SENG2021 21T1

Software Architecture & Initial Software Design

Team Half Stack

Timothy Devine z5316844

Ethan Dickson z5309251

Lachlan Fraser z5258840

Michael Agius z5257081

Guy Chilcott z5263967

Ellen Duan z5210986

Table of Contents

Software Architecture

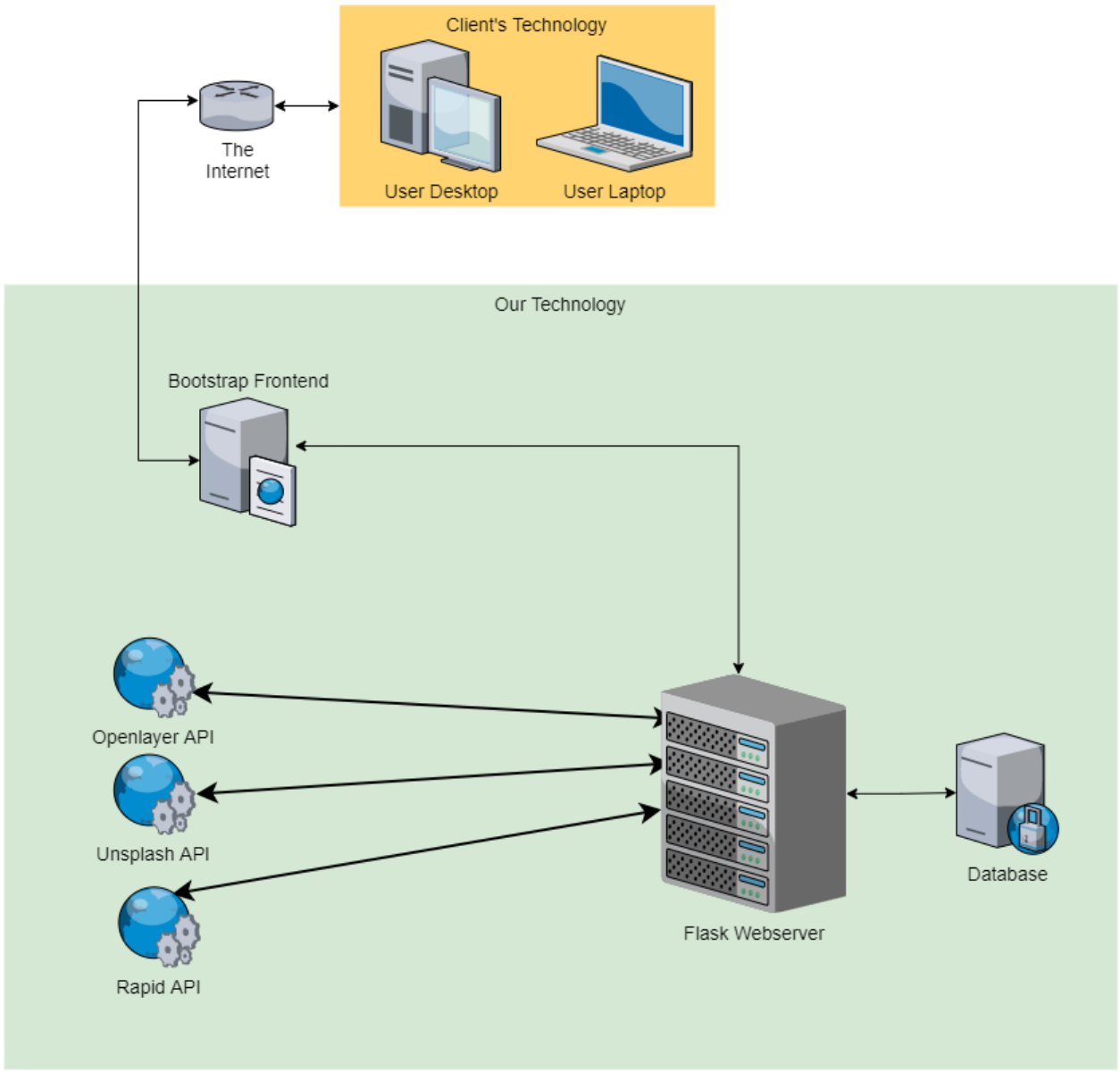
External Data Sources

In order to complete this project successfully, we will be using various APIs to help us get information needed for our application. Table 1 summaries the APIs we will be using.

Table 1: Summaries of APIs that will be used

API	Purpose
OpenLayers 6	An easy way to build interactive maps in our web pages and it can calculate the length of any path drawn on the map
Rapid API - Fitness Calculator	Calculate health data and information such as the number of calories burned from a specific activity
Unsplash	Help with the visual design and user interface of our application to make it more user-friendly and enjoyable to use

Software Components & Choice of Technology



Frontend

Backend

Database

Frontend

Table 1: Comparison of possible frontend software development libraries/frameworks for Javascript

	Vanilla JS	React	Angular JS

Pros	• No features are abstracted out, or hidden in development,
	  ○ Makes debugging easier,
	  ○ Code is easier to understand.
	• Strong group familiarity, everyone is able to develop in Vanilla JS.
		• Includes features to synchronise UI with a constantly changing set of data.
		• Basic JS functionality is abstracted out, reducing development time.
			• A full JS Framework, handles all the different aspects of web development.
			• Built on the basis of high quality, declarative programming.
			  ○ Fast development time.

Cons	• Poor scalability • Harder to maintain for growing applications. • No features are abstracted out, making for more repetitive and time consuming development.	• Unnecessarily complex for web applications with low amounts of user interaction. • Basic JS functionality is abstracted out, making debugging and understanding the code harder • Worse group familiarity, only a few members have used it before.	• Since it is a full framework, Angular JS effectively tells you how you must construct the piece of software. • Worst group familiarity, no members have used it before.

Due to the nature of our software solution, with a small level of user interaction, and no need for scalability, we have opted to develop the frontend in Vanilla JS. With total group familiarity, it will have the fastest development time altogether.

Backend

Comparison of backend languages:

Table 1: Comparison of common backend programming languages

Language	Speed	Group Familiarity	Frameworks & Libraries support	Integrations, APIs, SDKs
Python	Moderate	5/5	High	3/5
JavaScript	Fast	3/5	Moderate	3/5
Node	Fast	1/5	High	3/5
Java	Moderate	5/5	Low	4/5
C#	Very Fast	2/5	Minimal	2/5

From Table 1, Python will be chosen as the best option among these languages because of the group familiarity and moderate-high average score. 100% group familiarity also means that the learning curve is least and we can spend more time on adding additional features to the web stack.

Table 2: Comparison of some popular Python frameworks

	Django	Flask	Tornado
Pros	• Django includes prevention of common attacks like Cross-site request forgery (CSRF) and SQL Injections. • It includes its own Object Relation Mapping (ORM) layer for handling database access, sessions, routing, and multi-language support. It also takes care of security while handling requests.	• Flexible and extensible: huge range of external libraries and add-ons can be used. • Simplicity in the flask framework enables the developer to navigate around and create the application easily.	• Real-time functionality: supports large numbers of concurrent connections. • High performance
Cons	• Steeper learning curve • Built-in features and modules, offers far less freedom and control • Not for smaller projects: functionality of Django comes with lots of code. It takes server's processing and time.	• Scalability: it will handle every request in turns, one at a time. • Using more modules is seen as a third party involvement which could be a major breach in security.	• Smaller community • Stateless: no server side sessions

From Table 2, we will use Flask framework.

Database

Table 3: Comparison of Database Types

	Relational	Non-relational
Pros	• Have a well-designed predefined schema for structured data. • A good fit for complex queries as SQL has a standard interface for handling queries. • Excellent vendor support and community support is available for all SQL databases.	• Dynamic schema for unstructured data • NoSQL databases suit best for hierarchical data storage as it follows the key-value pair method for storing the data. • Cheaper to scale when compared to relational databases.
Cons	• SQL databases do not suit well for hierarchical data storage. • Costly to scale.	• Not a good fit for complex queries as there is no standard interface in NoSQL for handling queries. • NoSQL databases properly follow Brewers CAP theorem (Consistency, Availability, and Partition tolerance).

