Technological University Dublin Tallaght
Department of Computing

April 2021, Fourth year project

Employee Connectivity Tracker

Pavel Ivanov – x00149863

Contents

Introduction

With the COVID-19 outbreak, most workplaces closed their offices and sent their employees to work from home. And with the reduction of face-to-face interactions said employees are more likely to start feeling isolated and close themselves off.

The 'Employee Connectivity Tracker' application provides the Managers/Team leaders with the required tools to detect such behaviour on time and reach out to teammates that may be isolated.

Logical architecture

The logical architecture (Figure 1) includes a few components:

- Blazor client – the user interface of the application.
- .NET 5 Web API – contains most of the processing logic.
- Azure App Services – where the application is hosted as a web server.
- Microsoft Graph API – the communication data is pulled from this API.
- Azure SQL Database – cloud database of choice.
- Azure Active Directory – user authentication.
- GitHub – code repository and issue tracking.
- Azure Pipelines – continuous integration pipeline.

The logical architecture on Figure 1 is how the application can be scaled up to handle an increase in demand. The Azure plan for students does not allow for adding scaling options to App Services or databases, so the actual architecture does not include scaling.

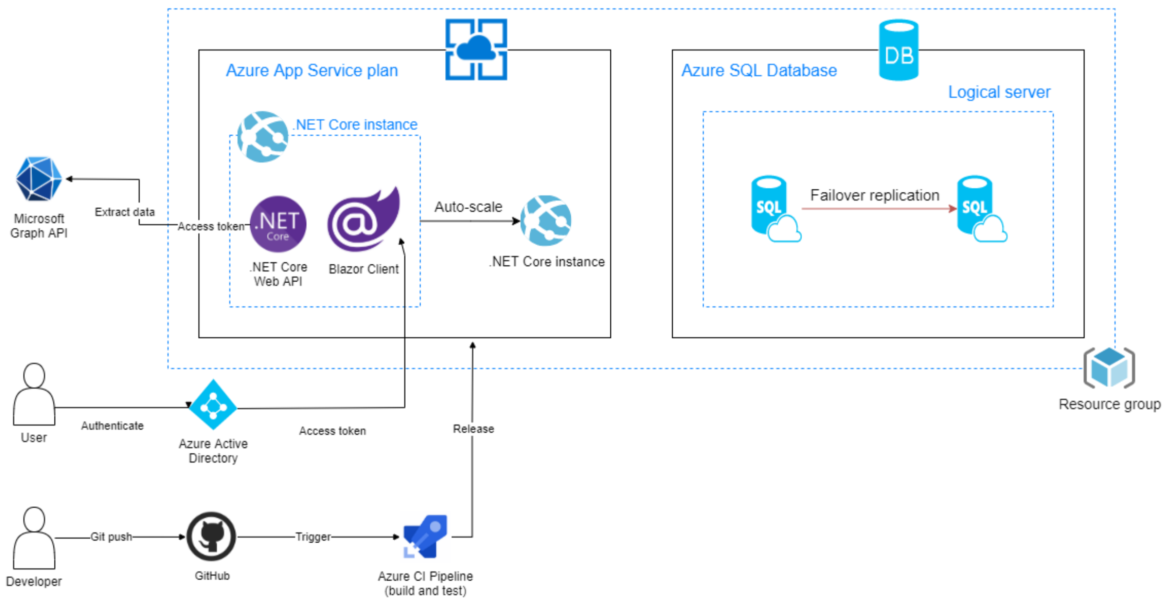


Figure 1. Potential logical architecture.

Database schema

The schema was initially designed to contain the users' information – the last time they signed in, and their communication data – emails and scheduled Outlook meetings. At a later stage in-app teams were added, and more information needed to be stored for each user, such as their email address and full name. Administrators are distinguished by simply having a record in the Administrators table with a valid foreign key pointing to a user.

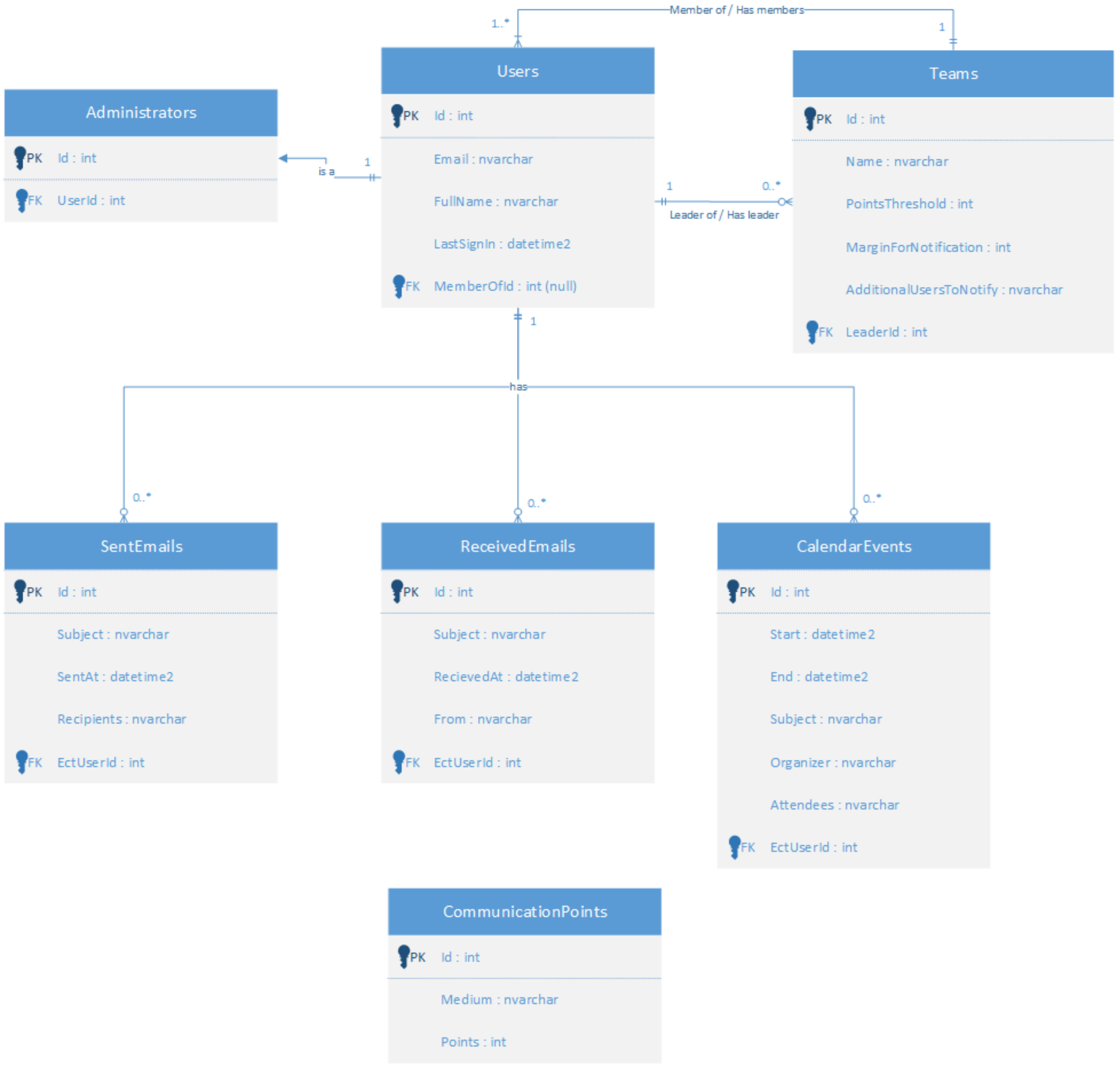


Figure 2. Database schema.

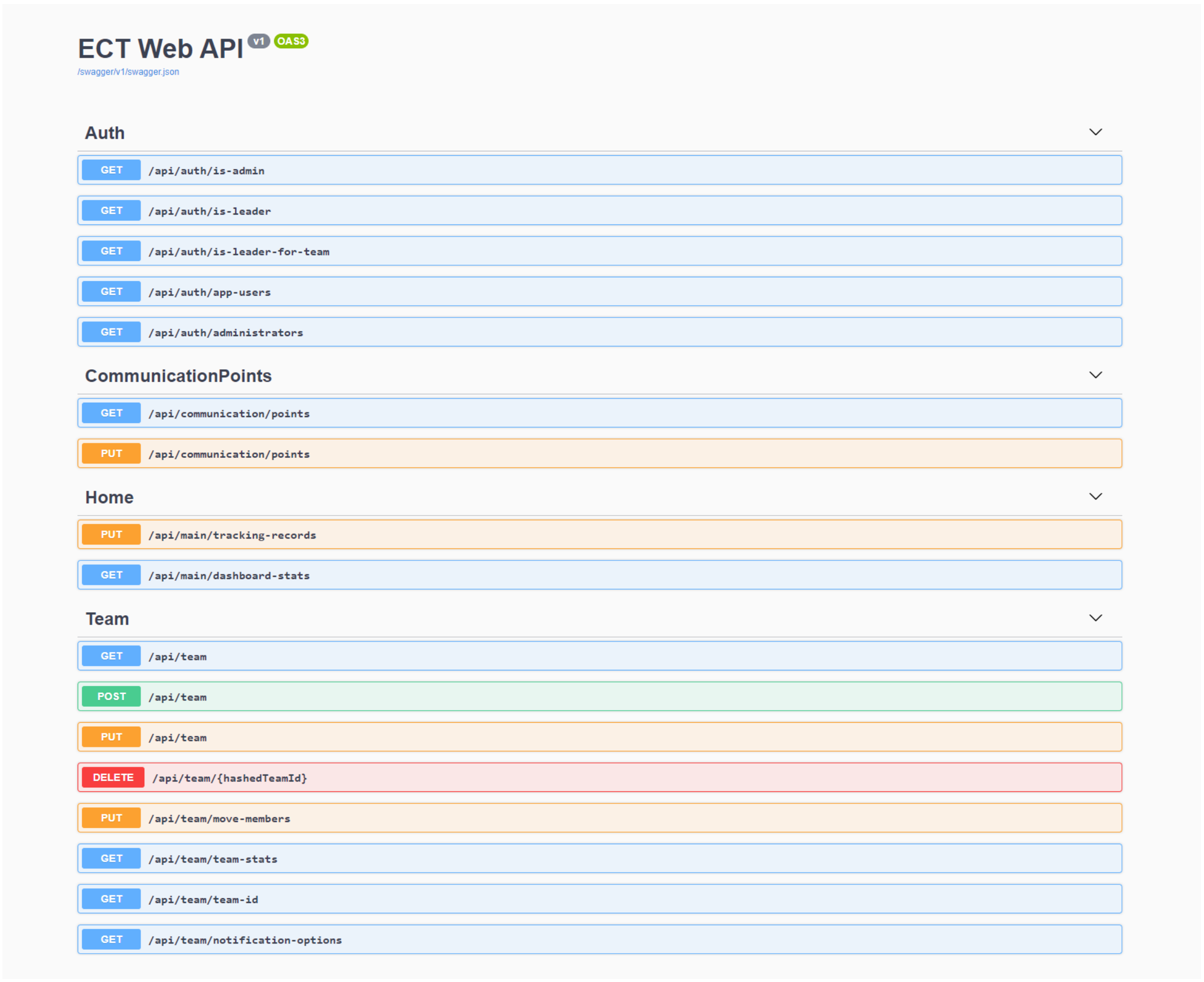
API structure

The .NET 5 Web API includes various endpoints to control access and perform CRUD operations. Endpoints are grouped logically in controllers (Figure 3):

- Auth

  Contains HTTP GET operations that return a true/false value.
  (except for */is-leader-for-team* which returns either null if the user does not have access or an object if they do)
- CommunicationPoints

  Contains operations to read out and update the communication points table.
- Home
  - HTTP PUT endpoint that triggers the retrieval of the user's communication data from Microsoft Graph API.
  - HTTP GET endpoint to retrieve the communication records from the database for a specific user.
- Team
  - Contains all endpoints that support full CRUD functionality for in-app teams.
  - Contains utility endpoints that return the stats for a full team, notification options and hashed team id.



ECT Web API v1 OAS3
/swagger/v1/swagger.json

Auth
GET /api/auth/is-admin
GET /api/auth/is-leader
GET /api/auth/is-leader-for-team
GET /api/auth/app-users
GET /api/auth/administrators

CommunicationPoints
GET /api/communication/points
PUT /api/communication/points

Home
PUT /api/main/tracking-records
GET /api/main/dashboard-stats

Team
GET /api/team
POST /api/team
PUT /api/team
DELETE /api/team/{hashedTeamId}
PUT /api/team/move-members
GET /api/team/team-stats
GET /api/team/team-id
GET /api/team/notification-options

Figure 3. Open API – Swagger page.

