CREATivators: Web and Mobile Development for Advertisement 1.0

POSITIVE PATHWAYS

Success Within Reach

Team Members: Mikaila Daniel, Michael Quiros, Alejandro Thornton
Product Owner: Dr. Steven Rios
Mentors: Drs. Margo Berman and Francisco Ortega
Instructor: Dr. Masoud Sadjadi

Problem

- Members currently use traditional communication methods such as e-mail and phone.

- No dedicated repository of information about the Positive Pathways Program (PPP).

- Agendas and minutes of monthly meetings are not centralized.

Solution

- Dedicated mobile application for communication and information.
- Pages dedicated to the programs, members, partners, and other information about PPP.
- Agendas of future monthly meetings and minutes of past meetings collected in one location.

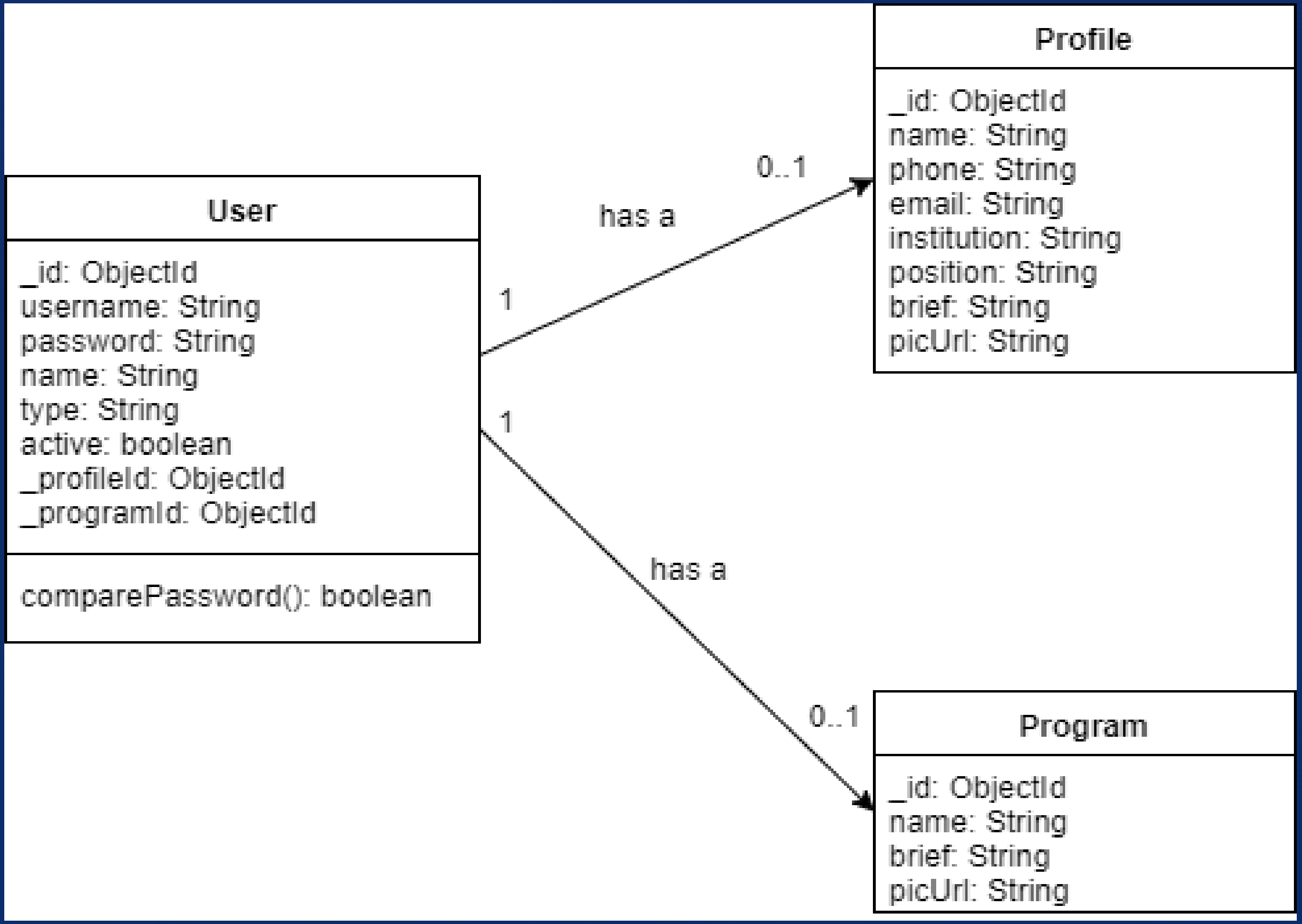
Requirements

- General users can view profiles, programs, partners, and training materials without an account.
- With an account, Members can edit their individual profile pages as well as their respective program pages.
- Additionally, Members can view and edit meeting agendas and minutes.

Implementation

- MongoDB for database organization.
- Mongoose for database operations.
- mLab for remote database hosting.
- Express for routing requests and responses to/from server.
- NodeJS for server operations.
- JSON Web Token for encrypting and decrypting tokens.

Object Design



User

_id: ObjectId
username: String
password: String
name: String
type: String
active: boolean
_profileId: ObjectId
_programId: ObjectId

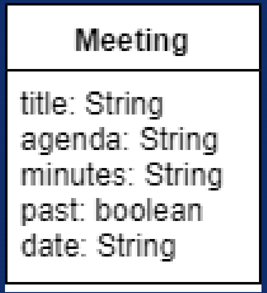
comparePassword(): boolean

Profile

_id: ObjectId
name: String
phone: String
email: String
institution: String
position: String
brief: String
picUrl: String

