Cloud

T-CLO-900

To Do List

Software Architecture Specifications

Table of contents

[Introduction 3](#_Toc12884248)

[1. Project context 3](#_Toc12884249)

[2. Global architecture 3](#_Toc12884250)

[3. Component description 3](#_Toc12884251)

[a. Component Controller/dashboard.js 3](#_Toc12884252)

[b. Component Models/composition.js 3](#_Toc12884253)

[c. Component Controller/composition.js 3](#_Toc12884254)

[d. Component Models/monitor.js 3](#_Toc12884255)

[e. Component Controller/ressource.js 3](#_Toc12884256)

[4. Traceability matrix 4](#_Toc12884257)

# Introduction

The aim of this software architecture specification (SAS) is to present the technical elements necessary for the To Do List project.

# Project context

The provider will have to provide a fully workable and manageable to-do-list software with the capabilities to manage users and tasks with some administrator capabilities.

Furthermore, the customer would like to have a mobile application of the software in the future, so you the provider need to implement a re-usable and “developer-friendly” backend.

It’s not a mandatory but the customer will really enjoy a mobile version of the software, the provider is free to do it or not

# Global architecture

Global view of the project, what are the different components and how they interact between each other.

# Component description

## Component back/src/api/controller/authController.js

This crontoller file provide the auth verification and auth function of a user through the API

## Component back/src/api/controller/taskController.js

This controller file provide a full CRUD management of all task through the API

## Component back/src/api/controller/userController.js

This controller file provide a full CRUD management of all iser through the API

## Component front/src/app/pages/HomePage.js

This component contains all the sub-component & actions required in the project

## Component front/src/app/components/layout/LoginModal.js

This component provide a Login Formular in a modal

## Component front/src/app/components/layout/TaskItem.js

This component provide a view of a task and all the CRUD actions

## Component front/src/app/components/layout/VerticalTaskList.js

This component display the task component verticaly

## Component front/src/app/components/layout/UserItem.js

This component provide a view of a user and all the CRUD actions

## Component front/src/app/components/layout/ VerticalUserList.js

This component display the user component verticaly

## Component front/src/app/components/layout/RegisterForm.js

This component provide a register Formular in a modal (it’s also allow to create an account for an administrator in the admin panel)

## Component front/src/app/components/layout/NewTaskModal.js

This component provide a Formular in a modal that allow to create a new task

# Traceability matrix

This matrix makes the correspondence between components, classes, functions and requirements developed in the request for proposal.

|  |  |  |  |
| --- | --- | --- | --- |
| Id requirement | ***Requirement description*** | ***Component*** | ***Function / action*** |
| REQ\_FUNC\_010 | You must be able to register / log-in on the software | LoginModal.js RegisterForm.js | Allow all the authentication action for user (login & logout) |
| REQ\_FUNC\_020 | You must be able to add / delete / edit / list the tasks | HomePage.js TaskItem.js | Allow all the CRUD action |
| REQ\_FUNC\_030 | You must be able to pass a task done or undone it | taskController.js TaskItem.js | Allow all the CRUD action |
| REQ\_FUNC\_040 | As administrator, you must be able to list all the users | userController.js VerticalUserList.js | Display all userItem.js in a vertical list |
| REQ\_FUNC\_050 | As administrator, you must be able to edit / remove / add / ban the users | userController.js UserItem.js | Allow all the CRUD action |
| REQ\_FUNC\_060 | As administrator, you must be able to list all the tasks | taskController.js VerticalTaskList.js | Display all taskItem.js in a vertical list |
| REQ\_FUNC\_070 | As administrator, you must be able to edit / remove / add the tasks | taskController.js TaskItem.js | Allow all the CRUD action |
| REQ\_FUNC\_080 | A user (non-administrator) is not able to see the tasks of another user | authController.js | All the Verify function manage the role access through the API |
| REQ\_IHM\_010 | You must provide a login page | LoginModal.js | Allow the user to connect himself, it will make a request on the API for getting a jwt |
| REQ\_IHM\_020 | You must provide a tasks management page | HomePage.js | The HomePage manage through the user jwt the contents to display on the screen |
| REQ\_IHM\_030 | You must provide an administrator a task management page | HomePage.js | The HomePage manage through the user jwt the contents to display on the screen |
| REQ\_IHM\_040 | You must provide an administrator a user management page with these features: | HomePage.js | The HomePage manage through the user jwt the contents to display on the screen |
| REQ\_DATA\_010 | The users and tasks information need to be stored in a persistent database | cloud.mongodb.com | Database is saved into MongoDB Atlas (Cloud) |