Access to Microsoft Graph API

Whenever a user clicks on ‘Log in’ or tries to load a protected page that requires authentication without the required credentials, they are presented with a Microsoft login pop-out (Figure 4). If the user’s browser does not allow pop-ups, then they are instead redirected to a Microsoft login page (Figure 5).
After entering their credentials, Microsoft verifies them against the defined rules in the Azure App Registration for the client (Figure 6). If all the validations pass, and the user’s organization has allowed the user to log into the application (Figure 7), then they will be redirected to the web application and the third-party libraries handle creating and maintaining the session.

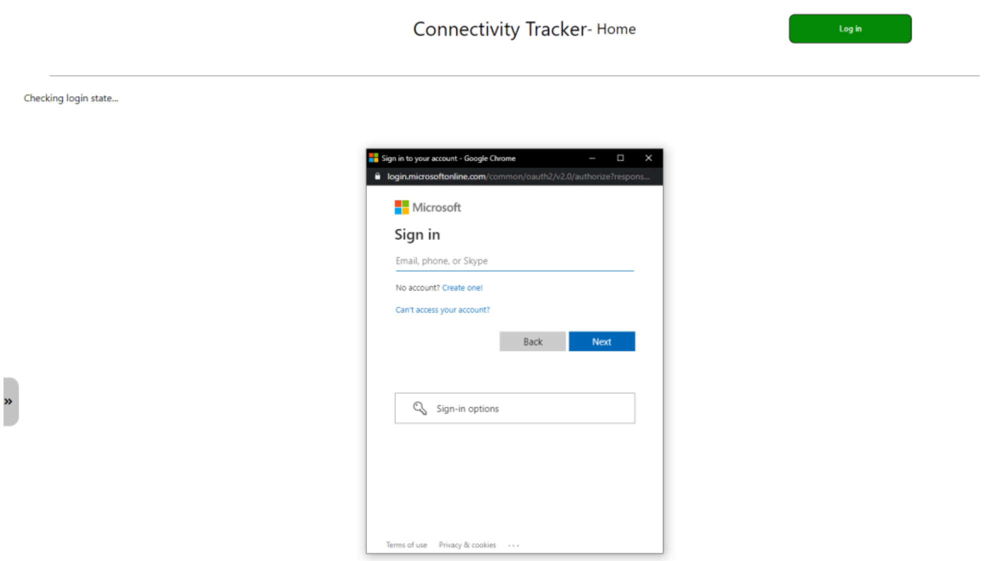

Connectivity Tracker- Home
Log in
Checking login state...
Sign in to your account - Google Chrome
login.microsoftonline.com/common/oauth2/v2.0/authorize?respons...
Microsoft
Sign in
Email, phone, or Skype
No account? Create one!
Can't access your account?
Back
Next
Sign-in options
Terms of use
Privacy & cookies

Figure 4. Login with Microsoft – pop-out.

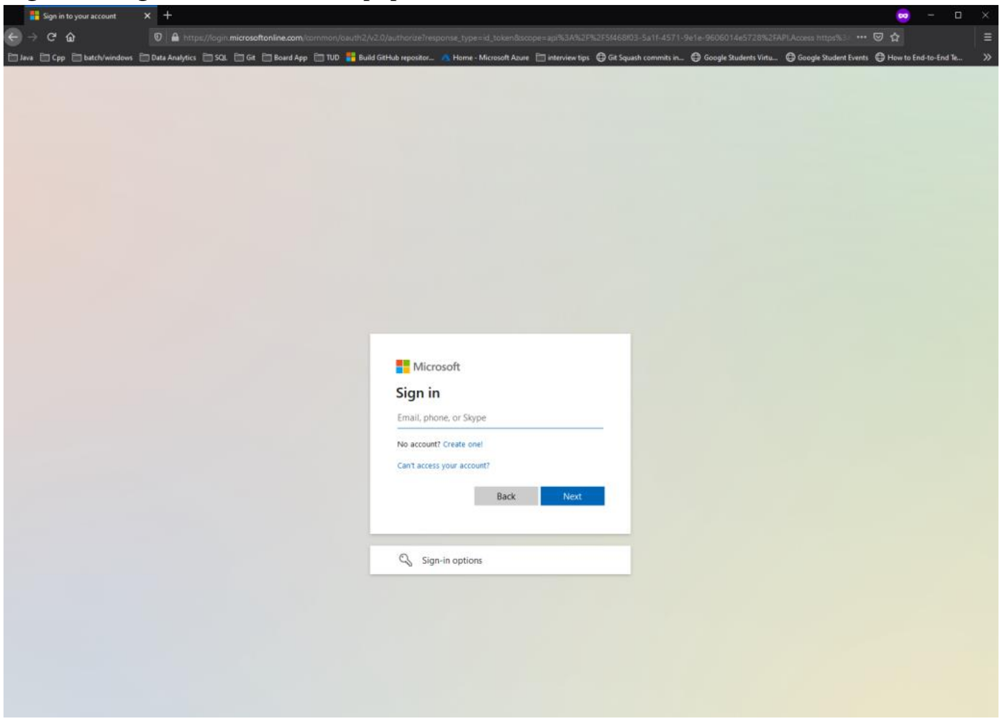
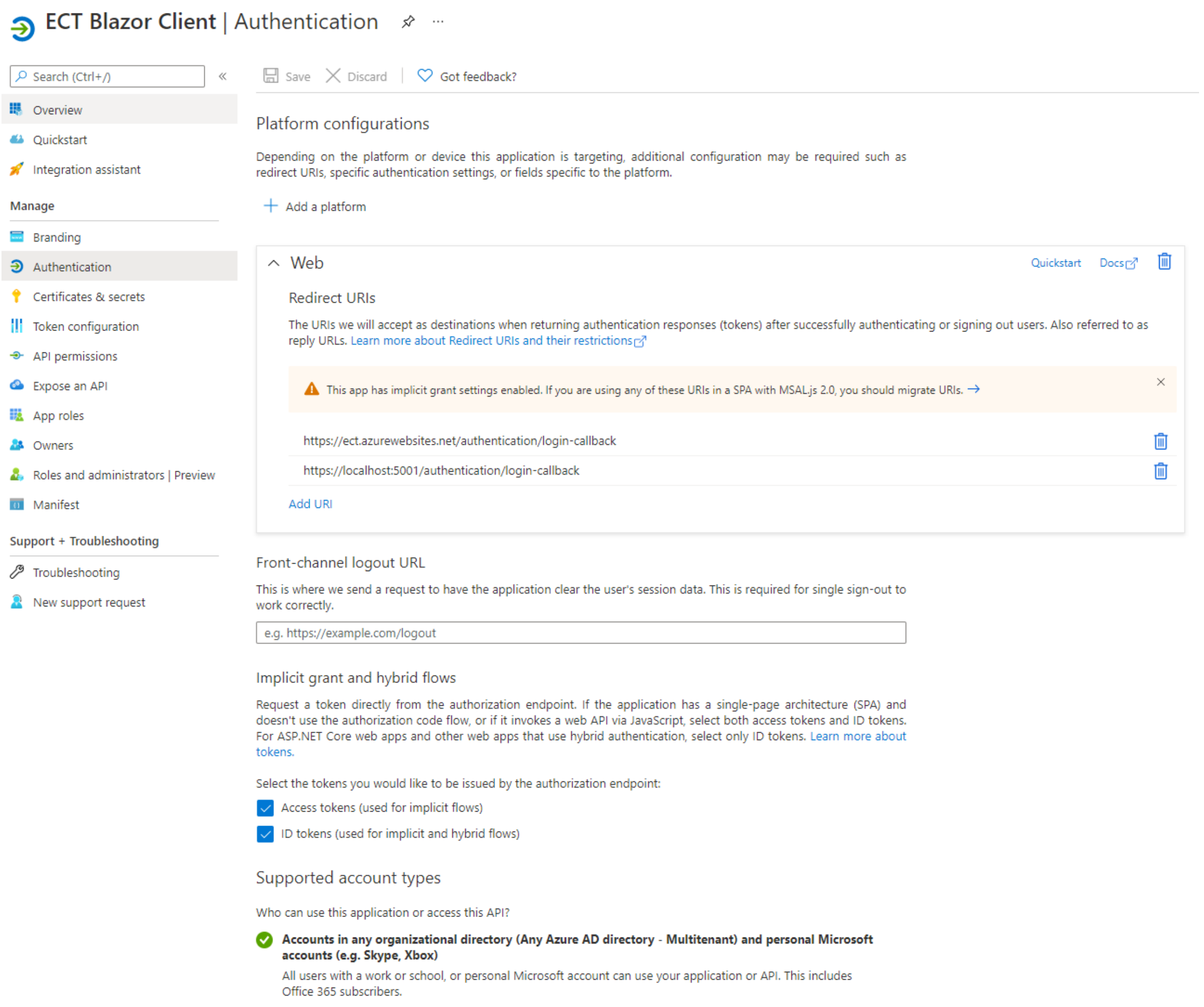


Figure 5. Login with Microsoft – redirect.



Platform configurations

Depending on the platform or device this application is targeting, additional configuration may be required such as redirect URIs, specific authentication settings, or fields specific to the platform.

Web

Redirect URIs

The URIs we will accept as destinations when returning authentication responses (tokens) after successfully authenticating or signing out users. Also referred to as reply URLs. Learn more about Redirect URIs and their restrictions

This app has implicit grant settings enabled. If you are using any of these URIs in a SPA with MSAL.js 2.0, you should migrate URIs. →

https://ect.azurewebsites.net/authentication/login-callback

https://localhost:5001/authentication/login-callback

Front-channel logout URL

This is where we send a request to have the application clear the user's session data. This is required for single sign-out to work correctly.

e.g. https://example.com/logout

Implicit grant and hybrid flows

Request a token directly from the authorization endpoint. If the application has a single-page architecture (SPA) and doesn't use the authorization code flow, or if it invokes a web API via JavaScript, select both access tokens and ID tokens. For ASP.NET Core web apps and other web apps that use hybrid authentication, select only ID tokens. Learn more about tokens.

Select the tokens you would like to be issued by the authorization endpoint:

- [✓] Access tokens (used for implicit flows)
- [✓] ID tokens (used for implicit and hybrid flows)

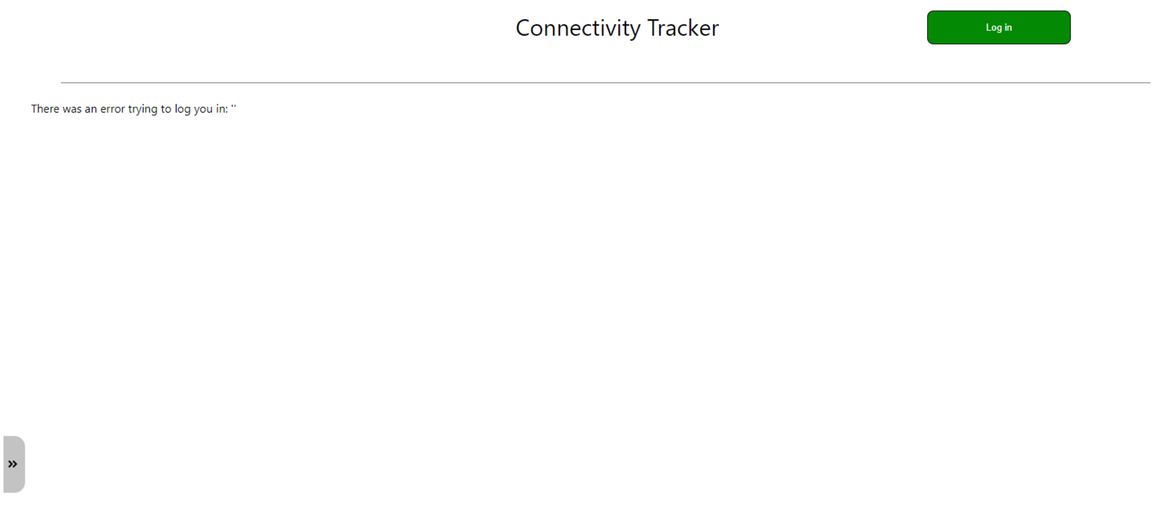
Supported account types

Who can use this application or access this API?

✓ **Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)**

All users with a work or school, or personal Microsoft account can use your application or API. This includes Office 365 subscribers.

Figure 6. Blazor Client – Azure App registration.



