



Digital Adoption by AppNavi

PRODUCT DESCRIPTION

As of: 08th July 2022



CONTENT

1	About this product description.....	4
2	What is AppNavi?.....	4
2.1	Advantages through AppNavi	5
2.2	Supported systems & surface technologies	5
2.3	Supported customer situations	5
2.4	Integration strategies	6
2.5	Compatibility.....	6
2.6	Service areas.....	6
2.7	Product components.....	7
2.8	User types.....	7
3	Architecture / Components.....	8
3.1	Tenant.....	8
3.1.1	Enterprise Authentication.....	8
3.1.2	Expert mode.....	8
3.1.3	Media library.....	8
3.1.4	User management.....	9
3.2	Workspaces.....	9
3.3	Apps.....	9
3.3.1	Settings.....	9
3.3.2	Customizing options.....	9
3.4	Content.....	10
3.4.1	Versioning.....	10
3.4.2	Assignment to an app.....	10
3.4.3	Copying contents.....	10
3.4.4	Publishing in an app.....	10
3.5	Roles & permissions	10
4	Routes	11
4.1	Display	11
4.2	Structure & Components	11
4.3	Creating routes.....	12
4.4	Element detection	12
4.5	Triggers & Modes	13
4.6	Variance & Learning	13
4.7	Automation & RPA	14
4.8	Versioning & Publication.....	14
4.9	Visibility and segmentation	14
4.10	Other features.....	15
5	News & Announcements.....	16
5.1	Display	16
5.2	Structure & Components	16
5.3	Triggers & Modes	17



6	Hotspots	18
6.1	Display	18
6.2	Triggers & Modes	18
7	Trainings & Collections	19
8	Analytics	20
8.1	Data acquisition.....	20
8.2	Metrics.....	20
8.3	Diagrams	21
8.3.1	<i>General</i>	21
8.3.2	<i>KPIs for the workspace</i>	21
8.3.3	<i>User activity</i>	21
8.3.4	<i>Most used applications</i>	22
8.3.5	<i>Content usage</i>	23
8.3.6	<i>Time spent</i>	23
8.3.7	<i>Overview of applications</i>	24
8.3.8	<i>Used browser</i>	24
8.3.9	<i>Used language</i>	25
8.3.10	<i>Route problems</i>	25
8.3.11	<i>KPIs for a route</i>	25
8.3.12	<i>Route progress</i>	26
8.3.13	<i>KPIs for a news item or an announcement</i>	26
8.3.14	<i>News</i>	26
9	User Behavior Mining	28
9.1	Managing User Behavior Mining Campaigns.....	28
9.2	Interaction objects	28
9.3	Sessions	29
9.4	Behavior Flows.....	29
9.5	Automated creation of routes.....	29
9.6	Data security.....	29
10	Possibilities for customizing.....	30
10.1	Standard Text Styling	30
10.2	Custom Theming.....	30
10.3	Custom Coding	31
11	Data protection.....	33
12	IT-Security	33
13	Software ergonomics.....	33
14	Boundaries and limitations.....	34
15	Other service areas.....	35



1 About this product description

Thank you for your interest in AppNavi. With this product description we would like to give you an overview of the functionalities of AppNavi.

As we continuously develop AppNavi according to the needs of our customers, there may be temporary differences between the description and the product - we reserve the right to make changes in the description as well as the product accordingly.

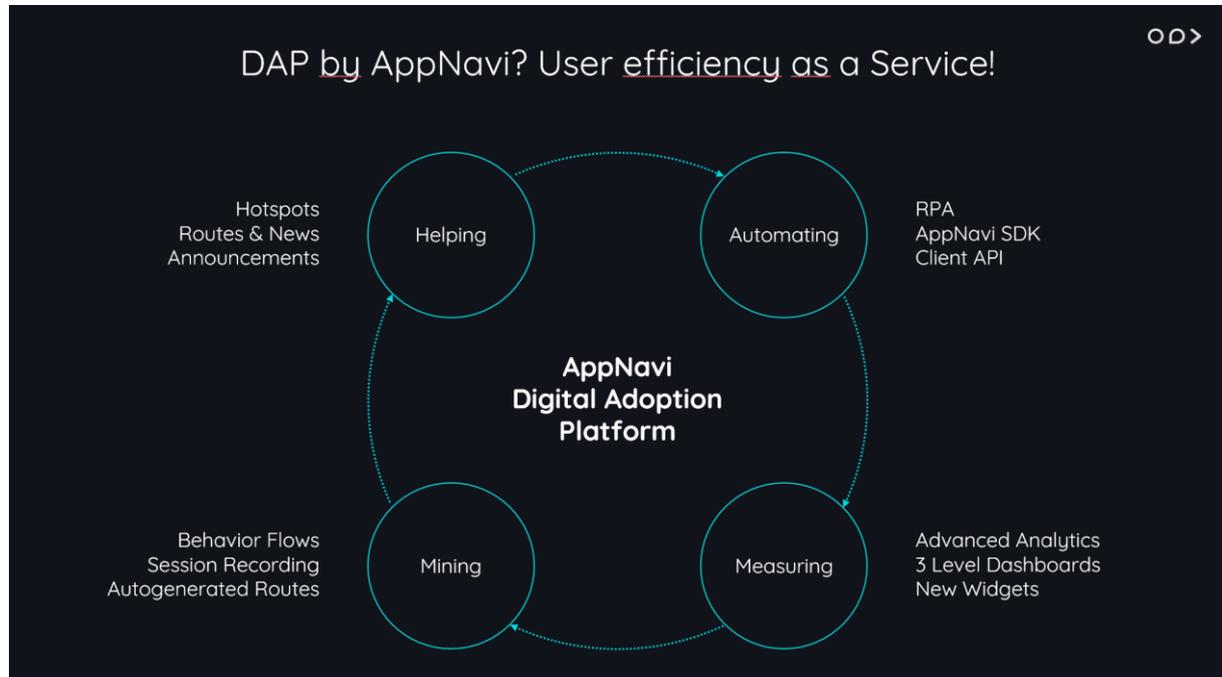
2 What is AppNavi?

AppNavi is a digital adoption platform. The special focus of AppNavi is the support of large corporate solutions. As the "easiest to use Digital Adoption Platform", AppNavi is very easy to use in the field through single IT projects or large transformation programs.

By "easy" we mean every aspect of working with our DAP. Here, the focus is not only on the end users, but also on the authors and all other roles within a professional company organization (such as IT and works council).

With AppNavi, either individual applications or entire software landscapes can be supported. The flexible architecture of AppNavi allows to map any desired customer organization and to support end users in the best possible way.

The platform consists of 4 service areas and supporting features:





2.1 Advantages through AppNavi

Especially in complex or rarely used applications, users often feel lost. As a result, a lot of time is wasted through unnecessary training, futile attempts or expensive inquiries to colleagues or the service desk.

At this point, AppNavi supports by empowering, supporting and accelerating users. Employees are trained "on the job". This reduces training costs, as training occurs at the time of the problem, rather than targeting fleeting stock knowledge. Necessary tool knowledge & process knowledge is made available "just in time". AppNavi is used in the context of digitalization initiatives and achieves quantitative and qualitative results.

Quantitative results

- > Reduced training costs
- > Increased end user efficiency
- > Lower support costs
- > Higher first-time-right
- > Higher conversion

Qualitative results

- > Support of strategic measures such as cost reduction programs, M&A processes, expansion strategies, etc.
- > Improved user centricity
- > Higher acceptance

2.2 Supported systems & surface technologies

AppNavi can be integrated into any browser-based software and supports both HTML and SVG elements.

Depending on the integration strategy, both in-house developed applications and SaaS solutions come into question here. Common use cases are:

- > Employee Self-Service Portals
- > Collaboration Systems
- > BPM, ERP & CRM Solutions
- > Reporting Apps & Dashboards
- > Customer Apps

Some examples: SAP SuccessFactors, SAP Ariba, Salesforce, Office 365 | M365, SharePoint, SharePoint Online, Outlook Online, Monday.com, Jira, Confluence, ServiceNow, Trello, Bitrix24, COYO, Slack, Workday, MicroStrategy, and more.

2.3 Supported customer situations

AppNavi distinguishes three typical customer situations:

- > End customers
- > SaaS provider
- > Consultations

AppNavi supports users in web-based software. The more complex this is, the better the strengths of AppNavi can be played out. The classic use case for AppNavi is therefore a company that wants to support its internally or externally used application landscape. Employees, applicants, customers as well as interested parties of this customer company are guided through the systems with the help of AppNavi.

In addition to the companies themselves, providers of SaaS solutions can also use AppNavi. Here, the support is created centrally by the provider and delivered to all of the provider's customers.

In addition to SaaS providers, training providers or consultancies can also create content for selected applications and play it out to their customers.



In the latter two customer situations, in addition to the centrally provided content, individual content can also be created and delivered for each customer.

2.4 Integration strategies

AppNavi is integrated into an existing browser-based software on the client side. This is done centrally by the customer's IT department. End users do not have to do anything - AppNavi becomes visible to them without any action on their part. With regard to the integration of AppNavi, two strategies can be distinguished:

- > Direct integration
- > Integration via browser extension

Both strategies lead to a client-side integration of AppNavi into the target application. With direct integration, a JavaScript reference is embedded in the target application. If you want to use AppNavi simultaneously in different applications within an organization or if you do not have access to the target application, integration via browser extension is available. This enables the browser user to use AppNavi within an application. Our browser extensions are rolled out via group policy and can also be used by end users immediately and without any intervention.

