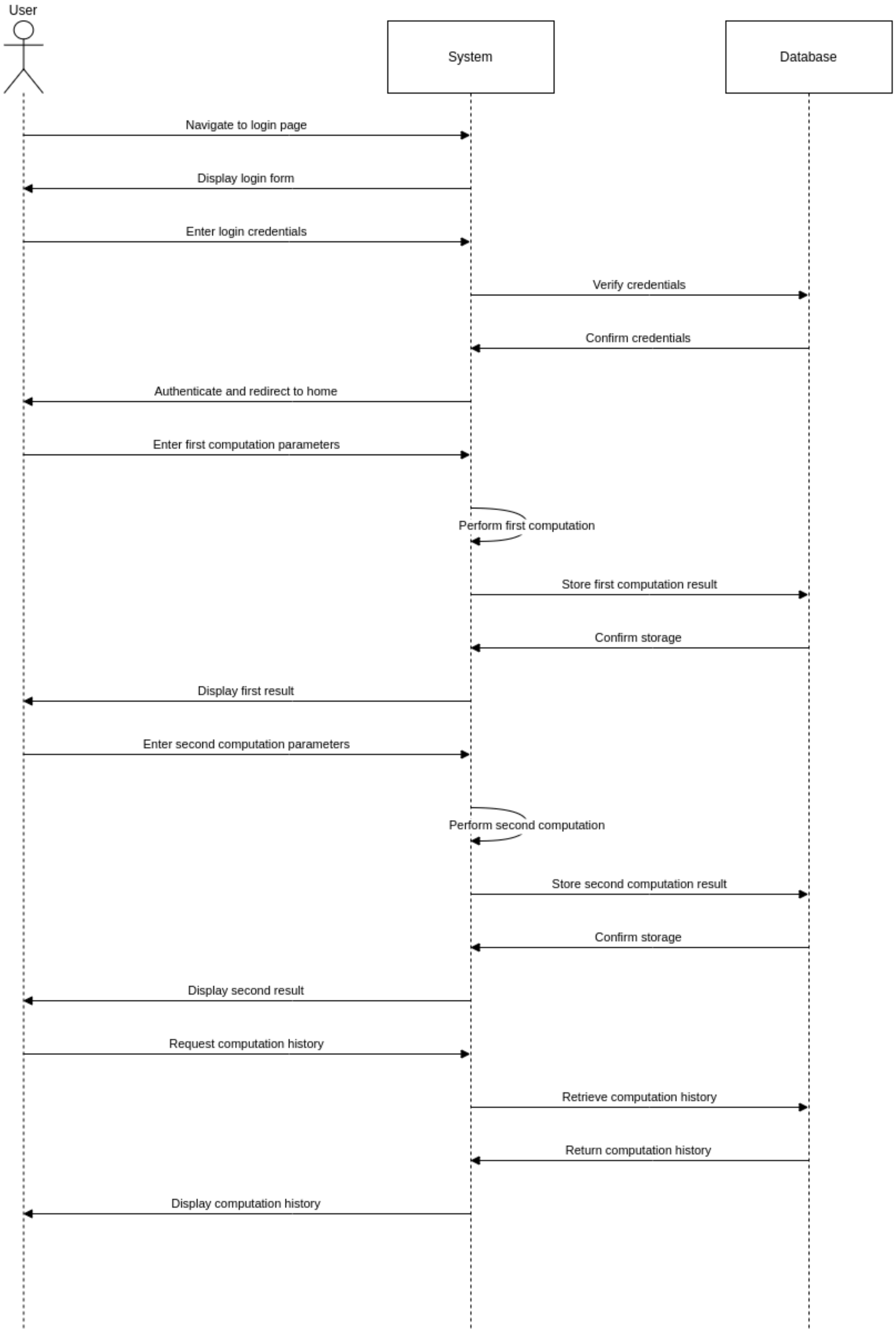
The authenticated user navigates to the history page.
The system retrieves the user's computation history from the database.
The system displays the history, showing details of past computations.
The user can view details of specific computations if desired.

