Use Cases and Logical Architecture

- XID: X00149863
- Name: Pavel Ivanov
- Project Title: Employee Connectivity Tracker

Use Cases

Title (goal)	Sign in with Microsoft account
Primary Actor	Administrator, Employee, Team Leader
Story	The user: 1. Opens the Web Application. 2. Clicks on the button “Sign in with Microsoft”. => Microsoft login form appears. 3. Enters credentials and submits. => Redirect to the Home page of the Web Application. => Web app requests any missing data from Microsoft Graph API, processes it and persists to the database. --- …. 3. Enters incorrect credentials and submits. => Error message is displayed. => User is not redirected to the Web Application and logged in.

Title (goal)	Assign and re-assign weight to communication application
Primary Actor	Administrator
Story	1. Navigate to “System settings”. => All available applications and current weights are displayed. 2. Drag the sliding selector beneath one of the applications. => Old value / new value comparison is displayed on the right-hand side of the screen. 3. Click “Save”. => Message stating that the action was successful is displayed and new values are persisted. => Every notification after this edit takes the updated values into account.

Title (goal)	Create a team
Primary Actor	Administrator
Story	1. On the home page, click on “Manage teams”. => All active teams are presented on the screen. 2. On the sub-menu, click “Create team”. => A form is displayed. 3. Enter team name. 4. Enter team leader’s email address. * 5. Enter one email address per member. * 6. Click on “Create”. => Message stating that the action was successful is displayed and new values are persisted. => All team members are persisted to the database and are tracked from that point on. _____ … 3. Leave field ‘Team name’ empty. … 6. Click on “Create”. => Error message stating that the action was unsuccessful is displayed and no values are persisted to the database. _____ … 4. Leave field ‘Team lead email’ empty. … 6. Click on “Create”. => Error message stating that the action was unsuccessful is displayed and no values are persisted to the database. _____ … 5. Leave field ‘Member emails’ empty. 6. Click on “Create”. => Error message stating that the action was unsuccessful is displayed and no values are persisted to the database. * Only email addresses associated to a Microsoft account will be tracked.

Title (goal)	Add a new team member
Primary Actor	Administrator
Story	1. On the home page, click on “Manage teams”. => All active teams are presented on the screen. 2. Click on one of the teams. => Team members are displayed. 3. Click on “Add member”. => An input field appears. => “Done” button appears but greyed out and unclickable. 5. Enter a single email address. => “Done” button turns green and is now clickable. 6. Click on “Done”. => Message stating that the action was successful is displayed and new values are persisted to the database. => New member is tracked from that point on.

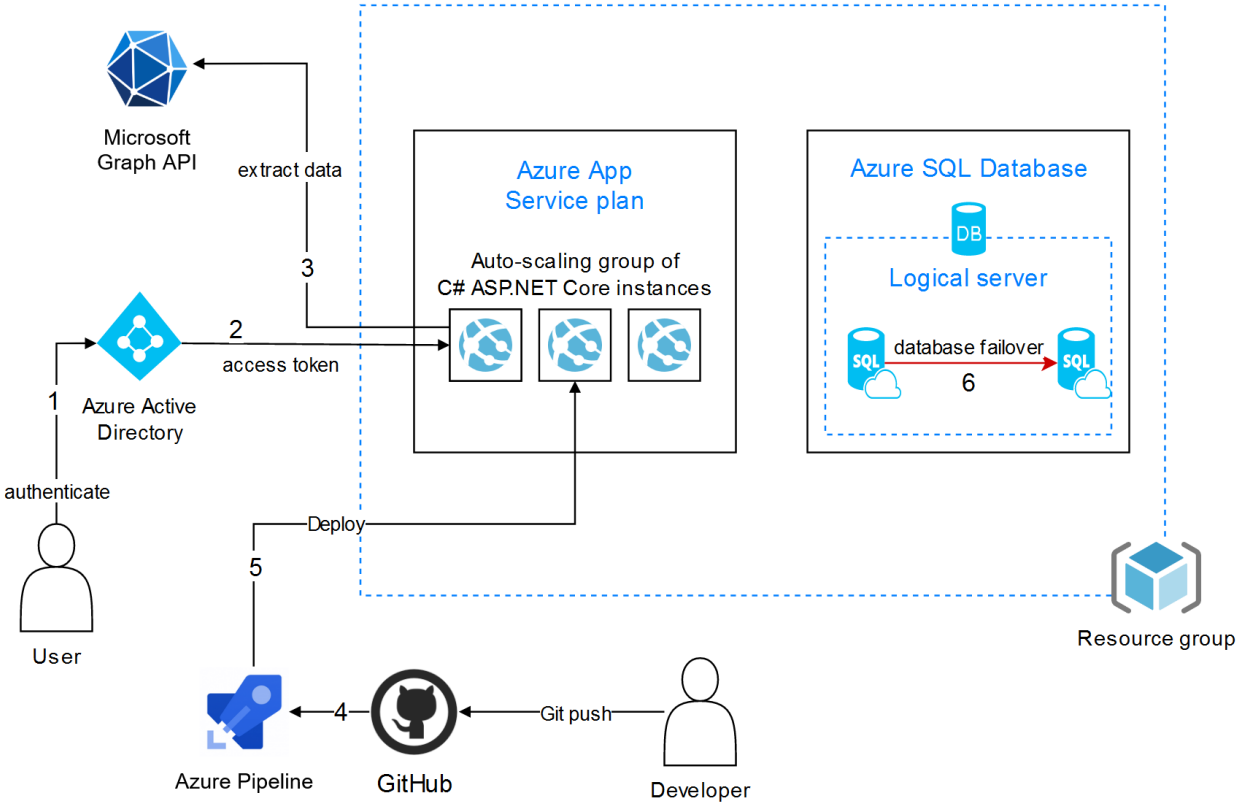
Title (goal)	Remove a team member
Primary Actor	Administrator
Story	1. On the home page, click on “Manage teams”. => All active teams are presented on the screen. 2. Click on one of the teams. => Team members are displayed. 3. Click on “Remove” beside the member. => A popup is displayed asking for confirmation. 4. Click “Yes”. => Message stating that the action was successful is displayed and new values are persisted to the database. => User is no longer a member of the team – team leader and administrator will no longer see them on that team. _____ … 4. Click “No”. => Popup closes. => User is still a member of the team.