2.5 Compatibility

AppNavi works as a navigation system for all browser-based applications. It does not matter on which device type (laptop, tablet or cell phone) or operating system (Windows, Mac OS, Linux, etc.) your users want to use AppNavi. For mobile devices, adjustments usually have to be made, which are taken into account in the initial integration project.

The object detection of AppNavi works on HTML and SVG elements as well as in all underlying frameworks (Vanilla, React, Vue, Angular, etc.).

A (non-exhaustive) list of supported browsers can be found here: <https://docs.appnavi.eu/docs/browser-compatibility>

2.6 Service areas

AppNavi consists of a total of 4 service areas. Due to the architecture of AppNavi, these interlock in an intelligent way and thus support each other.

To enable end users to make the best possible use of a system, AppNavi provides features for help & information live in the workflow. Thus, interfaces and processes can be explained or important information can be brought live and situationally controlled into the user's workflow. These features are often used in on-boarding scenarios.

In order to provide users with the best possible support even after on-boarding, numerous features from the field of RPA (Robotic Process Automation) are available. Here, the focus of the application is not on training, but on accelerating users in their daily work.

In order to evaluate the effectiveness of the implemented measures, a sophisticated set of analysis tools is available. Here, the use of the content provided by AppNavi can be monitored - and measures can be derived from it.

In addition to the simple provision and evaluation of possible measures in the area of digital adoption, AppNavi is also intended to support its customers in planning. Here, user behavior mining provides exact insights into how users use the target software.

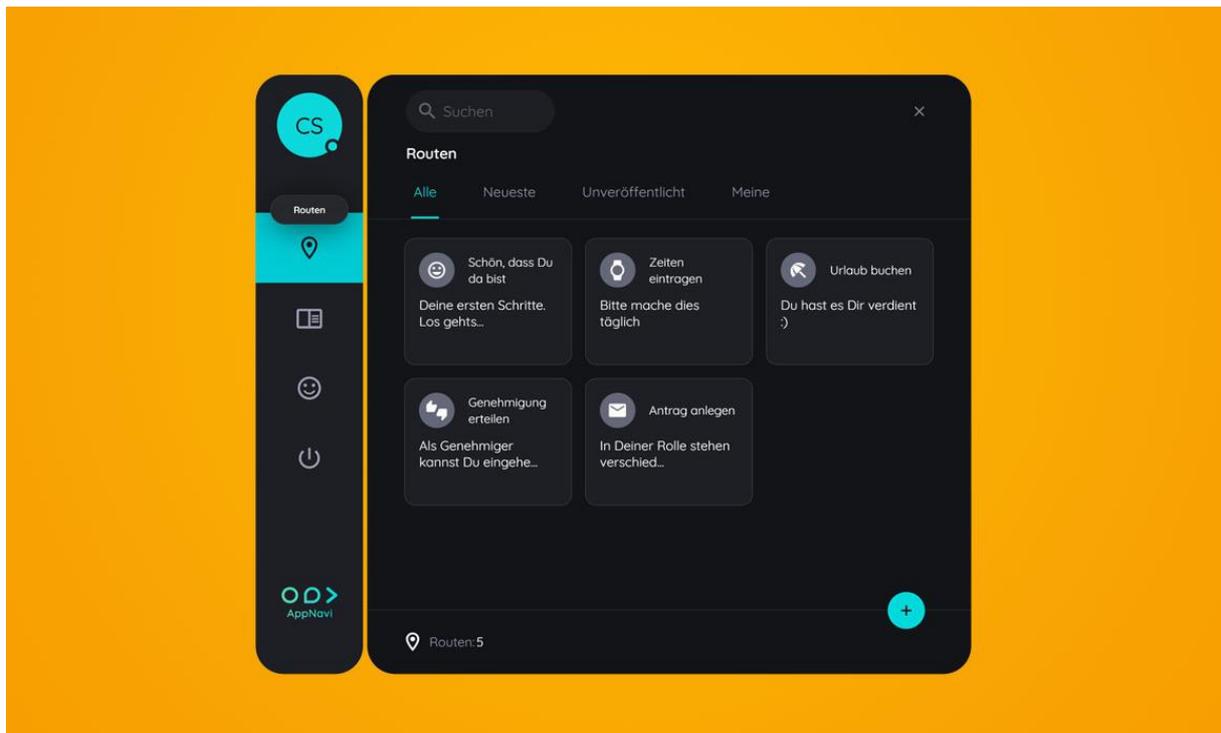
In addition to the 4 core service areas, AppNavi also provides a number of other features. For example, a content store allows users to easily obtain ready-made content via the AppNavi community.

2.7 Product components

AppNavi maps each customer in a logically separate tenant. All your apps, users, content, etc. can be managed in this tenant. The tenant can be accessed via the **customer portal**. This is a website that can be accessed via the browser.

With the help of a created app, the **AppNavi widget** becomes visible within the target system. Our widget is the visual presentation of AppNavi for your users within your systems.

Depending on the role, a user can open AppNavi's **planner & recorder** within the widget. This is available for editing and creating routes. With the help of a graphical user interface, authors can create routes very easily and without IT know-how.



2.8 User types

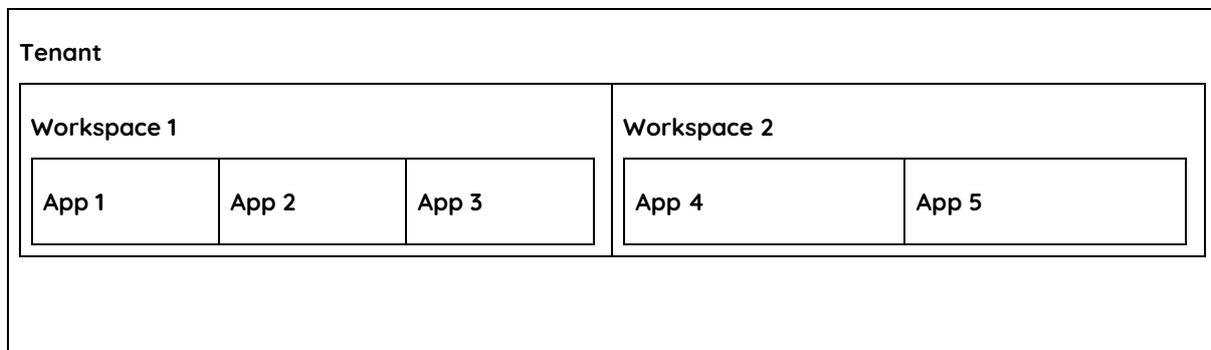
AppNavi distinguishes between end users and authors. End users are supported by AppNavi and can use the content provided by authors. Thus, when using AppNavi, there is a large group of end users and a small number of authors.

Authors have a login to AppNavi. This can be used both in the AppNavi widget (within your target system) and in the AppNavi customer portal. End users are completely anonymous and do not need an AppNavi login - for them AppNavi appears when they move around the target system.



3 Architecture / Components

AppNavi has a multi-client capable architecture. The topmost unit is the so-called ten-ant. Workspaces are available within a tenant, which in turn contain apps.



3.1 Tenant

AppNavi has a tenant structure. Each customer is mapped within a tenant. Within a tenant, so-called workspaces represent the organizational structure of a customer. Apps are the applications supported by AppNavi and contents are the content provided there, such as routes or news.

3.1.1 Enterprise Authentication

Enterprise Authentication allows content such as routes, messages or hotspots within your organization to be available only to authenticated users. Two different modes are available for authentication:

- > Email Verification
- > Azure Active Directory Authentication.

For email verification, users verify themselves with a verification code that they receive via email from the AppNavi system when they are authorized to do so. For AppNavi Enterprise Authentication in conjunction with Azure Active Directory, a new app registration in Azure Active Directory is required (web app type).

3.1.2 Expert mode

Within individual applications and individual steps of a route, customizations can be stored in the form of JavaScript and CSS. Against the background of internal IT security, this option can be activated or deactivated at a higher level in the tenant.

3.1.3 Media library

Each tenant has a media library. This can hold images in PNG, JPEG and SVG format. Images uploaded in the route planner are stored in this library and receive a publicly accessible link. The media library fulfills two objectives:

First, it allows images used in routes to be visible to any user. Second, images go through a virus scan when uploaded to the library. This ensures that the images are used in a trustworthy manner.



3.1.4 *User management*

All authors within a tenant are created as users. For this purpose, an e-mail address, a password, and a first and last name are stored for each user. Each user can then be assigned workspaces and a permission within these.

Workspace owners and contributors only see colleagues and content of their own workspace(s). This enables the use of AppNavi within different organizational sub-structures. Tenant owners can see and edit all users and all contents of the tenant.

3.2 Workspaces

Workspaces are used to build logical structures within the tenant. Depending on the customer situation (see chapter Supported customer situations) there are different examples for possible structures:

- > Parts of the company
- > Locations
- > Individual projects
- > End customers of the AppNavi ISV
- > Etc.

Workspaces have a name and can be assigned users. For each workspace there is a group of owners and a group of contributors. A single user can therefore have different permissions to different workspaces.

