Specificații proiect

1. [Aplicație web acordarea anonimă de note 2](#_Toc124705692)
2. [Aplicație web pentru gestionarea rezolvării bug-urilor 6](#_Toc124705694)
3. [Aplicație web pentru partajarea experiențelor utilizării mijloacelor de transport 9](#_Toc124705696)
4. [Aplicație web pentru a preveni risipa de alimente 12](#_Toc124705698)
5. [Aplicație web pentru gestionarea notițelor de curs/seminar 14](#_Toc124705700)
6. [Aplicație web acordarea de feedback continuu 18](#_Toc124705702)
7. [Template general 20](#_Toc124705704)

Aplicație web acordarea anonimă de note

Obiectiv

Realizarea unei aplicații web care să permită acordarea de punctaje anonime de catre un juriu anonim de studenti proiectului altor studenti.

Descriere

Aplicația trebuie să permită acordarea unui punctaj unui proiect de către un juriu anonim de colegi.

Platforma este bazată pe o aplicație web cu arhitectură de tip Single Page Application accesibilă în browser de pe desktop, dispozitive mobile sau tablete (considerând preferințele utilizatorului).

Funcționalități (minime)

Ca student membru în echipa unui proiect (MP) pot să îmi adaug un proiect și să definesc o serie de livrabile partiale ale proiectului. La înscriere devin automat și parte din grupul de posibili evaluatori. **1p**

Ca MP pentru un livrabil partial pot adăuga un video demonstrativ pentru proiect sau un link la un server unde poate fi accesat proiectul. **1p**

La data unui livrabil parțial, ca student care nu este MP pot fi selectat aleatoriu să fac parte din juriul unui proiect. Pot acorda o notă proiectului doar dacă am fost selectat în juriul pentru el. **1p**

Nota la un proiect este anonimă, iar nota totală se calculează omițând cea mai mare și cea mai mică notă. Notele sunt de la 1-10 cu până la 2 cifre fracționare. **1p**

Ca profesor, pot vedea evaluarea pentru fiecare proiect, fără a vedea însă identitatea membrilor juriului. **1p**

Aplicația are și un sistem de permisiuni. Doar un membru al juriului poate să adauge/modifice note și doar notele lui pe o perioadă limitată de timp. **1p**

Anonymous grading web application

Objective

Developing a web application which allows student projects to be graded by anonymous juries of colleagues.

Description

The application should allow students' projects to be graded by anonymous juries of peers.

The application is built on a Single Page Application architecture and is accessible from the browser on a desktop, mobile device or tablet (depending on user preference).

(Minimal) functionality

As a student member of a project team (PM), I can add a project and define a series of partial project deliverables. By registering I automatically become eligible as an evaluator. **1p**

As a PM I can add a demonstrative video or a link to a server hosting the deployed project for any partial deliverable. **1p**

When a partial deliverable is due, any student who is not a PM for a particular project can be randomly selected to be part of the project jury. A student which is a member of said jury can grade the project. **1p**

The grade for the project is anonymous and the total grade is calculated by omitting the lowest and highest grades and then averaging the remaining ones. The grades are 1-10 with at most 2 fractional digits. **1p**

As a professor, I can see the results of the evaluation for each project, without being able to see the identity of the jury members. **1p**

The application has a system of permissions. Only a member of the jury can add/modify grades and they can only modify their grades. Grades can only be modified for a limited period of time. **1p**

Aplicație web pentru gestionarea rezolvării bug-urilor

Obiectiv

Realizarea unei aplicații web care să permită gestionarea bug-urilor într-o aplicație.

Descriere

Aplicația trebuie să permită comunicarea dintre membri unei echipe a bug-urilor dintr-o aplicație

Platforma este bazată pe o aplicație web cu arhitectură de tip Single Page Application accesibilă în browser de pe desktop, dispozitive mobile sau tablete (considerând preferințele utilizatorului).

Funcționalități (minime)

Ca student trebuie să pot sa ma conectez la aplicație cu un cont bazat pe o adresă de email. **1p**

Ca student membru în echipa unui proiect (MP) pot să înregistrez un proiect software pentru a fi monitorizat prin aplicație, specificând repository-ul proiectului și echipa de proiect. **0.5p**

Ca student care nu face parte dintr-un proiect înregistrat pot să mă adaug ca tester (TST) la proiect. **0.5p**

Ca TST pot înregistra un bug în aplicație. Bug-ul conține o severitate, o prioritate de rezolvare, o descriere și un link la commit-ul la care se referă. **0.5p**

Ca MP pot vedea bug-urile înregistrate pentru proiectele din care fac parte. **0.5p**

Ca MP îmi pot aloca rezolvarea unui bug. Un singur MP poate să aibă alocată rezolvarea unui bug la un moment dat. **1p**

Ca MP după rezolvarea unui bug pot adăuga un status al rezolvării cu un link la commit-ul prin care s-a rezolvat. **1p**

Aplicația are și un sistem de permisiuni. Un MP poate adăuga și modifica un proiect, poate actualiza status-ul unui bug. Un TST poate adăuga un bug. **1p**

Exemple

Bugzilla => <https://www.bugzilla.org/>

Web application for bug tracking

Objective

Developing a web application which provides bug management for an application.