Sequence diagrams for Use Cases

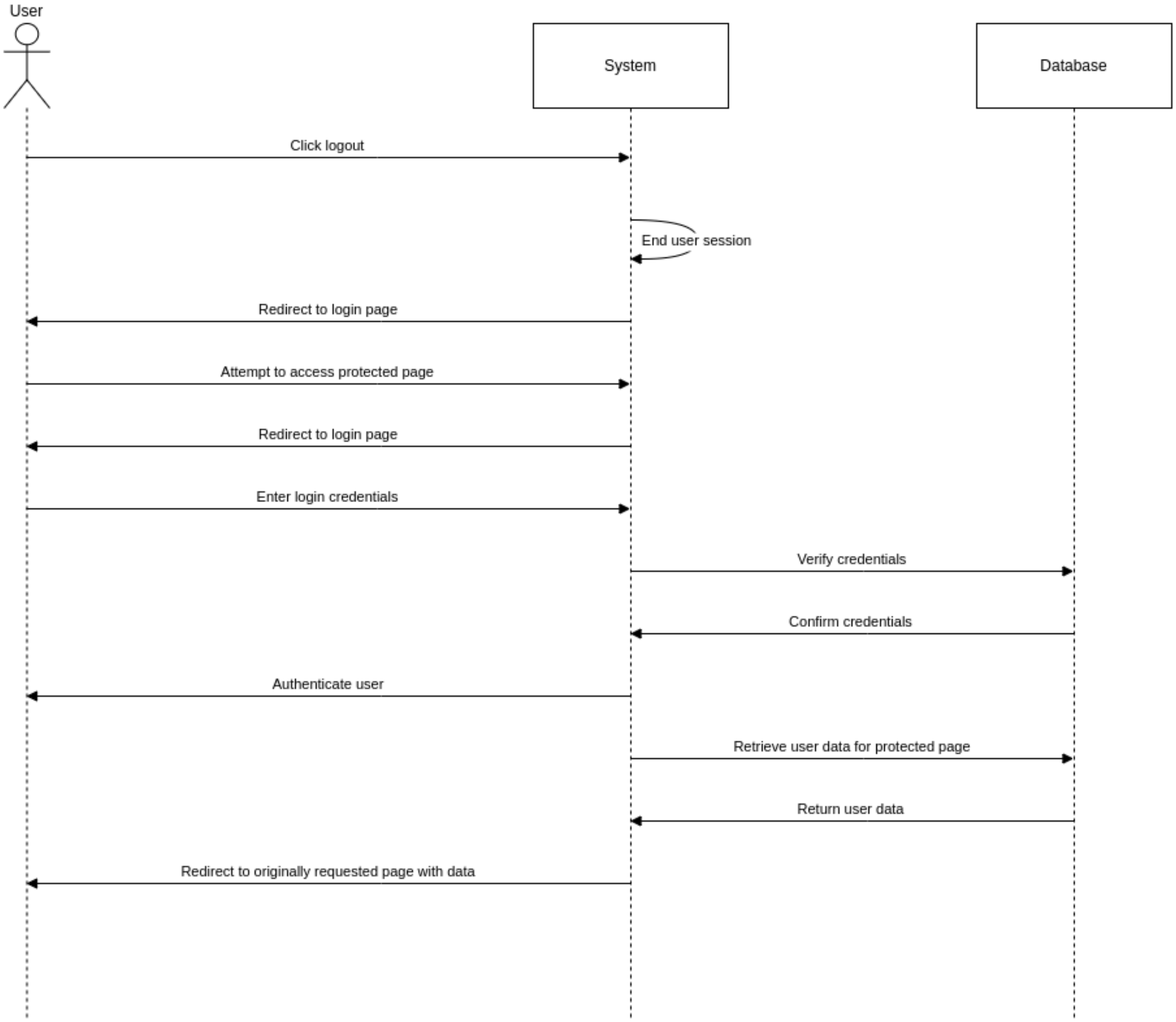
Register and test one computation


User
System
Database
Navigate to signup page
Display signup form
Submit registration details
Store new user account
Confirm storage
Redirect to login page
Enter login credentials
Verify credentials
Confirm credentials
Authenticate and redirect to home
Enter computation parameters
Perform computation
Store computation result
Confirm storage
Display computation result

