



System Design Document

MICROSOFT TEAMS INTEGRATION FOR THE WAZOKU
PLATFORM

AUTHOR: LUIS CRUZ - ENGINEERING MANAGER, WAZOKU

DATE: 11/08/2023

Contents

Introduction	2
Purpose of this document.....	2
Overview of the solution.....	3
Purpose and objectives.....	3
Key features	3
Target audience.....	3
Benefits	3
Stakeholders.....	4
System architecture	4
Components overview	5
Interaction flow.....	7
Communication protocols.....	8
Data model.....	9
Functional requirements.....	10
User interface design	11
Testing and quality assurance	12
Development workflow.....	12
Deployment process	13
Maintenance and support.....	13
Versioning.....	13
Training and support	14
Monitoring	14
Bug tracking	15
Escalation procedure and bug fixing timeline.....	15
Regression testing	16
Documentation and knowledge base	16

Introduction

Purpose of this document

The purpose of this System Design Document (SDD) is to provide a comprehensive and detailed description of the design and architecture of the Microsoft Teams integration for Wazoku's platform. It serves as a reference for developers, stakeholders, and other project team members involved in the development and implementation of the system.

This document aims to achieve the following objectives:

1. **System understanding:** The SDD aims to establish a clear understanding of the system's structure, components, and interactions. It provides insights into how the system is designed and organized, enabling effective communication and collaboration among project stakeholders.
2. **Design documentation:** The SDD serves as a central repository for documenting the system's architecture, data model, user interface design, and other design-related aspects. It provides a comprehensive overview of the system's technical specifications, facilitating development, maintenance, and future enhancements.
3. **Requirements traceability:** The SDD links the system design to the functional and non-functional requirements outlined in the system requirements document (SRD). It ensures that the design decisions align with the specified requirements and allows for easy verification and validation of the system's compliance.
4. **Support development efforts:** The SDD provides developers with the necessary information and guidelines to implement the system accurately. It outlines the technical details, design patterns, and best practices to be followed during the development phase, reducing ambiguity and ensuring consistent implementation.
5. **Facilitate collaboration:** The SDD acts as a reference point for stakeholders, allowing them to assess the system's design, provide feedback, and make informed decisions. It promotes effective collaboration among project team members, including developers, designers, testers, and project managers.

Overview of the solution

The Microsoft Teams app for Wazoku is a pivotal extension of our broader platform, dedicated to empowering organizations to harness the collective intelligence of their teams. This app integrates with Microsoft Teams, creating a powerful synergy that enhances productivity while fostering active engagement and participation among users.

Purpose and objectives

At the core of Microsoft Teams app for Wazoku lies a dual purpose: to drive productivity and to cultivate a culture of innovation and collaboration within the organization. By providing a personal app for each user and leveraging Microsoft Teams' collaborative features, we aim to streamline idea submission, feedback collection, and decision-making, fuelling a continuous cycle of ideation and implementation.

Key features

- **Personalized Idea submissions:** Through the Microsoft Teams app, users gain access to their personalized idea management interface, enabling them to effortlessly discover challenges or ideas and submit new ideas directly within the familiar Teams environment.
- **Real-time engagement and notifications:** The app facilitates dynamic interactions, ensuring users receive real-time notifications on idea status updates, comments, and voting capabilities. This ensures prompt engagement and keeps participants informed throughout the ideation process.
- **Interactive commenting bot:** A responsive bot facilitates engaging conversations around ideas. Users can effortlessly comment, vote, provide feedback, and collaborate on proposals without leaving the Teams app.

Target audience

The Microsoft Teams app for Wazoku caters to forward-thinking organizations and teams seeking to cultivate a culture of innovation, openness, and inclusivity. It benefits every user eager to actively contribute to the organization's growth and success.

Benefits

- **Boosted productivity:** By integrating seamlessly with Microsoft Teams, the app minimizes disruptions and enhances productivity, allowing users to ideate and collaborate efficiently within their regular workflow.

- **Amplified user engagement:** The app's user-centric approach and interactive features drive higher engagement, encouraging wider participation in idea generation and meaningful discussions.
- **Accelerated innovation:** Leveraging the collective intelligence of employees, the app empowers organizations to discover transformative ideas faster, leading to accelerated innovation and improved decision-making.

Stakeholders

The successful implementation of the solution inside Microsoft Teams involves collaboration and coordination among various stakeholders, each contributing their expertise and insights. Key stakeholders include:

- **Delivery team:** The delivery team and specially the Integration Squad is tasked with integrating the Wazoku platform and Microsoft Teams. Their deep understanding of APIs and system integrations ensures a smooth and cohesive user experience, allowing idea management to become a part of the Teams ecosystem.
- **Clients and end-users:** Our valued clients play a central role in the app's adoption and success. They represent the driving force behind the culture of innovation within their organizations. Their feedback, needs, and expectations are integral to shaping the app's features and functionality. Engaging with end-users ensures that the app aligns with their specific workflows, enhancing their overall experience and encouraging active participation in idea generation and evaluation.

