

|  |
| --- |
| **Project**  **GitHub** |

|  |  |  |
| --- | --- | --- |
| **Student Number** | **Student Name** | **N** |
| **442051268** | **Homoud aldosar** | **1** |
| **442050179** | **Saad alduhimi** | **2** |
| **442051101** | **Nasser alhumihim** | **3** |

**Supervised by: Dr.Mohammed alassiri**

**Year: 2023**

1. **Feasibility Study &Project Proposal:**
2. **Introduction:** The primary objective of our Software Engineering Project Applying course concepts and implementing them on a GitHub website , researching and looking at the details of the website, and understanding how it works.
3. **Problem:** software developers faced challenges in collaborating on projects and managing version control. There was no centralized platform for code hosting and collaboration, leading to difficulties in tracking changes, coordinating efforts, and maintaining code integrity.
4. **Background:** GitHub is a web-based platform that provides a centralized location for version control and collaboration on software development projects. It allows developers to store, manage, and share their code repositories and collaborate with other team members on software projects.
5. **Proposed solution:** Provide a web-based platform that enables developers to store their code on a central server manage and share their code repositories and collaborate with other team members on software projects.
6. **Work plan:** Identification of system requirements and scope, design of system structure, selection of technologies, development environment, control of versions, design, implementation of database chart, development of user management characteristics (registration, documentation and authorization), building of repository management functions (establishment, reproduction and deletion of warehouses), and many other complex matters.
7. **Project requirements**
8. Functional requirements **(FR):**

|  |  |  |
| --- | --- | --- |
| **N.** | Functional | Description |
| **1** | User Registration | * The system should allow users to create an account by providing information such as username, email address, and password. * The system should validate the uniqueness of the email address and username. * Users should receive a confirmation email to verify their account. |
| **1** | **Login** | * The system allows for users to log into their account by entering their email and password. * The system should authenticate the user's credentials and grant access upon successful login. * Users should have the option to stay logged in with a "Remember Me" feature. |
| **2** | **Search** | The system allows the user to a search box may appear for him and he will type about which he wants to obtain information. |
| **3** | **Create and manage repositories** | * Users should be able to create and manage repositories, which are collections of code files. * The website should provide options to set repository visibility (public or private) and manage access permissions for collaborators. * Users should be able to delete repositories if they are no longer needed. |
| **4** | **Commit and push changes to repositories.** | Users should be able to commit and push changes to repositories, which allows other users to view and collaborate on the code. |
| **5** | **Pull Requests** | * Users should be able to create and manage pull requests to propose changes to a repository. * Pull requests should include information such as source and target branches, description, and reviewers. |
| **6** | **Profiles Setting** | The system allows for users to customize and manage your personal information, security settings, email notifications, and other preferences related to account. |
| **7** | **GitHub CLI** | CLI is an open source tool for using GitHub from your computer's command line. When you're working from the command line, you can use the GitHub CLI to save time and avoid switching context. |
| **8** | **Fork** | A fork is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project. |