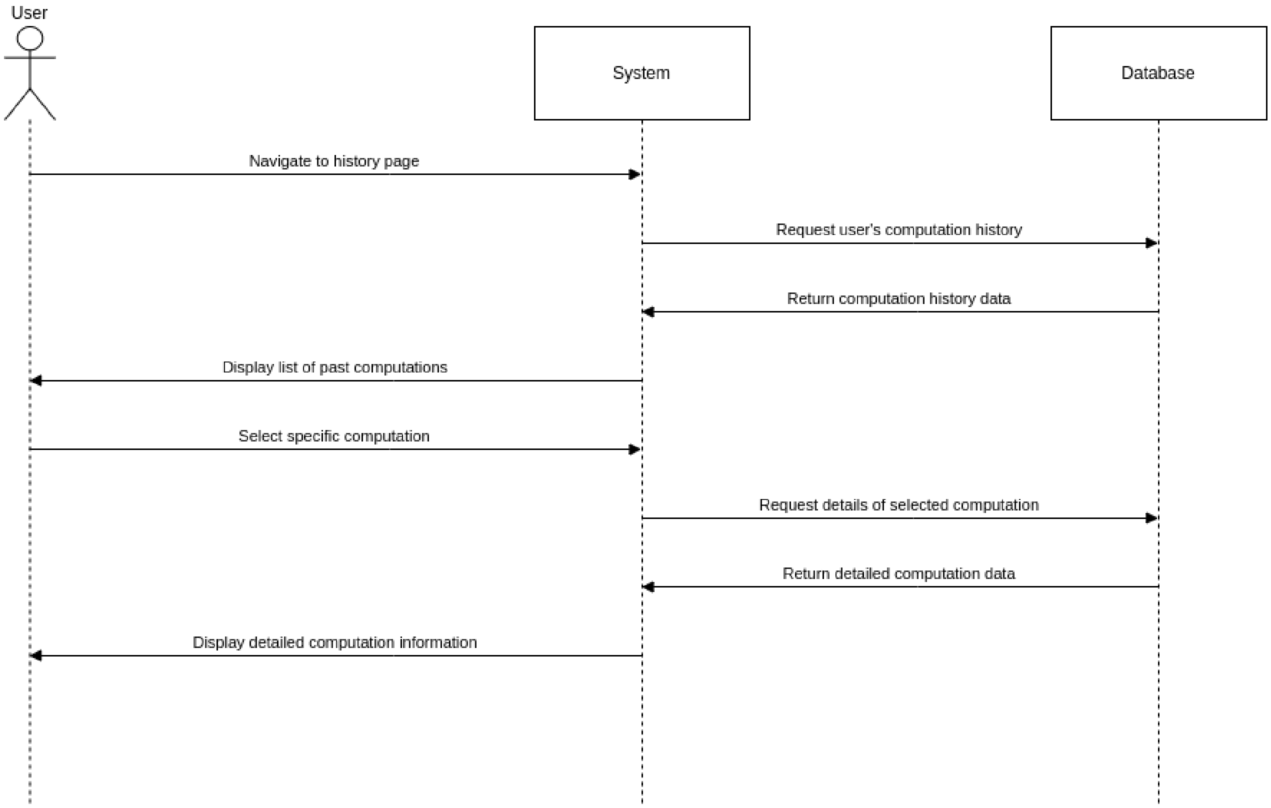
Login and make multiple computation


User
System
Database
Navigate to login page
Display login form
Enter login credentials
Verify credentials
Confirm credentials
Authenticate and redirect to home
Enter first computation parameters
Perform first computation
Store first computation result
Confirm storage
Display first result
Enter second computation parameters
Perform second computation
Store second computation result
Confirm storage
Display second result
Request computation history
Retrieve computation history
Return computation history
Display computation history

Logout and protected pages access



Viewing History


User
System
Database
Navigate to history page
Request user's computation history
Return computation history data
Display list of past computations
Select specific computation
Request details of selected computation
Return detailed computation data
Display detailed computation information