By collaborating with these dedicated stakeholders, we aim to create a transformative and user-centric Microsoft Teams app that revolutionizes Wazoku's platform, driving growth, innovation, and collective success across organizations.

System architecture

The Microsoft Teams app for Wazoku follows a modular and scalable architecture to ensure flexibility, maintainability, and integration with the Microsoft Teams platform. The architecture is designed to offer a personalized and engaging experience for users, leveraging the powerful capabilities of Teams.

Components overview

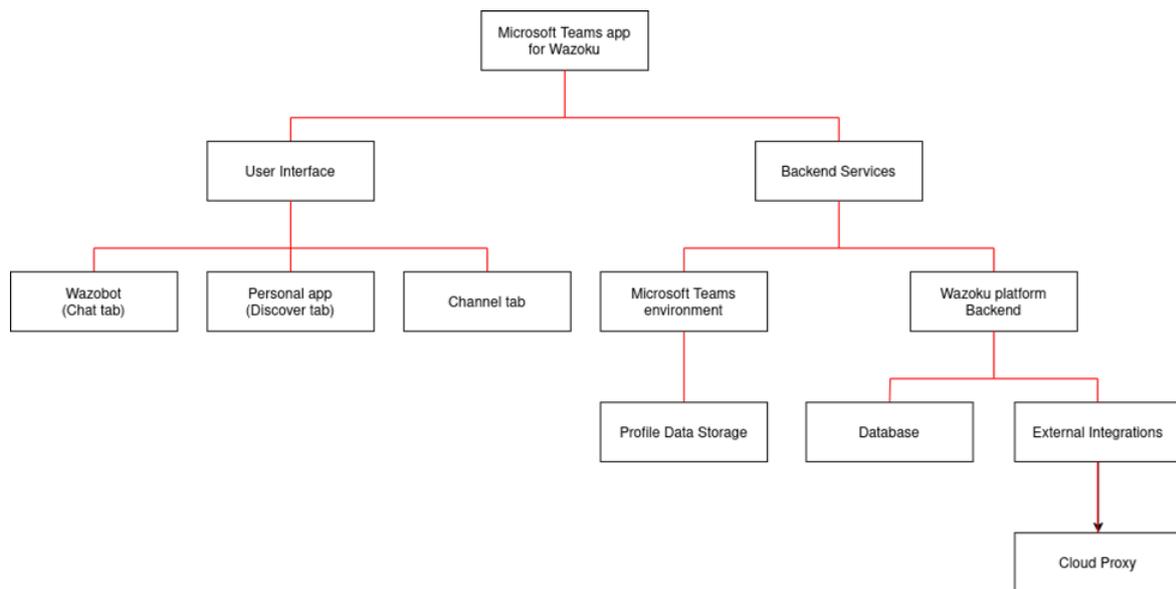


Figure 1: A diagram showing the components of the Microsoft Teams app for the Wazoku Platform.

Below you will find a brief explanation of each component of the Microsoft Teams app for the Wazoku Platform.

- **Microsoft Teams app interface:** The front-facing component of the app is the personalized interface embedded within Microsoft Teams. This interface allows users to access their discover page to search for ideas or challenges, create and submit ideas, view idea statuses.
- **WazoBot:** At the core of the app's interaction lies the WazoBot. This bot serves as a bridge between users and the Wazoku's platform backend. In its first version, it facilitates interactions around ideas, providing real-time notifications, and offering a user-friendly interface to gather comments.
- **Wazoku Platform backend:** The Wazoku platform Backend is the backbone of the app, responsible for processing and managing ideas, user data, and app-specific configurations. It ensures data integrity, implements business logic, and orchestrates communication with external services, APIs, and databases.
- **Database and data storage:** To handle user profiles, the integration utilizes a file storage system. This data storage component stores and retrieves information securely, ensuring efficient access and data management.
- **External integrations:** The Microsoft Teams app for Wazoku integrates our broader platform, leveraging existing functionalities and services. The integration with Microsoft Teams is facilitated through official APIs and SDKs, allowing for secure communication and seamless user experiences.

- **Cloud proxy:** The Cloud Proxy is a specialized intermediary service that functions as a central hub for communication between our multitenant platform, various third-party tools, and social media platforms. Its primary role is to facilitate seamless and secure interactions, data exchange, and integration between the platform's multiple domains and the diverse external services. The cloud proxy acts as a single point of contact for outbound and inbound communications from our multitenant platform, allowing efficient management and coordination of data and interactions with third-party tools, APIs, or social media platforms. By serving as a social broker, it streamlines the complexities of handling numerous connections and APIs, enhancing the platform's functionality while ensuring a consistent and unified experience for tenants across different domains.