Table 4: Comparison of Database Software Distributions

	PostgreSQL	SQLite	MySQL
Pros	• A fully open-source project, PostgreSQL's source code is developed by a large and devoted community. • PostgreSQL has been fully ACID-compliant and implements multiversion currency control to ensure that data remains consistent, making it a strong choice of RDBMS when data integrity is critical. • Postgres supports query plans that can leverage multiple CPUs in order to answer queries with greater speed.	• A server-less database and is self-contained • An open source project available in the public domain • SQLite directly stores info in a single file, making it easy to copy. • No configurations are required • Suitable for basic development and testing	• An open source project which is owned by Oracle • MySQL has a well-constructed user management system which can handle multiple users and grant various levels of permission. • Easily scalable and can handle a bigger database with less effort
Cons	• Require a high learning curve • For simple read-heavy operations, PostgreSQL is typically less performant than	• Not suitable for big databases • Lacks user management and security features • Not easily scalable	• Requires some technical expertise to setup • Slightly different syntax as compared to conventional

	other RDBMSs, like MySQL.		SQL

From Table 3 and Table 4, we have decided to use SQLite as our database.

Choice of a Platform:

The final system will run its backend server on Linux. Specifically, we plan to implement this code on a CSE machine or emulator for consistent system requirements and capability. The application itself is a web app and will be tested to ensure it runs successfully on both Firefox and Google Chrome.

Summary of the Key Benefits/Achievements of Architectural Choices

When choosing the desired implementations for this project, we considered a number of factors. To create a successful overall software architecture, we made sure that our implementation would have high cohesion and low coupling by using technologies that we know integrate well together. We also chose software tools that the team was previously familiar with in order to streamline development and a platform that would ensure uniformity between group members as well as a stable development environment. We also plan to make use of APIs, as listed above, to help with the creation of a meaningful project.

Frontend:
When creating the frontend, tools such as Layoutit / Bootstrap are used to maintain consistency between pages and ensure our web application is themed with a practical user interface. Html and css are used as all group members are familiar with these languages and have the understanding to review and improve web pages. Html has the benefit of small changes not having major impacts as each webpage is locally contained. Css can be kept consistent between all webpages and a major scc document for all pages. Html and css also work across all web browsers making them compatible in all situations. We have also opted to use Vanilla JS over other frontend frameworks or JS libraries, for our familiarity with it, and the fact it produces easy to understand code. Other solutions would be simply overkill.

Backend:
For our Backend, we plan to use the Flask web server framework, in Python, which will process and deliver content from the database to the frontend via a HTTP request API (GET, POST, etc). Alternatives to Flask include Django, or Node.js. In arriving at our decision, we considered these frameworks.
Django:

> Whilst Django has a much greater feature set, it abstracts out many of these features, making it more difficult for first time developers like us.

Likewise, Django's prioritises scalability, and maintainability for complex programs, which makes customisation for a much simpler software solution more difficult.
Compared to Django, Flask is a much more lightweight framework, and is better suited to applications with a simple feature set like ours.

Node.js

For our purposes, Node.js would perform similarly to Flask, a more flexible feature set makes first time development easy, as we can better understand what is going on in the code, as less is abstracted out by the framework. One major benefit of Node is it's native async features, making it best for real time applications, of which we have no use for due to the nature of our software solution.
Likewise, as a team we have far less experience developing in Node than Python, so it makes more sense to develop in Python.

Database:
After considering possible database implementations, we decided that the scope of the application and project are small enough so we will use Sqlite as our database

API:
While choosing which API's we could implement, we decided that it was important that they did not over-complicate our design. For example, there were a plethora of options for the map API, including some from large organisations such as Google and Apple. However, we felt that the majority of these would distract from the goal of the application which is to provide a simple way for users to input their workout data onto a map. The Unsplash API was selected as it makes the process of finding suitable free images which will help with the visual design and user interface of our application to make it more user-friendly and enjoyable to use.

Initial Software Design

Updated User Stories

Sign up/Log in

In Scope:

Feature 1: Sign Up by Creating an Account

As a fitness enthusiast
I want to make an account on this service
So that I can use Full Body Grin to help me achieve the fitness level I want
am shown only data relevant to me, and not others unsolicited.

(1) Scenario: A new user wants to sign up and fills in the registration form correctly.
    (a) Given the user is on the landing page
    (b) When the user clicks the 'Sign Up' button
    (c) Then the user is directed to the Sign up page
    (d) When the user enters all required fields in the correct format (username and email that have not been signed up before, valid date of birth, valid height, valid weight, valid password) and clicks the 'Submit' button;
    (e) Then a modal window pops up informing the user they have signed up successfully, they are added to the database, and they are directed to the Log in page.
(2) Scenario: A new user wants to sign up and fills in an invalid field.
    (a) Given the user is on the Sign up page;
    (b) When the user enters either username/email that has been signed up before, invalid date of birth, invalid height, invaid weight, invaid password or 2 different passwords when confirming the password and clicks the 'Submit' button;
    (c) Then a modal window pops up informing the user that they have failed the registration due to an invalid field, they are not added to the database, and they stay on the Sign up page.
(3) Scenario: A new user changes their mind while filling the registration form.
    (a) Given the user is on the Sign up page;
    (b) When the user clicks the 'Cancel' button;
    (c) Then the user is not added to the database, and they are directed to the landing page.

Feature 2: User Login with Email Address and Password
As a fitness enthusiast who has previously signed up
I want to log into the website
So that I can access all my information and activities.

(1) Scenario: A registered user wants to log into their account and fills in the email address and password correctly.
    (a) Given the registered user is on the landing page
    (b) When the user clicks the 'Log In' button;
    (c) Then the user is directed to the Log in page
    (d) When the registered user enters a valid email address and a valid password and clicks the 'Submit' button;
    (e) Then the registered user is signed into their account, and the user is directed to the Dashboard page.

(2) Scenario: A registered user wants to log into their account but provides invalid email address or password.
    (a) Given the registered user is on the Sign in page;
    (b) When the registered user enters an invalid email address or an invalid password in and clicks the ‘Submit’ button;
    (c) Then a modal window pops up informing the user that the email address/password entered is invalid.

Future Goal:

Feature 3: Recover password

As a fitness enthusiast who has previously signed up
I want to recover my password
So that I can access all my information and activities.

(1) Scenario: A registered user wants to log into their account but forgot their password.
    (a) Given the user is on the Sign In page;
    (b) When the registered user clicks the ‘Recover Password’ hyperlink and enters their email address;
    (c) Then an email that allows the user to recover their password is sent to the registered email address, and a modal window pops up informing the user that the email was sent.

Planning workouts

Feature 1: Plan new workourts

As a fitness enthusiast
I want to plan new workouts
So that I can get motivations to improve my fitness level

(1) Scenario: A logged-in user wants to plan a new workout and fills in all fields correctly
    (a) Given the user is on the Dashboard page;
    (b) When the user clicks ‘Plan New Workout’ button;

(c) Then they are directed to the New workout page on which they can fill in workout details.
(d) When the user selects a valid completion date, selects a workout type (either running or cycling) from a dropdown list, draws the path they plan to take on the m
(e) Then path length, calories will be burnt and time that is predicted to be spent are all calculated automatically and corresponding fields are filled with the calculated values.
(f) When the user clicks the 'Submit' button;
(g) Then a modal window pops up informing the user that the new workout has been created

(3) Scenario: A logged-in user wants to plan a new workout but at least one of the required fields are filled in incorrectly.

(a) Given the user is on the New workout page;
(b) When the user provides either empty or invalid completion date, empty workout type or doesn't draw a path on the map
(c) Then path length, calories will be burnt and time that is predicted to be spent are not calculated and corresponding fields appear empty.
(d) When the user clicks the 'Submit' button;
(e) Then a modal window pops up informing the user that some inputs are invalid.

(4) Scenario: A logged-in user changes their mind while planning new workouts and doesn't want to plan a new workout.
(a) Given the user is on the New workout page;
(b) When the user clicks the 'Back' button;
(c) They are directed back to the Dashboard page.

Feature 2: View upcoming workouts
As a fitness enthusiast
I want to view all upcoming workouts
So I can get reminded and allocate time wisely
(1) Scenario: A logged-in user wants to view all upcoming workouts
(a) Given the user is on the Dashboard page;
(b) When the user clicks the 'Show All' button in the 'Upcoming Workouts' section;
(c) Then they are directed to the Upcoming workouts page and general summaries of planned workouts with completion date being either today or after are listed in chronological order.

