EduTask

Context, Requirements, and System Specification

Version: v1.0.0

Project Name	EduTask	
Project Lead	Julian Frattini	
Created on	22.03.2022	
Last changed	02.05.2022	
State	X	In process
		Submitted
		Completed
AMDiRE Version	1.0 (2020), merged	

Context Specification

Scope

The internet has become a vault of easily accessible and freely available educational content. However, this abundance of content resulted in a shift of challenges when it comes to consumption: it is now not difficult anymore to find resources, but rather to identify and keep track of them.

While commercial learning platforms like Skillshare, Udemy, or brilliant.org have gained in popularity over the recent years, free platforms like YouTube remain to be among the most prominent sources for education, for example in the software engineering or more specifically web development context.

However, learning is not a primary use case supported on YouTube and therefore it is easy to lose track or focus of the learning process. YouTube's playlists, where users can save videos in a linear queue, and chapters, where a content creator can subdivide a video into smaller sections, are the only mechanisms that allow to assemble and decompose learning content.

Given the amount of available learning content, these features are not sufficient to facilitate an effective learning process. Learning a complex topic requires to

- Gather and store eligible resources and associate them to each other
- Associate gathered resources with concrete todo lists.

We need a platform that connects the already available educational resources with a management system, which allows to track and dissect the resources.

Objectives and Goals

The system EduTask shall fulfill the following goals:

1. Offer a central platform to store eligible educational resources.
2. Associate each resource with todo items and keep track of their progress.

Furthermore, it will be necessary to associate resources with each other. We consider two scenarios:

1. One resource may presuppose knowledge from another resource, e.g., because a video in a series relies on having seen all previous videos.
2. Several resources may belong to the same category.

When the number of stored resources grows, these associations become essential in order to maintain an effective learning process.

Requirements Specification

Versions and Features

We break down the accomplishment of the goals into multiple versions, where each version contains a number of features. The aim of this scoping is to incrementally deliver all features in time.

Version	Features
v1.0.0	Simple signup and login using just email addresses, adding tasks, manipulating todo lists of tasks, viewing tasks
v1.1.0	Secure signup and login using encrypted passwords
v1.2.0	Adding prerequisites to tasks (i.e., other tasks that need to be completed before)
v1.3.0	Adding categories to tasks

Prioritization

The requirements are prioritized on a four-tier scale using the MuSCoW approach:

- MUST HAVE: Project (release) cannot do without them
- SHOULD HAVE: Must haves long-term; not vital but add significant value
- COULD HAVE: Nice to have in a future release; small impact if left out
- WOULD HAVE: Get back to them in better days; no real impact

Functional Requirements

ID	Requirement	Priority	Version
R1	As a user I want to sign up using my email address.	MUST	v1.0.0

ID	R1UC1
Primary Actor	End user
Preconditions	The user is on the signup page of the system.
Main Success Scenario	1. If the user has entered a valid email address that is not yet associated to any account in the system, a first name, and a last name, and clicks “signup”, a new account is created.
End Condition	The user is forwarded to the task view of the system (see R6)
Alternative Scenarios	1.b If an email address is already associated to an existing user in the system, it cannot be used again for signup and the system instead prompts the user to log in with that email address. 1.c If an email address is invalid, i.e., does not contain exactly one @-symbol as well as a dot symbol in the internet domain (substring after the @-symbol), the “signup” button remains disabled. 1.d If either the first name or last name is empty, the “signup” button remains disabled.

ID	Requirement	Priority	Version
R2	As a user I want to protect my account with a secure password	MUST	v1.1.0
R3	As a user I want to login using an email and password combination.	MUST	v1.0.0

Use cases:

ID	R3UC1
Primary Actor	End user
Preconditions	The user has a valid account in the system
Main Success Scenario	1. When the user enters the website, the system prompts the user to enter his/her credentials. 2. When the user enters a valid email-password combination and clicks “Login”, the system authenticates the user.
End Condition	The user is forwarded to an overview of the tasks associated to him/her.
Alternative Scenarios	2.b When the email is not yet registered, the system prompts the user to signup instead. 2.c When the password is incorrect, the system prompts the user to try again.