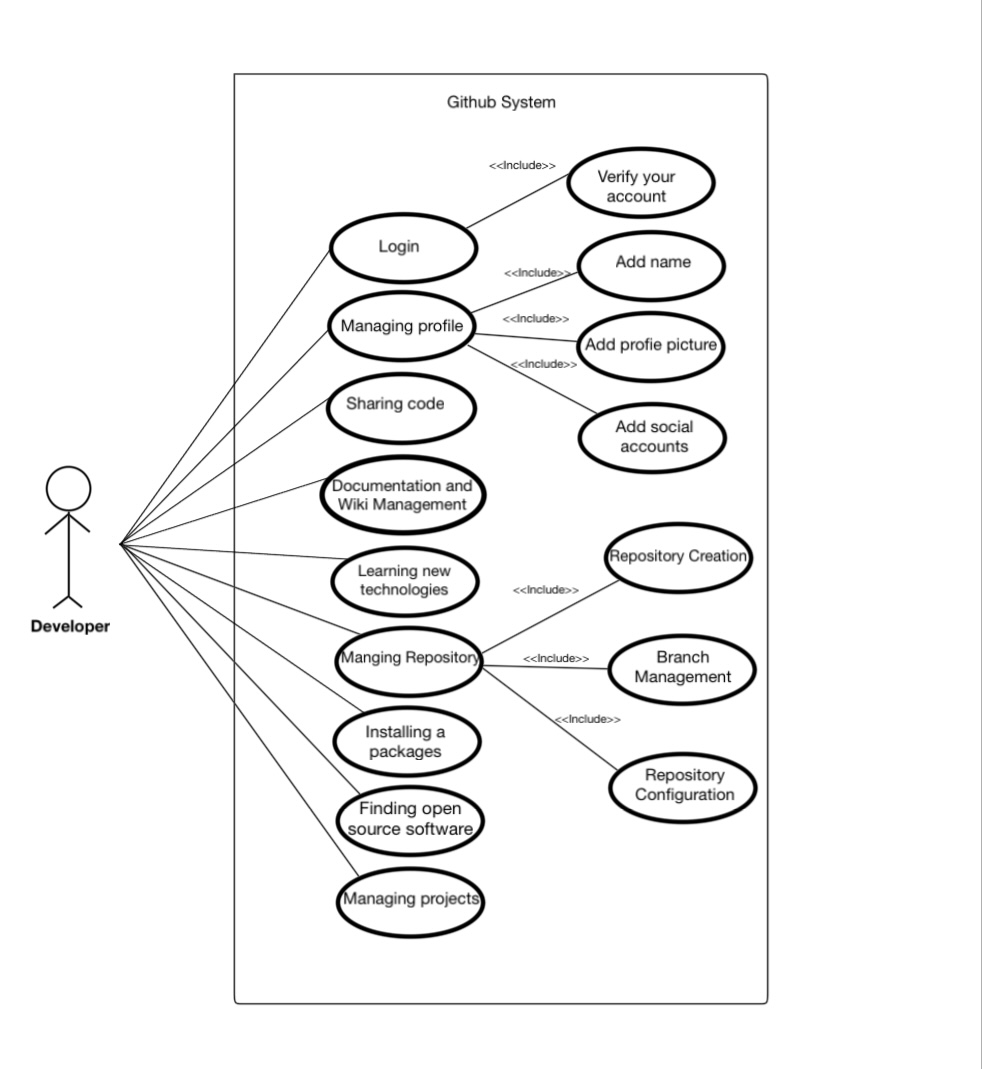
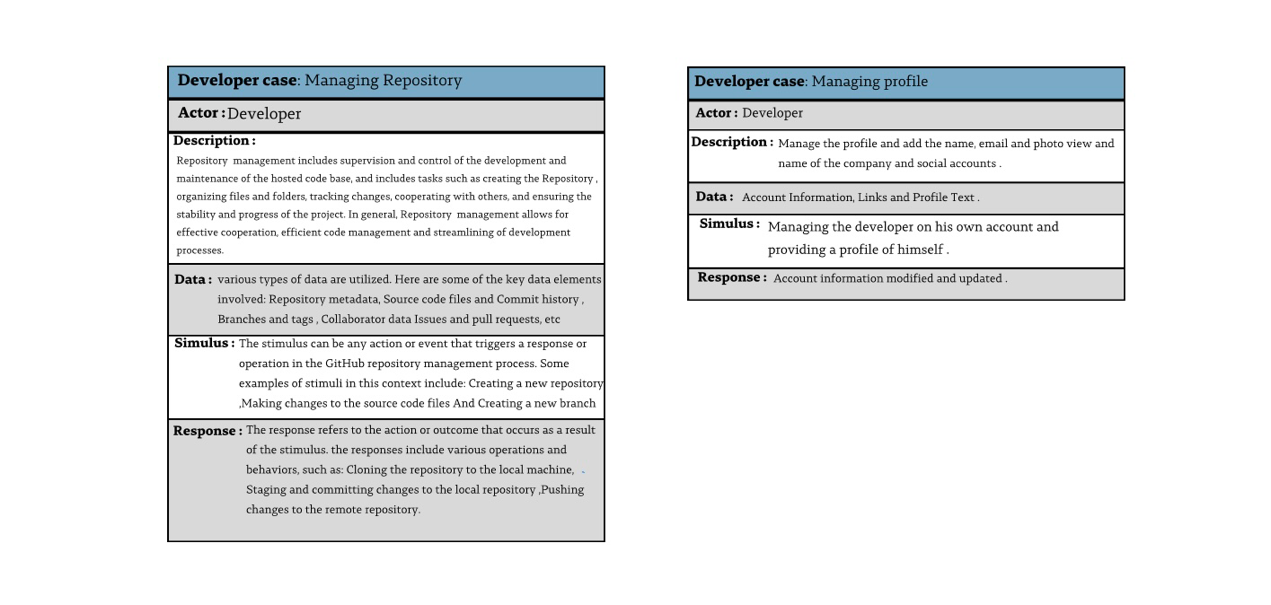
1. Non-Functional requirements **(NFR):**

|  |  |  |
| --- | --- | --- |
| **N**. | Non-Functional | Description |
| **1** | Performance | The system should be responsive and scalable to handle a large number of users and repositories. |
| **2** | Availability | The system must available 24/7. |
| **3** | Ease of use | The system should be easy to use and understand. |
| **4** | Security | The system should be secure and protect users' data from unauthorized access. |