(2) Scenario: A logged-in user wants to view details of a planned workout
(a) Given the user is on the Upcoming workouts page;
(b) When the user clicks the 'View' button of a specific planned workout;

(c) Then they are directed to the Planned workout data page on which they can view details of the selected planned workout.

Feature 3: Edit planned workouts

As a fitness enthusiast

I want to edit a planned workout

So the workout becomes more achievable for me given my current fitness level and time availability

(1) Scenario: A logged-in user wants to edit a planned workout's details and all the required fields are valid after the edit is made.
   (a) Given the user is on the Planned workout data page of a specific planned workout and has clicked the 'Edit' button
   (b) When the completion date is valid, either running or cycling is selected from the dropdown list as the workout type, and a valid path is drawn on the map after edits are made
   (c) Then path length calories will be burnt and time that is predicted to be spent are calculated again and corresponding fields are filled with calculated values.
   (d) When the user clicks the 'Save' button;
   (e) Then a modal window pops up informing the user that the workout has been updated and the user stays on the Planned workout data page with the updated workout details.

(2) Scenario: A logged-in user wants to edit a planned workout's details but at least one of the required fields are invalid after the edit is made.
   (a) Given the user on the Planned workout data page of a specific planned workout and has clicked the 'Edit' button
   (b) When either the completion date is invaid, no workout type is selected from the dropdown list or no path is drawn on the map after edits are made
   (c) When the user clicks the 'Save' button;
   (d) Then a modal window pops up informing the user that some information is invalid and the user stays on the Planned workout data page.

(3) Scenario: A logged-in user wants to discard edits made to the workout.
   (a) Given the user is on the Planned workout data page and has clicked the 'Edit' button;
   (b) When the user clicks the 'Cancel' button;
   (c) Then a modal window pops up asking the user to confirm that they want to discard the edits made to the workout.
   (d) When the user clicks the 'Discard changes' button;
   (e) Then the user stays on the Planned workout data page and workout details remain unchanged.

Feature 4: Delete planned workouts

As a fitness enthusiast

I want to delete an upcoming workout
So that I don't need to spend time on that

(1) Scenario: A logged-in user wants to delete a workout.
    (a) Given the user is on the Planned workout data page;
    (b) When the user clicks the 'Delete' button;
    (c) Then a modal window pops up asking the user to confirm that they want to delete this workout.
    (d) When the user clicks 'Yes, I want to delete this workout' button;
    (e) Then a modal window pops up informing the user that the workout has been deleted, and the user is directed to the Upcoming workouts page.

(2) Scenario: A logged-in user changes their mind when trying to delete a workout.
    (a) Given the user is on the Planned workout data page and has clicked the 'Delete' button;
    (b) When the user clicks 'No, I don't want to delete this workout' button in the modal window;
    (c) Then the user stays on the Planned workout data page of that workout.

Feature 5: Filter upcoming workouts
As a fitness enthusiast
I want to filter upcoming workouts based on my interest
So only workouts I am interested in are shown to me

(1) Scenario: A logged-in user wants to view planned workouts in a specific time period and enters valid start date and/or end date
    (a) Given the user is on the Upcoming workouts page;
    (b) When the user enters start date and/or end date correctly and clicks the 'Search' button;
    (c) Then only workouts with completion date in the time range are displayed in chronological order.

(2) Scenario: A logged-in user wants to view planned workouts in a specific time period but enters invalid start/end date
    (a) Given the user is on the Upcoming workouts page;
    (b) When the user enters start date or end date incorrectly and clicks the 'Search' button;
    (c) Then a modal window pops up informing the user they have entered invalid start/end date

(3) Scenario: A logged-in user wants to filter planned workouts by workout type
    (a) Given the user is on the Upcoming workouts page;
    (b) When the user selects one workout type from the dropdown list and clicks the 'Search' button;
    (c) Then only workouts of the select workout type are displayed on the Upcoming workouts page in chronological order.

(4) Scenario: A logged-in user wants to clear any filters applied before.
    (a) Given the user is on the Upcoming workouts page;
    (b) When the user clicks the ‘Clear Filter’ button;
    (c) All upcoming workouts are displayed in chronological order.

Feature 6: Mark planned workouts as completed
As a fitness enthusiast
I want to add a planned workout to workout logs automatically
So I don’t need to log it again manually

(1) Scenario: A logged-in user wants to log a planned workout automatically
    (a) Given the user is on the Planned workout data page;
    (b) When the user clicks the ‘Complete’ button;
    (c) Then a modal window pops up informing the user that this workout is added to the workout log, and the workout is removed from upcoming workouts.

Track workouts

Feature 1: Log previous workouts
As a fitness enthusiast
I want to log my previous workouts
So I can have an idea of what I have completed.

(1) Scenario: A logged-in user wants to log a previous workout and enter all required fields correctly.
    (a) Given the user is on the Dashboard page;
    (b) When the user clicks the ‘Log Previous Workout’ button;
    (c) Then the user is directed to the Log previous workout page where they can enter workout details.
    (d) When the user selects a valid date, enters a positive integer for time spent field, selects a workout type from the dropdown list and draws the path they travelled on the map
    (e) Then path length, calories burnt and pace are calculated and corresponding fields are filled with calculated values.
    (f) When the user clicks the ‘Submit’ button;
    (g) Then a modal window pops up informing the user that the workout has been logged successfully.

(2) Scenario: A logged-in user wants to log a previous workout and enter at least one required field incorrectly.
    (a) Given the user is on the Log previous workout page;
    (b) When the user either selects an invalid or empty date, enters an invalid input for time spent field, doesn’t select a workout type from the dropdown list or doesn’t draw a path on the map

(c) Then path length, calories burnt and pace are not calculated and corresponding fields appear empty.
(d) When the user clicks the ‘Submit’ button;
(e) Then a modal window pops up informing the user that some inputs are invalid.

(4) Scenario: A logged-in user changes their mind while logging a previous workout.
(a) Given the user is on the Log previous workout page;
(b) When the user clicks the ‘Back’ button;
(c) Then the user is directed to the Dashboard page.

Feature 2: View recently logged workouts
As a fitness enthusiast
I want to view my logged workouts
So I can know if I have improved

(1) Scenario: A logged-in user wants to view all previous logged workouts
(a) Given the user is on the Dashboard page
(b) When the user clicks on the ‘Show All’ button in the ‘Recent Workout‘ section
(c) Then they are directed to the Workout log page and general summaries of their recently logged workouts are listed in chronological order.

(2) Scenario: A logged-in user wants to view details of a previous logged workouts
(a) Given the user is on the Workout log page
(b) When the user clicks the ‘View’ button after each logged workout
(c) Then they are directed to the Workout data page on which they can view details of the selected previous workout.

Feature 3: Filter previous workouts
As a fitness enthusiast
I want to filter logged workouts based on my interest
So only workouts I am interested in are shown to me

(1) Scenario: A logged-in user wants to sort the previous workout by the amount the calories burnt
(a) Given the user is on the Workout log page
(b) When the user selects ‘Low to High’/‘High to Low’ from the ‘Sort by Calories burnt’ dropdown list
(c) Then the Workout log page refreshed and previously logged workouts are shown in the specified order.

(2) Scenario: A logged-in user wants to view logged workouts in the specific time range and enters start date and/or end date correctly
(a) Given the user is on the Workout log page;

(b) When the user enters start date and/or end date correctly and clicks the ‘Search’ button;
(c) Then the Workout page refreshes and only workouts that were logged in the time range are displayed in chronological order.

(3) Scenario: A logged-in user wants to view workouts that were logged in the specific time range but enters start date or end date incorrectly.
(a) When the user is on the Workout log page;
(b) When the user enters start date and end date incorrectly and clicks the ‘Search’ button;
(c) Then a modal window pops up informing the user they have entered invalid start/end date.

(4) Scenario: A logged-in user wants to filter logged workouts by workout type.
(a) Given the user is on the Workout log page;
(b) When the user selects one workout type from the dropdown list and clicks the ‘Search’ button;
(c) Then only workouts of the select workout type are displayed on the Workout log page in chronological order.

(5) Scenario: A logged-in user wants to clear any filters applied before.
(a) Given the user is on the Workout log page;
(b) When the user clicks the ‘Clear Filter’ button;
(c) All previous workouts are displayed in chronological order.

Feature 4: Reuse previous workout

As a fitness enthusiast
I want to reuse a previous workout as one of my upcoming workouts automatically.
So that I don’t need to enter the same details again while planning a new workout.

