**TECHNICAL DOCUMENTATION**

**[Plasma]**

TEAM: Authors

DATA: 04.10.2024

**History of technical documentation**

**Authors**

|  |  |  |
| --- | --- | --- |
| Fac.No | Name | Contact (e-mail) |
| 273… | Vadym | vponomarenko@tu-sofia.bg |
| 273222008 | Kiril Shyian | kshiyan@tu-sofia.bg |

**Version history**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Data | Author | Descripton |
| V1 | 10.10.2024 | Vadym | We established functional and non-func. requirements. |
| V2 | 18.10.2024 | Vadym | We changed functional, non-functional req-s. and user stories |
| V3 | 06.11.2024 | Vadym |  |
| V4 | 18.11.2024 | Kiril | Clarified certain user stories and acceptance criterias, added use case diagram |
| V5 | 21.11.2024 | Kiril | Added ER diagram and implementation diagram |
| V6 | 11.12.2024 | Vadym | Additional information added to the project |

**Content**

* **Introduction**
* **System Objectives**

*This technical documentation provides the reader with understanding of Plasma collaborative editor purpose and its stages of development.*

*The document includes definition and description of the functional and non-functional requirements, user stories, acceptance criterias for the Plasma collaborative editor application. This application is designed to facilitate real-time, seamless collaboration among users for creating, editing, and sharing text-based files or documents.*

* **System scope**

*The scope of this project encompasses the comprehensive development, deployment, and ongoing maintenance of a cutting-edge collaborative editor application. This platform is designed to enable users to create, edit, and manage documents within a shared, real-time environment, fostering enhanced teamwork and productivity. The application will cater to a broad range of use cases, including academic collaborations, professional content development, and organizational documentation needs.*

*The system’s primary goal is to provide a seamless and intuitive collaboration experience, ensuring that multiple users can work simultaneously on the same document without conflicts or interruptions. To achieve this, the project will address and implement a range of functional and non-functional requirements, user stories and their acceptance criteria. These functionalities will empower users to create and manage content with precision and efficiency.*

*In addition to core functional capabilities, the project will also prioritize adherence to critical non-functional requirements, which are integral to delivering a high-quality user experience.***Requirements specification**

1. **System requirements**
   1. **Functional requirements**

*Functional requirements specify the detailed capabilities and features a system must possess to meet the needs of its users and stakeholders. These requirements focus on the tasks the system must perform, the conditions under which it must operate, and the interactions it must support. Functional requirements serve as the foundation for system design, development, and testing, ensuring alignment with business objectives and user expectations.*

*By clearly defining functional requirements, teams can avoid misunderstandings, ensure all stakeholder needs are met, and deliver a system that is both effective and efficient.*

|  |  |  |
| --- | --- | --- |
| **No** | **Requirement** | **Priority** |
| 1 | System shall allow user to choose a project to work with. | Medium |
| 2 | System must present all files in given project. |  |
| 3 | System shall allow user to open multiple files. |  |
| 4 | System cannot be accessed without logging into an existing account. | High |
| 5 | System shall allow user to log into and existing account. | High |
| 6 | System shall allow user to create an account. |  |
| 7 | System must track and log all changes made by all users. | Medium |
| 8 | System must synchronize all changes and show current state of files to all concerned users. | High |
| 9 | Interface will allow user to choose/switch a theme (dark/white mode) |  |
| 10 | System shall allow owner to add and revoke an access to the project for other users. |  |
| 11 | System shall save all projects on local & remote environment that given user has worked on. |  |
| 12 | Open terminal inside of UI. | Low |
| 13 | System shall present user with inteface to create a new project or choose one |  |