3.3 Apps

To display AppNavi in a new system, a new app is created within a workspace. This is first defined by the URL of the system. Both absolute and relative patterns can be specified here:

- > <https://your-URL.com/> (AppNavi is only displayed on the start page)
- > <https://your-URL.com/page1/> (AppNavi is only displayed on page 1)
- > https://your-URL.com/* (AppNavi is displayed on all pages of your system)

Apps can also be shared between workspaces. In such a scenario, an app is always the responsibility of one workspace, but is also visible and usable in other workspaces. In this way, apps can be provided with content jointly by different teams - at the same time, however, the responsibility is anchored in one workspace.

3.3.1 *Settings*

Each app has a number of settings. In addition to the URL pattern of an app, the visibility of the app can be defined first.

Here, the app can be activated and deactivated in principle - or the visibility for individual roles within the target system can be defined.

To ensure the best possible support for end users in the respective target app, the use of individual AppNavi features can be planned as part of a roadmap. For this purpose, individual menu items (e.g. routes or news) can be activated or deactivated in the AppNavi widget.

Analytics options are available to determine the usage of AppNavi content. These can also be turned on or off as needed.

3.3.2 *Customizing options*

Apps can be customized in terms of their functionality and appearance. For this purpose, AppNavi offers a custom code as well as a custom UI module.

Regarding the appearance, the 11 basic colors of AppNavi can be adjusted with simple color pickers. Further and more profound customizations can be done very easily with CSS.



For functional customizing the AppNavi Client API is available. This is a set of methods and events that can be controlled by JavaScript via AppNavi or the target application.

Both the CSS provided by the author and the JavaScript are integrated into the scope of the page and executed at runtime. This happens before the start of AppNavi, so that customizations already take effect before the widget becomes visible.

3.4 Content

With AppNavi, various contents (news, routes, etc.) are available to support the use of your systems. Contents are initially assigned to a workspace within the tenant. The assignment of a content to an app defines the editability of the content by the users of the tenant. Only the users assigned to a workspace can edit contents within this workspace.

3.4.1 Versioning

Saving a content creates a new version of the same. For each version, the date of the change and the name of the changing author are saved.

3.4.2 Assignment to an app

Contents can be assigned to one or more apps within a workspace. By assigning them, they become visible within the target app in the AppNavi widget.

3.4.3 Copying contents

To achieve reusability in a very simple way, contents can also be copied within (or between) workspaces.

3.4.4 Publishing in an app

To make content visible to end users, the current version of the content can be published. This is optionally done on the basis of various criteria for the visibility of the content in the respective app (segmentation).

Subsequent versions of the content become visible to end users by republishing. Unpublished intermediate versions remain invisible to end users.

3.5 Roles & permissions

To enable appropriate content management, the following roles are available to authors:

- > **Tenant Owner:** Highest role
- > **Workspace Owner:** Can manage content of one or more workspaces and make structural changes to the workspace, apps, and content.
- > **Workspace Contributor:** Can create and edit content within a workspace.
- > **Workspace Translator:** Can be authorized for translation of individual language packages within a workspace.

A user within a tenant can have different permissions for different workspaces. This allows a very simple staff of individual authors in different projects.

4 Routes

The central content type is routes. Routes guide your users along a defined use case step by step through the respective live application. At each point of a route, tooltips provide help for the user to correctly execute the next step. In on-boarding scenarios, routes can also be assembled chapter by chapter and made available to users in the form of so-called collections.

Your end users can find routes in the AppNavi widget under the heading of the same name.

AppNavi provides other content types besides routes. Depending on the customer situation, the focus in using AppNavi is to support, inform, accelerate, etc. News and collections therefore complement the feature set accordingly.

4.1 Display

Within the AppNavi widget, users are shown all routes relevant to them in a menu. Within this menu, routes can be selected and started.

Routes are displayed sorted by the last modification date or based on an order defined by the authors. In addition, routes can be searched and found using a full text search.

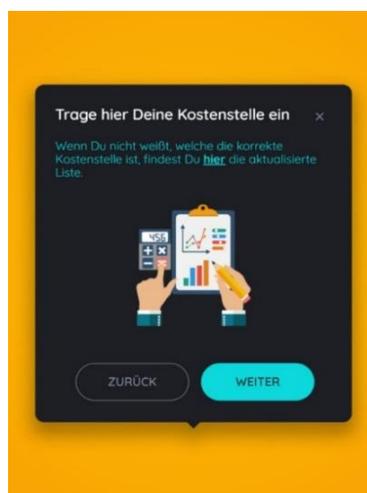
Routes always have a front and a back side in the routes menu. On the front side is the title and the icon of the route. On the back is a description that becomes visible as soon as the end user moves the mouse over the tile of a route.

4.2 Structure & Components

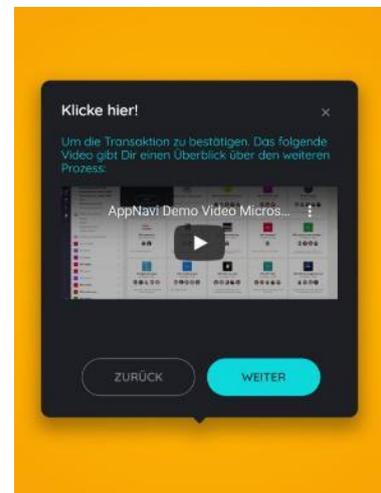
A route is identified by a title, a description and an icon. The content of a route is defined by a string of individual steps. Within AppNavi, two types of steps are available. One is intermediate information, the other is information linked to elements. Both step types are represented by tooltips. These consist of a heading and a content area.



Tooltip with text



... with images



... with video

Tooltips can be customized in size (height and width) and shape. Within tooltips different content can be included:

- > Text (WYSIWYG-Editor)
- > Images
- > Videos
- > Iframes
- > Links



With the help of the WYSIWYG editor, formatting in the tooltips can be done very easily. With the help of a freely definable style template, uniform UI concepts can also be realized across larger AppNavi projects. This is a very useful feature, especially for projects with a high division of labor across topics and countries.

With the help of these options, tooltips can be designed very individually - in addition, for example, with iframes etc., existing content can also be used within routes.

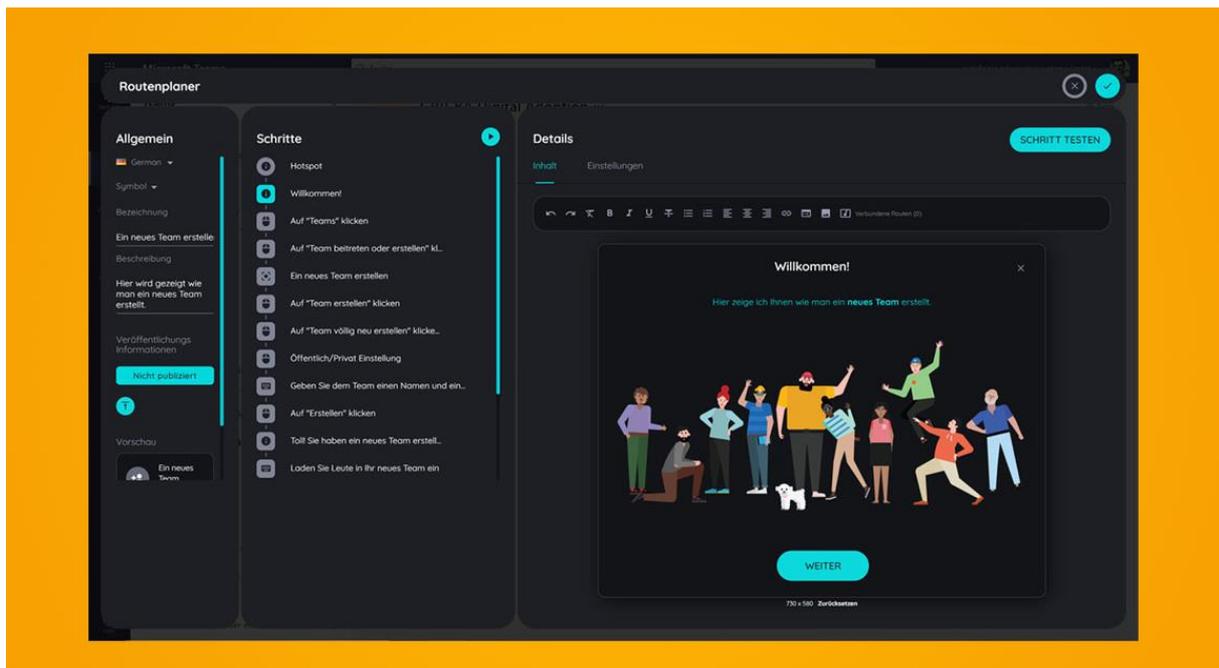
Each step within a route has a so-called interaction type. This defines how end users can move from one step to the next within the route:

- > **Text input** within the application - click on **"Next"** in the tooltip
- > **Left click** within the application
- > **Right click** within the application
- > **Enter** within the application
- > **Hover** within the application
- > **Highlighting** an element within the application - clicking on **"Next"** in the tooltip

Individual steps can be provided with an automation. This allows the respective step to click, tap, etc. independently when starting a route in speed mode.

4.3 Creating routes

The route planner is available for creating routes. This allows you to create routes very easily with the help of a graphical user interface. Technical understanding of the underlying application is not required.