(1) Scenario: A logged-in user wants to reuse a previous workout.
(a) Given the user is on the Workout data page;
(b) When the user clicks the ‘Reuse’ button;
(c) Then a modal window pops up informing the user that this workout has been added to the upcoming workouts.

Goal Setting

Feature 1: Add new goals

As a fitness enthusiast
I want to create new goals
So that I can get motivations and achieve the fitness level I desire

(1) Scenario: A logged-in user wants to add a new goal and all fields in the Goal form are entered correctly

(a) Given that the user is on the Workout log page;
(b) When the user clicks on the “Create Goal” button;
(c) Then the user is directed to the Create new goal page on which they can enter details of a new goal
(d) When the user selects a valid completion date, enters a positive integer for at least one of these fields: total calories burnt, total time spent and total distance travelled, and selects a workout type from the dropdown list (optional) and clicks the ‘Submit’ button
(e) Then a modal page pops up informing the user that the goal has been created successfully, and the user is directed to the Workout log page.

(3) Scenario: A logged-in user wants to add a new goal and all fields in the Goal form are entered incorrectly

(a) Given the user is on the Create new goal page
(b) When the user either selects an invalid completion date, enters an invaid input for at least one of these fields: total calories burnt, total time spent and total distance travelled, or doesn’t select a workout type from the dropdown list and clicks the ‘Submit’ button
(c) Then a modal window pops up informing the user that the new goal has been created successfully.

(4) Scenario: A logged-in user changes their minds while creating a new goal.

(a) Given the user is on the Create new goal page
(b) When the user clicks the ‘Back’ button;
(c) Then the user is directed to the Workout log page.

Feature 2: View goals and workout recommendations

As a fitness enthusiast
I want to view my goals and workout recommendations
So I know if I have completed my goal by the deadline I set and how to achieve them

(1) Scenario:

(a) Given that user is on the Workout log page;
(b) When the user clicks the ‘View Recommendations’ button;
(c) Then the user is directed to the Recommendation & Goal page on which goals with different status (completed, need to be completed, uncompleted

by their deadlines) are shown in different colors and workout recommendations are displayed.

Feature 3: Delete existing goals

As a fitness enthusiast
I want to delete existing goals
So that I don't need to complete a goal that is beyond my current ab

(1) Scenario:
- (a) Given that I am on the Recommendation & Goal page
- (b) When I click the "Delete" button
- (c) Then the goal is removed from the Recommendation & Goal page

Feature 4: Add recommendations to upcoming workouts

As a fitness enthusiast
I want to add workout recommendations to upcoming workouts automatically
So that I don't need to enter the same details again while planning a workout

(1) Scenario: A logged-in user wants to add a recommendation to upcoming workouts.
- (a) Given the user is on the Recommendation & Goal page;
- (b) When the user clicks the 'Use' button of a specific recommendation;
- (c) Then a modal window pops up informing the user that the recommendation has been added to upcoming workouts successfully, and this recommendation is removed from the Recommendation & Goal page.

Update User Info

Feature 1: Change personal stats

As a fitness enthusiast
I want to change my personal stats (weight and height)
So that calories burnt can be calculated correctly

(1) Scenario: A logged-in user wants to update personal stats and fills in all required fields correctly.
- (a) Given the user is on the Dashboard page
- (b) When the user clicks the 'Settings' button;
- (c) Then the user is directed to the Settings page;
- (d) When the user clicks the 'Update Your States' button

(e) Then the user is directed to the Change your stats page
(f) When the user enters valid positive numbers for height and weight and clicks the ‘Submit’ button;
(g) Then a modal window pops up informing the user that personal stats have been updated successfully.

(2) Scenario: A logged-in user wants to update personal stats but fills in at least one required fields incorrectly

(a) Given the user is on the Change your stats page;
(b) When the user enters an invalid input for height or weight, and clicks the ‘Submit’ button;
(c) Then a modal window pops up informing the user that at least one input is invalid.

Feature 2: A logged-in user want to update email/password

(1) Scenario: A logged-in user wants to update email/password and fills in all required fields correctly.

(a) Given the user is on the Dashboard
(b) When the user clicks the ‘Settings’ button;
(c) Then the user is directed to the Settings page;
(d) When the user clicks the ‘Update Email/Password’ button
(e) Then the user is directed to the Update Email/Password page
(f) When the user enters a valid new email and/or a valid new password, the same new email and/or new password while confirming new email and/or password, and clicks the ‘Submit’ button;
(g) Then a modal window pops up informing the user that email and password have been updated successfully.

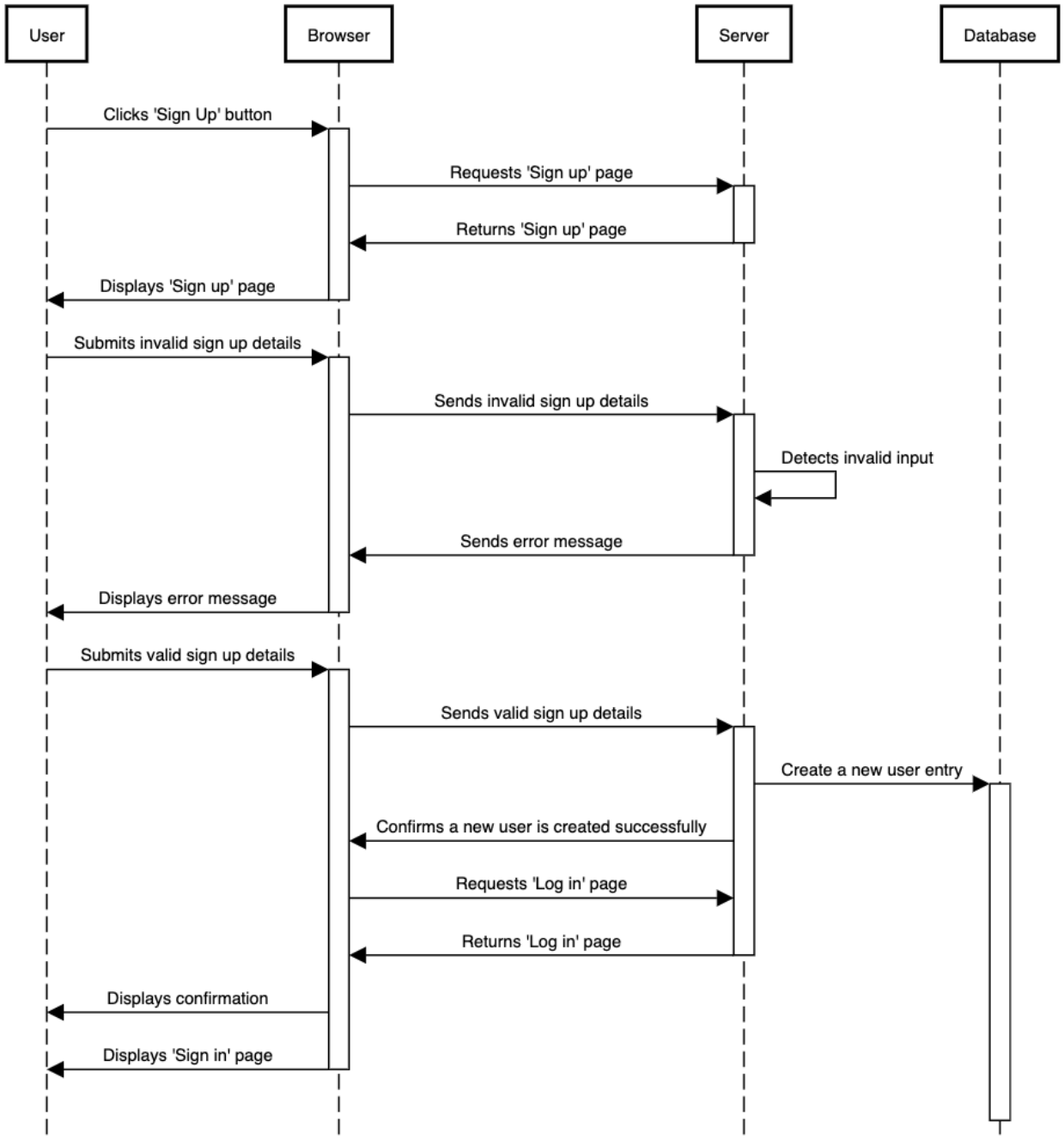
(2) Scenario: A logged-in user wants to update email/password and fills in at least one of the required fields incorrectly.