* 1. **Non-functional requirements**

*Non-functional requirements define the quality attributes, system constraints, and operational standards that a system must meet to ensure its effectiveness, reliability, and usability. Unlike functional requirements, which specify what the system should do, non-functional requirements focus on how the system performs its functions.*

|  |  |  |
| --- | --- | --- |
| **No** | **Requirement** | **Priority** |
| 1 | 10 clients should be able to connect the system simultaneously. |  |
| 2 | All communication should involve a secure protocol. |  |
| 3 | All data must be securely stored using AES encryption. | High |
| 4 | Windows application is allowed to access the system. |  |
| 5 | Files up to 10 MB are allowed to be uploaded to the system. |  |
| 6 | Response time of the system should be up to 100 ms with 10 Mb/sec. |  |
| 7 | Uptime of the system should be 99.999%. | High |

*These requirements serve as critical guidelines during the system's design, development, testing, and deployment phases. Properly defined and documented non-functional requirements help ensure that the system meets stakeholder expectations and provides a satisfactory user experience.*

1. **Study of user requirements**

*User requirements are an important part of the development process. A proper definition of user requirements facilitates the further stages of development and saves time that is crucial during development.*

*This section consists of user stories that were used to create Plasma project and its current version.*

* 1. **User stories**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **As a** **[persona]** | **I [want to]** | **[so that]** | **Owner** | **Status** | **Estimated effort** |
| 1 | Customer | As a Customer, I want to be able to browse all files of a project inside my local environment. |  |  |  |  |
| 2 | Customer | As a Customer, I want to be able to log in to the app. |  | Kiril | Done | 5 |
| 3 | Customer | As a Customer, I want to see logs of changes to a given file made both by me and my co-workers. |  |  |  |  |
| 4 | Customer | As a Customer, I want to have dark/white mode for the interface. |  | Ansor |  |  |
| 5 | Customer | As a Customer, I want to have different roles to separate user accessibility. |  | Vadym | Done | 5 |
| 6 | Customer | As a Customer, I want to be able to open/edit/save all files of a project on both my local and remote environment. |  |  |  |  |
| 7 | Customer | As a Customer, I want to be able to open the system terminal inside the application and run the files in it. |  |  |  |  |
| 8 | Customer | As a Customer, I want to be able to chat with other Customers. |  | Boris | Done |  |
| 9 | Customer | As a Customer, I want to edit given files simultaneously with my co-workers. |  | Kiril | Testing | 20 |
| 10 | Admin | As an Admin, I want to be able to add and delete Customers to the project |  | Vadym | Done | 5 |
| 11 | Customer | As a Customer, I want to be able to create a new project and become an admin of the project |  |  |  |  |
| 12 | Customer | As a Customer, when I create a new project, I want to have templates for popular languages. |  |  |  |  |
| 13 | Customer | As a Customer, I want to be able to browse and open projects that I have previously worked on. |  |  |  |  |
| 14 | Customer | As a Customer, I want to see who else is working on given file and see his cursor position |  | Kiril | Testing | 10 |

* 1. **Acceptance Criteria for User Stories**

*The following acceptance criteria are used to determine whether User Stories were implemented correctly and satisfy the requirements that were established at the beginning of development.*