There was an error trying to log you in: ''

Figure 7. Organization denied access to application.

Extracting data from Microsoft Graph API

Whenever the user logs in successfully they are redirected to the home page of the app (Figure 8). Once the page has loaded, the Blazor client initiates a request to the API. This request contains a personalized token that includes read permissions for email and Outlook Calendar data (Figure 9). The API endpoint then uses that token to retrieve all the emails and scheduled meetings for that user since their last log in and persist them to the database.

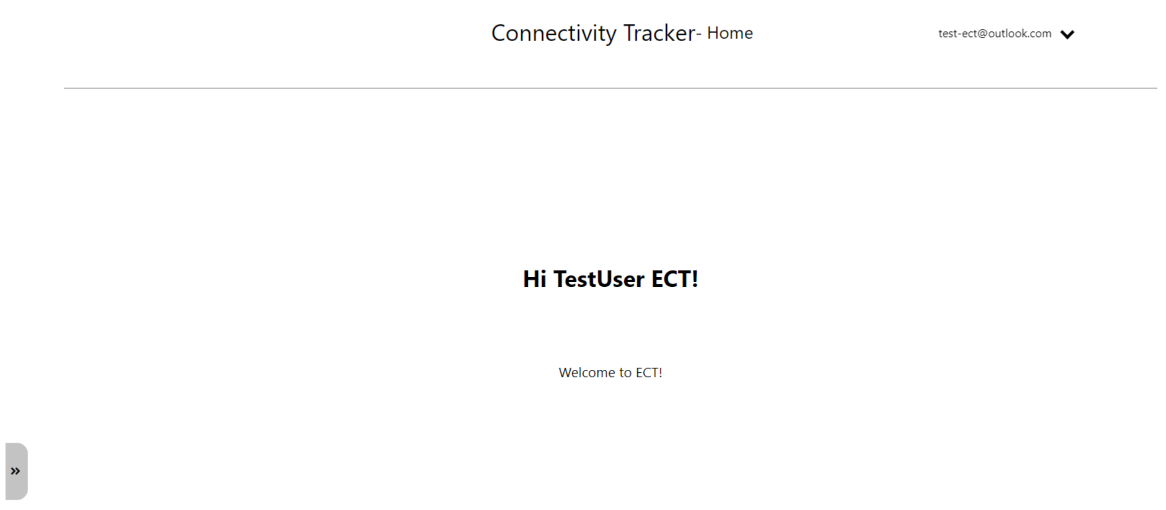
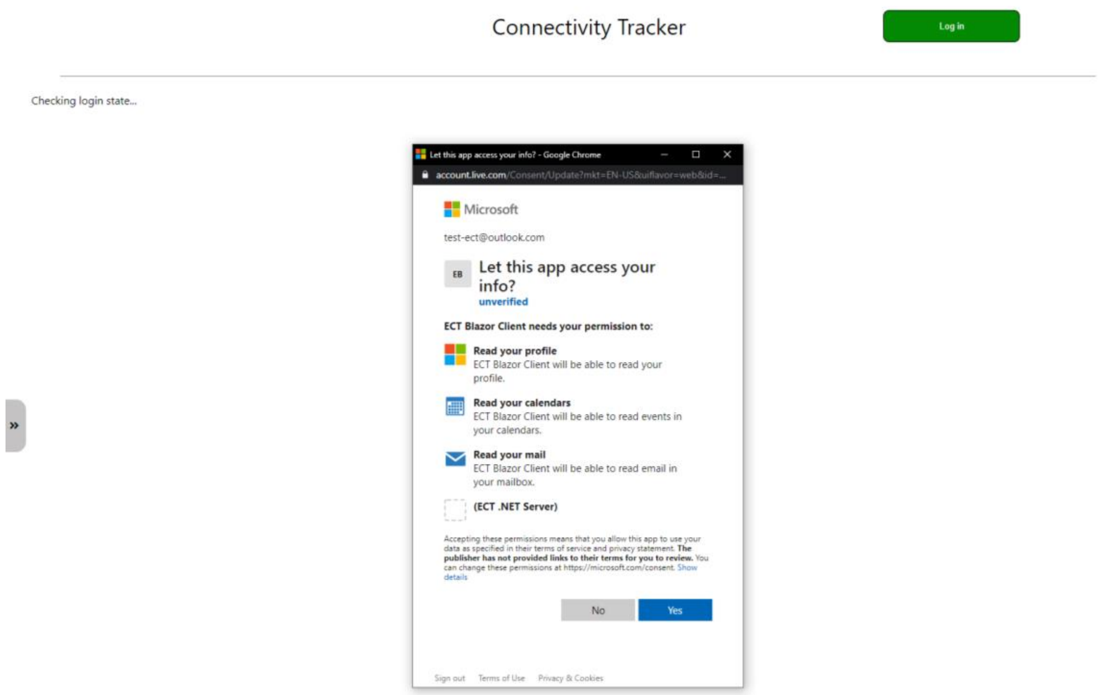


Figure 8. Home page.



Connectivity Tracker

Log in

Checking login state...

Let this app access your info? - Google Chrome

account.live.com/Consent/Update?mkt=EN-US&uiflavor=web&id=...

Microsoft

test-ect@outlook.com

EB Let this app access your info?

unverified

ECT Blazor Client needs your permission to:

Read your profile
ECT Blazor Client will be able to read your profile.

Read your calendars
ECT Blazor Client will be able to read events in your calendars.

Read your mail
ECT Blazor Client will be able to read email in your mailbox.

(ECT .NET Server)

Accepting these permissions means that you allow this app to use your data as specified in their terms of service and privacy statement. **The publisher has not provided links to their terms for you to review.** You can change these permissions at https://microsoft.com/consent. Show details

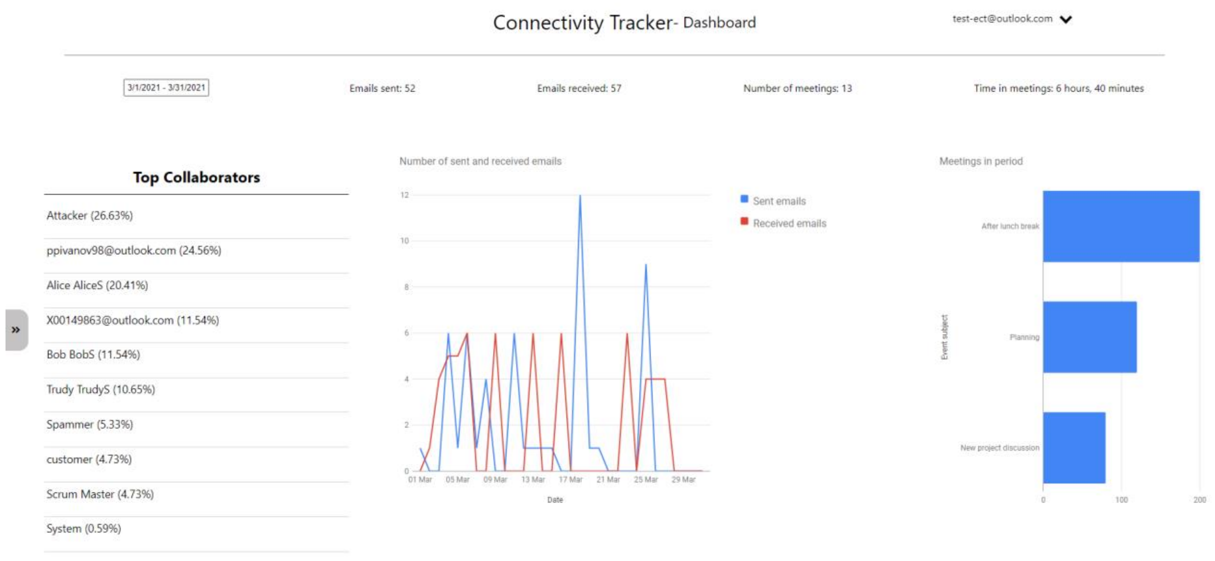
No Yes

Sign out Terms of Use Privacy & Cookies

Figure 9. Microsoft prompts new users to explicitly give the application read access to emails and calendars.

Viewing the data

Each user has access to the data the application has retrieved for them. The data itself is presented in visualisations for easier consumption. Every user has access to a personal dashboard (Figure 10), where they can view the data as 2 graphs – a double line graph to represent the sent and received emails, and a horizontal bar chart to display the scheduled meetings.
In the top-left side of the dashboard page, an input field in the form of a date range picker can be found (Figure 11) which filters the data displayed on the dashboard.



Connectivity Tracker- Dashboard

test-ect@outlook.com

3/1/2021 - 3/31/2021

Emails sent: 52

Emails received: 57

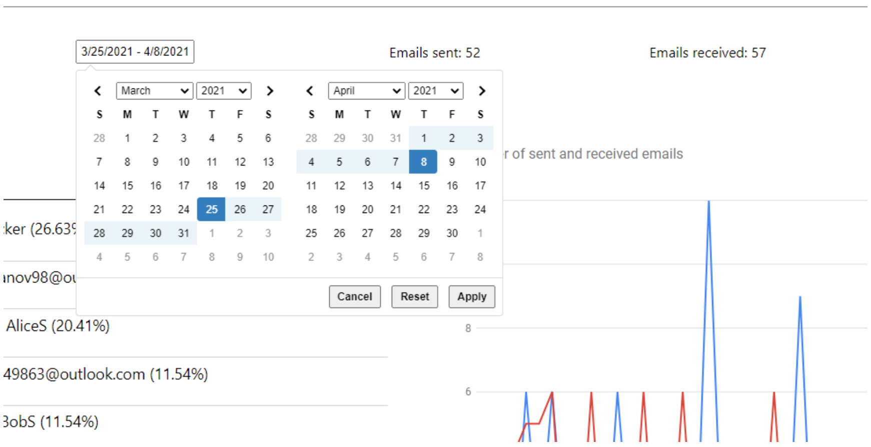
Number of meetings: 13

Time in meetings: 6 hours, 40 minutes

Top Collaborators

- Attacker (26.63%)
- ppivanov98@outlook.com (24.56%)
- Alice AliceS (20.41%)
- X00149863@outlook.com (11.54%)
- Bob BobS (11.54%)
- Trudy TrudyS (10.65%)
- Spammer (5.33%)
- customer (4.73%)
- Scrum Master (4.73%)
- System (0.59%)

Figure 10. TestUser ECT's Personal dashboard for all data in March.



3/25/2021 - 4/8/2021

Emails sent: 52

Emails received: 57

ker (26.63

nov98@ou

AliceS (20.41%)

49863@outlook.com (11.54%)

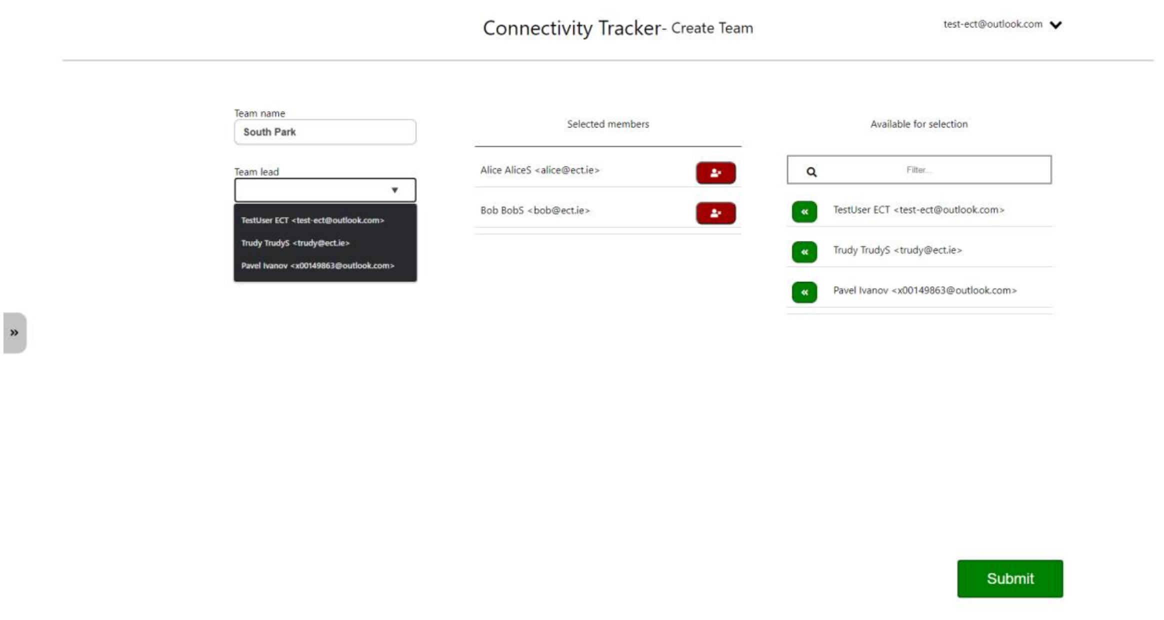
3obS (11.54%)

Figure 11. Date range picker. Open-source package: https://github.com/jdtcn/BlazorDateRangePicker
NuGet: https://www.nuget.org/packages/BlazorDateRangePicker/

Creating in-app teams

In-app teams can be registered by the administrators (Figure 12).
The page is designed with user experience in mind. It comprises of two input fields:
one for the team's name and one for the team leader. The team leader field is linked with a data list populated with all available users that can be selected. Typing in the field filters the options and saves the user double checking or scrolling through before making the selection.