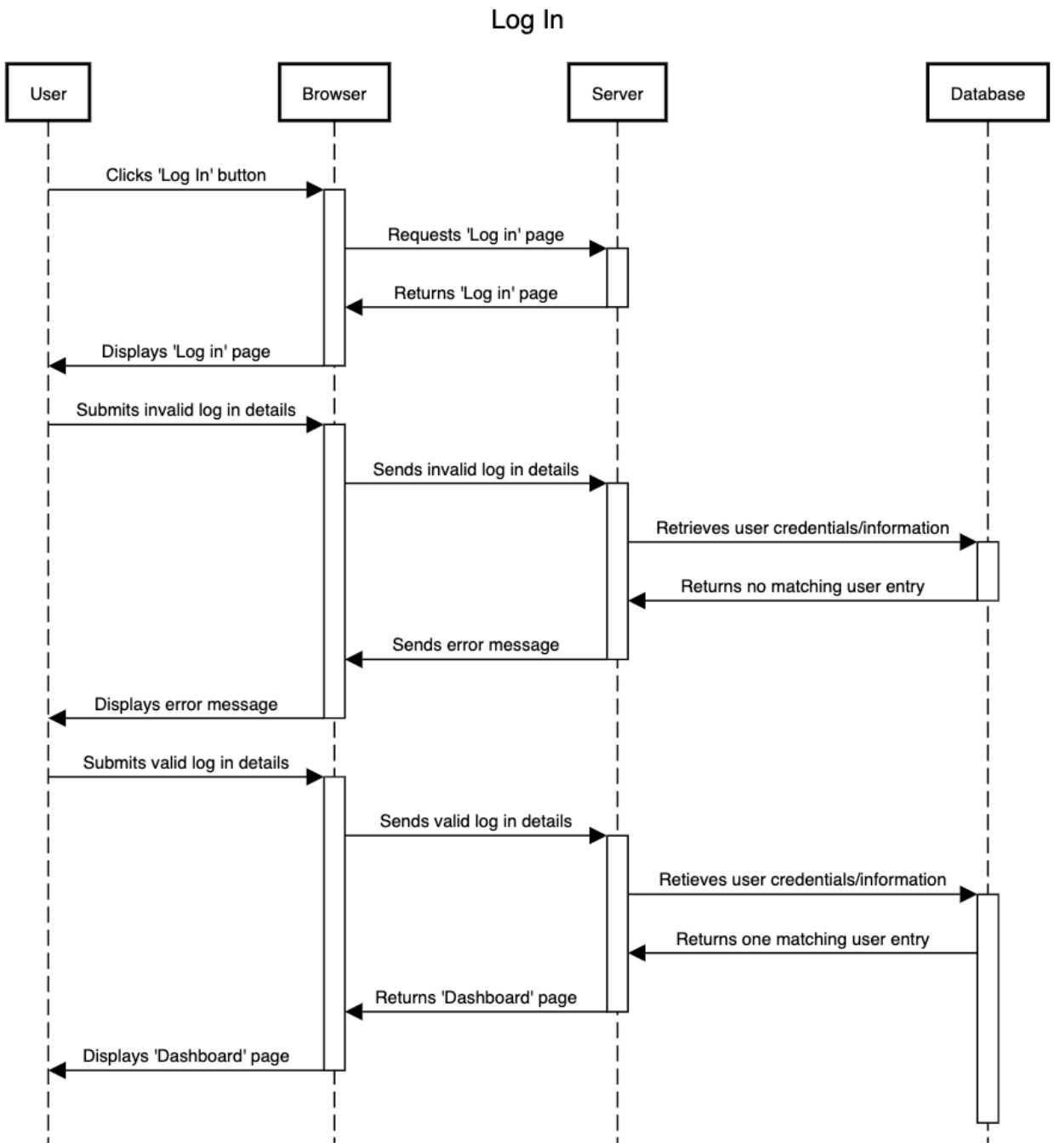
(a) Given the user is on the Update Email/Password page;
(b) When the user either enters an invalid email/password or enters different email/password while confirming new email/password, and clicks the ‘Submit’ button;
(c) Then a modal window pops up informing the user that at least one input is invalid.

