|  |  |
| --- | --- |
| Project name | Go notes |

|  |  |  |
| --- | --- | --- |
| Project author | | |
| № | Names of Participants | FN |
| 1 | Kaloyan Mehandzhiyski | 5MI0700025 |

|  |
| --- |
| 1. Short project description (Business needs and system features) |
| Every busy student from FMI needs to keep some notes because they can forget them – dates for exams, themes for projects, conspectus, etc. Let’s introduce Go Notes! The basic idea for the user is to write down and keep notes on a computer so they don’t forget something or let their PC remember for them. Why use a computer instead of paper? Because it’s an eco-friendly way, reducing a lot of paper, and because there will be a challenge for the developers to do it! The implementation of this application uses Golang, JSON, HTML, and Chrome/Firefox browser for its interface. It will implement a web-based front-end client using Golang’s html/template package, HTTP queries, and local JSON files for a database. There will be one main JSON database file containing all registered users with their passwords. Every user has a unique username and password. The username will be a unique key for the database, and there are no restrictions for the password for the purpose of the app. The application only supports registering a new user (so the user cannot change their username or password or delete their account). The username and password are strings. After logging in, a user can create their own notes. Each note has a unique name and content, both of which are strings. A user can see all their notes and content, add new notes, delete old notes, and change the content of notes. • Anonymous User – can only view the information page, register, or log in to the system. • Logged-in User – can view, create, edit, or delete their own notes. |
|  |

|  |  |  |
| --- | --- | --- |
| 1. Main Use Cases / Scenarios | | |
| **Use case name** | **Brief Descriptions** | **Actors Involved** |
| * 1. **Register** | Anonymous User can register in the system by providing a unique username and choosing a password. If the username doesn’t exist in the database, the new user will be registered successfully.  Actors Involved: Anonymous User | *Anonymous User* |
| * 1. **Login** | User can try to log in with a username and password. If they are correct, the user will be logged in to the system and can see/add/edit/delete their own notes. Otherwise, if the username and password don’t match any registered user, the Anonymous User can quit the application or try to register a new profile. | *Anonymous User, Registered User* |
| * 1. **Change content** | *User who is successfully logged in can view/add/delete/change their notes.* | *Registered User* |

|  |  |  |
| --- | --- | --- |
| 1. Main Views (Frontend) | | |
| **View name** | **Brief Descriptions** | **URI** |
| * 1. **Home** | Presents the introductory information for the purpose of the system as well as detailed instructions on how to start using it. Prominently offers the ability to register or login. | / |
| * 1. **User Registration** | Presents a view allowing the Anonymous Users to register. | */register* |
| * 1. **Login** | Presents a view allowing the users to login. | */login* |
| * 1. **User Data** | *Provides the ability to view and edit personal notes.* | */notes*  */edit\_note* |

|  |  |  |
| --- | --- | --- |
| 1. API Resources (Backend) | | |
| **View name** | **Brief Descriptions** | **URI** |
| * 1. **Users** | GET user data for all registered users. Only accessible by administrators. | */api/users* |
| * 1. Register | POST register a new user by providing a unique username and password. | */api/register* |
| * 1. **Login** | POST login with a username and password. If successful, the user gains access to their notes. | */api/login* |
| * 1. **Logout** | POST log out the current user, clearing the session and any associated cookies. | */api/logout* |
| * 1. **Notes** | GET, POST retrieve all notes for the logged-in user. Users can also create, edit, or delete their notes. | */api/notes* |
| * 1. Edit Note | POST edit an existing note for the logged-in user. | */api/edit* |
| * 1. Delete Note | POST delete a specified note for the logged-in user. | */api/delete* |