ID	Requirement	Priority	Version
R4	As a user I want to be able to log out of the system.	SHOULD	v1.0.0

Use cases:

ID	R4UC1
Primary Actor	End user
Preconditions	The user has a valid account in the system and is currently authenticated
Main Success Scenario	1. When the user clicks on the arrow in the NavBar, a dropdown menu with the option to logout opens. 2. When the user clicks on logout, his authentication will be voided
End Condition	The user is forwarded to the landing page of the System
Alternative Scenarios	2.b When the user instead clicks on the arrow in the NavBar again, the dropdown menu closes

ID	Requirement	Priority	Version
R5	As an authenticated user I want to add a new task by entering a name and a YouTube viewkey.	MUST	v1.0.0

Use cases:

ID	R5UC1
Primary Actor	End user
Preconditions	The user has a valid account in the system and is currently authenticated
Main Success Scenario	1. When the user enters a title and a valid YouTube viewkey in the task creation form and clicks create, the form will be reset and a new task with one default todo item named “Watch Video” will be created.
End Condition	A new task card appears in the task view.
Alternative Scenarios	1.b If either the title or the viewkey input is empty, the “create” button remains disabled 1.c If the YouTube viewkey is invalid and the user clicks “create”, a warning message appears.

ID	Requirement	Priority	Version
R6	As an authenticated user I want to see the list of all of my tasks.	MUST	v1.0.0

Rationale: An overview of the tasks is essential to facilitate tracing the work that an end-user associates to his/her resources.

Use cases:

ID	R6UC1
Primary Actor	End user
Preconditions	The user has a valid account in the system, is currently authenticated, and has at least one tasks associated to him
Main Success Scenario	2. If the user hovers over a task card, the name of the task appears. 3. If the user clicks on the task card, a detailed view of the task is opened in a popup window.
End Condition	A popup window containing the title, description, link to the resource, and list of todo items is shown on top of the list of tasks
Alternative Scenarios	

ID	R6UC2
Primary Actor	End user
Preconditions	The user has a valid account in the system, is currently authenticated, has at least one tasks associated to him, and has opened that task in detail view (see R6UC1)
Main Success Scenario	1. If the user clicks on the close button of the popup window, the detail view closes
End Condition	The popup window disappears, and the list of all tasks is unobstructed again.
Alternative Scenarios	1b. If the user hits the “Escape” key, the detail view closes also.

ID	Requirement	Priority	Version
R7	As an authenticated user I want to change the title and description of an added task.	SHOULD	v1.0.0

Use cases:

ID	R7UC1
Primary Actor	End user
Preconditions	The user has a valid account in the system, is currently authenticated, has at least one tasks associated to him, and has opened that task in detail view (see R6UC1)
Main Success Scenario	1. When the user clicks on the title, the text gets replaced by an input form with the current content and a “save” button appears next to it. 2. When the user clicks “save”, the button disappears, and the input form gets replaced by a text field.
End Condition	The title is replaced by the text the user entered in the input form.
Alternative Scenarios	2.b If the user presses the “Escape” key, the “save” button disappears, the input form gets replaced by a text field, but the text is reset to the original text before the edit.

ID	R7UC2
Primary Actor	End user
Preconditions	The user has a valid account in the system, is currently authenticated, has at least one tasks associated to him, and has opened that task in detail view (see R6UC1)
Main Success Scenario	1. When the user clicks on the description, the text gets replaced by an input form with the current content and a “save” button appears next to it. 2. When the user clicks “save”, the button disappears, and the input form gets replaced by a text field.
End Condition	The description is replaced by the text the user entered in the input form.
Alternative Scenarios	2.b If the user presses the “Escape” key, the “save” button disappears, the input form gets replaced by a text field, but the text is reset to the original text before the edit.