Sequence Diagrams

Sign Up & Sign In

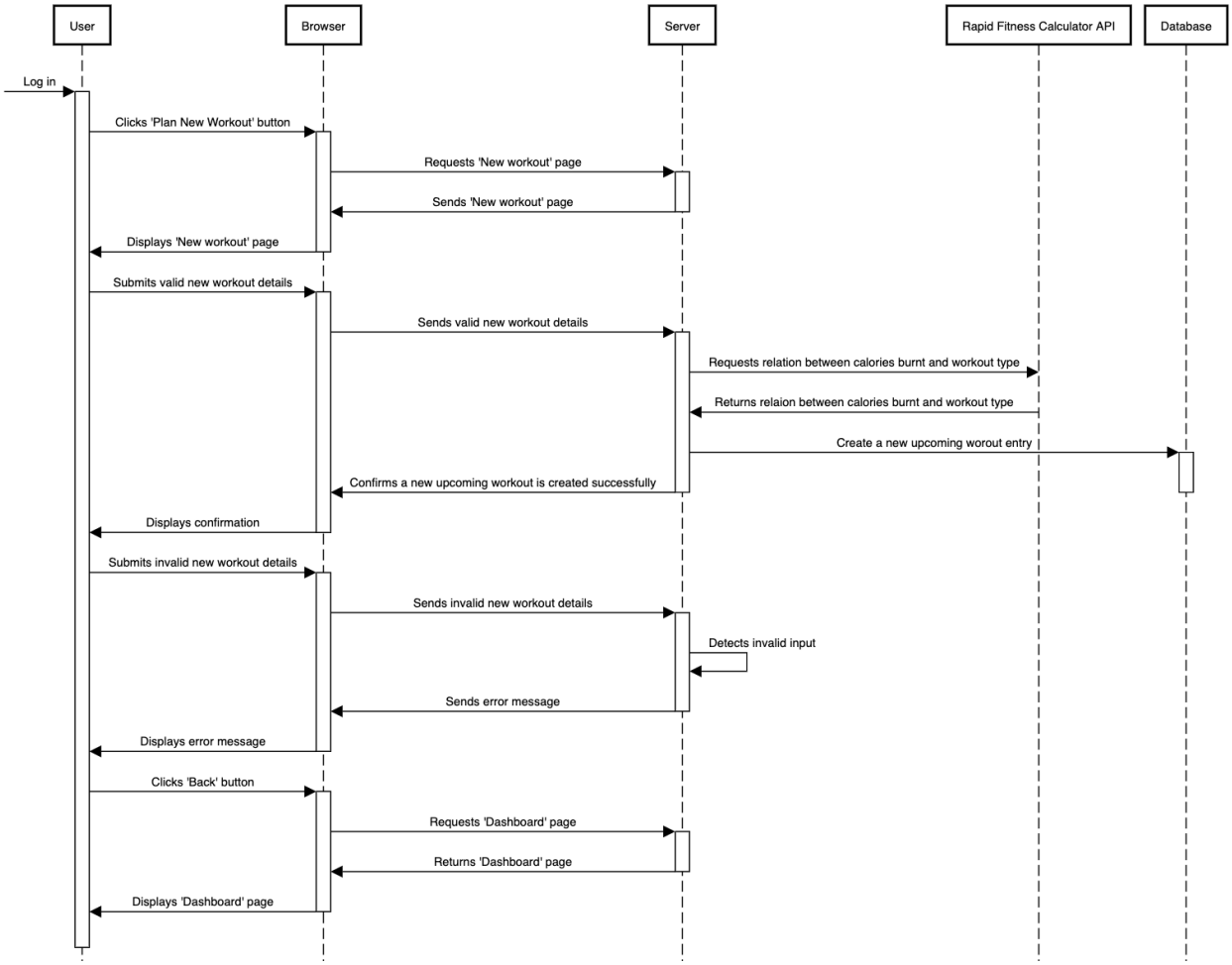
Sign Up

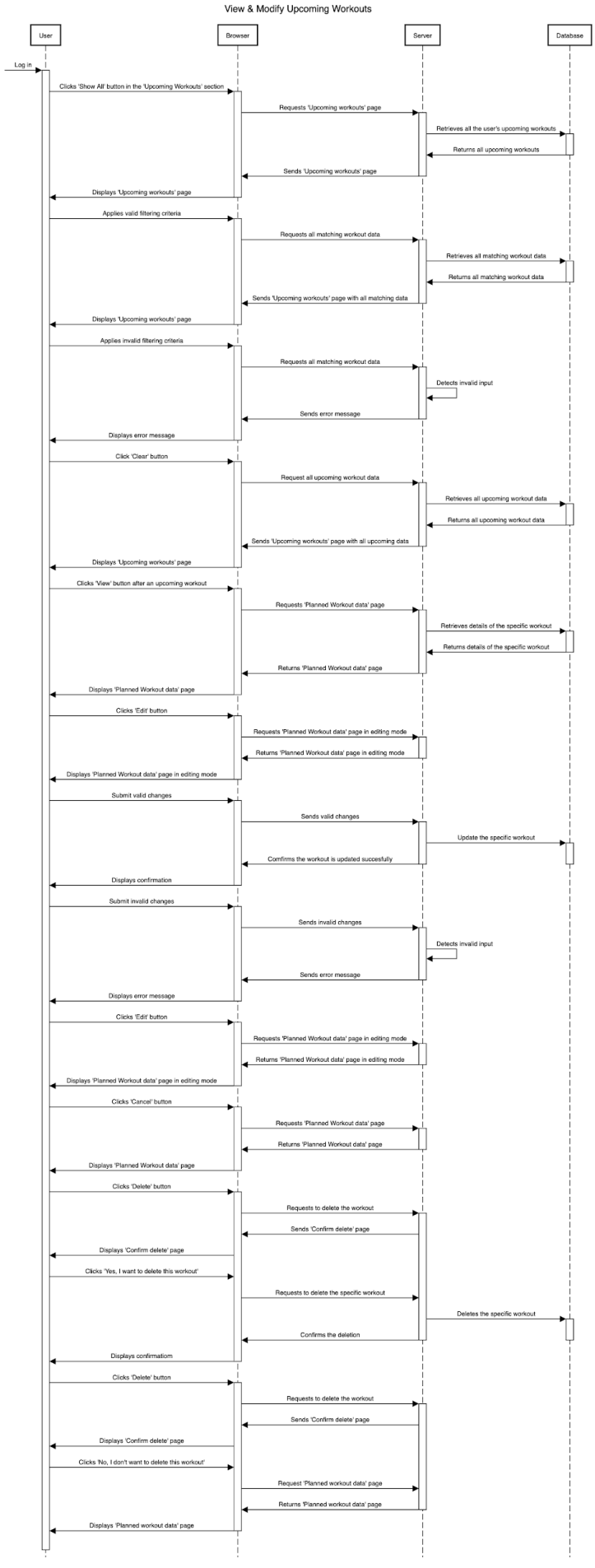

User
Browser
Server
Database
Clicks 'Sign Up' button
Requests 'Sign up' page
Returns 'Sign up' page
Displays 'Sign up' page
Submits invalid sign up details
Sends invalid sign up details
Detects invalid input
Sends error message
Displays error message
Submits valid sign up details
Sends valid sign up details
Create a new user entry
Confirms a new user is created successfully
Requests 'Log in' page
Returns 'Log in' page
Displays confirmation
Displays 'Sign in' page