Program

_id: ObjectId
name: String
brief: String
picUrl: String

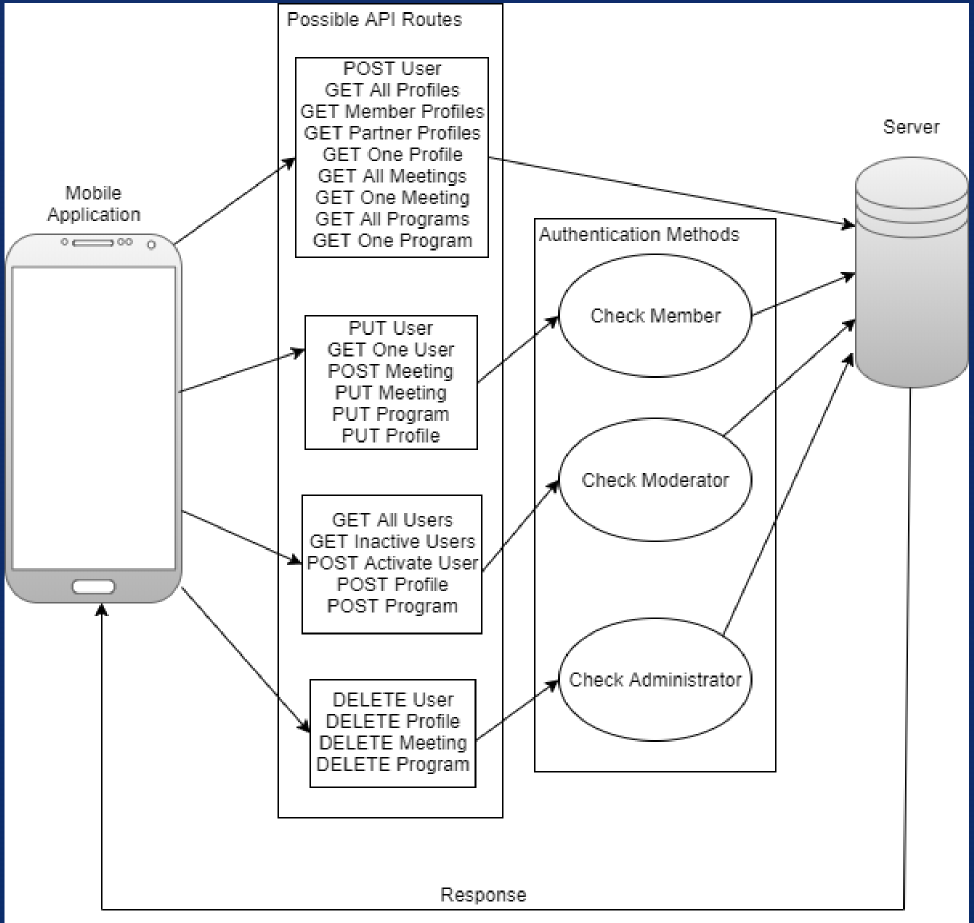


Meeting

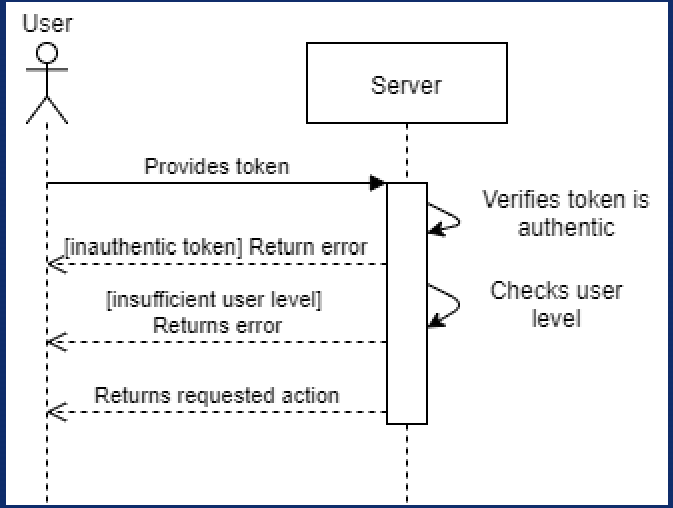
title: String
agenda: String
minutes: String
past: boolean
date: String

Class Diagrams for User, Profile, Program and Meeting objects.