Members can be added to the roster by simply clicking on the button beside them. The list of selectable users can be filtered via the 'Filter…' field above them.



Team name: South Park

Team lead:

TestUser ECT <test-ect@outlook.com>
Trudy TrudyS <trudy@ect.ie>
Pavel Ivanov <x00149863@outlook.com>

Selected members

Alice AliceS <alice@ect.ie>
Bob BobS <bob@ect.ie>

Available for selection

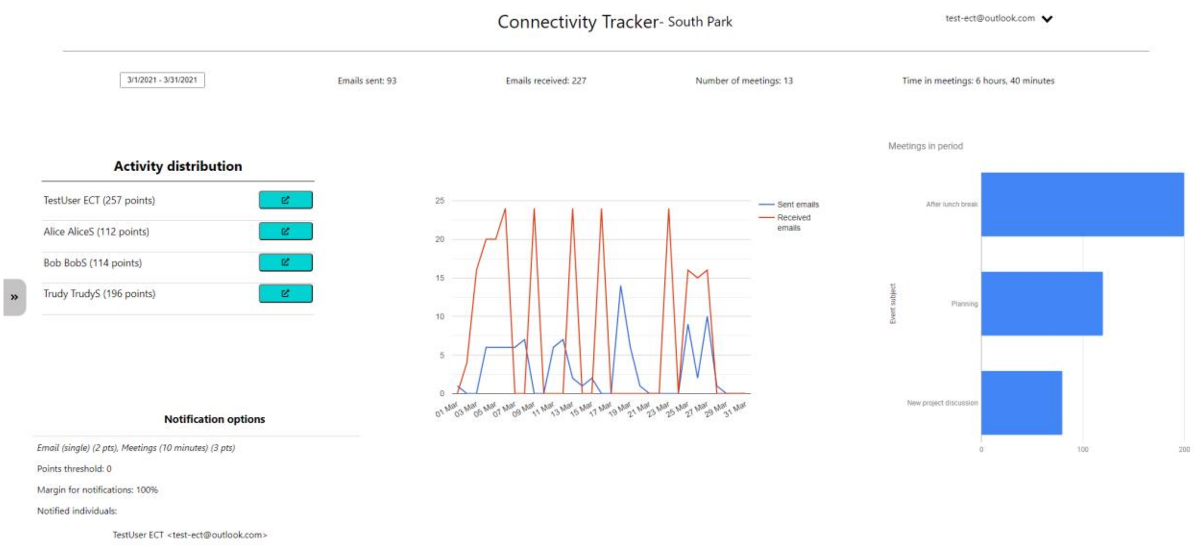
TestUser ECT <test-ect@outlook.com>
Trudy TrudyS <trudy@ect.ie>
Pavel Ivanov <x00149863@outlook.com>

Figure 12. Create Team page.

Viewing the data generated by team members

Team leaders have access to more pages and have additional options on the navigation menu. One of these options is “My Team”, where they are presented with a dashboard, that looks very similar to the Personal dashboard view with the difference that My Team summarises the communication data for all members within that team (Figure 13).

On this page the team leader can get an overview of how their team is doing. The interactive graphs provide some detail about the underlying data per individual (Figure 14). If the user sees some inconsistencies or simply decides to view how a single member has been doing recently - they can click on the light-blue button beside their name (Figure 13) which will open a new browser tab and display them a dashboard only for that person (Figure 15).


Connectivity Tracker- South Park

test-ect@outlook.com

3/1/2021 - 3/31/2021 Emails sent: 93 Emails received: 227 Number of meetings: 13 Time in meetings: 6 hours, 40 minutes

Activity distribution

TestUser ECT (257 points)
Alice AliceS (112 points)
Bob BobS (114 points)
Trudy TrudyS (196 points)

Notification options

Email (single) (2 pts), Meetings (10 minutes) (3 pts)
Points threshold: 0
Margin for notifications: 100%
Notified individuals:
TestUser ECT <test-ect@outlook.com>

Figure 13. My Team – summarised team information.

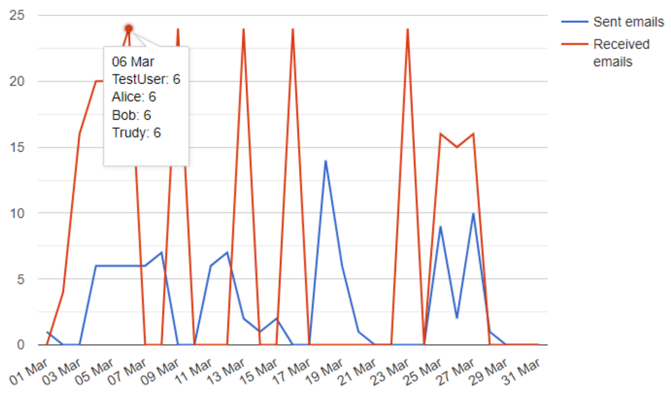
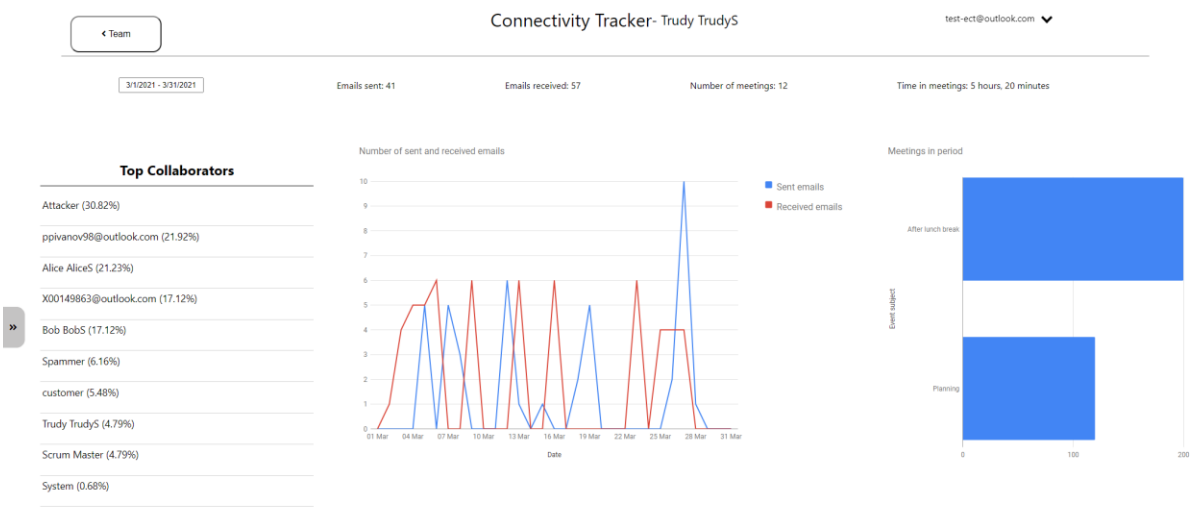


Figure 14. My Team – interaction with line graph.



3/1/2021 - 3/31/2021

Emails sent: 41 | Emails received: 57 | Number of meetings: 12 | Time in meetings: 5 hours, 20 minutes

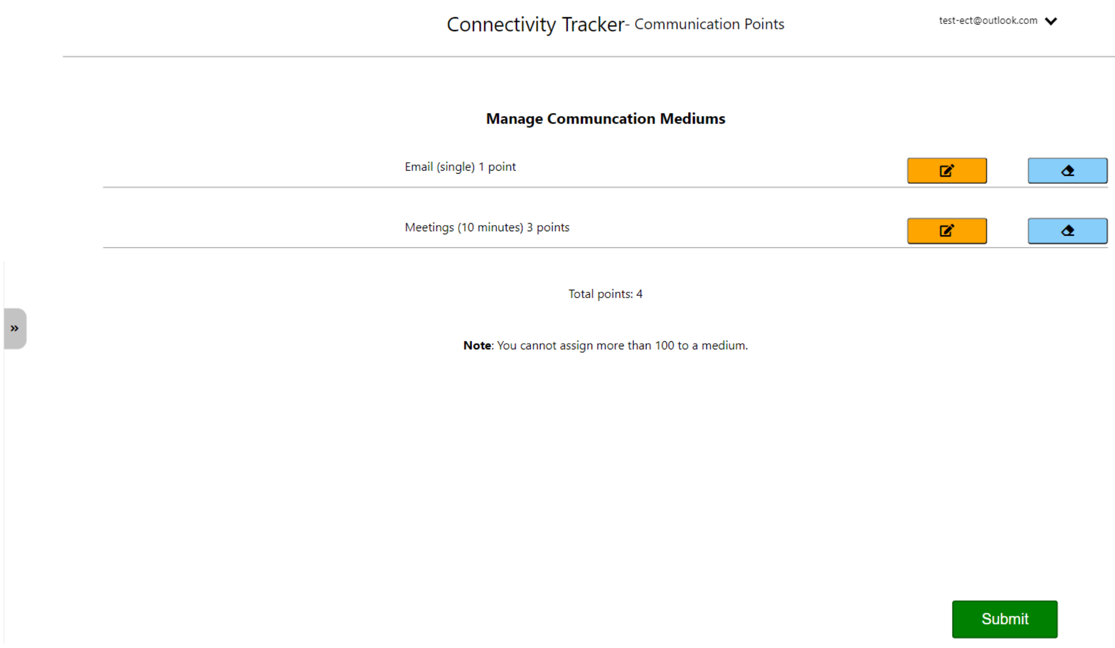
Top Collaborators

Attacker (30.82%)
ppivanov98@outlook.com (21.92%)
Alice AliceS (21.23%)
X00149863@outlook.com (17.12%)
Bob BobS (17.12%)
Spammer (6.16%)
customer (5.48%)
Trudy TrudyS (4.79%)
Scrum Master (4.79%)
System (0.68%)

Figure 15. Drilling into a team member to view more details.

Different forms of communication – different weights

As per the initial requirements for this project, it was necessary to assign different weights to different types of communication in order to identify teammates who may be becoming isolated. Administrators are authorized to access the page ‘Manage Mediums’ (Figure 16) where they can modify the weights in points for each communication medium that the application supports. Each medium can be assigned between 0 and 100 points that will then be used for automated email notifications (see Email notifications and notification options for more). These configurable values are also used to determine the top collaborators of a user (Figures 10 and 15) as well as to display the total points for a team member on the My Team view (Figure 13).



Manage Communcation Mediums

Email (single) 1 point

Meetings (10 minutes) 3 points

Total points: 4

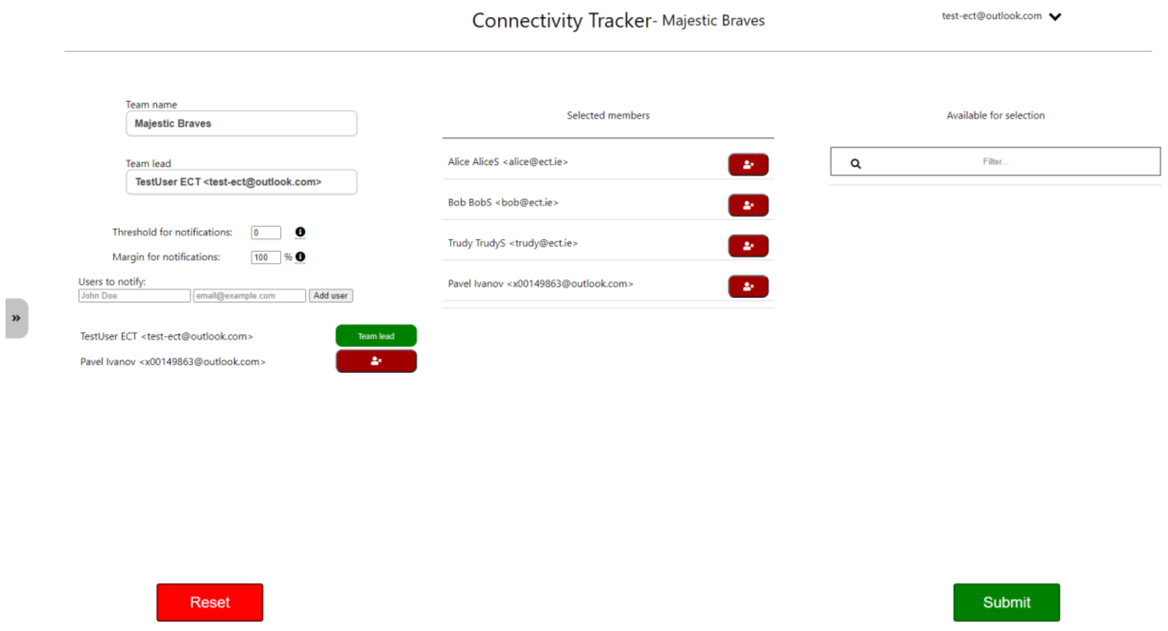
Note: You cannot assign more than 100 to a medium.

