Documentation

Content

TOPIC.......................................................................................................……...3

STEP OF REALIZATION..................................................................................3

USING TECHNOLOGY………………………………..................................... 3

AIM………………............................................................................................ 4

CODE…………………………………………………………………………..4

TOPIC

The topic of the project is to create an application for managing the operations of an IT company.

STEP OF REALIZATION

**Planning**

Planning is one of the hardest parts. Initially, I divided my work by starting with the choice of the way of saving information, and then I divided my work on the code into parts. I started with the features for users, then teams, projects, and tasks.

**Realization**

The project was implemented using a program written in the C++ programming language. When problems arose, I consulted the help of the Internet.

**Testing**

Code testing is done by manual checking with multiple options that can go wrong. The bugs I was able to find have been fixed.

USING TECHNOLOGY

The technologies that were used to create this project are:

I used Visual Studio 2019 to create program written in the C++ programming language

I used PowerPoint to create my presentation

I used Microsoft Word to create my documentation

AIM

The goal of the project is to make a program for managing the operations of an IT company, thus making the work of this company more orderly and easier for people to become familiar with.

Code

Chart

Description automatically generated

|  |  |  |  |
| --- | --- | --- | --- |
| **Type of Function** | **Name of function** | **Parameters** | **Description** |
| vector<pm::type::User> | getAll() |  | Get all information about users from file. |
| vector<pm::type::User> | create() | vector<pm::type::User> users, size\_t idOfCreator, bool registerOrNot | Create new user and add inforamtion about him\her into file |
| vector<pm::type::User> | remove() | vector<pm::type::User> users, size\_t idOfUser | Remove user from file. After that get all information from file again. |
| vector<pm::type::User> | update() | std::vector<pm::type::User> users, size\_t userIndex, size\_t idOfUserChange | Update information about user and after that get all information from file again. |
| pm::type::User | getById() | std::vector<pm::type::User> user, size\_t idOfUser | Get user with given id. |
| size\_t | getByEmail() | std::vector<pm::type::User> user, std::string email | Get user with given email. |
| bool | checkId() | std::vector<pm::type::User> user, size\_t idOfUser | Check if there is user with given id. |
| void | displayUsers() | std::vector<pm::type::User> users | Display all users. |
| vector<pm::type::Team> | getAll() |  | Get all information about teams from file. |
| vector<pm::type::Team> | create() | vector<pm::type::Team> teams, pm::type::User currentUser | Create new team and add information about team into file |
| vector<pm::type::Team> | remove() | vector<pm::type::Team> teams, size\_t idTeam | Remove team from file from file and after that get all information from file again |
| vector<pm::type::Team> | update() | vector<pm::type::Team> teams, size\_t indexOfTeam, size\_t currentUserId | Update information about team and after that get all information from file again |
| size\_t | getById() | vector<pm::type::Team> teams, size\_t idTeam | Get team with given id. |
| size\_t | getByTitle() | vector<pm::type::Team> teams, string titleOfTeam | Get team with given title. |
| void | displayTeams() | vector<pm::type::Team> teams | Display all teams. |
| bool | checkForAdded() | pm::type::Team team, size\_t idOfUser | Check if user is added to given team. |
| bool | checkId() | vector<pm::type::Team> teams, size\_t idTeam | Check if there is team with given id. |
| vector<pm::type::Team> | asignToTeam() | vector< pm::type::Team> teams, size\_t indexOfTeam, size\_t idOfUser | Asign user to given team. |
| vector<pm::type::Project> | getAll() |  | Get all information about projects from file. |
| pm::type::Project | create() | vector<pm::type::Project> projects, size\_t idOfCreator | Create new project and add information about project file. |