Interaction flow

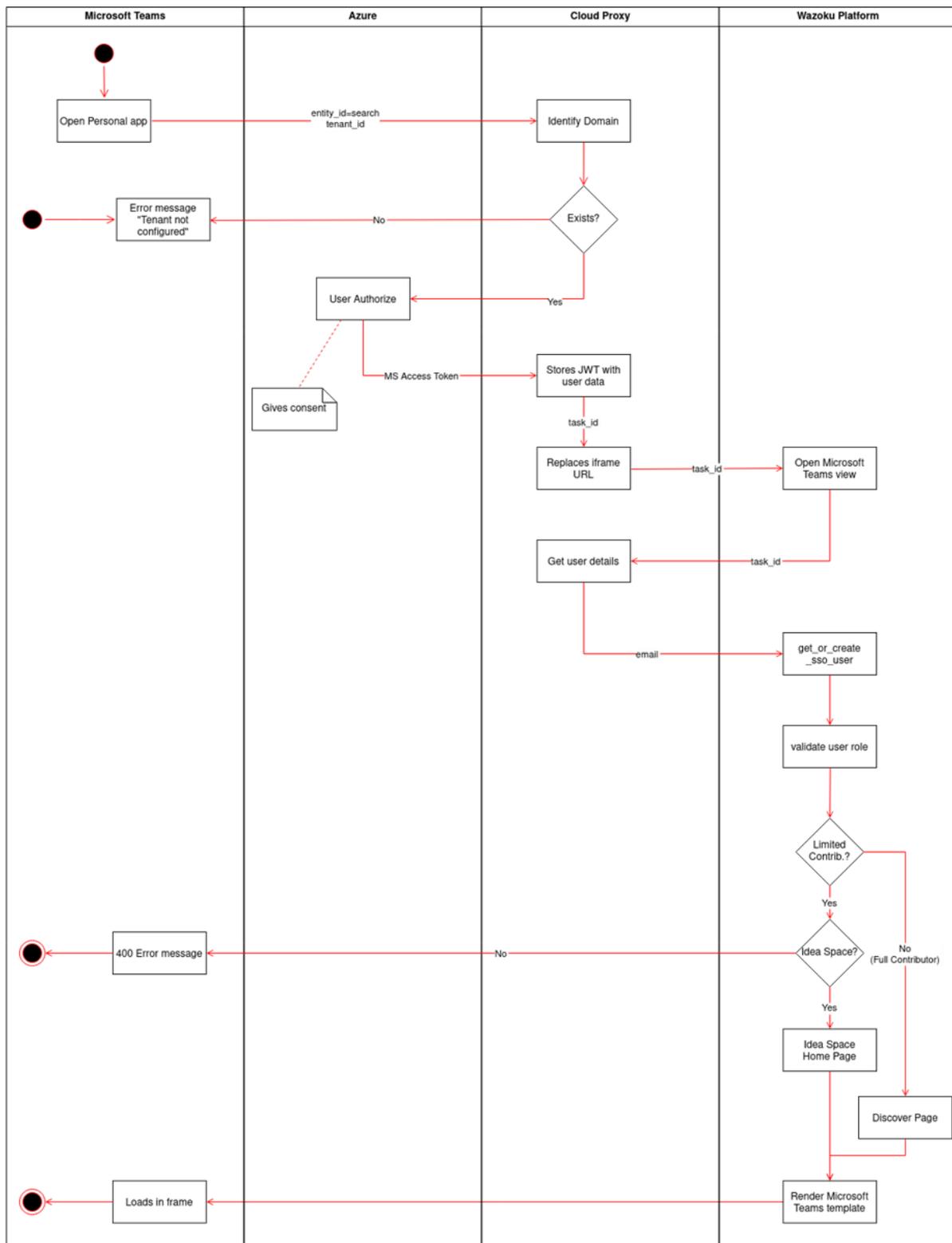


Figure 2: A diagram showing the interaction flow of the Personal App.