Routenplaner

In addition to being able to create routes using the route planner, there is also the option of having routes created automatically based on the recordings made as part of user behavior mining.

4.4 Element detection

The elements for a step are recognized using a fuzzy score algorithm, which recognizes elements within a page based on a statistical probability model. During the search for a step, this compares all HTML elements occurring in a page (DOM structure) with one (or more) record(s) of the element.



As a result, the most reliable element is displayed if it can be unambiguously determined in the page and fulfills certain constraints - e.g. the identified element must currently be visible in the page.

The special feature of this method is its high reliability. In contrast to common approaches, AppNavi does not store CSS selectors, but context information about an element. Thus, elements are found even in the case of dynamic attributes and IDs as well as, for example, in the case of position changes of the elements.

The element recognition of AppNavi works within the respective HTML document as well as in iframes (Same Origin) and shadow roots within the page.

4.5 Triggers & Modes

Routes can be started in a total of four different ways:

- > Click on the **tile** in the route menu
- > About **URL parameters**
- > Click on a **hotspot** (small "sticker" on an element)
- > **Program-controlled** via the AppNavi Client API

With regard to the modes, two variants can be distinguished. On the one hand, a route can be started in learning mode. Here, all tooltips are run through step by step. The user clicks and taps in the application and thus progresses.

In addition, AppNavi has a speed mode. This allows previously automated steps to be clicked or text typed automatically through AppNavi.

4.6 Variance & Learning

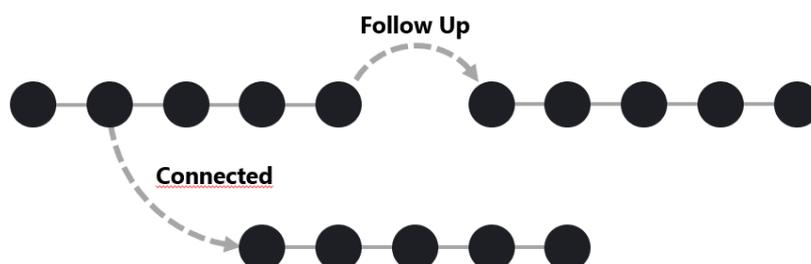
To add intelligence to routes, several options are available. On the one hand, different records can be included in the HTML structure of the page for each step. In addition, individual steps can also be taught. This is useful, for example, in situations where the respective step is defined slightly differently depending on the user, language or similar. Authors can let the step learn the different characteristics of an element. A single recording is then sufficient to map variances in the page.

If individual steps within a route are to be skipped or otherwise automatically controlled, the AppNavi Client API can be used. Here, conditions can be defined and checked for certain events (e.g. On Before Render) - depending on the result, AppNavi then skips individual steps or jumps to a corresponding position in new routes. This can be done completely automatically.

Furthermore, especially in complex scenarios, it is often helpful to create sequences of routes that can then be linked together. This makes it possible to reduce the number of different routes to a minimum and to use the overlapping parts several times. AppNavi offers the possibility to connect routes in two different ways:

- > Connecting routes within a route via "Connected Routes"
- > Connecting routes at the end of each route via "Follow Ups"

The following graphic illustrates the difference between the two options:





In the case of a Follow Up Route, end users will not notice that they are working on two different routes in succession. End users will also not have to select a follow up route, it will simply be launched. It is also possible to select routes from different applications, so you can seamlessly connect use cases across applications.

4.7 Automation & RPA

In order to increase the KPIs "increase first-time-right" and "increase efficiency", AppNavi has a functionality for Attended RPA. The focus here is on the user and his speed in processing his tasks in the respective system.

Within a route, individual or all steps of a route can be automated. Information steps are not automated.

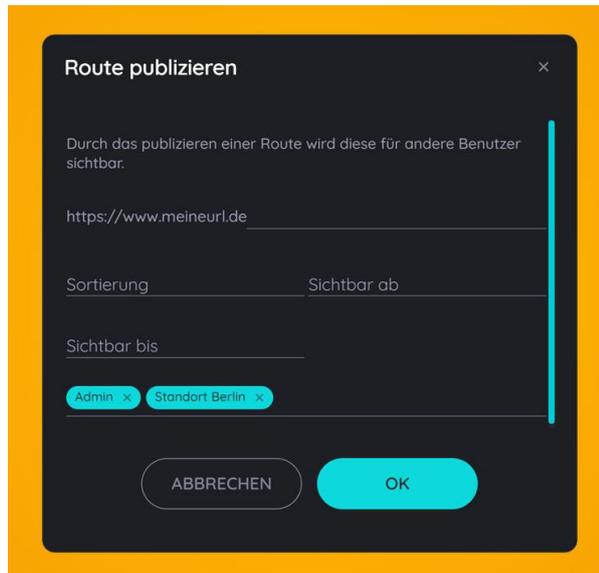
Depending on the start mode of a route, these steps are then run through in learning or speed mode. The default mode valid for the respective route can be set with the help of the route planner - the respective alternative mode can then be reached by end users within the route tile.

Even if routes are started via the API, both modes are available.

4.8 Versioning & Publication

AppNavi has a two-step publication mechanism. A route created by an author can first be saved and thus made available to the other authors. Each saving creates a new version of the route.

As soon as a processing status has reached a certain maturity, the route can be published. This makes it visible to users in the current version. In the background, the authors can continue editing and generate new working versions, which can be made available to the users with a renewed publication.



4.9 Visibility and segmentation

The publication of a route (the same applies to all other content) can be linked to rules. Routes can thus be delivered to certain segments of end users.

Due to its privacy-friendly architecture, AppNavi first delivers all AppNavi content on the page when the end user calls it up - the client then decides which of the contents to show or hide. This enables a targeted delivery of content without requiring information about the respective user.



Typical segmentations:

- > By user role
- > By location
- > By country

For the client-side segmentation, criteria located in the page as well as information external to the system (e.g. via client API) can be used.

For simple segmentation that can be reused across a tenant, target audiences can be created. These are rule sets (e.g. HR Business Partner / Region South), which make it very easy to target specific audiences.

4.10 Other features

A number of other features are available for routes within AppNavi. Here is a selection:

Multilingualism: Routes can be translated into different languages. This is possible both in the route planner and in the customer portal. In addition to the respective languages, dialects can also be offered as a separate package. A route is recorded only once across all languages - i.e. if changes are made, a single adjustment is automatically reflected in all languages.

Automatic translation: A total of 29 languages can be translated automatically (at no additional cost) using DeepL integration.

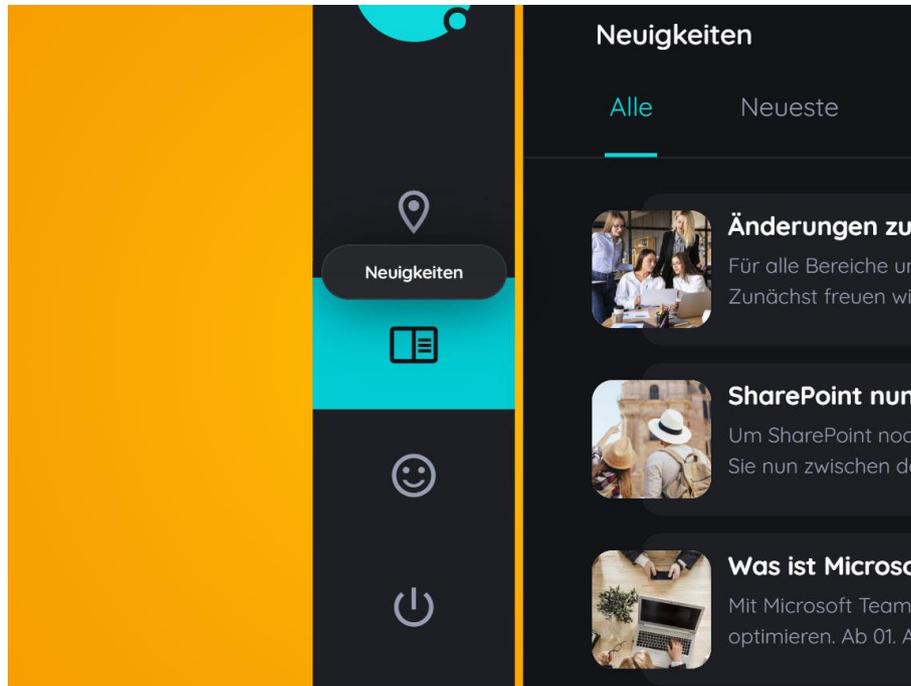
Automatic translation: A total of 29 languages can be translated automatically (at no additional cost) using DeepL integration.

Auto Scrolling & Tracking: If the next element is not in the visible area, our algorithm scrolls to the right place: Horizontally and vertically, in the page as well as in iframes.

5 News & Announcements

News makes it possible to display small articles for users in a targeted manner. Here, too, segmentation takes effect so that the articles are only displayed to the relevant users and only at the relevant place in the system.

News can be displayed from the AppNavi widget, as an announcement or based on user-defined triggers (e.g. initiated when the user visits the page for the first time).



Just like all other AppNavi content, news can be created once centrally within a workspace and then used in different apps. This allows a central creation and maintenance, but at the same time a decentralized use of news.