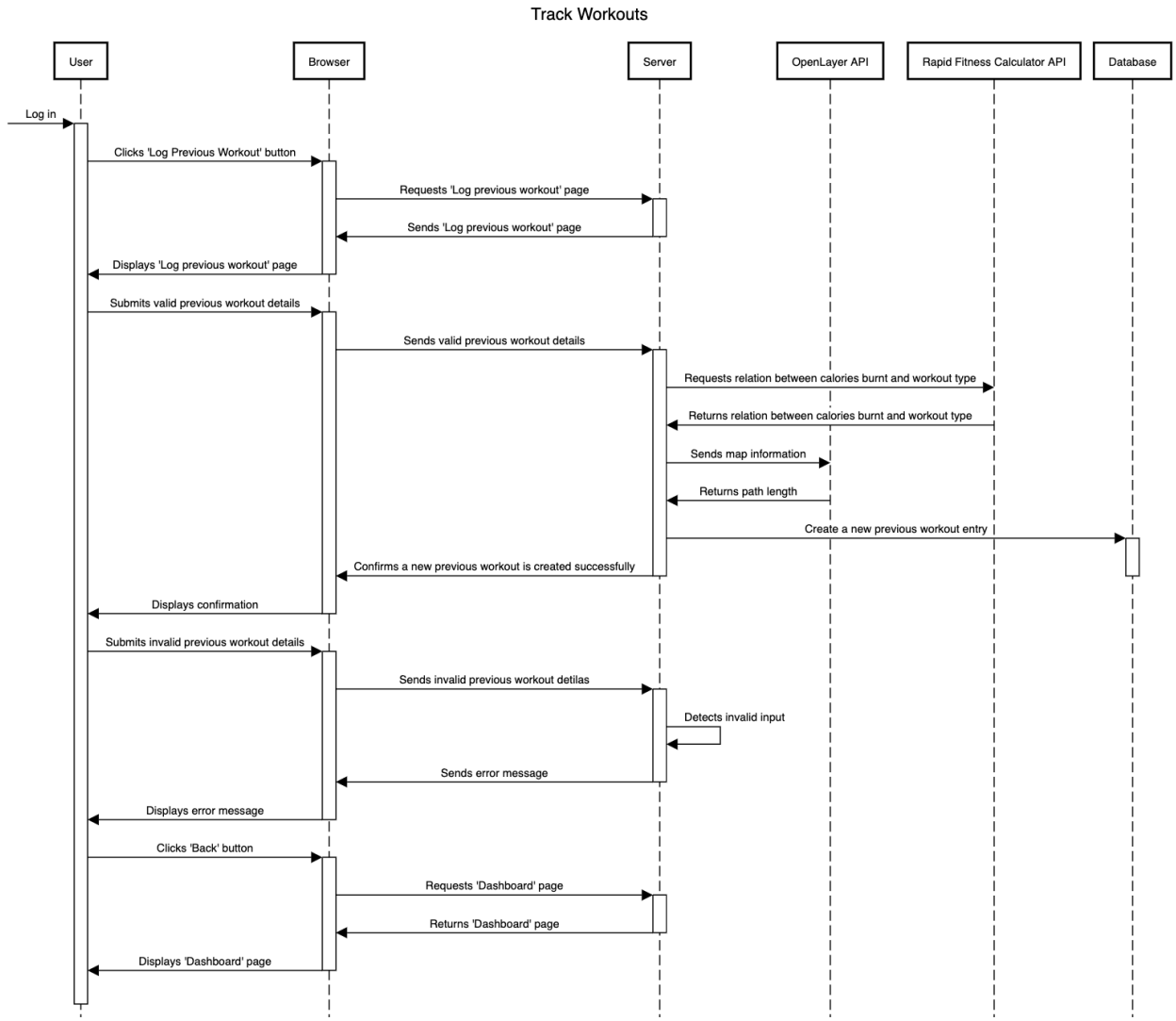
Planning Workouts

Plannig Workouts



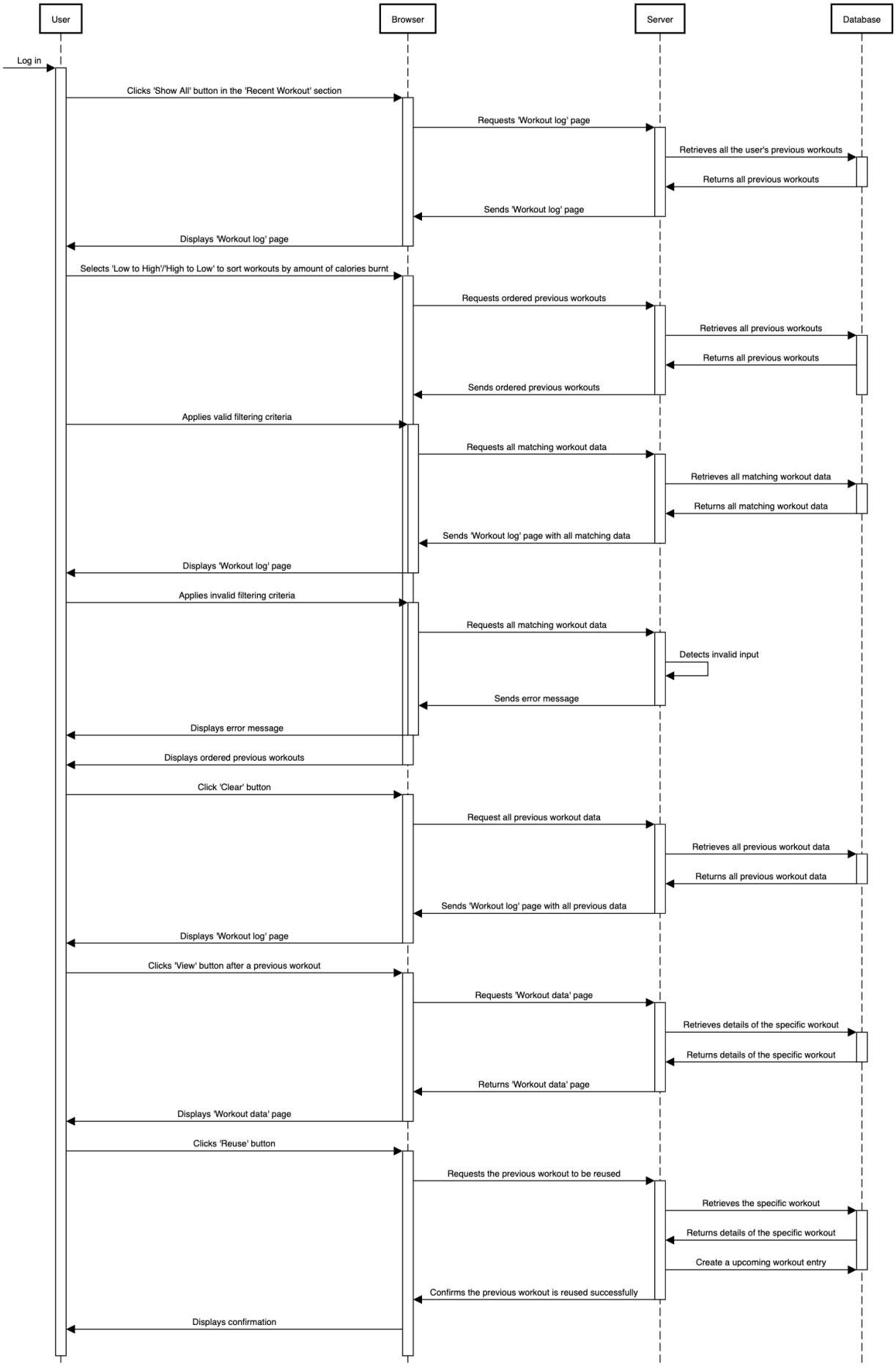

View & Modify Upcoming Workouts
User
Browser
Server
Database
Log in
Clicks 'Show All' button in the 'Upcoming Workouts' section
Requests 'Upcoming workouts' page
Retrieves all the user's upcoming workouts
Returns all upcoming workouts
Sends 'Upcoming workouts' page
Displays 'Upcoming workouts' page
Applies valid filtering criteria
Requests all matching workout data
Retrieves all matching workout data
Returns all matching workout data
Sends 'Upcoming workouts' page with all matching data
Displays 'Upcoming workouts' page
Applies invalid filtering criteria
Requests all matching workout data
Detects invalid input
Sends error message
Displays error message
Click 'Clear' button
Request all upcoming workout data
Retrieves all upcoming workout data
Returns all upcoming workout data
Sends 'Upcoming workouts' page with all upcoming data
Displays 'Upcoming workouts' page
Clicks 'View' button after an upcoming workout
Requests 'Planned Workout data' page
Retrieves details of the specific workout
Returns details of the specific workout
Returns 'Planned Workout data' page
Displays 'Planned Workout data' page
Clicks 'Edit' button
Requests 'Planned Workout data' page in editing mode
Returns 'Planned Workout data' page in editing mode
Displays 'Planned Workout data' page in editing mode
Submit valid changes
Sends valid changes
Update the specific workout
Comfirms the workout is updated succesfully
Displays confirmation
Submit invalid changes
Sends invalid changes
Detects invalid input
Sends error message
Displays error message
Clicks 'Edit' button
Requests 'Planned Workout data' page in editing mode
Returns 'Planned Workout data' page in editing mode
Displays 'Planned Workout data' page in editing mode
Clicks 'Cancel' button
Requests 'Planned Workout data' page
Returns 'Planned Workout data' page
Displays 'Planned Workout data' page
Clicks 'Delete' button
Requests to delete the workout
Sends 'Confirm delete' page
Displays 'Confirm delete' page
Clicks 'Yes, I want to delete this workout'
Requests to delete the specific workout
Deletes the specific workout
Confirms the deletion
Displays confirmatiom
Clicks 'Delete' button
Requests to delete the workout
Sends 'Confirm delete' page
Displays 'Confirm delete' page
Clicks 'No, I don't want to delete this workout'
Request 'Planned workout data' page
Returns 'Planned workout data' page
Displays 'Planned workout data' page