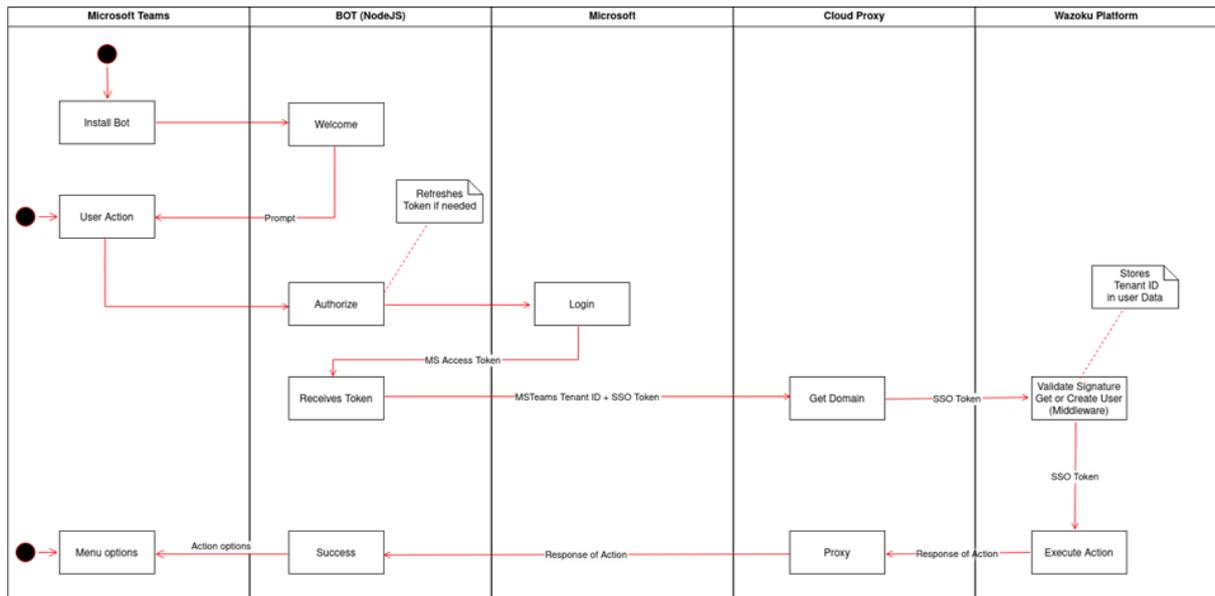


Figure 3: A diagram showing the WazoBot authentication flow.

Communication protocols

The Microsoft Teams app for Wazoku prioritizes security and user convenience through the implementation of robust communication protocols. These protocols ensure the integrity and confidentiality of data exchanged between various app components. Key communication protocols include:

- HTTPS (Hypertext Transfer Protocol Secure):** All communication between the app's front-end interface, idea management bot, backend services, and external integrations takes place over HTTPS. This ensures that data transmitted is encrypted, safeguarding sensitive user information and preventing unauthorized access.
- OAuth 2.0 (Open Authorization):** The app incorporates OAuth 2.0 as the standard authorization protocol for user authentication and secure access to external services and APIs. This enables users to log in securely to the app using their Microsoft Teams credentials, leveraging Single Sign-On (SSO) capabilities. SSO streamlines the authentication process, reducing the need for multiple logins and enhancing the overall user experience.
- JSON Web Tokens (JWT):** JWTs are used for securely transmitting information between parties as compact and URL-safe JSON objects. The app leverages JWTs for transmitting authenticated user information and to verify the integrity of messages exchanged within the app.

The adoption of these communication protocols ensures that the Microsoft Teams app for Wazoku maintains a high level of data security, protecting sensitive user data and interactions from

unauthorized access and potential threats. By incorporating OAuth 2.0 and SSO, the app provides a secure user experience, eliminating the need for repeated logins and improving overall usability.

Data model

There are some models in Cloud Proxy, regarding the Microsoft Teams integration exclusively:

- **CompanyMSTeamsConfiguration:** It maps the `tenant_id` with the domain.
- **TeamsJWTToken:** Stores the claims coming from the SSO, validating they come from Microsoft. Alongside it stores the domain, email and token itself. Later, the `couch_id` of this instance is used to query the email from Spotlight, through a secured call.

In Wazoku platform, a couple of fields are added to the User Profiles in this Microsoft Teams integration for the app to work:

- **msteams_tenant:** Same tenant id of the organization the user belongs to.
- **msteams_oid:** Unique identifier of the User and the Personal App conversation. Used to send the notifications to that user.

Both are set when user signs in with the bot and removed when they sign out. They are also removed if the tenant is removed from the Admin Panel.

On a high level, the bot is a NodeJS application built using express, that is connected to the Wazoku platform. This application contains the necessary logic for receiving messages from the user chat, display content, and interact with the actions taken by the users on the cards.

Below, you can find a flow diagram describing how the bot handles these actions on a high level, which are considered as activities.