Figure 16. Manage Communication Mediums page.

Email notifications and notification options

Apart from viewing the team members' data and making a decision based on what they see, the team leaders can configure 'notification options' for a particular team (Figures 13 and 17). These notification options include a minimum number of 'points' to be generated by each member within a week's timeframe. Another configurable option is to set a maximum percentage of drop in the communication data. To clarify what these options are doing, they are both accompanied by a help text which is displayed only when the user hovers over the information icon, i.e., a tooltip (Figure 18).

At the end of every week a background job processes each of the registered in-app teams and determines which teammate should be reported as potentially isolated by taking the aforementioned notification options into account. If a team member has less points than the points threshold or their communication has dropped more than what the team leader has specified, then they will be silently added to the weekly email report sent to the team leader (Figure 19).
The recipients of said report are also configurable, except for the team leader who will always receive it (see *Users to notify* in Figure 17).



Team name: Majestic Braves

Team lead: TestUser ECT <test-ect@outlook.com>

Threshold for notifications: 0

Margin for notifications: 100 %

Users to notify: John Doe | email@example.com

TestUser ECT <test-ect@outlook.com>

Pavel Ivanov <x00149863@outlook.com>

Selected members:

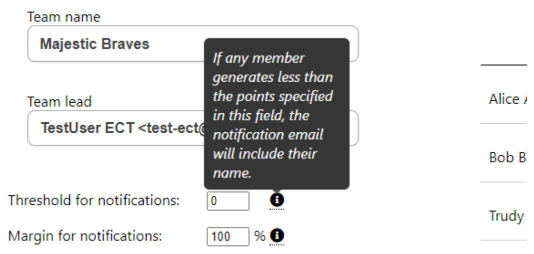
Alice AliceS <alice@ect.ie>

Bob BobS <bob@ect.ie>

Trudy TrudyS <trudy@ect.ie>

Pavel Ivanov <x00149863@outlook.com>

Figure 17. Manage Team page.



Team name: Majestic Braves

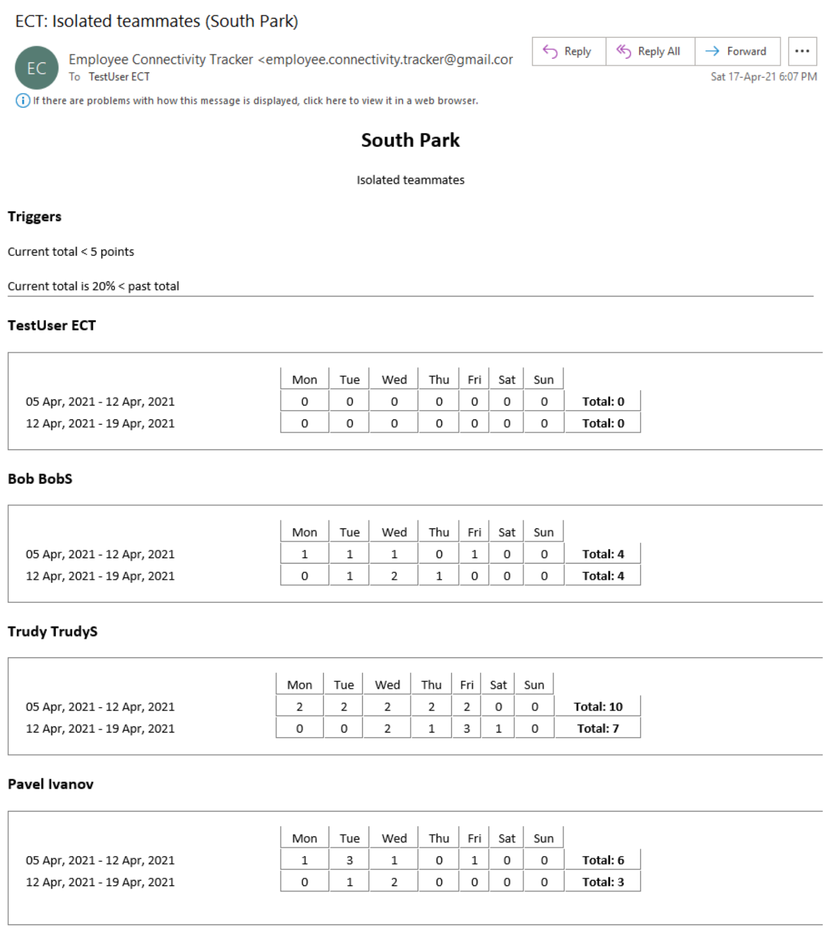
Team lead: TestUser ECT <test-ect(

If any member generates less than the points specified in this field, the notification email will include their name.

Threshold for notifications: 0

Margin for notifications: 100 %

Figure 18. Notification options tooltip.



ECT: Isolated teammates (South Park)

Employee Connectivity Tracker <employee.connectivity.tracker@gmail.cor
To TestUser ECT

Sat 17-Apr-21 6:07 PM

If there are problems with how this message is displayed, click here to view it in a web browser.

South Park

Isolated teammates

Triggers

Current total < 5 points

Current total is 20% < past total

TestUser ECT

	Mon	Tue	Wed	Thu	Fri	Sat	Sun	
05 Apr, 2021 - 12 Apr, 2021	0	0	0	0	0	0	0	**Total: 0**
12 Apr, 2021 - 19 Apr, 2021	0	0	0	0	0	0	0	**Total: 0**

Bob BobS

	Mon	Tue	Wed	Thu	Fri	Sat	Sun	
05 Apr, 2021 - 12 Apr, 2021	1	1	1	0	1	0	0	**Total: 4**
12 Apr, 2021 - 19 Apr, 2021	0	1	2	1	0	0	0	**Total: 4**

Trudy TrudyS

	Mon	Tue	Wed	Thu	Fri	Sat	Sun	
05 Apr, 2021 - 12 Apr, 2021	2	2	2	2	2	0	0	**Total: 10**
12 Apr, 2021 - 19 Apr, 2021	0	0	2	1	3	1	0	**Total: 7**

Pavel Ivanov

	Mon	Tue	Wed	Thu	Fri	Sat	Sun	
05 Apr, 2021 - 12 Apr, 2021	1	3	1	0	1	0	0	**Total: 6**
12 Apr, 2021 - 19 Apr, 2021	0	1	2	0	0	0	0	**Total: 3**

Figure 19. Email notification for team.

Data behind the notification:

Alice AliceS has:

- 10 points for dates between 5th and 12th April.
- 9 points for dates between 12th and 19th April.

(*She is above the threshold and within the 20% margin. Hence, she is* ***not included*** *in the report*)

Bob BobS has:

- 4 points for dates between 5th and 12th April.
- 4 points for dates between 12th and 19th April.

(*He is* ***below*** *the threshold and within the 20% margin*)

Trudy TrudyS has:

- 10 points for dates between 5th and 12th April.
- 7 points for dates between 12th and 19th April.

(*She is above the threshold and has* ***30% drop*** *in communication*)

Pavel Ivanov has:

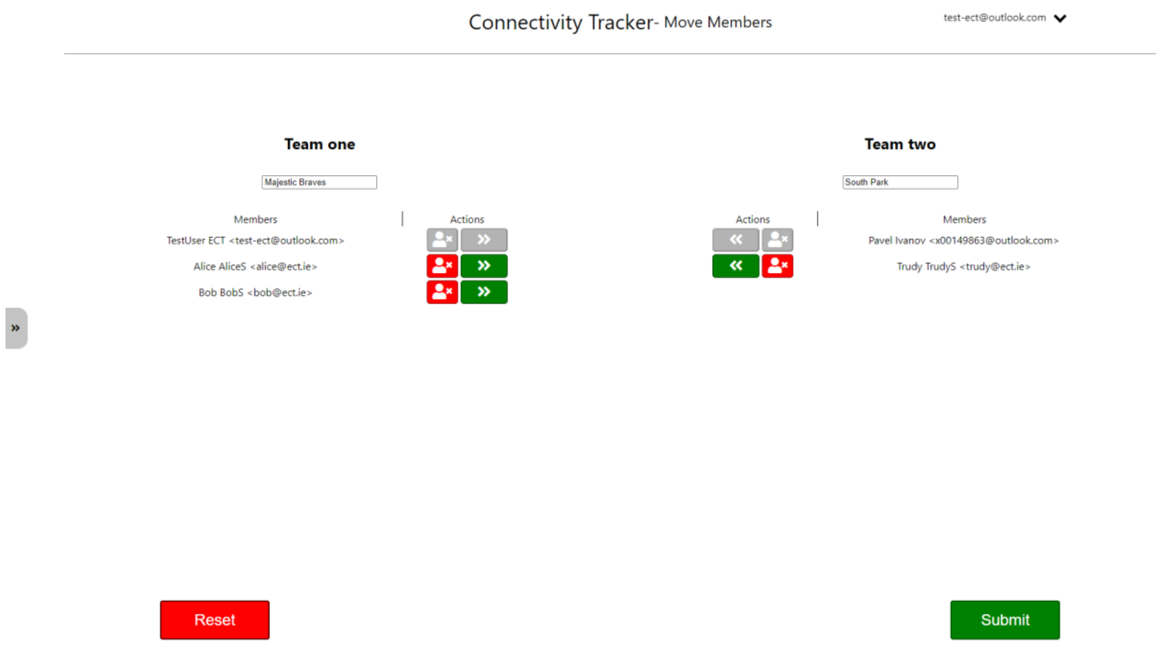
- 6 points for dates between 5th and 12th April.
- 3 points for dates between 12th and 19th April.

(*He is* ***below*** *the threshold and has* ***30% drop*** *in communication*)

Additional functionality for administrators

Moving members between two teams

Apart from creating teams and managing the medium weights, the administrators also have the ability to move members between two teams (Figure 20). The page consists of two input fields to select the teams, and once a team is selected the members appear with 2 buttons beside them – Move and Remove.



Team one

Majestic Braves

Members	Actions
TestUser ECT <test-ect@outlook.com>	
Alice AliceS <alice@ect.ie>	
Bob BobS <bob@ect.ie>	

Team two

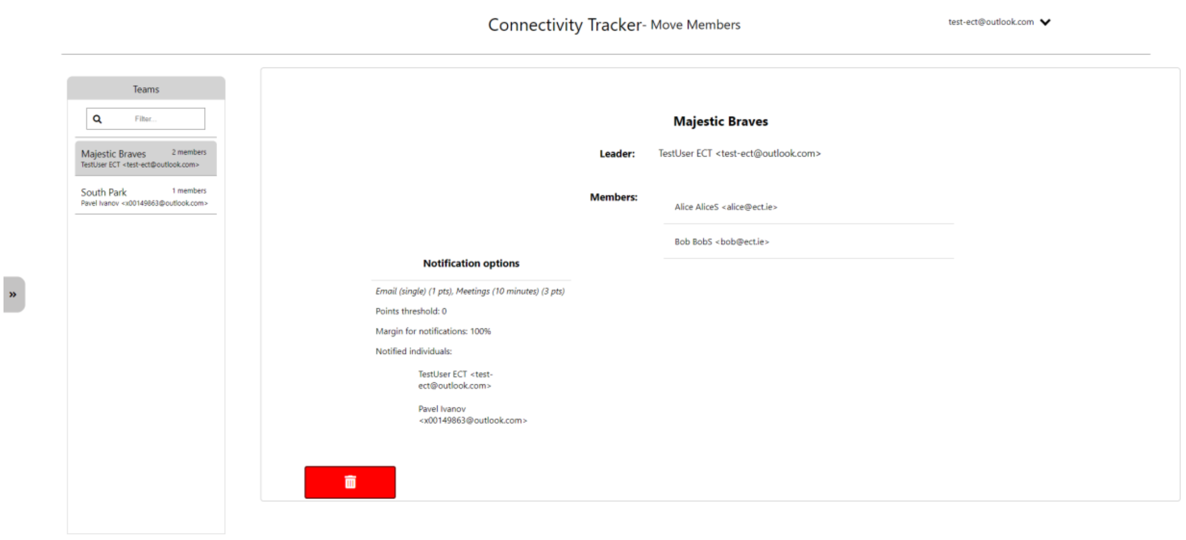
South Park

Actions	Members
	Pavel Ivanov <x00149863@outlook.com>
	Trudy TrudyS <trudy@ect.ie>

Figure 20. Move Members page.