|  |  |
| --- | --- |
| **User story Number** | **Acceptance Criteria** |
| 1 | User can view all files of a project in a hierarchical or list view format.  File names, types, and sizes are displayed for each file.  User can navigate through folders and subfolders within the project.  An error message appears if any files cannot be accessed. |
| 2 | User can create account using unique and correctly formatted credentials. User cannot create account using already taken or incorrectly formatted credentials User cannot login using invalid credentials. User can successfully log in using valid credentials. User can provide either email or username in the same field. |
| 3 | User can view a changelog/history for each file.  The log includes timestamps, changes made, and the author of each change. Logs can be filtered by date, author, or change type.  Only changes made by authorized users are logged. |
| 4 | User can toggle between dark and light mode in the interface settings.  The selected mode is applied to all UI components.  The selected mode persists across sessions until changed by the user. |
| 5 | User roles (e.g., Admin, Editor, Viewer) can be assigned.  Permissions vary based on role (e.g., only Admin can edit user roles).  Users can only access features permitted by their role.  Error messages appear if a user attempts to access restricted features. |
| 6 | User can open files stored locally or on a remote server.  User can edit files and save changes to the original location (local or remote). Files opened from a remote environment are synced with the local environment.  An error message is shown if connection to the remote environment fails. |
| 7 | User can open an integrated system terminal within the application.  User can navigate the file system and run scripts directly from the terminal.  Terminal displays output and error messages of executed commands. User receives an error message if terminal access is restricted.16:25 |
| 8 | Customers can initiate real-time, multi-user chats with message status indicators, chat history, notifications, and encrypted messages. |
| 9 | Multiple users can edit the same file at the same time without any loss of data or overwriting of each other's changes. Changes made by one user are visible to all collaborators in real-time Changes must persist even after user leaves webpages |
| 10 | Project admins can add or remove customers with notifications sent to all members and immediate access revocation for deleted customers. |
| 11 | Customers can create a new project where they are assigned admin rights, access project settings, and see the project on their dashboard. |
| 12 | Customers can select from language-specific templates when creating a project, which populates the project with relevant files that are editable. |
| 13 | The system must show the real-time position of each collaborator’s cursor within the document. As a user moves their cursor or makes changes to the document, all other users should see the updates in real time The system must dynamically update the position of the cursor across all users' screens without requiring a page refresh. The system should support showing additional details (like name) if necessary, for the collaborating user. |