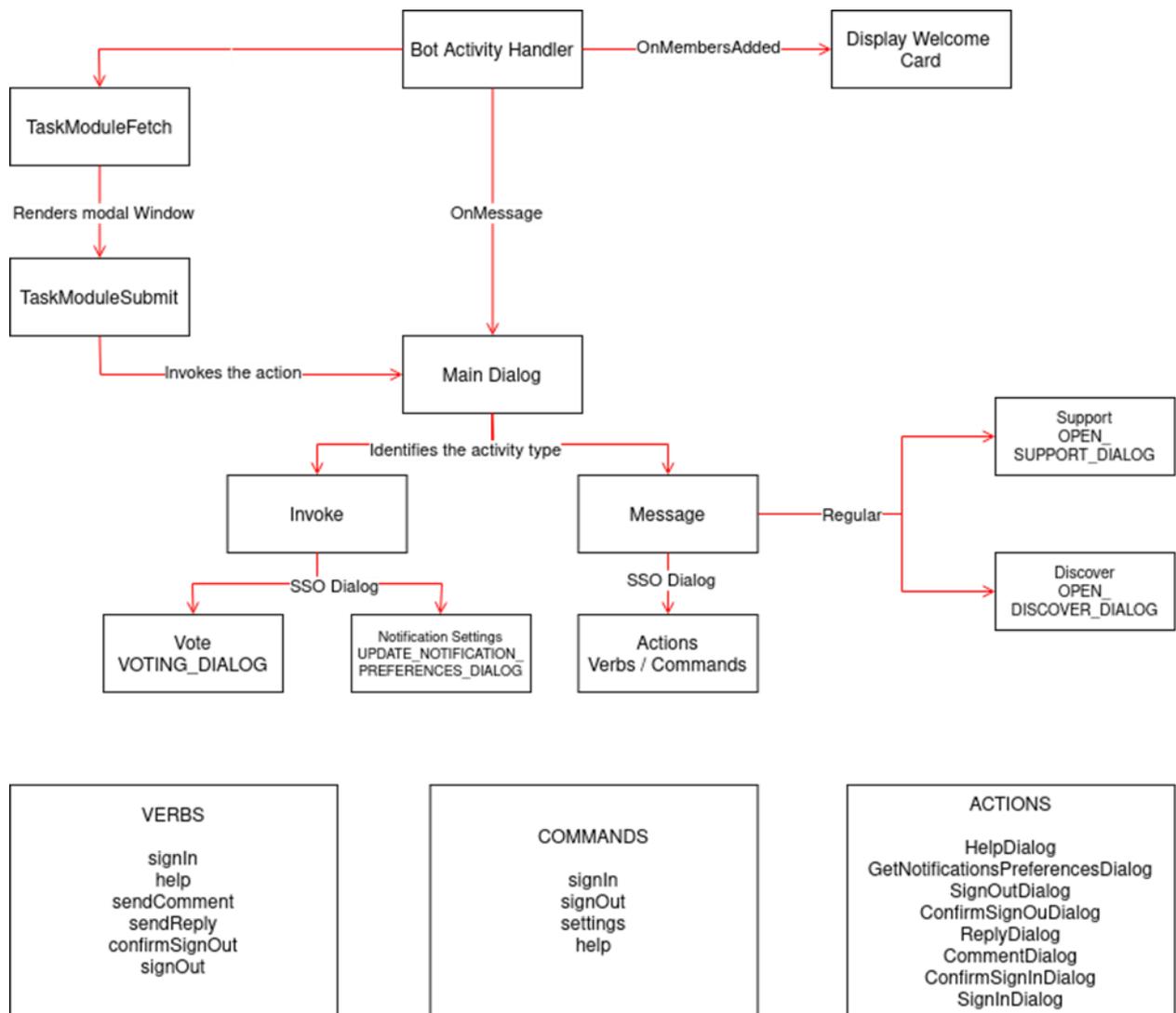


Figure 4 - A diagram showing the bot logic.

Each activity is handled as a Dialog, which can be considered as a “Conversation turn” where the bot sends content to the users. Once the Dialog is completed, the bot waits for the next user interaction which can be either a message, or an action on a card.

Functional requirements

In order to understand the comprehensive functionality of the Microsoft Teams app for Wazoku, please refer to the following support article: <https://support.wazoku.com/connect-microsoft-teams-to-your-platform> This document provides detailed descriptions of user interactions, features, and workflows, offering insights into how the app enhances idea management and user engagement.