ID	Requirement	Priority	Version
R8	As an authenticated user I want to manipulate the todolist associated to a task, i.e., add a new, toggle an existing, or delete an existing todo item.	MUST	v1.0.0

Rationale: todo items are the result of decomposing an educational task associated with a resource (e.g., a YouTube video) into smaller, manageable subtasks. A todo item consists of a description and its status, which is either *active* or *done*. Manipulating a todolist (i.e., a list of todo items) is of major importance in order to ensure that each user can capture the result of his/her personal decomposition process of a task.

Use Cases:

ID	R8UC1
Primary Actor	End user
Preconditions	The user is authenticated, has at least one task associated to his account, and views this task in detail view mode.
Main Success Scenario	1. The user enters a description of a todo item into an empty input form field. 2. If the description is not empty and the user presses “Add”, the system creates a new todo item
End Condition	The new (active) todo item with the given description is appended to the bottom of the list of existing todo items.
Alternative Scenarios	2.b If the description is empty then the “Add” button should remain disabled.

ID	R8UC2
Primary Actor	End user
Preconditions	The user is authenticated, has at least one task with at least one todo item associated to his account, and views this task in detail view mode.
Main Success Scenario	1. The user clicks on the icon in front of the description of the todo item. 2. If the todo item was previously active, it is set to done.
End Condition	The toggled todo item is struck through.
Alternative Scenarios	2.b If the todo item was previously done, it is set to active. The toggled todo item is not struck through anymore.

ID	R8UC3
Primary Actor	End user
Preconditions	The user is authenticated, has at least one task with at least one todo item associated to his account, and views this task in detail view mode.

Main Success Scenario	1. If user clicks on the x symbol behind the description of the todo item, the todo item is deleted
End Condition	The todo item is removed from the todo list.
Alternative Scenarios	-

ID	Requirement	Priority	Version
R9	As an authenticated user I want to flag an existing task to be the prerequisite of another existing task.	COULD	v1.2.0
R10	As an authenticated user I want to filter my task list for tasks, where all prerequisites are fulfilled.	COULD	v1.2.0
R11	As an authenticated user I want to associate an existing task to a category	SHOULD	v1.3.0
R12	As an authenticated user I want to create a new category.	SHOULD	v1.3.0
R13	As an authenticated user I want to filter my task list for a specific subset of my categories	SHOULD	v1.3.0
R14	As an authenticated user I want to see all categories created by other users.	WOULD	v1.3.0
R15	As an authenticated user I want to express interest in an existing category and receive recommended tasks for that category.	WOULD	v1.3.0

Non-functional Requirements

ID	Requirement
NFR1	The system has to be accessible.
NFR2	The system has to be evolvable.
NFR3	The system has to be extensible
NFR4	The system has to be maintainable.
NFR5	The system has to be reliable.
NFR6	The system has to be safe.