Description

The application must ensure bug-fixing-related communication between team members of an application.

The application is built on a Single Page Application architecture and is accessible from the browser on a desktop, mobile device or tablet (depending on user preference).

(Minimal) functionality

As a student, I can connect to the application with an account based on an email address. **1p**

As a project team member (PM) student, I can register a software project in the bug tracking application, specifying a description, the repository where the project is hosted and the project team. **0.5p**

As a student that is not a member of the project team, I can register as a tester (TST) for the project. **0.5p**

As a TST I can register a bug in the bug-tracking application. The bug is registered with a severity, a description and a link to the commit that has been tested. **0.5p**

As a PM I can see the registered bugs for the projects I participate in. **0.5p**

As a PM I can allocate fixing a bug to myself. Only one PM can have a bug allocated at a particular time. **1p**

As a PM, after solving a bug, I can add a status to the solution with a link to the commit through which the bug was fixed. **1p**

The application has a permission system. A PM can add and modify a project and can change the status of a bug. A TST can add a bug. **1p**

Examples

Bugzilla => <https://www.bugzilla.org/>

Aplicație web pentru partajarea experiențelor utilizării mijloacelor de transport

Obiectiv

Realizarea unei aplicații web prin care utilizatorii pot împărtăși experiența din urma utilizării unuia din mijloacele de transport în comun.

Descriere

Aplicația trebuie să permită crearea unui cont prin care utilizatorul poate să partajeze o experiență, după ce a folosit un mijloc de transport în comun. Pentru utilizatorii anonimi, aplicația va permite căutarea și vizualizarea intrărilor în platforma.

Funcționalități (minime)

Modulul Utilizator **2p**

* Creare cont utilizator pe baza unor câmpuri sau cu ajutorul unui API extern (GMAIL, Facebook, Linkedin) **1p**
* Modificarea sau dezactivarea contului **0.5p**
* Resetarea parolei **0.5p**

Modulul de Partajare **2p**

* Crearea unei experiențe presupune completarea următoarelor câmpuri: **0.5p**
  + Punctul de plecare (A)
  + Punctul de sosire (B)
  + Mijlocul de transport folosit: bus, metro, tram, etc.
  + Ora plecare
  + Durata călătoriei
  + Gradul de aglomerare al mijlocului de transport
  + Observații
  + Nivelul de satisfacție (smiley faces)
* Modificarea intrărilor specifice utilizatorului; **0.5p**
* Listarea tuturor experiențelor create de un utilizator; **0.5p**
* Ștergerea unei experiențe; **0.5p**

Modulul de Căutare **2p**

* Modulul de căutare va trebui să permită utilizatorului introducerea unor cuvinte cheie, după care vor fi afișate rezultatele, sub formă de listă. **1p**
* Modulul va trebui să returneze rezultate relevante în funcție de locație, mijlocul de transport folosit sau destinație. **1p**

Web application for sharing public transport experiences

Objective

Develop a web application where users could share their experience after using public transportation.

Description

The application should allow the creation of an account where the user should share an experience after using public transportation. For anonymous users, the application would allow searching and viewing existing experiences from the platform.

Functionalities (minimal)

User Module **2p**

* Create an account based on some completed fields or based on an external API (Gmail, Facebook, Linkedin) **1p**
* The ability to modify or disable the account **0.5p**
* Password recovery **0.5p**

Sharing Module **2p**

* Sharing an experience would require completing the following fields: **0.5p**
  + Starting point (A)
  + Destination point (B)
  + Public transport that was used: bus, metro, tram, etc.
  + Departure hour
  + Trip duration
  + Crowdedness level of the public transport
  + Observations
  + Satisfaction level (smiley faces)
* Edit a shared experience of a user; **0.5p**
* Listing all the experiences shared by a user; **0.5p**
* Delete an experience; **0.5p**

Searching module **2p**

* The search module should allow the user to find an experience or a list of experiences based on one or multiple keywords. The results will be displayed as a list. **1p**
* This module should return the results based on location, used transportation or destination. **1p**

Aplicație web pentru a preveni risipa de alimente

Obiectiv

Realizarea unei aplicații web prin care utilizatorii pot să ofere alimente pe care nu le mai consumă

Descriere

Aplicația încurajează utilizatorii să conștientizeze risipa de alimente într-un mod distractiv și bazat pe comunitate