5.1 Display

Within the AppNavi widget there is a separate menu item for news. A list with all news (corresponding to the respective segmentation) is displayed as soon as this menu item is opened.

News is represented by an image, a title and a description in the list. If the user clicks on a news item, a detail area opens, which displays the content.

5.2 Structure & Components

A news item consists of a Hero image and the actual content of the news item. This can also be edited with a WYSIWYG editor:

- > Text (WYSIWYG-Editor)
- > Images
- > Videos
- > Iframes
- > Links



5.3 Triggers & Modes

News can be launched in a total of three different ways:

- > Click on the **tile** in the menu
- > Click on a **hotspot** (small "sticker" on an element)
- > **Program controlled** via the AppNavi Client API



6 Hotspots

Hotspots make it possible to enrich any browser-based software with additional information that supports the user in making optimal use of the application. In this way, elements can be supplemented with relevant application or process information, either briefly or permanently. The following types of hotspots can be distinguished:

- > **Show hints:** A single tooltip is displayed when the hotspot is clicked.
- > **Show Announcements:** An announcement is opened
- > **Start routes:** A route in an app to be selected is started at a selectable step

To make the creation and maintenance of hotspots as easy as possible, the design is largely based on the creation of routes.

6.1 Display

Hotspots can be stored as small round circles within the UI of the target application. Depending on the type, they have a corresponding icon.

6.2 Triggers & Modes

Hotspots are always searched for independently in the UI of the target system and drawn in as soon as they are found. Hotspots can be shown and hidden via the client API.

If a user clicks on a hotspot, it is triggered and, depending on the type, a corresponding action is triggered.



7 Trainings & Collections

Collections can be used as a feature from the OnBoarding section. Collections allow routes to be combined into chapters. A collection can in turn be assigned to an app and then published. Users can run routes within the collections and see the progress. Once all routes in a collection are completed, the collection is marked as complete.

When a collection is assigned to an app, all routes in it are also assigned to that app (if they are not already). Conversely, when a collection is removed from an app, the routes remain assigned.

When a collection is deleted, the user can specify whether (all) routes in the collection should also be deleted. If they are deleted, they will be deleted completely - with all their further assignments and publications in any other apps. If the user denies this, the routes remain available in the respective apps with unchanged assignments and publications. The collection itself is then deleted in both cases (including assignments and publications).

When a collection is published, all routes in the collection are also published. The following cases can be distinguished here:

1. one route of the collection is already published
2. one route of the collection is already published but currently invisible
3. a route is not yet published
4. a route is depublished (while it is published in a collection) in the app
5. the collection is depublished

Ad 1: The route appears in the route menu and in the collection.

Ad 2, 3, 4: The route remains invisible in the route menu, but appears in the collection.

To 5: The collection becomes invisible. Routes that were previously visible in the menu remain visible - all invisible routes remain invisible.



8 Analytics

AppNavi provides end users with various types of help when using new or existing systems. In order to give the authors of these helps a better insight into the usage of their content and thus improve the quality of the help, AppNavi provides a sophisticated analytics module.

Here, analytics data is collected on three levels and displayed within the AppNavi portal using dashboards:

- > Content level (route, news, etc.)
- > App level
- > Workspace level

8.1 Data acquisition

In order to display analytics data, AppNavi collects various user data. For this, the analysis function must be enabled. By default, the following user data is collected:

- > **Browser name**
- > **Language of the user**
- > **Session ID:** The session ID is generated by the system for each user session. A user session starts when an application is opened in the browser and ends when the user closes the application, e.g. by closing the tab or the browser.
- > **User ID:** By default, AppNavi generates a user ID for each user and stores it in the local memory of the user's browser. However, the user ID can also be set by custom code, e.g. to collect OKRs or similar.

8.2 Metrics

In AppNavi Analytics, various metrics are presented to show user behavior across different content. The basic metrics are described below.

| Active users

An active user is a user who has called an application at least once in a given period of time. If a user with a unique identifier (User ID) has called an application several times in a certain period of time, the user is counted as only one user.

| Interacting users

An interacting user is a unique user who has interacted with AppNavi at least once. An interaction could be:

- > Start a route
- > Resume a route
- > Read a news or announcement
- > Click on the avatar
- > Initiate a hotspot

| Recurring users

A returning user is a user who has used AppNavi at least twice within a certain period of time.



8.3 Diagrams

AppNavi uses descriptive diagrams to visualize analyses. The different types of diagrams are described below.

8.3.1 General

AppNavi uses a hierarchical analysis model: starting from the global level (all data of a workspace within a tenant), data can be viewed more granularly. This includes insights into application-level data, but also content-level data within an application.

- > **Dashboard on the start page**
After clicking on the "Home" menu item in the AppNavi menu of the customer portal, the dashboard appears. This shows all the data aggregated within a tenant and the workspaces selected within it.
- > **Dashboard for an app**
The charts displayed in the dashboard can also be viewed at the application level.
- > **Dashboard for a content item**
The charts displayed in the dashboard and at the application level can also be displayed at the content item level (routes, messages, etc.).

The individual standard diagram types are described below. The time period can be selected for most diagrams:

- > **Last 30 days:** The data of the last 30 days are displayed here.
- > **Total:** This setting visualizes all data since the creation of the workspace, the application or the respective content item on a monthly level

8.3.2 KPIs for the workspace

Small tiles within the dashboard show KPIs and their changes within the last 30 days.

- > **Active users**
This metric shows the number of active users. Active users are users who have used the underlying application in the selected time interval. In addition, the change in this key figure over the last 30 days is shown as a percentage change.
- > **Interacting users**
This figure shows all users who have used AppNavi and the change in this metric.
- > **Time used**
This figure shows the time of all interacting users of AppNavi and the change in this metric.

The Time Used is calculated based on the interaction time. The following times are included in the calculation:

- > The duration of a route run (duration of all step displays within a route).
- > The display duration of a news item or an announcement.
- > The display duration of a notice with the help of a hotspot.

8.3.3 User activity

| Content

This line graph represents the progression of three different user types:

- > **Active users**
This line shows the active users in the selected time interval. Active users are users who used the underlying application in the selected time interval.
- > **Interacting users**
This line shows all users who have used AppNavi.



> **Recurring users**

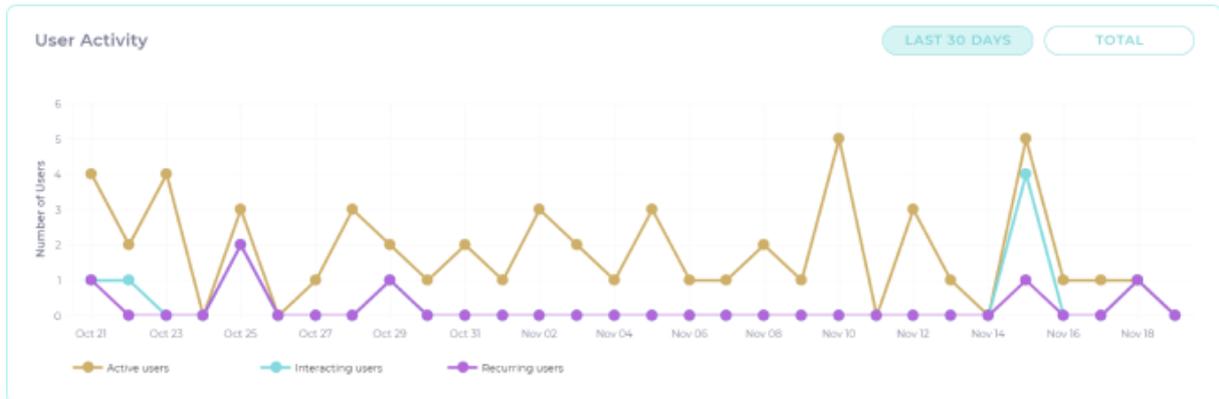
This line displays users who have been active multiple times in the selected period.

Options

- > Show or hide line
- > Select period

Note

If an application is disabled or no longer exists in the tenant or workspace, the data will not be included in the chart.



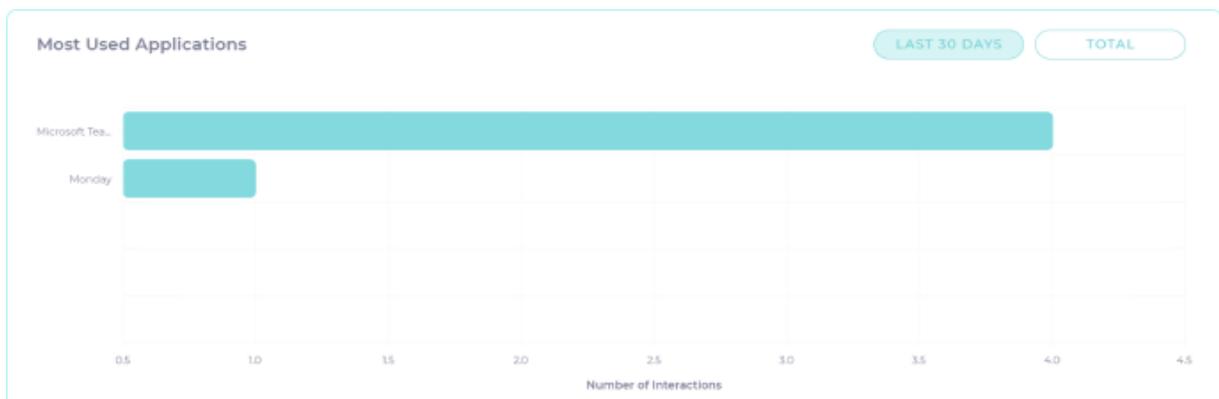
8.3.4 Most used applications

Content

This bar chart shows the top 5 most frequently used applications within Work-spaces. The ranking is based on the interacting users per application. All active and non-deleted applications within the workspace are taken into account.