System Design

System Architecture

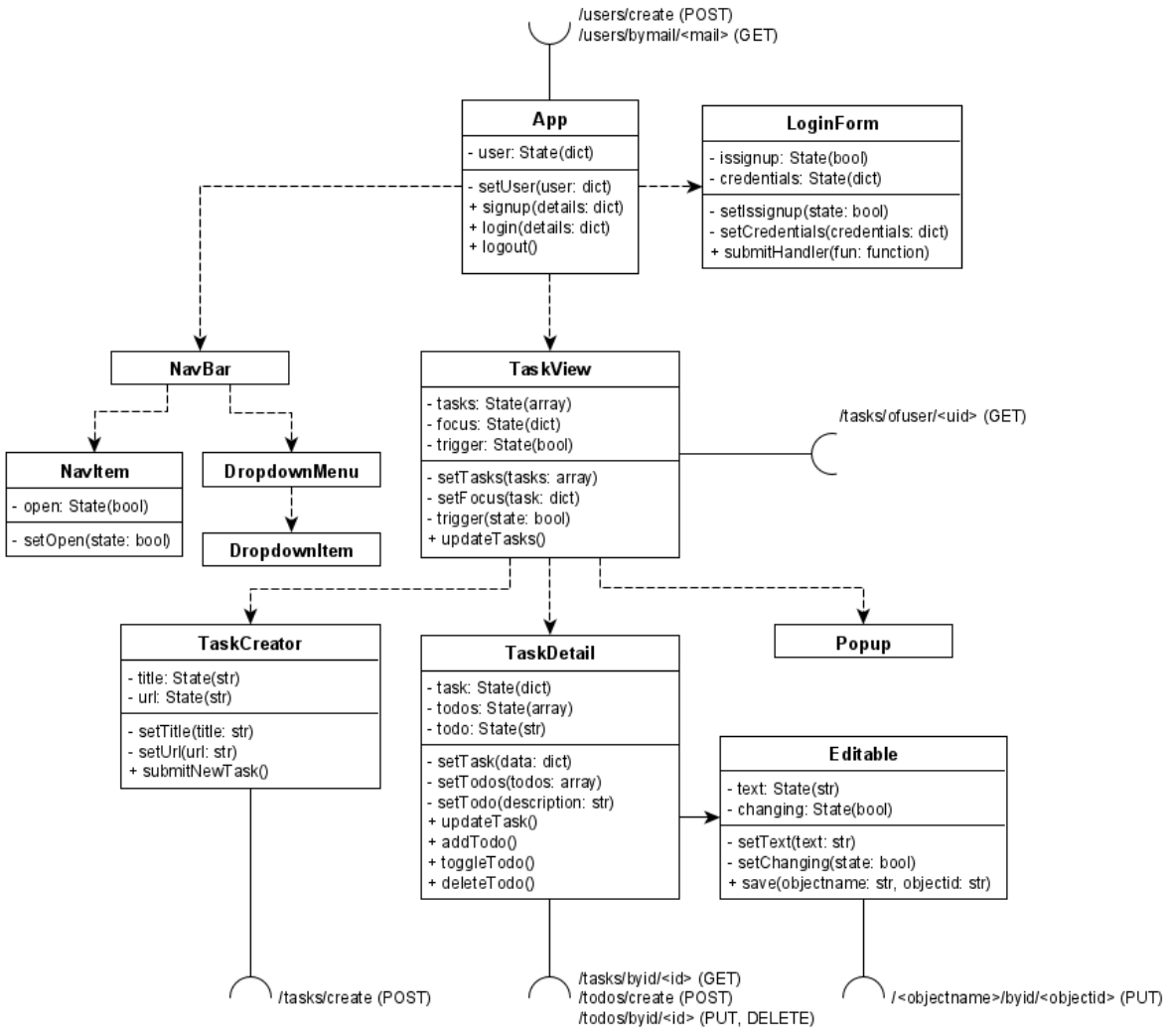
The system will be realized as a three-tier web application.

The technology of the three components is:

- Database: MongoDB, a NoSQL document-oriented database
- Server: Flask, a Python backend framework
- Client: React, a Javascript frontend framework