**Conceptual system design**

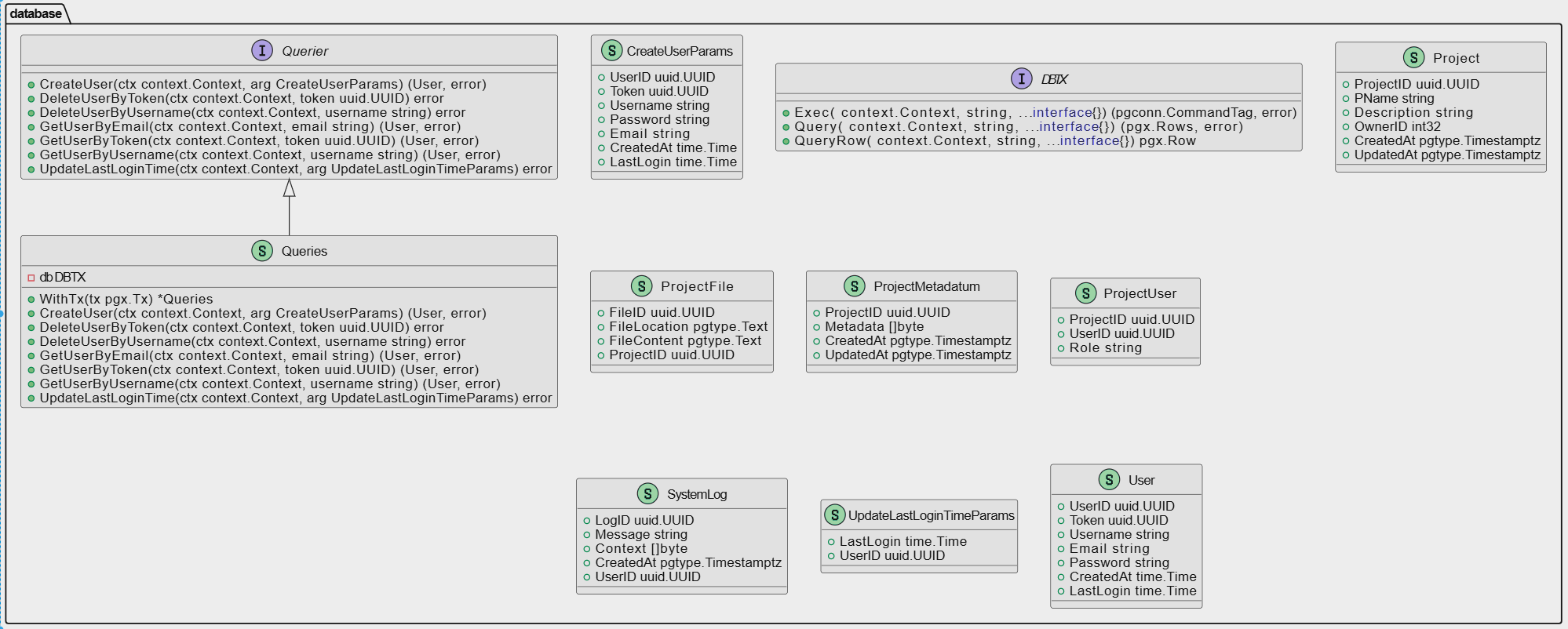
* **Use Case Diagram**

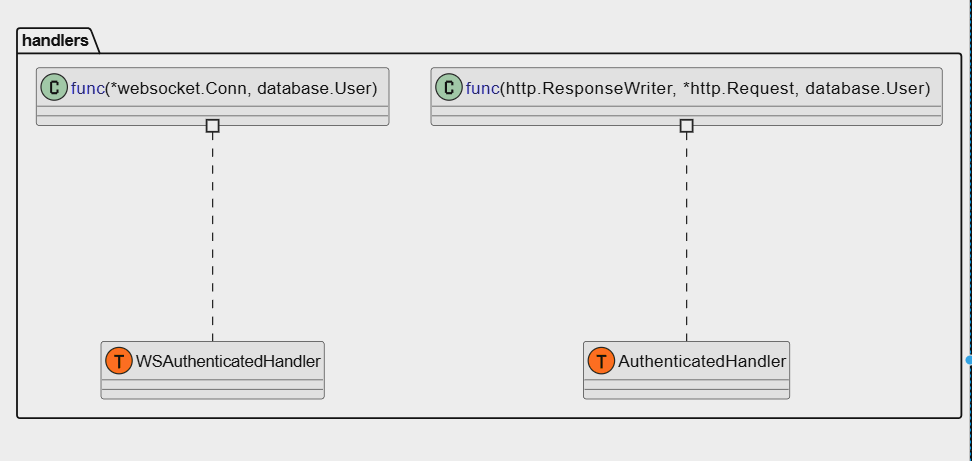
*The following diagram shows user functionalities. These functionalities are essential for convenient collaboration and communication between users. Each user has opportunity to:*

**A diagram of a project

Description automatically generated**

* **Software architecture**

**



* **Conceptual Data Base model**

*The conceptual data base model consists of 6 tables that are directly interconnected between each other. Each table was designed with a specific purpose.*

*The “****users****” table stores sensitive information, namely personal data of each user registered in Plasma System.*