Note

If you click on the title of a bar, you can view detailed data about the selected application.





8.3.5 Content usage

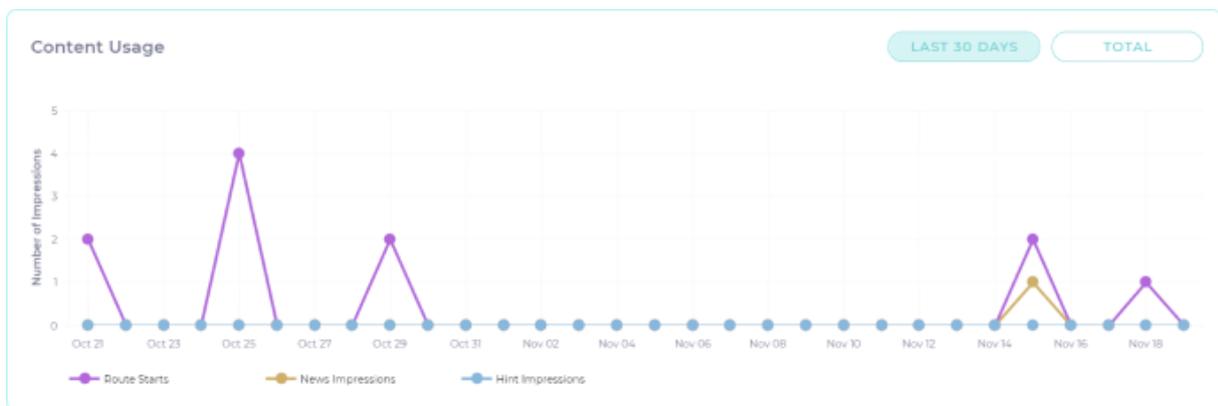
| Content

This line graph shows the usage of three different content types:

- > **Route starts**
The line shows the frequency with which routes were started in the selected period. The routes that were started via hotspots are also counted.
- > **News Impressions**
The line shows the frequency with which messages were started or announcements were displayed in the selected period. The messages or announcements initiated via hotspots are also counted.
- > **Note views**
The line shows the frequency with which notices about hotspots were initiated in the selected period.

| Options

- > Show or hide line
- > Select period



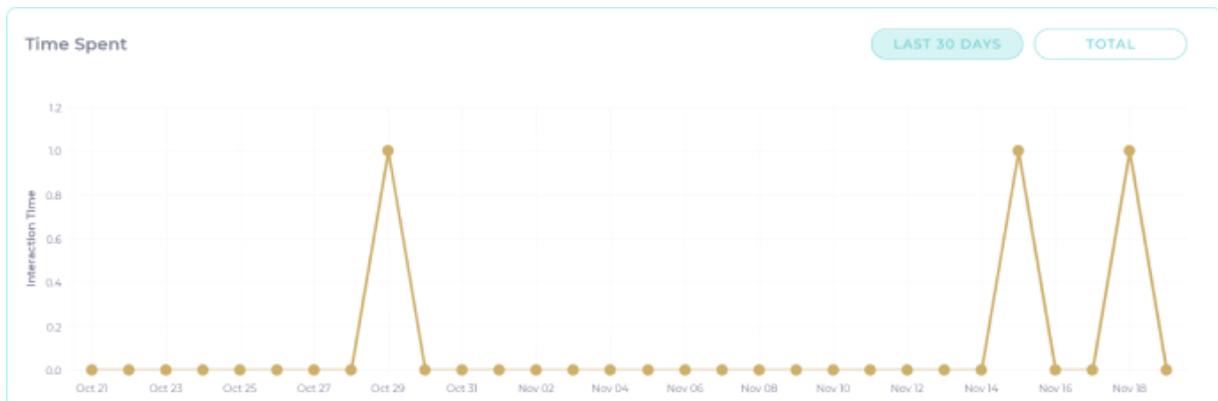
8.3.6 Time spent

| Content

The line shows how much time users have spent with AppNavi content in the selected period.

| Options

- > Select period





8.3.7 Overview of applications

| Content

The table shows all applications supported by AppNavi within the selected workspace. The following information can be found in the table.

- > **Name:** Title of the application
- > **Features:** Features used in the respective application (routes, messages, collections, hotspots).
- > **Routes:** Number of routes in the respective application
- > **Contents:** Number of contents (news, announcements, notices) in the respective application
- > **Analytics:** Indicates whether the Analytics are enabled or disabled.
- > **Theme:** Indicates whether the default dark theme or a custom theme is used
- > **Active:** Shows whether AppNavi is enabled or disabled in the respective application

| Options

- > Search by keyword
- > Add new app
- > Reload

Applications								
Search <input type="text" value=""/>								
+ ADD NEW RELOAD								
	NAME	FEATURES	ROUTES	CONTENTS	ANALYTICS	THEME	ACTIVE	
<input checked="" type="checkbox"/>	AppNavi Learning Center		3	2	<input checked="" type="checkbox"/>	dark	<input checked="" type="checkbox"/>	
<input checked="" type="checkbox"/>	AppNavi Portal		0	0	<input checked="" type="checkbox"/>	dark	<input checked="" type="checkbox"/>	
<input checked="" type="checkbox"/>	AppNavi Test		0	0	<input checked="" type="checkbox"/>	dark	<input checked="" type="checkbox"/>	
<input checked="" type="checkbox"/>	Hotspots		7	1	<input checked="" type="checkbox"/>	dark	<input checked="" type="checkbox"/>	
<input checked="" type="checkbox"/>	WATEV		2	3	<input checked="" type="checkbox"/>	dark	<input checked="" type="checkbox"/>	

Rows per page: 5 10 15

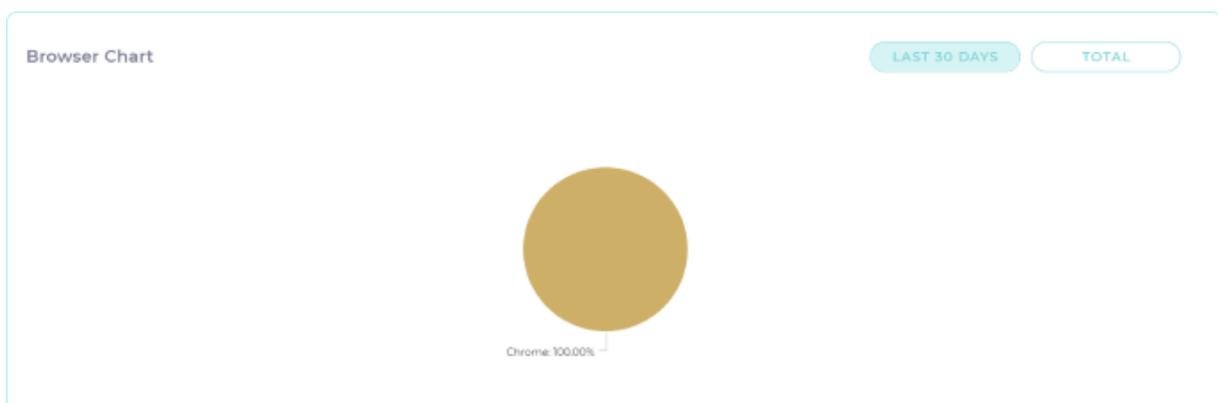
8.3.8 Used browser

| Content

The pie chart shows the percentage of usage of the application by different browsers.

| Options

- > Select period





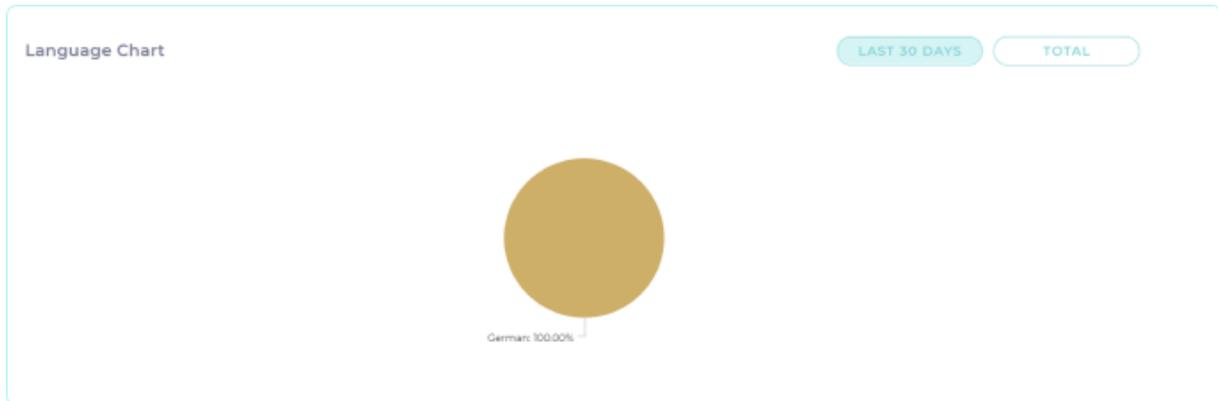
8.3.9 Used language

| Content

The pie chart shows the percentage of application usage in each browser language.

| Options

- > Select period



8.3.10 Route problems

| Content

This table contains information about possible route impairments. Authors can use it to get a quick overview of which routes may need to be reviewed. The following information can be seen in the table:

- > **Route title:** Title of the respective route
- > **Step title:** Title of the step that may need to be examined
- > **Error rate:** Calculated error rate of the corresponding step
- > **Criticality:** Calculated criticality of the potential error

| Options

- > Search by keyword
- > Reload

The figure shows a table titled "Route issues (last 30 days)". At the top right, there is a search bar labeled "Search route issues" and a "RELOAD" button. The table has four columns: "ROUTE TITLE", "STEP TITLE", "ERROR RATE", and "CRITICALITY". The table body is empty, with the text "No data available" centered below the header row.