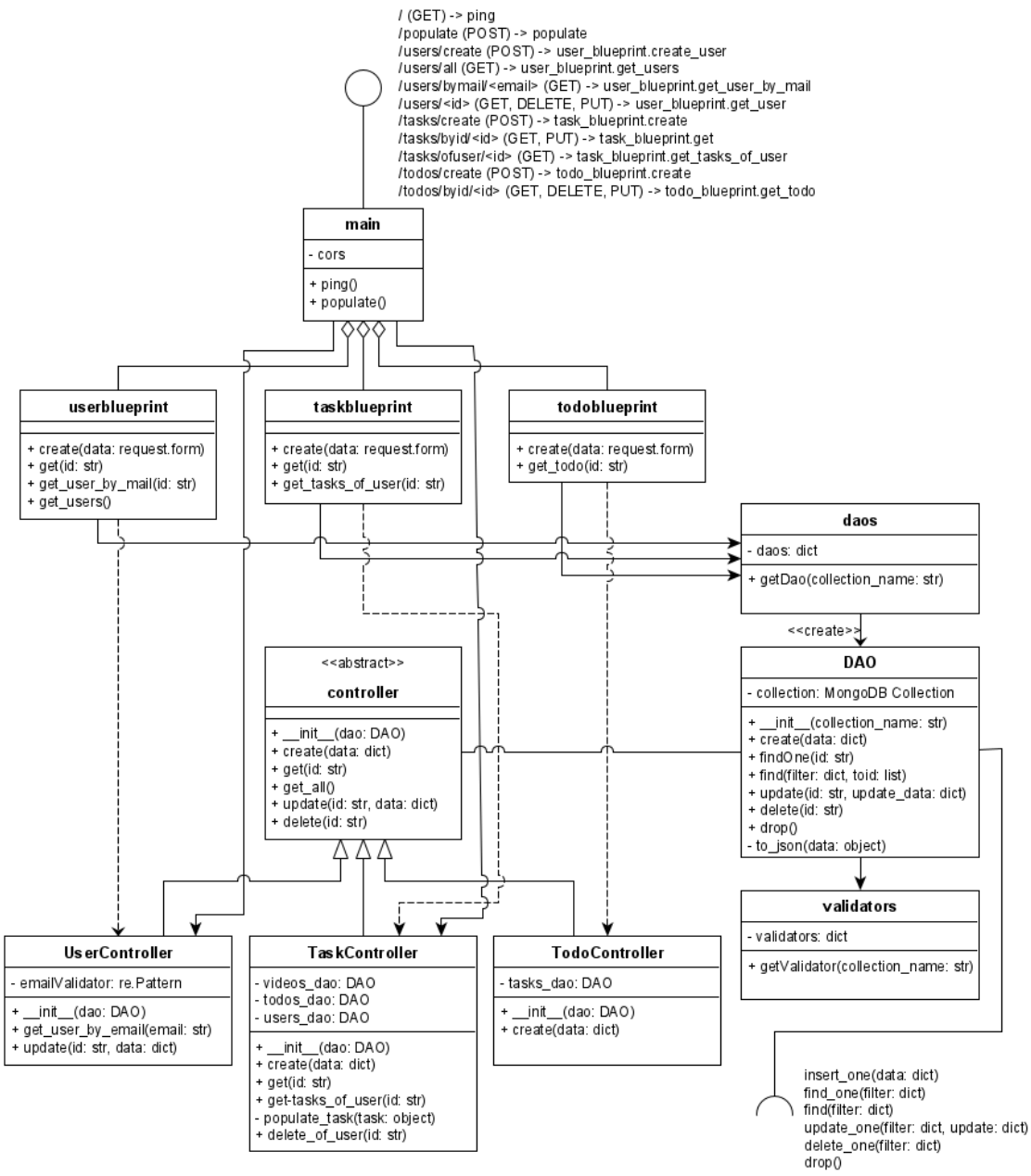
Frontend

The frontend is realized as a cluster of React components, which require an interface to a backend API to access and manipulate data.



Backend

The backend is realized as a Flask API. The *main* process exposes an API, which is realized by the *blueprints*. These blueprints call upon the *controllers* to perform data manipulation. The *data access object* (DAO) acts as a mediator between the backend and the MongoDB database.



Data Model

The data model realized in the MongoDB consists of the following entities. Each entity is realized as a collection in one central database, each entry in a collection is realized as a document. MongDB's integrated validators are used to ensure that the datatypes, required properties, and uniqueness of properties is ensured when inserting a new document