System Design

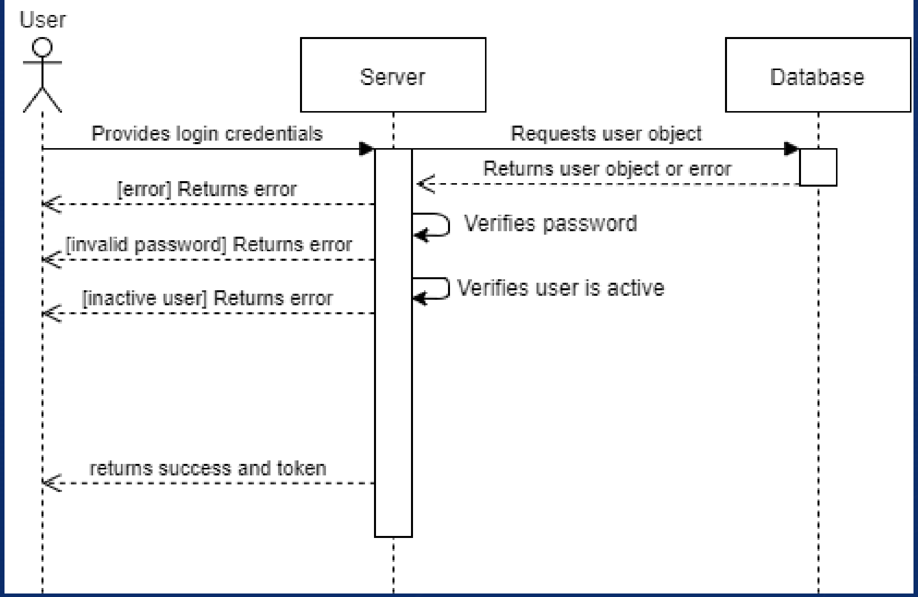


Authentication / Token Verification Design


User
Server
Provides token
Verifies token is authentic
[inauthentic token] Return error
[insufficient user level] Returns error
Checks user level
Returns requested action