8.3.11 KPIs for a route

Small tiles within the dashboard show KPIs within the last 30 days.

- > **Route starts**
This metric shows how many times the selected route has been displayed and how the number has changed over the last 30 days.



- > **Unique route starts**

This number shows how many times the selected route has been displayed by different users and how the number has changed in the last 30 days.

- > **Average display time**

This number shows how long individual steps of the route were displayed on average and how the number has changed in the last 30 days.

8.3.12 Route progress

| Content

This table contains information about steps in a route and their achievement in the course of route usage.

- > **Step title:** Title of the route step.
- > **Display duration (seconds):** Number of seconds a step is displayed on average.
- > **Displayed by user:** percentage of display of this step over all route runs.

| Options

- > Search by keyword
- > Reload

STEP TITLE	DISPLAY DURATION (SECONDS)	SHOWN BY USER
Willkommen	6	75%
SCHRITT 1: Klicken Sie auf "Chat"	6	75%
SCHRITT 2: Klicken Sie auf das Bearbeitungsymbol	32	75%
SCHRITT 3: Geben Sie einen Namen ein	6	10%
SCHRITT 4: Schreiben Sie eine Nachricht	0	0%
SCHRITT 5: Schreiben Sie Ihre Nachricht los	0	0%
Herzlichen Glückwunsch!	0	0%

8.3.13 KPIs for a news item or an announcement

Small tiles within the dashboard show KPIs and their changes within the last 30 days.

- > **Views**

This metric shows how many times the selected news item or announcement has been viewed and how the number has changed over the last 30 days.

- > **Unique views**

This number shows how many times the selected news or announcement was viewed by different users and how the number has changed in the last 30 days.

- > **Average display time**

This number indicates how long the news or announcements were displayed on average and how the value has changed in the last 30 days.

8.3.14 News

| Content

This line graph shows the number of views of news or announcements in the selected period:

- > **Views**

This line shows all calls to news and announcements by all users.

- > **Unique impressions**

This line displays all calls to news and announcements by individual users.

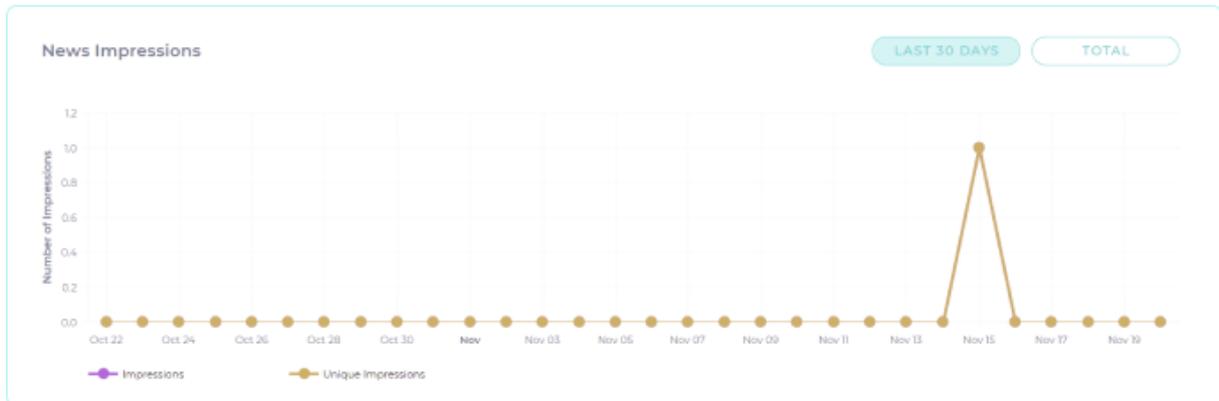
| Options

- > Show or hide rows
- > Select period



Note

If an application is disabled or no longer exists in the tenant or workspace, the data will not be included in the chart.





9 User Behavior Mining

With the help of user behavior mining, it is possible to obtain information about the behavior of users in an application. The evaluations of user behavior mining provide answers to the following questions, for example:

- > How many users use the software?
- > How long is the software used by a user?
- > How regularly and at what times is the software used?
- > Which functions and processes of a software are used?
- > At which points in the software do problems occur?

User behavior mining comes with additive costs. Before using it, you need to make sure that the feature is enabled.

9.1 Managing User Behavior Mining Campaigns

User Behavior Mining campaigns are managed via the context of the application in the application overview. The prerequisite for this is that a contract with activated user behavior mining is assigned to the application.

The recording of analysis data always takes place in so-called campaigns. A campaign has a fixed duration of 30 days. The start date of the campaign can be defined. There can always be exactly one active campaign per application for a given period of time. Campaigns can be paused and resumed at any time. When a campaign is paused, no user behavior data is collected.

9.2 Interaction objects

An interaction object is a UI element of the web application that the user has interacted with. By interactions we mean:

- > Left click
- > Right click
- > Keyboard input

The interaction object is captured during the first interaction. However, interaction objects are only ever captured once per application. Therefore, an action, such as changing the title of an interaction object, always affects all campaigns within an application. The following information is displayed in the UI interaction item overview:

- > **Title:** The captured or modified title of an interaction element.
- > **Page url:** The URL of the page where the interaction element was captured.
- > **Last Discovery:** The time when the element was captured.
- > **Recapture:** Indicates whether the element is in recapture mode.
- > **Active:** Indicates whether the element is active or not.

| Recovery

Each UI interaction element contains descriptive information about the element that is used to find an element. When this information needs to be updated (for example, because an element can no longer be found), retrieval mode can be enabled. When a user triggers an event on the element, the updated information is saved.

| Enable and disable

If an element is deactivated, it will no longer be searched for and displayed in the Label Wizard.



9.3 Sessions

Sessions offer the possibility to analyze the behavior of individual users in a session. In this way, a sense of how users behave in an application can be developed. A session begins when a browser window or tab is opened and ends when the browser window or tab is closed. All recorded data is anonymous. It is therefore possible to determine how a user behaves in an application, but not who that user is. The Session Details view shows the exact flow of a user session.

9.4 Behavior Flows

A Behavior Flow is a behavior pattern identified by an AI-powered mechanism. For this purpose, the behavior of all users in a campaign is analyzed and the behavior patterns with the highest frequency of execution are displayed. Within the profile of a campaign, the following information can be displayed using a dashboard (Behavior Flow Cockpit):

- > The number of complete process executions
- > The number of partial process executions
- > The average cycle time of the process
- > The complete and partial process executions over time
- > A heatmap diagram shows the progression of process executions by days of the week and time windows
- > Representation of the individual actions within the behavior pattern. The Time To Reach value shows how much time has passed on average from the execution of one step to the execution of the next step.

9.5 Automated creation of routes

Most common user behaviors can be converted into a route using behavioral flows. That is, the user can create routes based on the behavioral flows available in a campaign.

9.6 Data security

The data collected during an enabled User Behavior Mining campaign is not personal. The following Analytics data is collected:

- > Browser
- > Language
- > Action (left click / right click / text input)
- > Element information

No data is collected from the page itself, such as the text of a user input. For this purpose, a **data clearing algorithm** is used, which removes names, phone numbers, bank details, etc.



10 Possibilities for customizing

10.1 Standard Text Styling

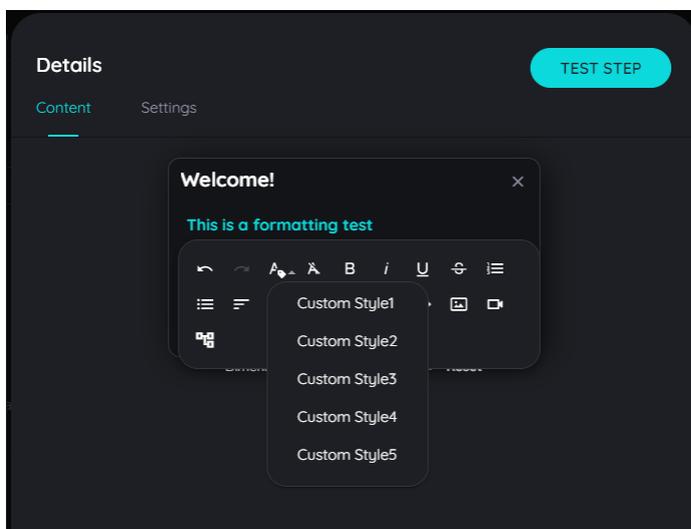
In some organizations it is required that the CI of the organization will be used (e.g. Arial font as default font).