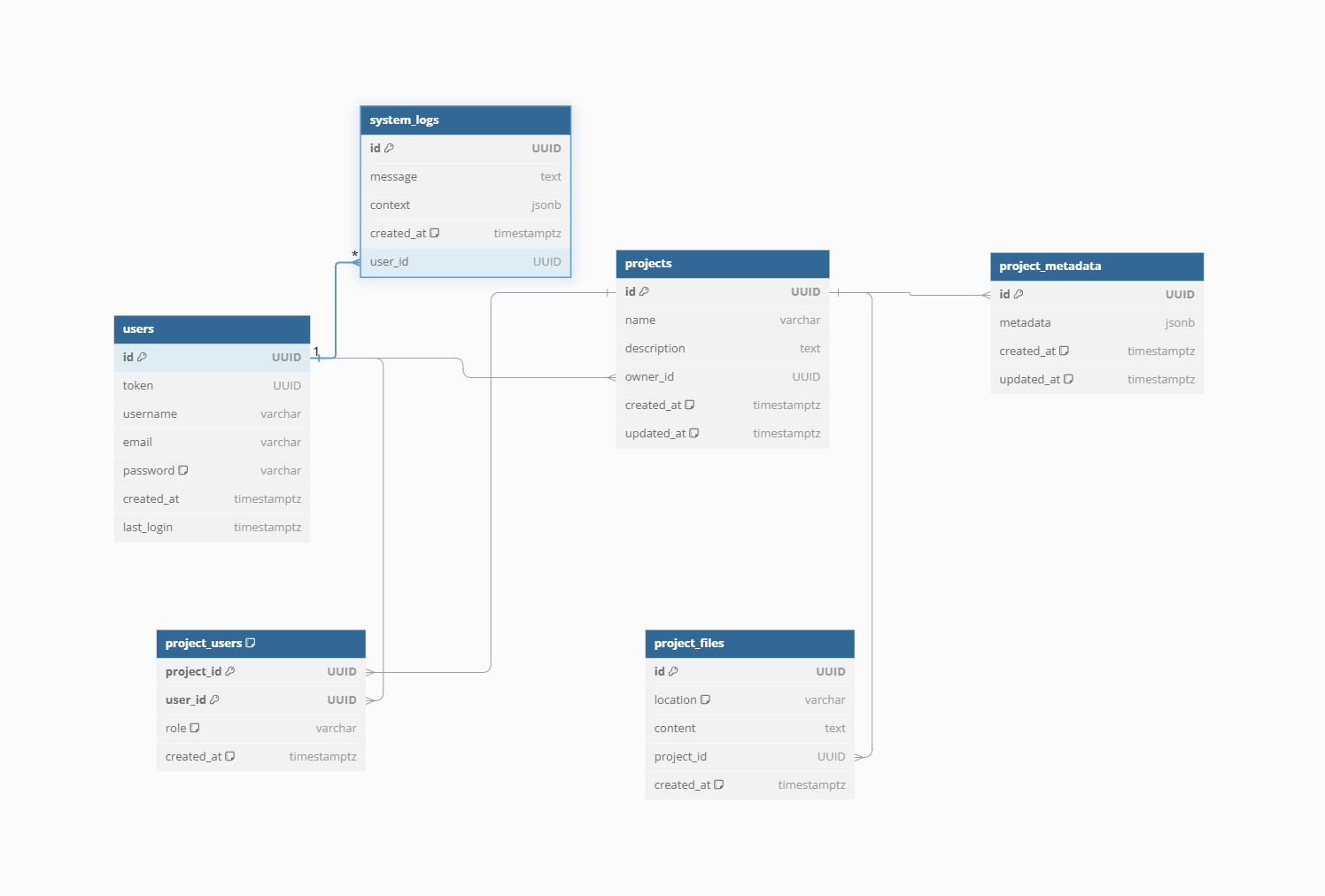
*The “****project\_users****” stores projects and users that are connected to those projects.*

*The “****system\_logs****” table was created to trace all changes / errors / issues that could take place during the actual working processes.*

*The “****projects****” table stores all the information about each project created by users.*

*The “****projects\_metadata****” is used to store additional information about each project that cannot be stored at the “****projects****” table.*

*The “****projects\_files****” stores the important information about the files that are related to the project.*

**

* **Resorces**

*The Plasma development team were using these resources to receive comprehensive understanding of fundamental knowledge such as Golang programming language, Software Architecture Design and Requirements Engineering.*

**Acronyms**

|  |  |
| --- | --- |
| **Acronym** | **Description** |
| **Golang Course** | **Go Programming – Golang Course with Bonus Projects |** [**Link**](https://www.youtube.com/watch?v=un6ZyFkqFKo&t=27169s) |
| **Software Architecture** | **Software Architects design solutions for complex back office enterprise applications by identifying the basic abstractions. |** [**Link**](https://www.youtube.com/watch?v=mCM6QVHD08c) |
| **Software Architecture Diagrams** | **This video presents how to build software architecture diagrams using C4 model and Structurizr tool. |** [**Link**](https://www.youtube.com/watch?v=_I0U1sZ9RJ8&t=1234s) |
|  |  |

**Others**

* [What Are Functional Requirements? Types and Examples - WINaTALENT Blog](https://winatalent.com/blog/what-are-functional-requirements-types-and-examples/)
* [A Guide to Functional Requirements (with Examples)](https://www.nuclino.com/articles/functional-requirements)
* [Nonfunctional Requirements: Examples, Types and Approaches](https://www.altexsoft.com/blog/non-functional-requirements/)