Tracking Workouts

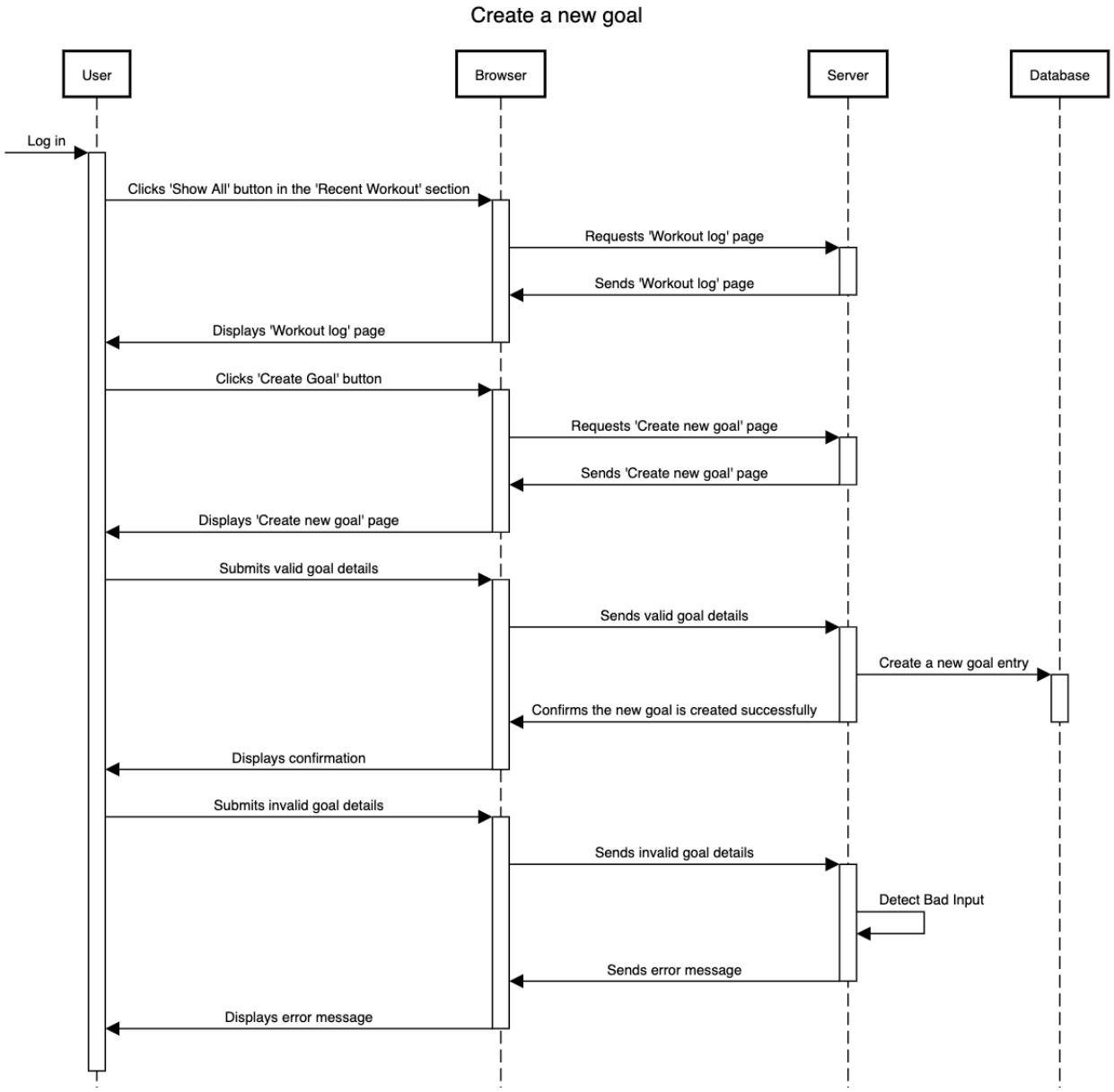


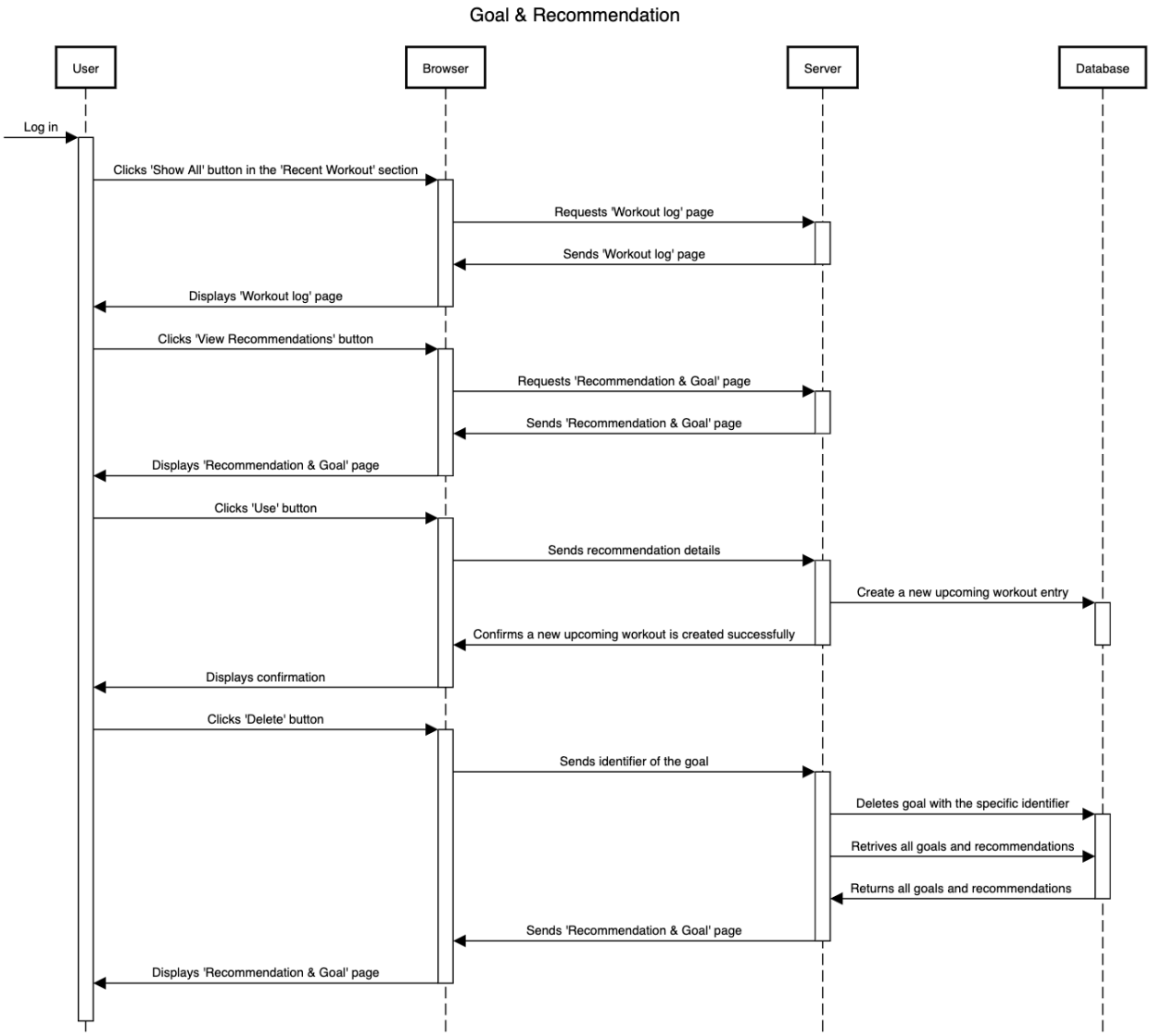
View Workouts & Reuse Old Workout

View Workouts & Reuse Old Workout



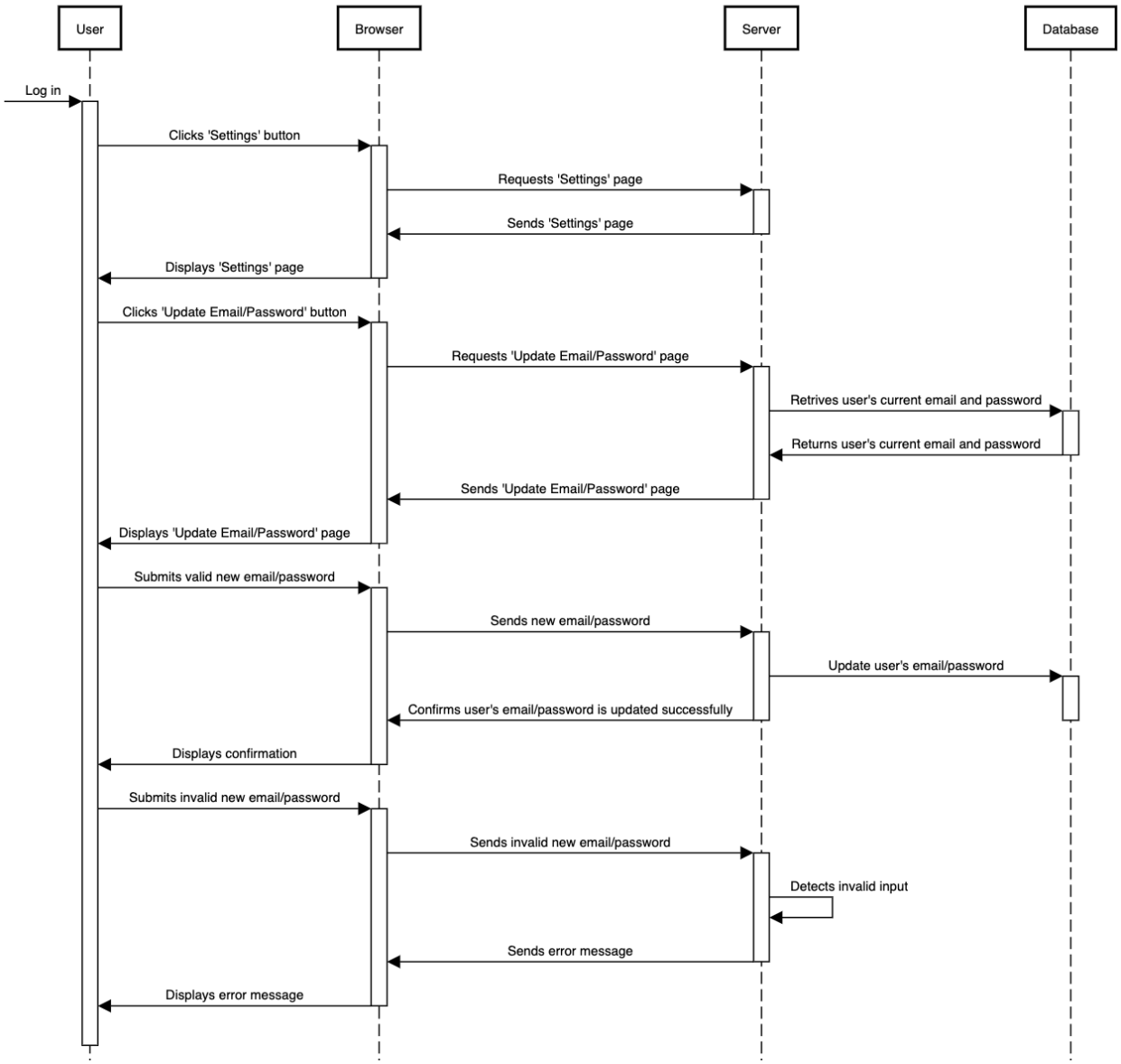
Goal Setting





Update User’s Info

Update Email/Password



Update User Info