Title (goal)	View own data
Primary Actor	Employee, Team Leader
Story	1. The home page shows a dashboard of the cumulative data generated by the user on that day such as: • Total emails sent. • Total emails received. • People you collaborate with. • Number of meetings on calendar. 2. Click on “Filter by” on the side menu. => All available options appear: • Time period: ○ Today (selected by default) ○ This week ○ This month ○ This year ○ By date • Application: ○ Outlook ○ Calendar ○ Teams 3. Click on the “By date” option. => The two date fields underneath it – start date and end date, become enterable. 4. Select or enter start date in the past. => No changes to the dashboard. 5. Select or enter end date in the past. => No changes to the dashboard. 6. Click on the “Filter” button underneath the options. => Dashboard is updated and displays only data within the specified period. _____ … 4. Select or enter a start date in the future. => Field is marked in red. => Error message is displayed, stating the start date must the in the past.

Title (goal)	View teammates' data
Primary Actor	Team Leader
Story	1. The home page shows a dashboard of the cumulative data generated by the user on that day. 2. Click on "My teams' data". => A list of teams appears where the user is assigned as team leader. 3. Click on top-most team. => Dashboard is updated. => Team members' names are displayed at the top of the dashboard. 4. Click on "Filter by" on the side menu. => All available options appear: • Time period: ○ Today (selected by default) ○ This week ○ This month ○ This year ○ By date • Application: ○ Outlook ○ Calendar ○ Teams 5. Click on the "Outlook" option. 6. Click on the "Filter" button underneath the options. => Dashboard is updated and displays only data related to emails, such as: • Total number of emails sent by all team members. • Total number of emails sent to people outside of the team. • Total number people reached.

Title (goal)	View all communication data
Primary Actor	Administrator
Story	1. The home page shows a dashboard of the cumulative data generated by the user on that day. 2. Click on “All teams” on the side menu. => Dashboard is updated to show data for each team. 3. Click on “Filter by” on the side menu. => All available options appear: • Time period: ○ Today (selected by default) ○ This week ○ This month ○ This year ○ By date • Application: ○ Outlook ○ Calendar ○ Teams • Team: ○ Team 1’s name ○ Team 2’s name ○ etc… 5. Click on “Team 1’s name” option. 6. Click on the “Filter” button underneath the options. => Dashboard is updated and displays only data generated by members of Team 1.

Logical Architecture



Logical Architecture Discussion

A user is interacting with the core features of the application, only when they are authenticated by Microsoft.

(1) Once the user submits the Microsoft login form and their credentials are correct, they will be redirected back to the ASP.NET Core Web Application, written in C#.
The application is hosted on Azure App Service, that offers auto-scaling for high availability and enables automated deployments from GitHub.

(2, 3) The application itself is not generating any data, instead it uses the user's access token, received from Azure Active Directory, to authenticate requests for information to Microsoft Graph API.

> "Microsoft Graph is a RESTful web API that enables you to access Microsoft Cloud service resources. After you register your app and get authentication tokens for a user or service, you can make requests to the Microsoft Graph API."
>
> Microsoft documentation,
> https://docs.microsoft.com/en-us/graph/use-the-api

(6) The Web Application processes the data retrieved from Microsoft Graph and persists it to an Azure SQL Database which replicates all data to a secondary database in case the main database fails. The persisted data is then read to determine dashboard content, user types and filtering options.

(4, 5) The Continuous Integration and Continuous Deployment process starts with a commit to the main repository hosted on GitHub. That push triggers the Azure Pipeline service and that kicks off all the specified tasks:

- Downloading latest code from the GitHub repository.
- Restoring dependencies.
- Compile the project.
- Run the unit tests.

That process produces an artifact which can then be released to Azure App Services.