Viewing all teams and their roster

Administrators also have access to a page where they can view all teams, their members, and the notification options that apply to them (Figure 21). An input field at the top of the list of teams serves the purpose of a filter, where the administrator can type in a keyword and all teams whose name partially matches the keyword will be displayed. This page also enables the administrators to disband a team, which frees up the members and makes them eligible for selection in another team (Figure 22).



Majestic Braves 2 members
TestUser ECT <test-ect@outlook.com>

South Park 1 members
Pavel Ivanov <x00149863@outlook.com>

Majestic Braves

Leader: TestUser ECT <test-ect@outlook.com>

Members: Alice AliceS <alice@ect.ie>
Bob BobS <bob@ect.ie>

Notification options

Email (single) (1 pts), Meetings (10 minutes) (3 pts)

Points threshold: 0

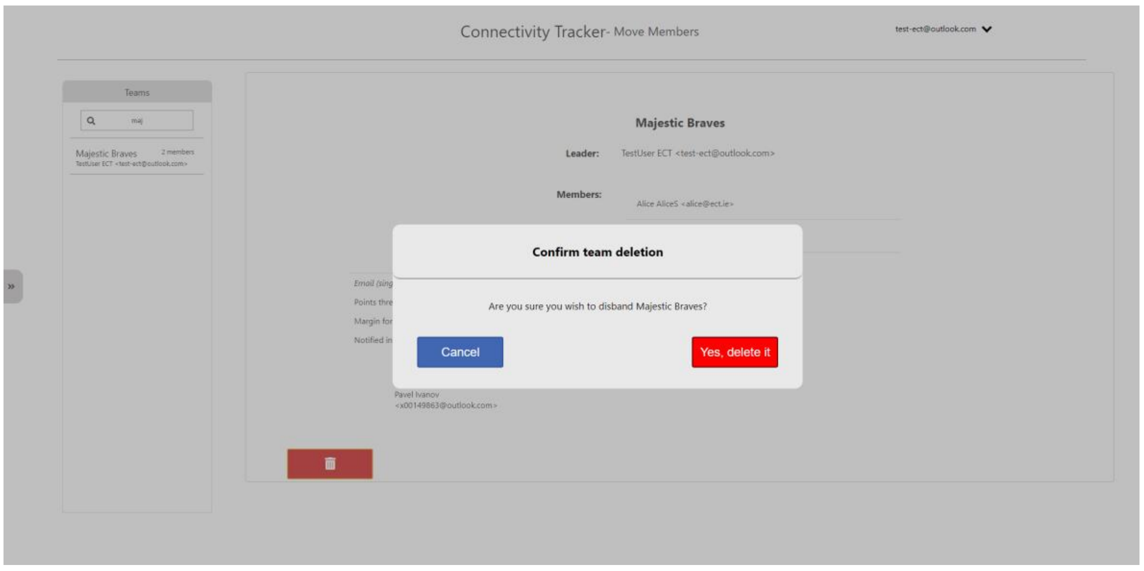
Margin for notifications: 100%

Notified individuals:

TestUser ECT <test-ect@outlook.com>

Pavel Ivanov <x00149863@outlook.com>

Figure 21. View Teams page.



Majestic Braves 2 members
TestUser ECT <test-ect@outlook.com>

Majestic Braves

Leader: TestUser ECT <test-ect@outlook.com>

Members: Alice AliceS <alice@ect.ie>

Confirm team deletion

Are you sure you wish to disband Majestic Braves?

Pavel Ivanov <x00149863@outlook.com>

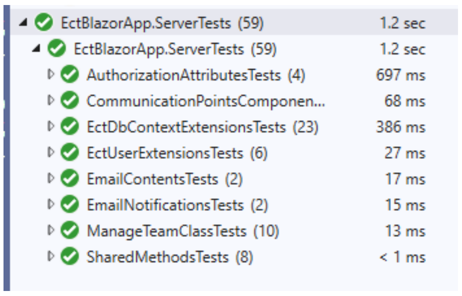
Figure 22. Delete team confirmation.

Unit tests

The application includes a unit test project utilizing the MSTest framework. This project contains 59 unit tests that verify the methods that support the core logic of the application are working as expected (Figure 23) and any changes that introduce a defect will most likely be caught by them.

The unit tests cover:

- Helper and utility methods.
- Custom authorization attributes (API authorization).
- Methods used in front-end logic.
- Methods fetching and updating the user’s communication records.
- Email notification methods.
- Other methods comprising the core functionality.



```
◢ ✓ EctBlazorApp.ServerTests (59)              1.2 sec
  ◢ ✓ EctBlazorApp.ServerTests (59)            1.2 sec
    ▷ ✓ AuthorizationAttributesTests (4)       697 ms
    ▷ ✓ CommunicationPointsComponen...          68 ms
    ▷ ✓ EctDbContextExtensionsTests (23)       386 ms
    ▷ ✓ EctUserExtensionsTests (6)              27 ms
    ▷ ✓ EmailContentsTests (2)                  17 ms
    ▷ ✓ EmailNotificationsTests (2)             15 ms
    ▷ ✓ ManageTeamClassTests (10)               13 ms
    ▷ ✓ SharedMethodsTests (8)                 < 1 ms
```

Figure 23. Local run of tests in ServerTests project.

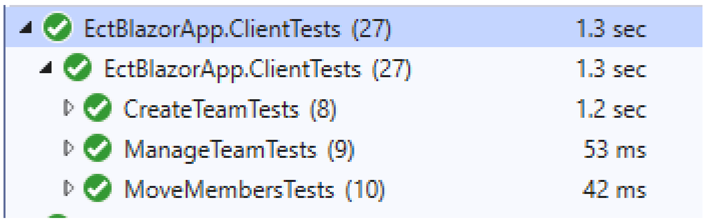
Component tests – bUnit

Since the front-end client is built using .NET Blazor framework, it could be tested with a new testing library – bUnit (see GitHub page for more). It enables the writing of tests that verify the contents of a component and conditional rendering, all without having to launch and/or host a server to run the tests against (unlike Selenium tests that require the application to be deployed).

The MSTest project that includes this library contains 27 unit tests that cover component access, filters, button operations, etc., for the following components:

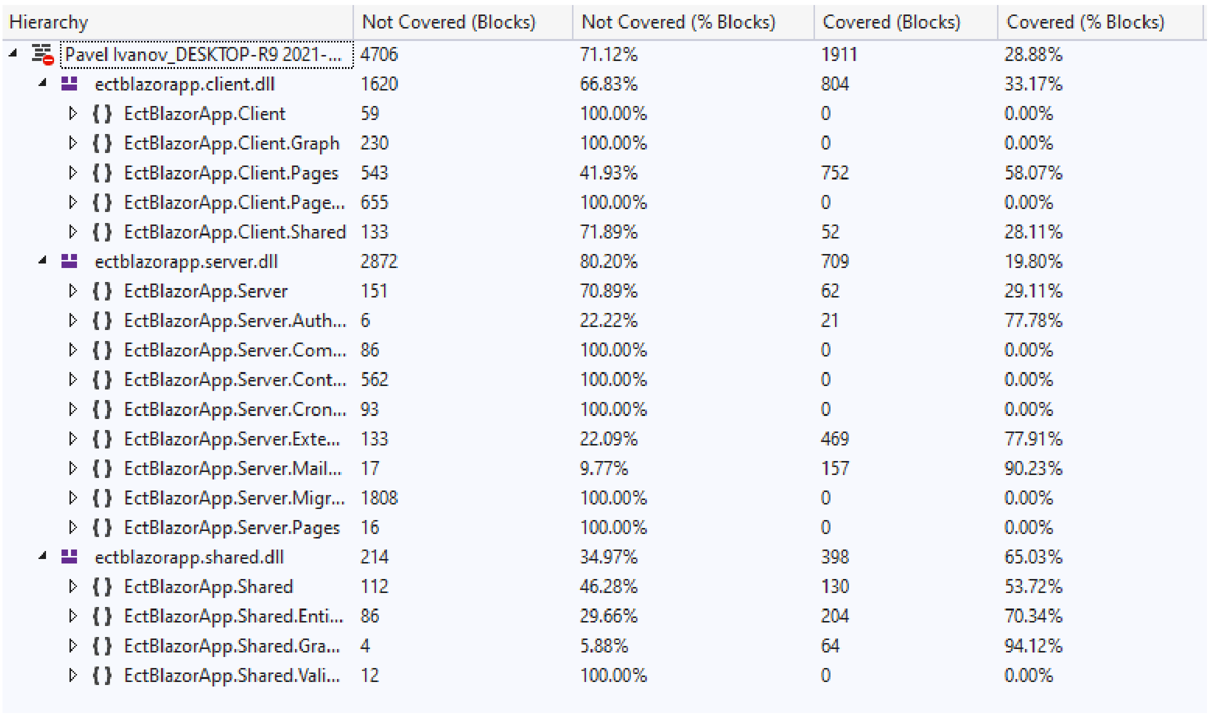
- Create Team page.
- Manage Team page.
- Move Members page.

The results from the code coverage analysis (Figure 25) show that the overall coverage is not near perfect, but the crucial code blocks where obscure bugs may occur are well covered (e.g., methods in the Server.Extensions namespace or the Server.MailKit namespace).



```
◢ ✓ EctBlazorApp.ClientTests (27)        1.3 sec
  ◢ ✓ EctBlazorApp.ClientTests (27)      1.3 sec
    ▷ ✓ CreateTeamTests (8)              1.2 sec
    ▷ ✓ ManageTeamTests (9)               53 ms
    ▷ ✓ MoveMembersTests (10)             42 ms
```

Figure 24. Local run of tests in ClientTests project.



Hierarchy	Not Covered (Blocks)	Not Covered (% Blocks)	Covered (Blocks)	Covered (% Blocks)
Pavel Ivanov_DESKTOP-R9 2021-...	4706	71.12%	1911	28.88%
ectblazorapp.client.dll	1620	66.83%	804	33.17%
EctBlazorApp.Client	59	100.00%	0	0.00%
EctBlazorApp.Client.Graph	230	100.00%	0	0.00%
EctBlazorApp.Client.Pages	543	41.93%	752	58.07%
EctBlazorApp.Client.Page...	655	100.00%	0	0.00%
EctBlazorApp.Client.Shared	133	71.89%	52	28.11%
ectblazorapp.server.dll	2872	80.20%	709	19.80%
EctBlazorApp.Server	151	70.89%	62	29.11%
EctBlazorApp.Server.Auth...	6	22.22%	21	77.78%
EctBlazorApp.Server.Com...	86	100.00%	0	0.00%
EctBlazorApp.Server.Cont...	562	100.00%	0	0.00%
EctBlazorApp.Server.Cron...	93	100.00%	0	0.00%
EctBlazorApp.Server.Exte...	133	22.09%	469	77.91%
EctBlazorApp.Server.Mail...	17	9.77%	157	90.23%
EctBlazorApp.Server.Migr...	1808	100.00%	0	0.00%
EctBlazorApp.Server.Pages	16	100.00%	0	0.00%
ectblazorapp.shared.dll	214	34.97%	398	65.03%
EctBlazorApp.Shared	112	46.28%	130	53.72%
EctBlazorApp.Shared.Enti...	86	29.66%	204	70.34%
EctBlazorApp.Shared.Gra...	4	5.88%	64	94.12%
EctBlazorApp.Shared.Vali...	12	100.00%	0	0.00%