To support this requirement, AppNavi provides the ability to set up to 5 custom style sheets in the portal. These can then be used by Planner to design route and hotspot content.

To create a custom style sheet, the following 5 classes can be defined in the Custom CSS section of the application:

- > .an-custom-style-1
- > .an-custom-style-2
- > .an-custom-style-3
- > .an-custom-style-4
- > .an-custom-style-5

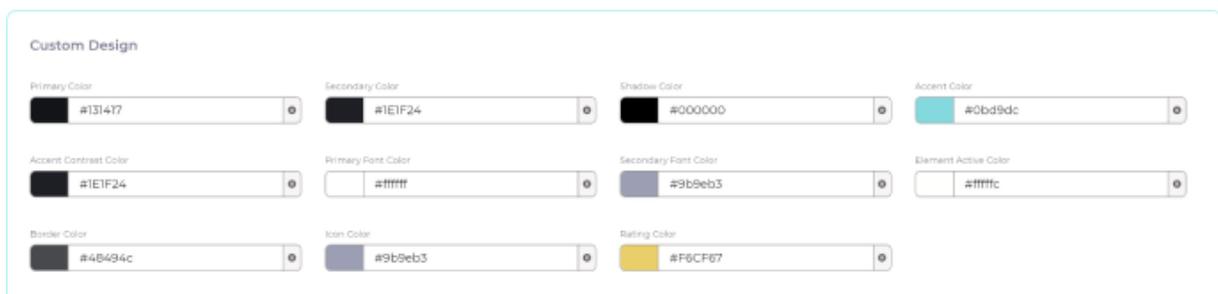
Each of the above classes defines how the corresponding content is displayed in Planner.



10.2 Custom Theming

In the profile of an app, a total of 11 colors can be defined for the AppNavi color scheme when Custom Theming is enabled. In addition, it is possible to make even more subtle changes to the UI with the help of Custom CSS.

Colorpicker





- > **Primary color**
The primary color affects the basic appearance of the AppNavi menu background as well as the background of tooltips, hotspots, news and announcements.
- > **Secondary color**
The secondary color affects the basic appearance of the AppNavi route tiles.
- > **Shadow color**
The shadow color affects all shadows in AppNavi. For example, the shadows for the route tiles.
- > **Accent color**
The accent color sets accents in AppNavi. It affects the color of buttons (for example, in tooltips, the route planner, or the AppNavi menu) and other highlighted elements.
- > **Accent contrast color**
The accent contrast color stands out from the accent color.
- > **Primary font color**
The primary font color is usually used in areas with the primary color, e.g. on route tiles, in the AppNavi menu.
- > **Secondary font color**
The secondary font color is usually used in all other areas.
- > **Element active color**
The element active color is used to highlight active or selected elements.
- > **Border color**
The frame active color is used for all frames.
- > **Icon color**
The icon color affects the appearance of all icons in AppNavi, for example the icons on the route tiles.
- > **Rating color**
The rating color controls the rating stars on the route tiles.

10.3 Custom Coding

With the help of an SDK, AppNavi provides the AppNavi Client API. This set of methods and events allows to connect and extend the functionality of AppNavi with the target application.

Within the AppNavi architecture, custom code can be used in two places. First, in the profile of an application, and second, in individual steps within the profile of a route.

The following options are available:

| General

- > setGuestUserId :method
- > setPreferredLanguage :method
- > hideAvatar :method
- > showAvatar :method
- > getCurrentApplicationId :method
- > getCurrentApplication :method
- > getCurrentSubscriptionId :method
- > getApplicationContainer :method
- > setTargetAudienceConditions :method
- > avatarLoad :event

| Announcement

- > getAvailableNews :method
- > showAnnouncement :method
- > newsShow :event
- > newsRead :event
- > announcementShow :event



| Hotspots

- > getAvailableHotspotCollections :method
- > getPageHotspots :method
- > hidePageHotspots :method
- > showPageHotspots :method
- > hideHotspotItem :method
- > showHotspotItem :method
- > startHotspotAction :method
- > stopHotspotAction :method
- > startRenderHotspots :method
- > stopRenderHotspots :method
- > hotspotActionStart :event
- > hotspotActionStop :event

| Automation

- > insertValue :method
- > skipAutomation :method
- > Analytics
- > setUserRole :method

| Route

- > startRoute :method
- > getActiveRouteId :method
- > getCurrentStepId :method
- > getStartMode :method
- > stopRoute :method
- > playStep :method
- > getCurrentElement :method
- > getAvailableRoutes :method
- > getTooltipContainer :method
- > routeStart :event
- > routeStop :event
- > playStep :event



11 Data protection

Basically, the functionality of AppNavi is based on a "data-blind" concept. I.e. AppNavi routes are started in the target system and know there the context of the elements to be found (element type, attributes, classes, approximate position, etc.). Completely unknown, however, is the content of the elements. Thus no input values, texts, or the like are stored.

At runtime, the fuzzy score algorithm determines the element with the highest match between recorded route definition and current screen situation.

The functionality of AppNavi (including the segmentation logic) is always executed on the client side - a transfer of information to the backend of AppNavi does not take place at any time.

12 IT-Security

As a SaaS solution, AppNavi is regularly tested by external pentests. These are based on common standards such as OWASP and check criteria such as session management, file upload and download, cross-site scripting, SQL injection, path traversal, parameter manipulation and other OWASP top 10 vulnerabilities.

In addition to the audits initiated by AppNavi itself, AppNavi was also tested in numerous other tests during customer implementations.

| Authorization concept

- > Access level according to the **need-to-know principle**.
- > User authorizations are defined and assigned by the respective departments
- > Dedicated user roles are available

| Data privacy

- > Data is always encrypted - during storage ("at rest") as well as during transmission ("at transit").
- > The integrity of the information is ensured by logging and version control of all user changes
- > The data is stored in an AWS data center (Frankfurt)

No personal data is stored. The analysis data is evaluated per user population. It is not possible to draw conclusions about individual users.

13 Software ergonomics

AppNavi and all its components are designed to be as simple and intuitive to use as possible.

In this area, various tests have been successfully passed at group works councils. The focus here is on software ergonomics topics such as:

- > Operability
- > Consistency
- > Accessibility

During development, we follow the standards and specifications according to DIN EN ISO 9241 (parts 11, 13-16, 110, 112, 125, 129, 143, 151, 154, 161, 171, 303, 306) and WCAG 2.1 Level AA (ISO/IEC 40500:2012).



14 Boundaries and limitations

The following limits and restrictions apply to the use of AppNavi. These can be adjusted if necessary:

Tenant	
Max. Workspaces per Tenant	100
Max. Images per Tenant	1GB
Max. Image size	3MB
Max. Authors per tenant	100
Max. Routes per Tenant	250
Max. News per Tenant	250
Max. Applications per Tenant	100
Max. Route-Collections per Tenant	50
Max. Length title	50
Max. Length Company	100
Max. Number of backups	5GB
Max. Number of User Profiles	50,000

Application	
Max. Number of assigned routes	50
Max. Number of assigned news	50
Max. Number of assigned route collections	25
Max. Length title	50
Max. Length category	50
Max. Length pattern	128
Max. Length url	128
Max. Length Custom Code	15,000
Max. Length Custom CSS	15,000
Max. Size image	3MB
Allowed image formats: .jpg, .jpeg, .png	3
Max. Number of published routes	30
Max. Number of published news	50
Max. Number of published route collections	10

News	
Max. Length title	50
Max. Length Description	100
Max. Length Content	1,000
Max. Size image	3MB
Allowed image formats: .jpg, .jpeg, .png	3

Users	
Max. Length first name	25
Max. Length last name	25
Max. Length phone number	25
Max. Length email address	50
Max. Length username	25

Workspaces	
Max. Routes per Workspace	50
Max. Applications per Workspace	50
Max. News per Workspace	100
Max. Route-Collections per Workspace	25
Max. Authors per Workspace	100
Max. Length title	50
Max. Length Description	100

Routes	
Max. Number of steps	30
Max. Length title per step	100
Max. Length Content per step	1,000
Max. Size per step	100KB
Max. Length Custom Code (OnBeforeRender) / Step	5,000
Max. Length Custom Code (OnAfterRender) / Step	5,000
Max. Number of captures per step	3
Max. Length Url (relative path)	512
Max. Size route	3MB
Max. Length Description	100

Routes-Collections	
Max. Length title	50
Max. Length Description	100
Max. Length Content	1,000
Max. Size image	3MB
Allowed image formats: .jpg, .jpeg, .png	3
Max. Number of assigned routes	15



15 Other service areas

In addition to providing software products, AppNavi offers other services:

| SW- Implementation

- > **Consulting:** requirements gathering, adoption design, learner journeys, KPI definition, etc.
- > **Set-Up Services:** Integration, customizing, styling, segmentations, content creation.
- > **Training:** Training of authors
- > **Go Live Support**

| SW maintenance

- > **Standard product support:** provision of releases, troubleshooting (incident management)
- > **Optional support services:** 1st/2nd level for standard applications, more extensive service hours, individual service levels, additional support languages, dedicated support