Token Verification Sequence Diagram

User Authentication Sequence Diagram

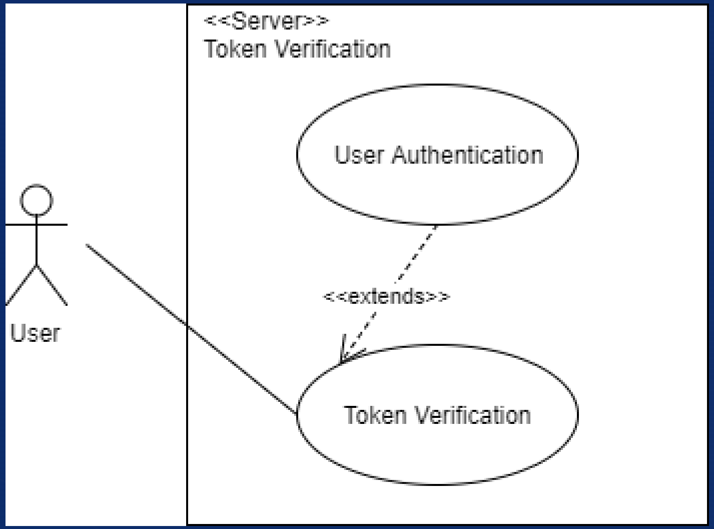

User
Server
Database
Provides login credentials
Requests user object
Returns user object or error
[error] Returns error
Verifies password
[invalid password] Returns error
Verifies user is active
[inactive user] Returns error
returns success and token

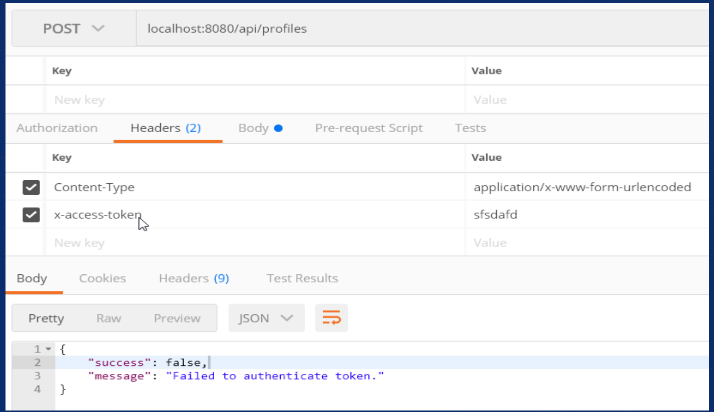
User Stories

Sprints	Completed User Stories
2	#666: Server Setup
3	#670: Create Profile Model #671: Create Profile API Routes #676: Create Main Menu Page
4	#678: Testing Calls to the Database #681: Create Meeting Model #683: Create Meeting API Routes #684: Create User Authentication Structure #688: User Authentication #690: Token Verification
5	#682: View Meetings #685: Create Program Model #686: Create Program API Routes
6	#680: Create New Meeting #691: View Member Profile #692: Log onto Server from Application #693: Search by Name or University #694: Edit Meeting #695: Edit Profile

Example User Story 1: #690: Token Verification

Use Case Name	Token Verification
Participating Actors	User, Server
Entry Condition	1. User provides token to Server.
Flow of Events	2. Server verifies token authenticity using predefined secret phrase. If token is not authentic: 2.1. Go to step X. If token is authentic: 2.2. Go to step 3. 3. Server checks the user level contained within the token. If the user level is insufficient for the requested action: 3.1. Go to step X. If the user level is sufficient for the requested action: 3.2. Go to step 4.
Exit Condition	4. Server either allows or prevents access. If token does not meet requirements for access: 4.1. Server returns error message. If token does meet requirements for access: 4.2. Server continues with requested action.