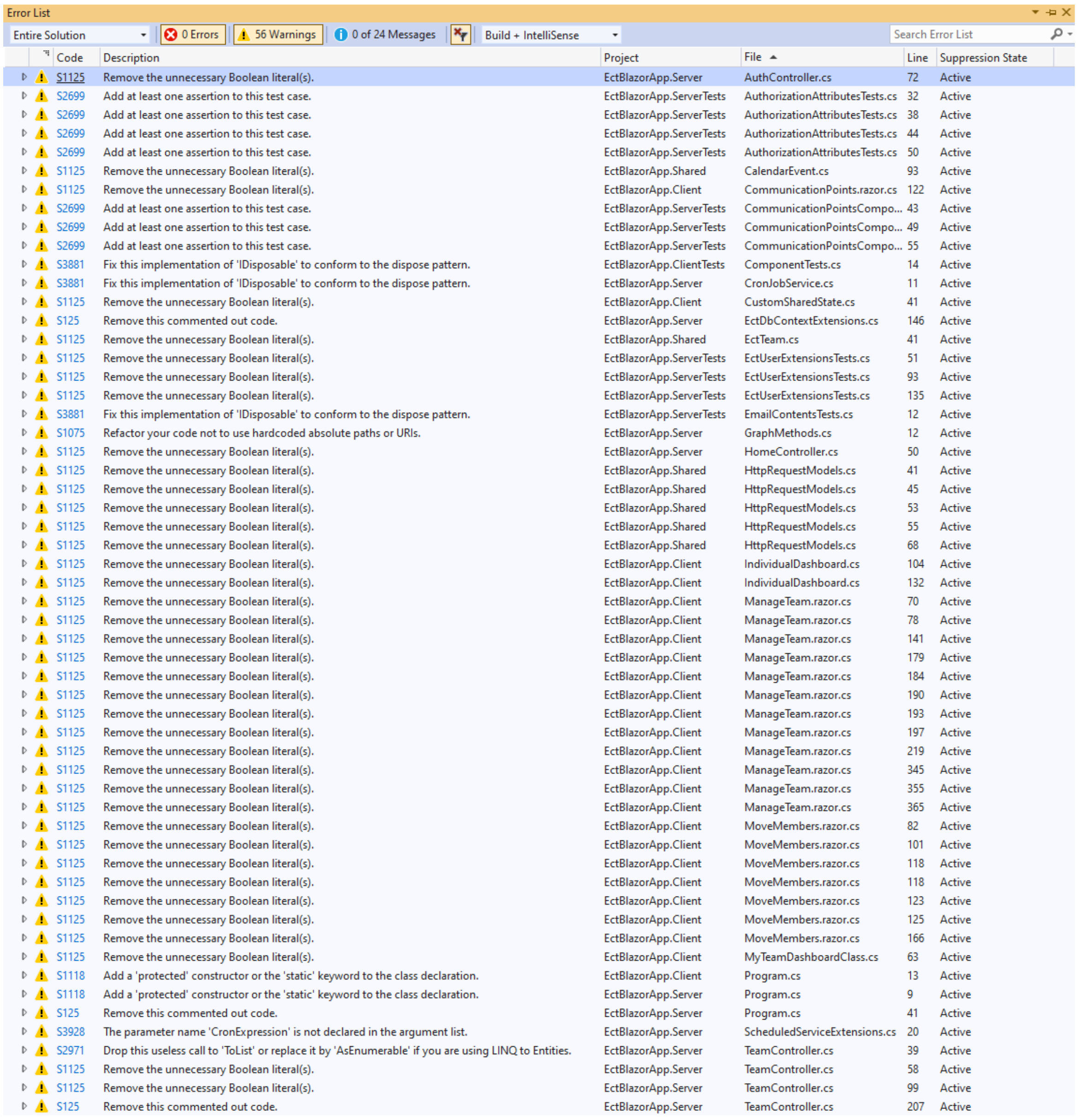
Figure 25. Results of code coverage analysis.

Lint tool warnings – SonarLint

The warnings reported by the SonarLint extension (which essentially returns whatever the SonarQube server reports), are mostly for ‘unnecessary’ Boolean literals in conditional expressions. These are purposely put in the expressions in order to be clear with the intentions behind the code, e.g., when comparing ‘*methodReturningBoolean() == false*’ and ‘*!methodReturningBoolean()*’, *== false* gets the point across way easier than the exclamation mark in front of the statement.

There are warnings about test cases without an assert statement, which is technically true, as these methods are using a template method where the assert statements are executed. These tests do run and produce the expected results.

Some code is intentionally left commented out as a preview of what a method would look like if a potential feature were to be implemented. Other cases involve multiple solutions to the same problem and a quick explanation as to why one would work better over the other.


Error List

Entire Solution | 0 Errors | 56 Warnings | 0 of 24 Messages | Build + IntelliSense | Search Error List

Code	Description	Project	File	Line	Suppression State
S1125	Remove the unnecessary Boolean literal(s).	EctBlazorApp.Server	AuthController.cs	72	Active
S2699	Add at least one assertion to this test case.	EctBlazorApp.ServerTests	AuthorizationAttributesTests.cs	32	Active
S2699	Add at least one assertion to this test case.	EctBlazorApp.ServerTests	AuthorizationAttributesTests.cs	38	Active
S2699	Add at least one assertion to this test case.	EctBlazorApp.ServerTests	AuthorizationAttributesTests.cs	44	Active
S2699	Add at least one assertion to this test case.	EctBlazorApp.ServerTests	AuthorizationAttributesTests.cs	50	Active
S1125	Remove the unnecessary Boolean literal(s).	EctBlazorApp.Shared	CalendarEvent.cs	93	Active
S1125	Remove the unnecessary Boolean literal(s).	EctBlazorApp.Client	CommunicationPoints.razor.cs	122	Active
S2699	Add at least one assertion to this test case.	EctBlazorApp.ServerTests	CommunicationPointsCompo...	43	Active
S2699	Add at least one assertion to this test case.	EctBlazorApp.ServerTests	CommunicationPointsCompo...	49	Active
S2699	Add at least one assertion to this test case.	EctBlazorApp.ServerTests	CommunicationPointsCompo...	55	Active
S3881	Fix this implementation of 'IDisposable' to conform to the dispose pattern.	EctBlazorApp.ClientTests	ComponentTests.cs	14	Active
S3881	Fix this implementation of 'IDisposable' to conform to the dispose pattern.	EctBlazorApp.Server	CronJobService.cs	11	Active
S1125	Remove the unnecessary Boolean literal(s).	EctBlazorApp.Client	CustomSharedState.cs	41	Active
S125	Remove this commented out code.	EctBlazorApp.Server	EctDbContextExtensions.cs	146	Active
S1125	Remove the unnecessary Boolean literal(s).	EctBlazorApp.Shared	EctTeam.cs	41	Active
S1125	Remove the unnecessary Boolean literal(s).	EctBlazorApp.ServerTests	EctUserExtensionsTests.cs	51	Active
S1125	Remove the unnecessary Boolean literal(s).	EctBlazorApp.ServerTests	EctUserExtensionsTests.cs	93	Active
S1125	Remove the unnecessary Boolean literal(s).	EctBlazorApp.ServerTests	EctUserExtensionsTests.cs	135	Active
S3881	Fix this implementation of 'IDisposable' to conform to the dispose pattern.	EctBlazorApp.ServerTests	EmailContentsTests.cs	12	Active
S1075	Refactor your code not to use hardcoded absolute paths or URIs.	EctBlazorApp.Server	GraphMethods.cs	12	Active
S1125	Remove the unnecessary Boolean literal(s).	EctBlazorApp.Server	HomeController.cs	50	Active
S1125	Remove the unnecessary Boolean literal(s).	EctBlazorApp.Shared	HttpRequestModels.cs	41	Active
S1125	Remove the unnecessary Boolean literal(s).	EctBlazorApp.Shared	HttpRequestModels.cs	45	Active
S1125	Remove the unnecessary Boolean literal(s).	EctBlazorApp.Shared	HttpRequestModels.cs	53	Active
S1125	Remove the unnecessary Boolean literal(s).	EctBlazorApp.Shared	HttpRequestModels.cs	55	Active
S1125	Remove the unnecessary Boolean literal(s).	EctBlazorApp.Shared	HttpRequestModels.cs	68	Active
S1125	Remove the unnecessary Boolean literal(s).	EctBlazorApp.Client	IndividualDashboard.cs	104	Active
S1125	Remove the unnecessary Boolean literal(s).	EctBlazorApp.Client	IndividualDashboard.cs	132	Active
S1125	Remove the unnecessary Boolean literal(s).	EctBlazorApp.Client	ManageTeam.razor.cs	70	Active
S1125	Remove the unnecessary Boolean literal(s).	EctBlazorApp.Client	ManageTeam.razor.cs	78	Active
S1125	Remove the unnecessary Boolean literal(s).	EctBlazorApp.Client	ManageTeam.razor.cs	141	Active
S1125	Remove the unnecessary Boolean literal(s).	EctBlazorApp.Client	ManageTeam.razor.cs	179	Active
S1125	Remove the unnecessary Boolean literal(s).	EctBlazorApp.Client	ManageTeam.razor.cs	184	Active
S1125	Remove the unnecessary Boolean literal(s).	EctBlazorApp.Client	ManageTeam.razor.cs	190	Active
S1125	Remove the unnecessary Boolean literal(s).	EctBlazorApp.Client	ManageTeam.razor.cs	193	Active
S1125	Remove the unnecessary Boolean literal(s).	EctBlazorApp.Client	ManageTeam.razor.cs	197	Active
S1125	Remove the unnecessary Boolean literal(s).	EctBlazorApp.Client	ManageTeam.razor.cs	219	Active
S1125	Remove the unnecessary Boolean literal(s).	EctBlazorApp.Client	ManageTeam.razor.cs	345	Active
S1125	Remove the unnecessary Boolean literal(s).	EctBlazorApp.Client	ManageTeam.razor.cs	355	Active
S1125	Remove the unnecessary Boolean literal(s).	EctBlazorApp.Client	ManageTeam.razor.cs	365	Active
S1125	Remove the unnecessary Boolean literal(s).	EctBlazorApp.Client	MoveMembers.razor.cs	82	Active
S1125	Remove the unnecessary Boolean literal(s).	EctBlazorApp.Client	MoveMembers.razor.cs	101	Active
S1125	Remove the unnecessary Boolean literal(s).	EctBlazorApp.Client	MoveMembers.razor.cs	118	Active
S1125	Remove the unnecessary Boolean literal(s).	EctBlazorApp.Client	MoveMembers.razor.cs	118	Active
S1125	Remove the unnecessary Boolean literal(s).	EctBlazorApp.Client	MoveMembers.razor.cs	123	Active
S1125	Remove the unnecessary Boolean literal(s).	EctBlazorApp.Client	MoveMembers.razor.cs	125	Active
S1125	Remove the unnecessary Boolean literal(s).	EctBlazorApp.Client	MoveMembers.razor.cs	166	Active
S1125	Remove the unnecessary Boolean literal(s).	EctBlazorApp.Client	MyTeamDashboardClass.cs	63	Active
S1118	Add a 'protected' constructor or the 'static' keyword to the class declaration.	EctBlazorApp.Client	Program.cs	13	Active
S1118	Add a 'protected' constructor or the 'static' keyword to the class declaration.	EctBlazorApp.Server	Program.cs	9	Active
S125	Remove this commented out code.	EctBlazorApp.Server	Program.cs	41	Active
S3928	The parameter name 'CronExpression' is not declared in the argument list.	EctBlazorApp.Server	ScheduledServiceExtensions.cs	20	Active
S2971	Drop this useless call to 'ToList' or replace it by 'AsEnumerable' if you are using LINQ to Entities.	EctBlazorApp.Server	TeamController.cs	39	Active
S1125	Remove the unnecessary Boolean literal(s).	EctBlazorApp.Server	TeamController.cs	58	Active
S1125	Remove the unnecessary Boolean literal(s).	EctBlazorApp.Server	TeamController.cs	99	Active
S125	Remove this commented out code.	EctBlazorApp.Server	TeamController.cs	207	Active

Figure 26. SonarLint warnings after local code analysis.

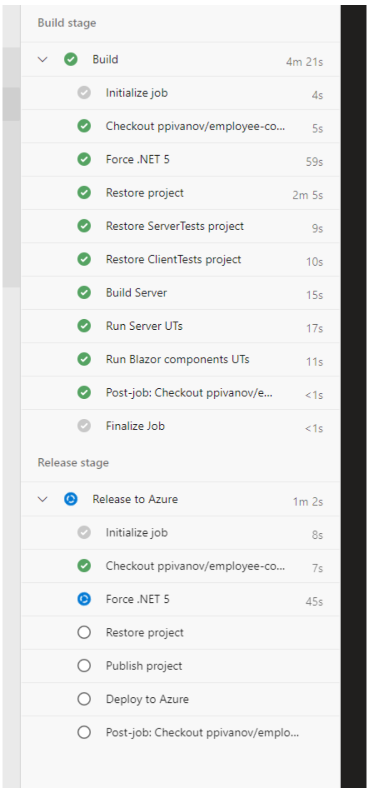
Continuous integration and deployment

The application includes a build-test-release pipeline, that is triggered whenever a new change has been pushed to the code repository in GitHub (Figure 27). The Pipeline resides in an Azure DevOps project and it is using Azure's integration with GitHub to kick it off.