User interface design

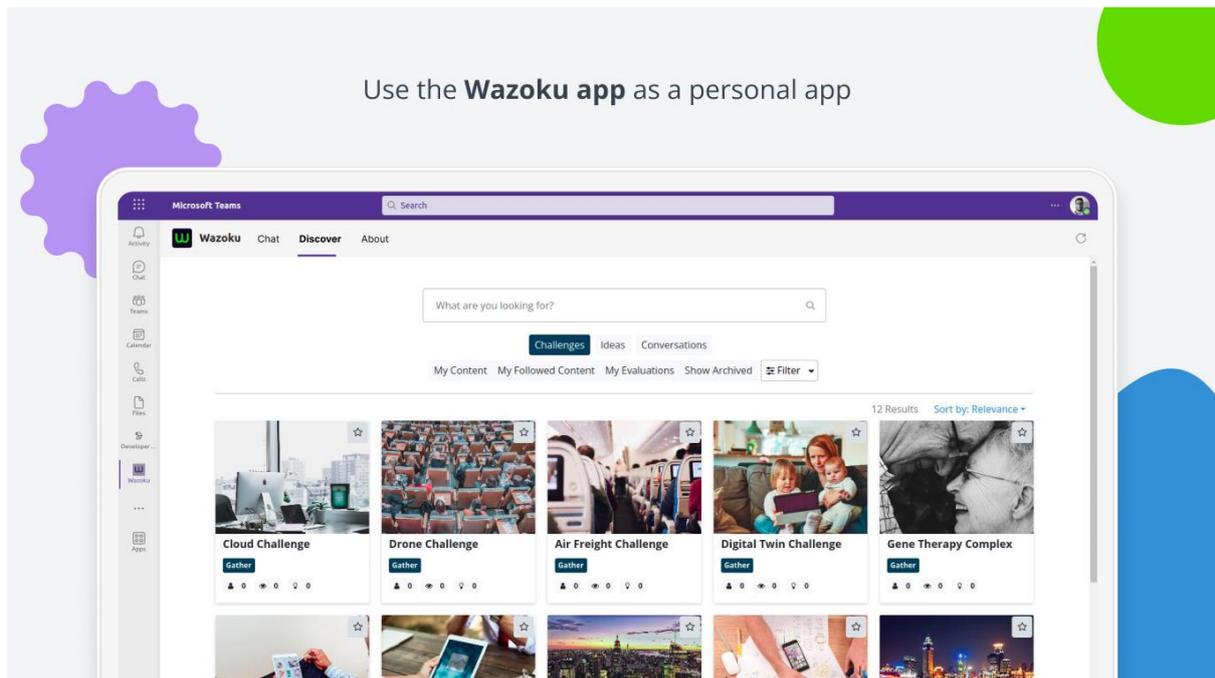


Figure 5: A design showing the Discover tab within the Personal App on Microsoft Teams.

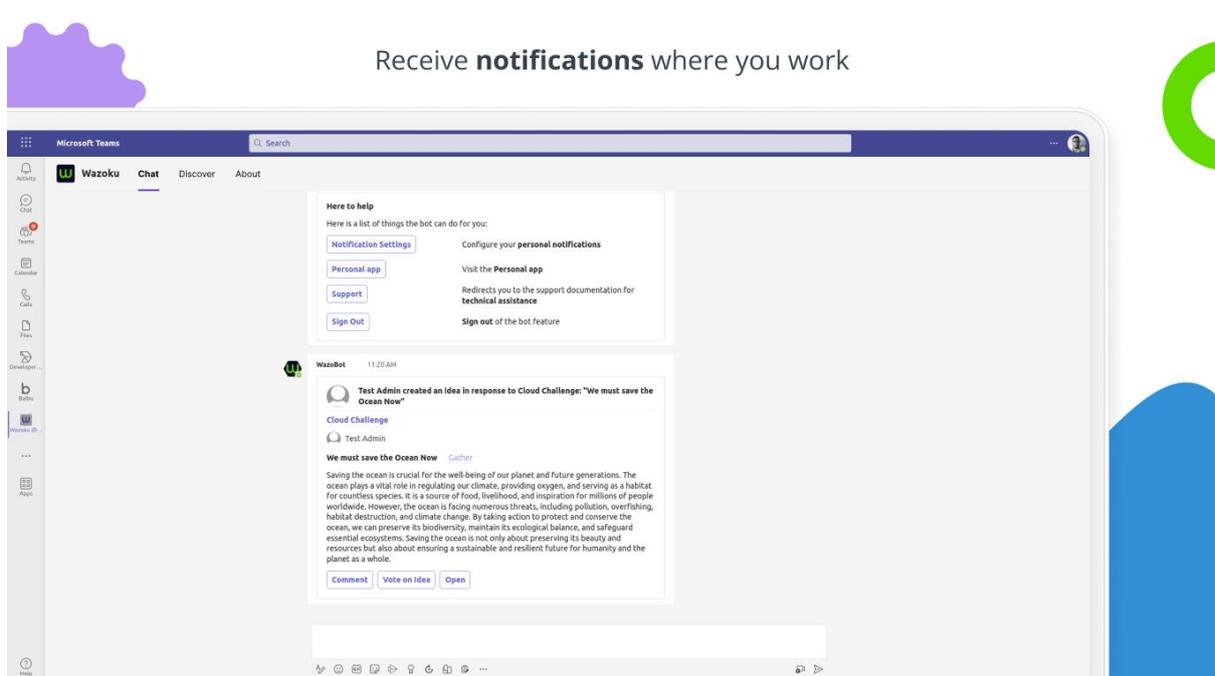


Figure 6: A design showing the WazoBot and Chat tab on Microsoft Teams.