localhost:8080/api/profiles

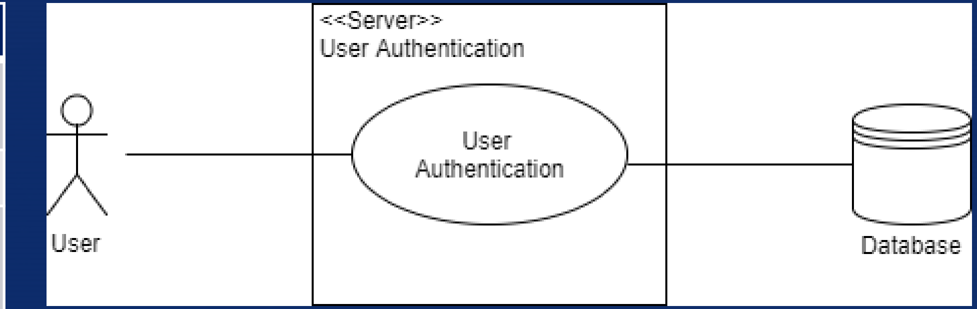
	Key	Value
	New key	Value

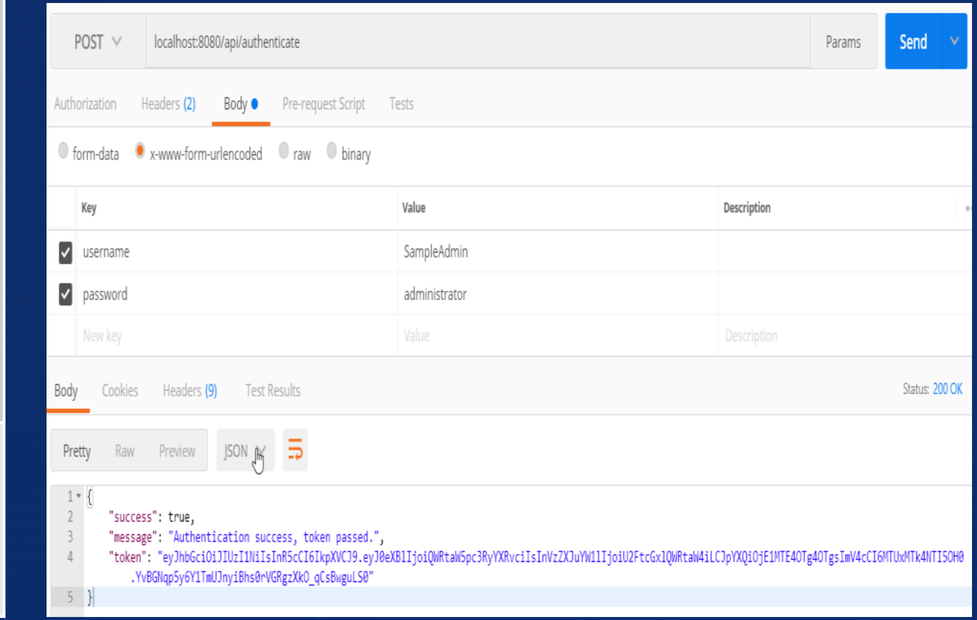
	Key	Value
☑	Content-Type	application/x-www-form-urlencoded
☑	x-access-token	sfsdafd
	New key	Value

```
{
    "success": false,
    "message": "Failed to authenticate token."
}
```

Example User Story 2: #688: User Authentication

Use Case Name	User Authentication
Participating Actors	User, Server, Database
Entry Condition	1. User provides credentials to Server.
Flow of Events	2. Server verifies that the username exists in the Database. If username does not exist in the Database: 3.1. Go to step 5.2. If username does exist in the Database: 3.2. Go to step 3. 3. Server verifies the provided password matches the password of the account associated with the username in the Database. If the passwords do not match: 4.1: Go to step 5.2. If the passwords do match: 4.2: Go to step 4. 4. Server verifies that the account is currently active. If the account is not active: 5.1: Go to step 5.2. If the account is active: 5.2: Go to step 5.1.
Exit Condition	5. Server returns response to User. If credentials were successfully authenticated: 6.1. Server returns success message and token to User. If credentials were not successfully authenticated: 6.2. Server returns error message to User.