The steps the pipeline executes include:

- Retrieving the newest code from the GitHub repository.
- Installing .NET 5 on the virtual agents if necessary.
- Restoring all dependencies.
- Building the application.
- Running the unit tests.
- Deploying the application to an Azure App Service.

If a step fails to execute successfully the build will be stopped and the pipeline run will fail.



```
Build stage
  Build                                  4m 21s
    Initialize job                           4s
    Checkout ppivanov/employee-co...         5s
    Force .NET 5                            59s
    Restore project                      2m 5s
    Restore ServerTests project              9s
    Restore ClientTests project             10s
    Build Server                            15s
    Run Server UTs                          17s
    Run Blazor components UTs               11s
    Post-job: Checkout ppivanov/e...        <1s
    Finalize Job                            <1s
Release stage
  Release to Azure                       1m 2s
    Initialize job                           8s
    Checkout ppivanov/employee-co...         7s
    Force .NET 5                            45s
    Restore project
    Publish project
    Deploy to Azure
    Post-job: Checkout ppivanov/emplo...
```

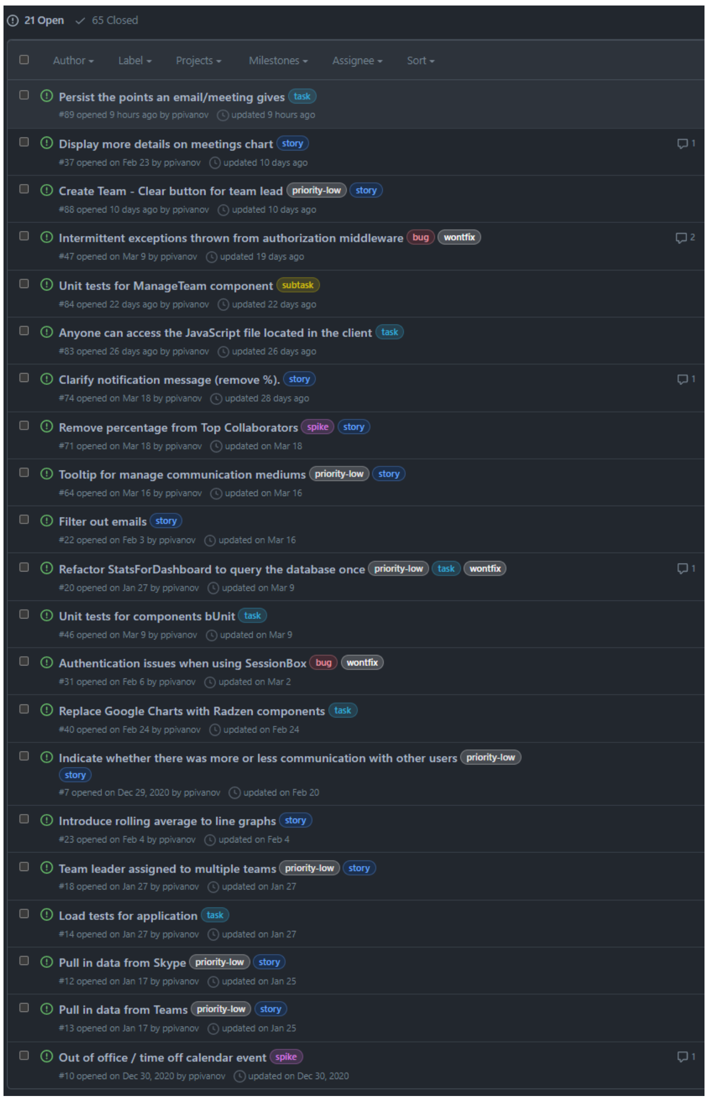
Figure 27. CI/CD pipeline steps in execution.

Issue tracking

GitHub’s integrated issue tracking system was used to keep track of all the outstanding tasks (Figures 28 and 29). Issues were labelled in a concise manner so that the tickets represent either a feature (user story), non-functional task, a bug, or a spike. Labels were also used for prioritization of the tasks, so that issues can be filtered by priority, e.g., show all tickets with a ‘priority-high’ label.

Meetings with Bank of America

Meetings with the mentor from Bank of America aided feature development as the mentors provided their feedback for almost everything: features developed since last meeting, bug fixes, user experience, non-functional work, etc.
These discussions provided every participant with the necessary insight to understand how the application functions.


21 Open ✓ 65 Closed

Author ▾ Label ▾ Projects ▾ Milestones ▾ Assignee ▾ Sort ▾

Persist the points an email/meeting gives task
#89 opened 9 hours ago by ppivanov · updated 9 hours ago

Display more details on meetings chart story — 1
#37 opened on Feb 23 by ppivanov · updated 10 days ago

Create Team - Clear button for team lead priority-low story
#88 opened 10 days ago by ppivanov · updated 10 days ago

Intermittent exceptions thrown from authorization middleware bug wontfix — 2
#47 opened on Mar 9 by ppivanov · updated 19 days ago

Unit tests for ManageTeam component subtask
#84 opened 22 days ago by ppivanov · updated 22 days ago

Anyone can access the JavaScript file located in the client task
#83 opened 26 days ago by ppivanov · updated 26 days ago

Clarify notification message (remove %). story — 1
#74 opened on Mar 18 by ppivanov · updated 28 days ago

Remove percentage from Top Collaborators spike story
#71 opened on Mar 18 by ppivanov · updated on Mar 18

Tooltip for manage communication mediums priority-low story
#64 opened on Mar 16 by ppivanov · updated on Mar 16

Filter out emails story
#22 opened on Feb 3 by ppivanov · updated on Mar 16

Refactor StatsForDashboard to query the database once priority-low task wontfix — 1
#20 opened on Jan 27 by ppivanov · updated on Mar 9

Unit tests for components bUnit task
#46 opened on Mar 9 by ppivanov · updated on Mar 9

Authentication issues when using SessionBox bug wontfix
#31 opened on Feb 6 by ppivanov · updated on Mar 2

Replace Google Charts with Radzen components task
#40 opened on Feb 24 by ppivanov · updated on Feb 24

Indicate whether there was more or less communication with other users priority-low story
#7 opened on Dec 29, 2020 by ppivanov · updated on Feb 20

Introduce rolling average to line graphs story
#23 opened on Feb 4 by ppivanov · updated on Feb 4

Team leader assigned to multiple teams priority-low story
#18 opened on Jan 27 by ppivanov · updated on Jan 27

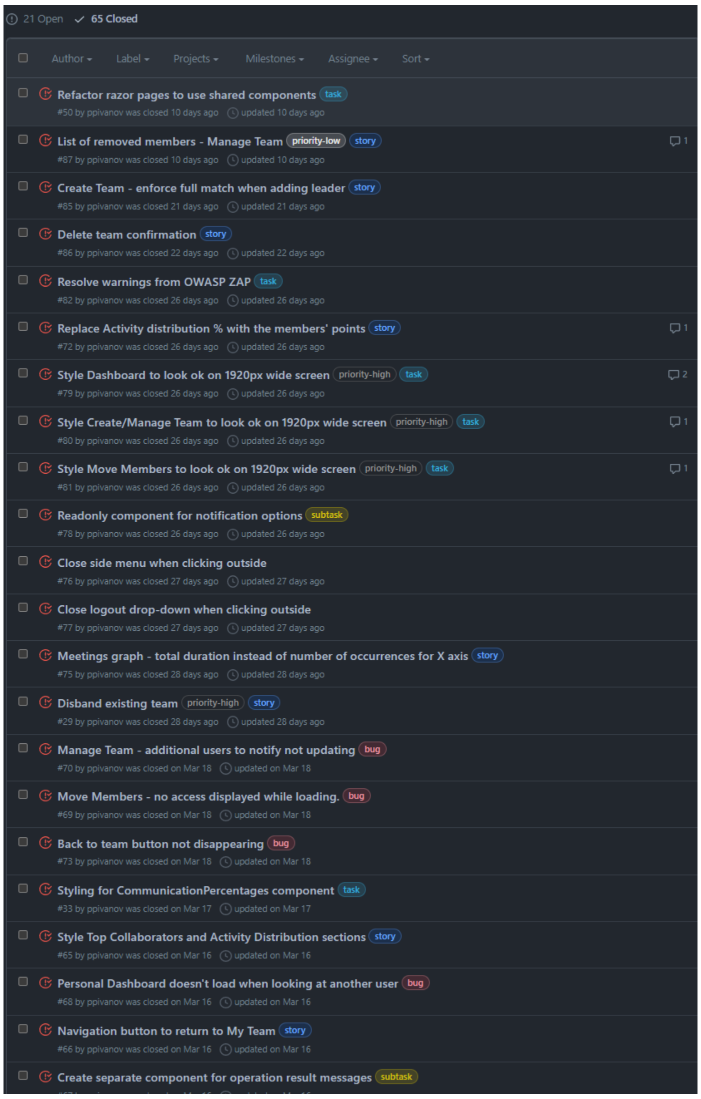
Load tests for application task
#14 opened on Jan 27 by ppivanov · updated on Jan 27

Pull in data from Skype priority-low story
#12 opened on Jan 17 by ppivanov · updated on Jan 25

Pull in data from Teams priority-low story
#13 opened on Jan 17 by ppivanov · updated on Jan 25

Out of office / time off calendar event spike — 1
#10 opened on Dec 30, 2020 by ppivanov · updated on Dec 30, 2020

Figure 28. Open issues in GitHub.


21 Open ✓ 65 Closed

Author ▾ Label ▾ Projects ▾ Milestones ▾ Assignee ▾ Sort ▾

Refactor razor pages to use shared components task
#50 by ppivanov was closed 10 days ago · updated 10 days ago

List of removed members - Manage Team priority-low story — 1
#87 by ppivanov was closed 10 days ago · updated 10 days ago

Create Team - enforce full match when adding leader story
#85 by ppivanov was closed 21 days ago · updated 21 days ago

Delete team confirmation story
#86 by ppivanov was closed 22 days ago · updated 22 days ago

Resolve warnings from OWASP ZAP task
#82 by ppivanov was closed 26 days ago · updated 26 days ago

Replace Activity distribution % with the members' points story — 1
#72 by ppivanov was closed 26 days ago · updated 26 days ago

Style Dashboard to look ok on 1920px wide screen priority-high task — 2
#79 by ppivanov was closed 26 days ago · updated 26 days ago

Style Create/Manage Team to look ok on 1920px wide screen priority-high task — 1
#80 by ppivanov was closed 26 days ago · updated 26 days ago

Style Move Members to look ok on 1920px wide screen priority-high task — 1
#81 by ppivanov was closed 26 days ago · updated 26 days ago

Readonly component for notification options subtask
#78 by ppivanov was closed 26 days ago · updated 26 days ago

Close side menu when clicking outside
#76 by ppivanov was closed 27 days ago · updated 27 days ago

Close logout drop-down when clicking outside
#77 by ppivanov was closed 27 days ago · updated 27 days ago

Meetings graph - total duration instead of number of occurrences for X axis story
#75 by ppivanov was closed 28 days ago · updated 28 days ago

Disband existing team priority-high story
#29 by ppivanov was closed 28 days ago · updated 28 days ago

Manage Team - additional users to notify not updating bug
#70 by ppivanov was closed on Mar 18 · updated on Mar 18

Move Members - no access displayed while loading. bug
#69 by ppivanov was closed on Mar 18 · updated on Mar 18

Back to team button not disappearing bug
#73 by ppivanov was closed on Mar 18 · updated on Mar 18

Styling for CommunicationPercentages component task
#33 by ppivanov was closed on Mar 17 · updated on Mar 17

Style Top Collaborators and Activity Distribution sections story
#65 by ppivanov was closed on Mar 16 · updated on Mar 16

Personal Dashboard doesn't load when looking at another user bug
#68 by ppivanov was closed on Mar 16 · updated on Mar 16

Navigation button to return to My Team story
#66 by ppivanov was closed on Mar 16 · updated on Mar 16

Create separate component for operation result messages subtask

Figure 29. Partial list of closed issues on GitHub.