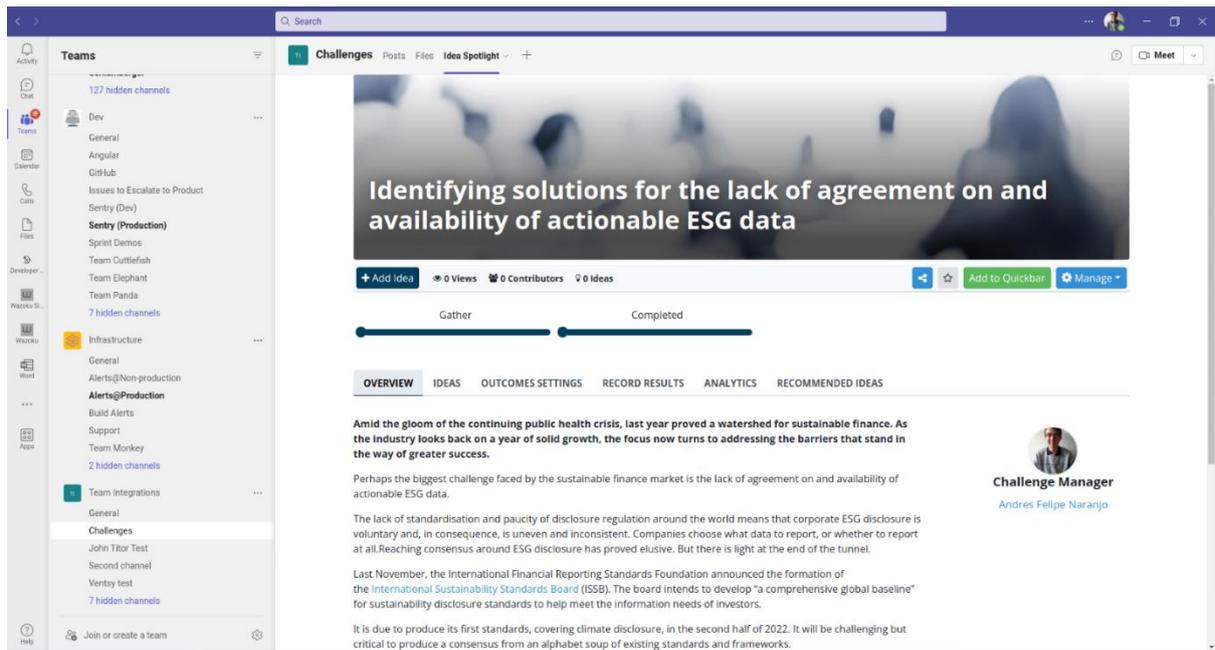


Figure 7: A design showing a Challenge within a Channel tab on Microsoft Teams.

Testing and quality assurance

In line with the testing and quality assurance practices adopted for the broader implementation of the Wazoku platform, the Microsoft Teams app for Wazoku undergoes comprehensive testing procedures, and adherence to Microsoft's standards. Detailed information on test cases, test environments, testing methodologies, and the criteria used to ensure the app's reliability, performance, and functionality can be found in the “Testing Strategy” section of the [Software Development Life Cycle \(SDLC\) document](#).

Once the testing phase is successfully completed, the app undergoes Microsoft's approval process, which involves submitting the app for review. Microsoft assesses the app's compliance with their platform's guidelines, security requirements, and user experience standards.

Development workflow

The implementation of the Microsoft Teams app for Wazoku follows the same step-by-step process that has been successful in the broader implementation of the platform. For a detailed breakdown of the deployment strategy, configuration steps, and other critical aspects, please refer to the “Development workflow” section of the [Software Development Life Cycle \(SDLC\) document](#).

For the development of the Microsoft Teams app for Wazoku, there's an additional git repository which hosts the NodeJS application for the Bot. This repository follows the same branching strategy,

development and testing strategy listed in [Software Development Life Cycle \(SDLC\) document](#) and used for the wider platform development.

Deployment process

The first step in deployment of the Microsoft Teams App for Wazoku involved publishing and deploying any changes to the Wazoku platform, Cloud-proxy server and NodeJS bot server. Steps for deploying these application is outlined in ““Software Packaging & Deployment Process” section of the [Software Development Life Cycle \(SDLC\) document](#).

The second step of the deployment process is to obtain the Microsoft’s approval for the app to be published in the store. This approval process includes:

1. **Submission:** Uploading the app to the Microsoft Teams platform for review.
2. **Review and testing:** Microsoft conducts a thorough review of the app's functionality, security, and user experience.
3. **Approval and publishing:** Upon successful review, Microsoft grants approval for the app's distribution on the Microsoft Teams platform.

It's important to note that obtaining Microsoft's approval is a crucial step to ensure the app's availability to users within the Microsoft Teams environment. The rigorous testing and adherence to Microsoft's guidelines contribute to a high-quality and reliable app experience.

