**IssueTracker**

**Definitions**

**Description:**

IssueTracker is an enterprise solution that aims to facilitate the process of tracking feature development and discovered bugs throughout the lifetime of a software product

**Main features:**

Users can create projects and add collaborators to them

Roles can be defined on a set of existing permissions and attributed to users in a project

If given permission, a user should be able to add new features or report bugs to the project

Users should be able to be assigned to different tasks, bugs and features

They should also be notified about changes to issues they are watching

Issues should provide ways to comment to portrait progress on them

**Typical users:**

Team Leaders

Technical support

Developers

**Analysis**

**Personas:**

**Vlad** is a Team Leader on many projects as he works in a well-known corporation. He will typically start a new project, invite all the required members that are assigned on that particular project, and start assigning tasks to some of them. He will also create some unassigned tasks that any suited member can pick up, he should be notified about all issues by default unless he decides not to follow some of them for some reason.

**Cornelia** is working for the technical support department in the same corporation. Once invited by Vlad, she should be able to report new bugs that are reported by clients through a different platform. Her interaction with the platform sums up to this, as she is not concerned with the overall development, assigned software tasks or even marking bugs as solved on the platform. She should, however, be notified about updates to the bugs she reported, unless she explicitly chooses to stop watching some of them.

**Ioana** is a software developer working for the same corporation and she will get assigned to some of the tasks, issues or bugs reported by the rest of the team. She might also assign herself to some of them, and she should be able to change the status to the issues she was assigned to. Based on the granted permissions, she might also be able to mark solved issues as closed. However, this is not the default, as Vlad will have to explicitly give her such permissions.

**User stories:**

**Administer Project**

As a project owner, I want to be able to administer my projects by inviting teammates and assigning issues to them.

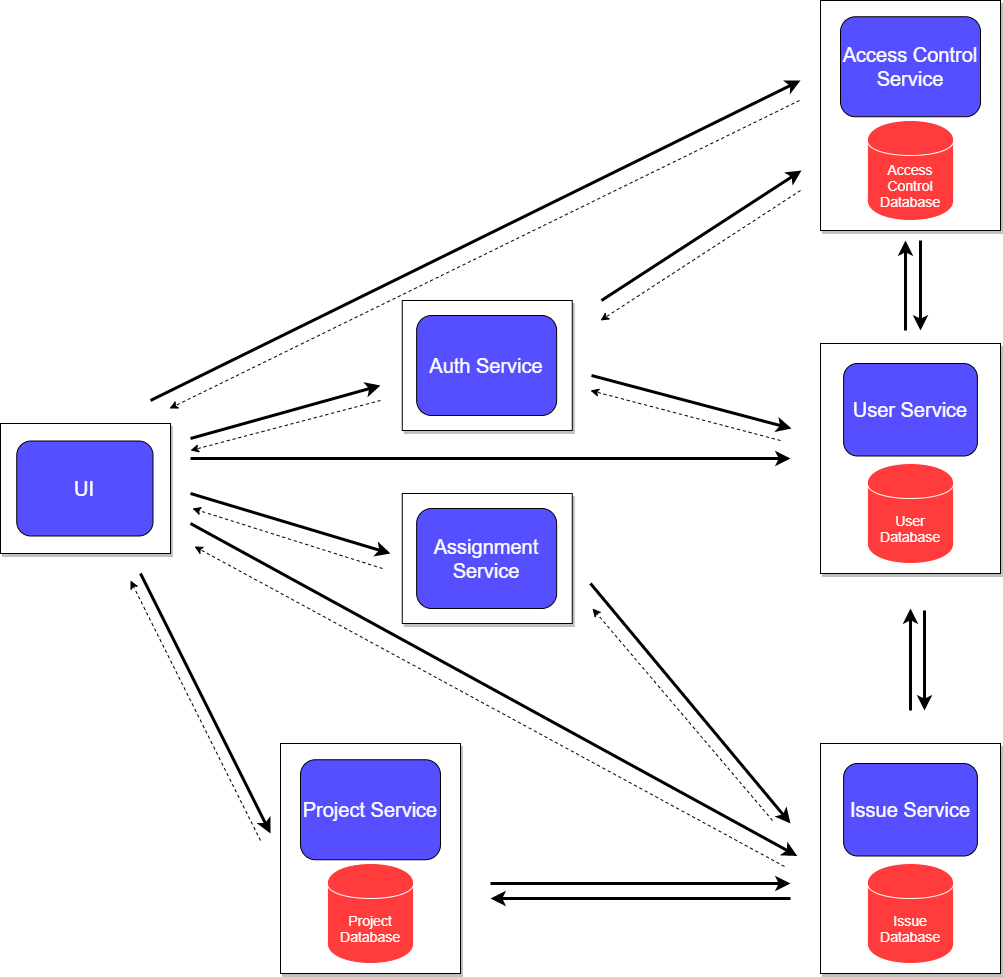
**Report Bugs**

As a technical support, I want to report new bugs and be notified about any updates on them by default, including comments, replies from other teammates or status change.

**Issue Assignment**

As a developer, I want to assign myself to issues and change status on issues I’m assigned to mark them as in progress or completed.

**Microservices design**



**REST APIs design**

|  |  |  |
| --- | --- | --- |
| Service name | Collections | Methods |
| Access Control | Roles, Permissions | getRoleById, getAllRoles, getRolesByProjectId, getRolesByGrantedPermissions, addRole, updateRole, removeRole, getAllPermissions, getPermissionById |
| User | Users | getById, addUser, updateUser |
| Issue | Issues | getById, getIssuesByProjectId, getIssuesByUserId, add, update, remove |
| Project | Projects | getById, add, update, remove, addUserToProject, removeUserFromProject,  assignRoleToUserOnProject,  removeRoleFromUserOnProject,  getUserIdsByProjectId,  getUserIdsByRoleOnProject,  getProjectsByUserId |
|  |  |  |
|  |  |  |
|  |  |  |

Comentariu: avem acces la un proiect si vrem sa vedem rolurile definite, deci facem request la **AccessControl** pentru **getRolesByProjectId** (stim déjà project id-ul). Avand rolurile, deci si id-urile lor, putem vedea, pentru fiecare rol in parte, ce useri au acest rol pe proiect facand request la **Project** pentru **getUserIdsByRoleId**.