1. **Activity diagram:**

صورة تحتوي على نص, رسم بياني, رسم, رسم تقني

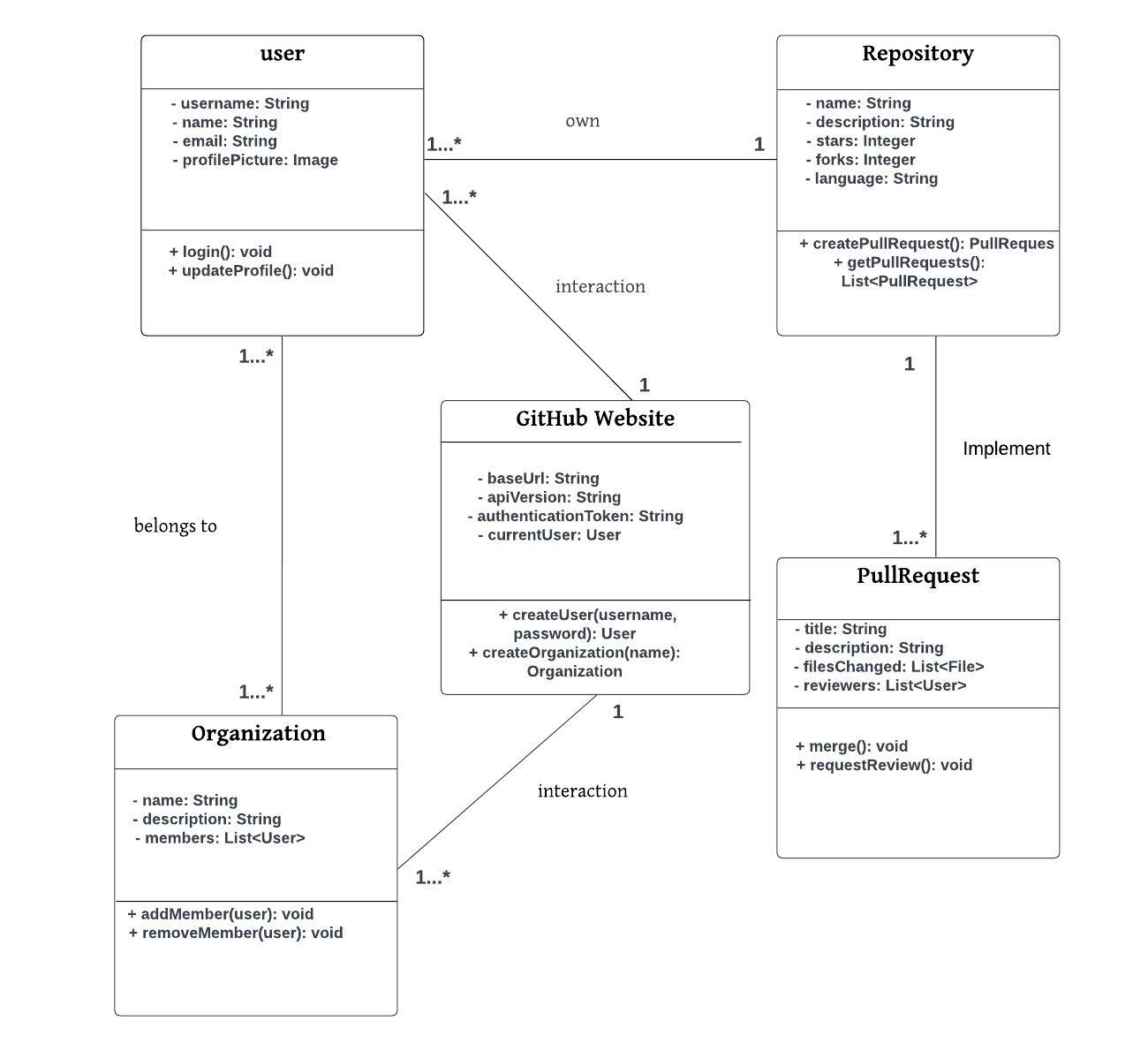
تم إنشاء الوصف تلقائياً

1. **Project Use Case Modeling:**
2. **Creating Sequence Diagrams:**

صورة تحتوي على نص, لقطة شاشة, رسم بياني, الخط

تم إنشاء الوصف تلقائياً

1. **Creating a Class Diagram:**

****