For specific details about Microsoft's approval process and guidelines, please refer to [Microsoft's official documentation](#) on app submission and review.

Maintenance and support

The maintenance plan encompasses various aspects, including versioning, training and support, monitoring, bug fixes, user support and documentation.

Versioning

The Microsoft Teams app for Wazoku follows the Semantic Versioning (SemVer) framework to manage version numbers and communicate changes to the software. SemVer consists of three components: MAJOR, MINOR, and PATCH, organized as MAJOR . MINOR . PATCH. Each component signifies specific types of changes:

- **MAJOR version:** Incremented for incompatible changes that may require users to adapt their workflows or configurations. These changes include significant updates, feature rewrites, or major architectural shifts.
- **MINOR version:** Incremented for backward-compatible feature additions or enhancements. This component is used for introducing new functionality without breaking existing features.
- **PATCH version:** Incremented for backward-compatible bug fixes and minor updates that don't affect existing features.

This versioning strategy enhances communication about the scope and impact of software changes, fostering a clear understanding of each release.

Training and support

- **End-user training:** Through webinars and targeted meetings, there are training sessions and documentation provided to guide users on how to use the app effectively within Microsoft Teams.
- **Support readiness:** Through company-wide meetings like Product Council and specific approaches, there is preparation for the support team to handle user queries and issues during and after the app's launch.

Monitoring

The Microsoft Teams app for Wazoku benefits from a comprehensive set of monitoring tools to ensure real-time visibility into its performance, user interactions, and system health. These monitoring tools include:

- **Grafana Dashboards:** Grafana provides interactive and customizable dashboards that visualize critical performance metrics, system health, and user engagement data. The Grafana dashboards empower stakeholders to monitor the app's key performance indicators (KPIs), identify trends, and respond promptly to any performance anomalies.
- **Elasticsearch, Logstash, Kibana (ELK) Enabled Logs:** The app leverages ELK stack, consisting of Elasticsearch, Logstash, and Kibana, to collect, parse, index, and visualize log data generated by the app and the underlying infrastructure. With ELK enabled logs, the platform gains the ability to conduct detailed log analysis, error tracking, and debugging, significantly enhancing the app's troubleshooting capabilities.
- **Real-time Alerts:** The monitoring tools are configured to generate real-time alerts based on predefined thresholds. These alerts notify the support and development teams of any

performance degradation, system errors, or security-related events, enabling prompt incident response and resolution.

- **Performance Metrics:** Monitoring tools track essential performance metrics, including response times, latency, request rates, and resource utilization. The analysis of these metrics facilitates proactive optimization and ensures that the app consistently meets performance goals.
- **Security Auditing and Compliance:** The app's monitoring tools also support security auditing and compliance checks, allowing the platform to monitor user access patterns, detect potential security breaches, and maintain regulatory compliance.

Bug tracking

Jira is utilized as the centralized bug tracking system to efficiently record, prioritize, and manage reported issues. The integration of Jira streamlines the bug resolution process, improving collaboration among development and support teams.

Escalation procedure and bug fixing timeline

Reported issues regarding our Microsoft Teams App will be resolved as per our standard support SLAs. See table below for details:

Ticket Severity Level	Criteria	Response Time	Resolution Time
1 High	Unplanned interruption rendering the SaaS Service unavailable (no work-around available)	1 hour	24 hours
2 Med	Critical issues with the service including serious bugs or issues (affecting all users / no work-around available)	5 hours	48 hours
3 Low	Non-critical issues with the service including unexpected system responses, or functionality issues experienced (work-around available)	5 hours	14 days
4 None	System admin general questions, requests, or non-system-related client queries	24 hours	N/A

Figure 8: A table showing Wazoku's standard support SLAs.

Regression testing

To ensure the continued functionality of the application after any code revision, update or optimization, we employ a comprehensive test suite. This suite covers the several flows that encompass our core functionalities, including end-to-end testing. These suites are executed both manually and through automation scripts.

Documentation and knowledge base

The Microsoft Teams app for the Wazoku Platform is supported by well-structured documentation and a knowledge base to ensure that developers and end users have access to comprehensive resources. These include:

- **Developer Wiki:** The Developer Wiki serves as a centralized resource for the app's development team. It contains detailed technical documentation, architectural insights, coding guidelines, and best practices. The Developer Wiki assists the team in maintaining consistent development standards, fostering collaboration, and expediting new feature implementation.
- **Knowledge Base:** The Knowledge base provides users of the Wazoku Platform with comprehensive documentation that covers app functionality, frequently asked questions, and troubleshooting guides. This resource empowers users to quickly find solutions to common issues, improving their overall app experience.
- **Continuous Updates:** Both the Developer Wiki and Knowledge base are continuously updated to reflect new features, enhancements, and any changes in the app's functionality. This ensures that developers and end-users have access to the latest information.