POST localhost:8080/api/authenticate

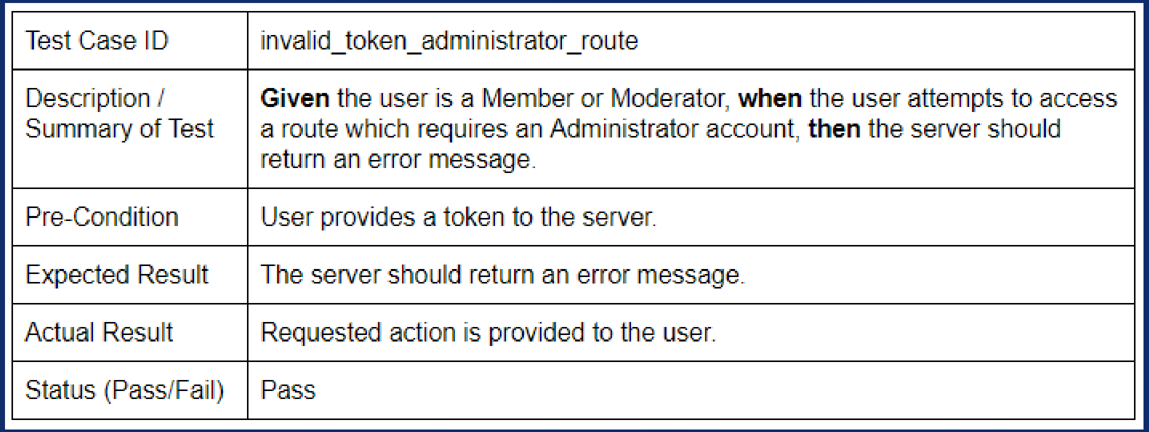
○ form-data ● x-www-form-urlencoded ○ raw ○ binary

	Key	Value	Description
☑	username	SampleAdmin	
☑	password	administrator	
	New key	Value	Description

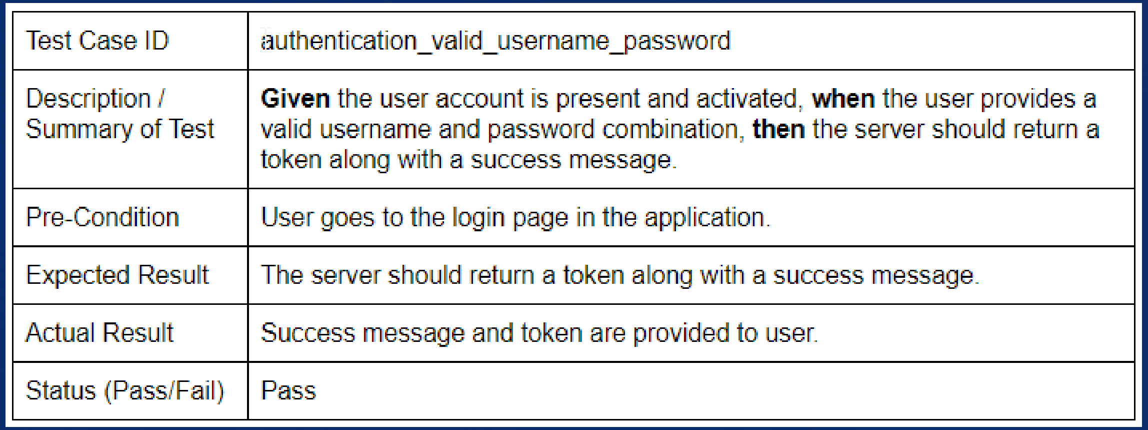
Status: 200 OK

```
{
    "success": true,
    "message": "Authentication success, token passed.",
    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0eXBlIjoiQWRtaW5pc3RyYXRvciIsInVzZXJuYW1lIjoiU2FtcGxlQWRtaW4iLCJpYXQiOjE1MTE4OTg4OTgsImV4cCI6MTUxMTk4NTI5OH0
      .YvBGNqp5y6Y1TmUJnyiBhs0rVGRgzXkO_qCsBwguLS0"
}
```

Verification (Example Test Cases)



Test Case ID	invalid_token_administrator_route
Description / Summary of Test	**Given** the user is a Member or Moderator, **when** the user attempts to access a route which requires an Administrator account, **then** the server should return an error message.
Pre-Condition	User provides a token to the server.
Expected Result	The server should return an error message.
Actual Result	Requested action is provided to the user.
Status (Pass/Fail)	Pass



Test Case ID	authentication_valid_username_password
Description / Summary of Test	**Given** the user account is present and activated, **when** the user provides a valid username and password combination, **then** the server should return a token along with a success message.
Pre-Condition	User goes to the login page in the application.
Expected Result	The server should return a token along with a success message.
Actual Result	Success message and token are provided to user.
Status (Pass/Fail)	Pass

Summary

- General users may find information about Positive Pathways’ programs, partners, members and meetings.

- Members may edit their profiles, programs, and meetings.

- Application data guarded against tampering through user authentication system.