| vector<pm::type::Project> | remove() | vector<pm::type::Project> projects, size\_t idOfUser, size\_t idOfProject | Remove information about project from file and after that get all infromation from file again. |
| void | update() | vector<pm::type::Project>& projects, size\_t idOfUser, size\_t idOfProject | Update information about project from file and after that get all infromation from file again. |
| void | displayProjects() | vector<pm::type::Project> projects, pm::type::User currentUser, vector<pm::type::Team> teams, bool ownOrNot | Display all projects. |
| void | displayProject() | pm::type::Project project, vector<pm::type::User> users | Display specific project. |
| void | asignToTeam() | vector<pm::type::Project>& projects, size\_t indexOfProject, size\_t indexOfTeam, vector<pm::type::Team> teams | Assign team to given project. |
| bool | checkForAdded() | pm::type::Project project, size\_t idOfTeam | Check if given team is assigned to given project. |
| bool | checkId() | vector<pm::type::Project> projects, size\_t idOfProject | Check if there is project with given id. |
| size\_t | getById() | vector<pm::type::Project> projects, size\_t idOfProject | Get project with given id. |
| size\_t | getByTitle() | vector<pm::type::Project> projects, string titleOfteam | Get project with given title. |
| vector<pm::type::Task> | getAll() |  | Get all task from file. |
| pm::type::Task | create() | std::vector<pm::type::Task> tasks, size\_t idOfCreator, std::vector<pm::type::User> users, std::vector<pm::type::Project> projects, std::vector<pm::type::Team> teams, pm::type::User currentUser | Create new task and assign to user. |
| void | remove() | std::vector<pm::type::Task>& tasks, size\_t taskId, size\_t userId | Remove task from file and after that get all task from file again. |
| void | update() | std::vector<pm::type::Task>& tasks, pm::type::Task& task, size\_t idOfUser, pm::type::Project project, std::vector<pm::type::Team> teams, std::vector<pm::type::User> users | Update task and file and after that get all task from file again. |
| void | displayCreated() | std::vector<pm::type::Task> tasks, pm::type::User currentUser, std::vector<pm::type::Project> projects, std::vector<pm::type::User> users, std::vector<pm::type::Team> teams | Display all tasks for given project which current user has created. |
| void | displayAllProjectsTasks() | std::vector<pm::type::Task> tasks, pm::type::User currentUser, std::vector<pm::type::Project> projects, std::vector<pm::type::User> users, std::vector<pm::type::Team> teams | Display all tasks . |
| void | displayProjectTasks() | size\_t idOfProject, std::vector<pm::type::Task> tasks, pm::type::User currentUser, std::vector<pm::type::Project> projects, | Display all tasks for given project. |
| size\_t | getById() | std::vector<pm::type::Task> tasks, size\_t idOfProject | Get task with given id. |
| bool | checkId() | std::vector<pm::type::Task> tasks, size\_t idOfProject | Check if there is a task with given id. |
| void | startMenu() |  | Start menu |
| void | managementMenuAdmin() | vector<pm::type::User>& users, pm::type::User& currentUser, vector<pm::type::Team>& teams | Menu with option only for admin users |
| void | managementMenuUser() | vector<pm::type::Project>& projects, pm::type::User& currentUser, vector<pm::type::Team>& teams, vector<pm::type::User>& users, vector<pm::type::Task>& tasks | Menu with options only for users |
| void | menuUserProject() | vector<pm::type::Project>& projects, pm::type::User& currentUser, vector<pm::type::Team>& teams, vector<pm::type::User>& users | Menu for options for projects only for users |
| void | menuUserTasks() | vector<pm::type::Task> tasks, pm::type::User currentUser, vector<pm::type::Project>& projects, vector<pm::type::User>& users, vector<pm::type::Team>& teams | Menu for options for tasks only for users |
| void | menuAdminTeam() | vector<pm::type::Team>& teams, vector<pm::type::User>& users, pm::type::User& currentUser | Menu for options for teams only for admin users |
| void | menuAdminUser() | vector<pm::type::User>& users, pm::type::User& currentUser | Menu for options for users only for admin users |

|  |  |  |  |
| --- | --- | --- | --- |
| **Type of Function** | **Name of function** | **Parameters** | **Description** |
| vector<pm::type::User> | getAll() |  | Get all information about users from file. |
| vector<pm::type::User> | create() | vector<pm::type::User> users, size\_t idOfCreator, bool registerOrNot | Create new user and add inforamtion about him\her into file |
| vector<pm::type::User> | remove() | vector<pm::type::User> users, size\_t idOfUser | Remove user from file. After that get all information from file again. |
| vector<pm::type::User> | update() | std::vector<pm::type::User> users, size\_t userIndex, size\_t idOfUserChange | Update information about user and after that get all information from file again. |
| pm::type::User | getById() | std::vector<pm::type::User> user, size\_t idOfUser | Get user with given id. |
| size\_t | getByEmail() | std::vector<pm::type::User> user, std::string email | Get user with given email. |
| bool | checkId() | std::vector<pm::type::User> user, size\_t idOfUser | Check if there is user with given id. |
| void | displayUsers() | std::vector<pm::type::User> users | Display all users. |
| vector<pm::type::Team> | getAll() |  | Get all information about teams from file. |
| vector<pm::type::Team> | create() | vector<pm::type::Team> teams, pm::type::User currentUser | Create new team and add information about team into file |
| vector<pm::type::Team> | remove() | vector<pm::type::Team> teams, size\_t idTeam | Remove team from file from file and after that get all information from file again |
| vector<pm::type::Team> | update() | vector<pm::type::Team> teams, size\_t indexOfTeam, size\_t currentUserId | Update information about team and after that get all information from file again |
| size\_t | getById() | vector<pm::type::Team> teams, size\_t idTeam | Get team with given id. |
| size\_t | getByTitle() | vector<pm::type::Team> teams, string titleOfTeam | Get team with given title. |
| void | displayTeams() | vector<pm::type::Team> teams | Display all teams. |
| bool | checkForAdded() | pm::type::Team team, size\_t idOfUser | Check if user is added to given team. |
| bool | checkId() | vector<pm::type::Team> teams, size\_t idTeam | Check if there is team with given id. |
| vector<pm::type::Team> | asignToTeam() | vector< pm::type::Team> teams, size\_t indexOfTeam, size\_t idOfUser | Asign user to given team. |
| vector<pm::type::Project> | getAll() |  | Get all information about projects from file. |
| pm::type::Project | create() | vector<pm::type::Project> projects, size\_t idOfCreator | Create new project and add information about project file. |
| vector<pm::type::Project> | remove() | vector<pm::type::Project> projects, size\_t idOfUser, size\_t idOfProject | Remove information about project from file and after that get all infromation from file again. |
| void | update() | vector<pm::type::Project>& projects, size\_t idOfUser, size\_t idOfProject | Update information about project from file and after that get all infromation from file again. |
| void | displayProjects() | vector<pm::type::Project> projects, pm::type::User currentUser, vector<pm::type::Team> teams, bool ownOrNot | Display all projects. |
| void | displayProject() | pm::type::Project project, vector<pm::type::User> users | Display specific project. |
| void | asignToTeam() | vector<pm::type::Project>& projects, size\_t indexOfProject, size\_t indexOfTeam, vector<pm::type::Team> teams | Assign team to given project. |
| bool | checkForAdded() | pm::type::Project project, size\_t idOfTeam | Check if given team is assigned to given project. |
| bool | checkId() | vector<pm::type::Project> projects, size\_t idOfProject | Check if there is project with given id. |
| size\_t | getById() | vector<pm::type::Project> projects, size\_t idOfProject | Get project with given id. |
| size\_t | getByTitle() | vector<pm::type::Project> projects, string titleOfteam | Get project with given title. |
| vector<pm::type::Task> | getAll() |  | Get all task from file. |
| pm::type::Task | create() | std::vector<pm::type::Task> tasks, size\_t idOfCreator, std::vector<pm::type::User> users, std::vector<pm::type::Project> projects, std::vector<pm::type::Team> teams, pm::type::User currentUser | Create new task and assign to user. |
| void | remove() | std::vector<pm::type::Task>& tasks, size\_t taskId, size\_t userId | Remove task from file and after that get all task from file again. |
| void | update() | std::vector<pm::type::Task>& tasks, pm::type::Task& task, size\_t idOfUser, pm::type::Project project, std::vector<pm::type::Team> teams, std::vector<pm::type::User> users | Update task and file and after that get all task from file again. |
| void | displayCreated() | std::vector<pm::type::Task> tasks, pm::type::User currentUser, std::vector<pm::type::Project> projects, std::vector<pm::type::User> users, std::vector<pm::type::Team> teams | Display all tasks for given project which current user has created. |
| void | displayAllProjectsTasks() | std::vector<pm::type::Task> tasks, pm::type::User currentUser, std::vector<pm::type::Project> projects, std::vector<pm::type::User> users, std::vector<pm::type::Team> teams | Display all tasks . |
| void | displayProjectTasks() | size\_t idOfProject, std::vector<pm::type::Task> tasks, pm::type::User currentUser, std::vector<pm::type::Project> projects, | Display all tasks for given project. |
| size\_t | getById() | std::vector<pm::type::Task> tasks, size\_t idOfProject | Get task with given id. |
| bool | checkId() | std::vector<pm::type::Task> tasks, size\_t idOfProject | Check if there is a task with given id. |
| void | startMenu() |  | Start menu |
| void | managementMenuAdmin() | vector<pm::type::User>& users, pm::type::User& currentUser, vector<pm::type::Team>& teams | Menu with option only for admin users |
| void | managementMenuUser() | vector<pm::type::Project>& projects, pm::type::User& currentUser, vector<pm::type::Team>& teams, vector<pm::type::User>& users, vector<pm::type::Task>& tasks | Menu with options only for users |
| void | menuUserProject() | vector<pm::type::Project>& projects, pm::type::User& currentUser, vector<pm::type::Team>& teams, vector<pm::type::User>& users | Menu for options for projects only for users |
| void | menuUserTasks() | vector<pm::type::Task> tasks, pm::type::User currentUser, vector<pm::type::Project>& projects, vector<pm::type::User>& users, vector<pm::type::Team>& teams | Menu for options for tasks only for users |
| void | menuAdminTeam() | vector<pm::type::Team>& teams, vector<pm::type::User>& users, pm::type::User& currentUser | Menu for options for teams only for admin users |
| void | menuAdminUser() | vector<pm::type::User>& users, pm::type::User& currentUser | Menu for options for users only for admin users |