Funcționalități

Ca utilizator pot sa fac o lista de alimente organizată pe categorii cu ce am în frigider. **1p**

Ca utilizator primesc alerte cand un produs se apropie de termenul de valabilitate. Am posibilitatea de a marca acele produse ca fiind disponibile. **1p**

Ca utilizator pot sa îmi definesc grupuri de prieteni și sa ii etichetez ca vegetarieni, carnivori, iubitori de zacusca, etc. **1p**

Îmi invit prietenii sa se uite pe lista de alimente disponibile. **1p**

Un alt utilizator poate sa faca claim pe un produs. **1p**

Integrare cu instagram si facebook pentru a posta produsele disponibile. **1p**

Anti-Food Waste App

Goal

To develop a web app that allows users to share food they no longer consume.

Descriere

The app makes users more aware of food waste in a fun way and based on community sharing.

Functionalities

As a user, I can make a list organized by categories with the product available in my fridge. **1p**

As a user, I get notified when a product reaches the availability term. The app allows me to mark products as available to be shared. **1p**

As a user, I can define groups of friends and label them according to their food preferences. **1p**

I can invite friends to see the list of available products. **1p**

Any user can claim products in my list. **1p**

The app allows sharing on social media platforms. **1p**

Aplicație web pentru gestionarea notițelor de curs/seminar

# Obiectiv

Realizarea unei aplicații web care să permită gestionarea notițelor luate la curs / seminar.

# Descriere

Aplicația trebuie să permită organizarea notițelor de către student în funcție de materiile la care participă și activitățile de studiu individual.

Platforma este bazată pe o aplicație web cu arhitectură de tip Single Page Application accesibilă în browser de pe desktop, dispozitive mobile sau tablete (considerând preferințele utilizatorului).

Editorul de notițe trebuie să fie ușor de folosit pentru a prelua informațiile în timpul cursului/seminarului. Implementează un sistem de **markdown** pentru a realiza formatarea textului.

# Funcționalități (minime)

Ca student vreau să mă loghez folosind contul instituțional (@stud.ase.ro) pentru a gestiona notițele de la curs/seminar. **1p**

Ca student vreau să am posibilitatea de a vizualiza, adăuga, edita și șterge notițele de curs ca să organizez informația mai bine. **1p**

Ca student vreau să am opțiunea de a adăuga atașamente (imagini, documente) la notițele de curs/seminar ca să am mai multe detalii. **1p**

Ca student vreau să am posibilitatea de a organiza notițe în funcție de materii, data, etichete (tag), cuvinte cheie ca să regăsesc cu ușurință informația. **1p**

Ca student vreau să am posibilitatea de a partaja notițe cu alți colegi. **0.5p**

Ca student vreau să am posibilitatea de a integra conținut din alte surse pentru a lua notițe (de exemplu când urmăresc un video pe youtube, citesc o carte pe kindle, particip la o conferință online). **1p**

Ca student vreau să organizez grupuri de studiu. Voi invita mai mulți colegi pe grup voi partaja notițe. **0.5p**

# Exemple

StuDocu

<https://www.studocu.com/>

Evernote

<https://evernote.com/>

Web app for taking notes during courses/labs

# Goal

To develop a web app that allows students to take notes during courses/labs

# Description

The application must allow students to organise their notes by classes they attend and individual study activities.

The platform is based on a Single Page Application architecture. It will be accessible on a desktop, mobile or tablet browser (depending on user preferences).

The note editor should be easy to use so that the student can take notes during class. It will implement a **markdown system** to allow simple text formatting.

## Minimal functionalities

As a student, I want to be able to log in with my institutional account (@stud.ase.ro) to manage my notes. **1p**

As a student, I want to be able to view, add, edit and delete notes so that I can better organize information. **1p**

As a student, I want the option to add attachments (images, documents) to the notes I take so that I can have more details about the subject. **1p**

As a student, I want to be able to organize notes based on classes, date, labels (tags) and keywords so that I can retrieve information easily. **1p**

As a student, I want to be able to share notes with other colleagues.  **0.5p**

As a student, I want to be able to integrate content from other sources and take notes (like for example while I listen to a youtube video, read a book on kindle, attending an online conference). **1p**

As a student, I want to be able to organize a study group. I can invite multiple colleagues that will share notes within the group. **0.5p**

# Examples

StuDocu

<https://www.studocu.com/>

Evernote

<https://evernote.com/>

Aplicație web acordarea de feedback continuu

Obiectiv

Realizarea unei aplicații web care să permită acordarea de feedback continuu unei activități.

Descriere

Aplicația trebuie să permită acordarea de feedback continuu la un curs sau seminariu.

Platforma este bazată pe o aplicație web cu arhitectură de tip Single Page Application accesibilă în browser de pe desktop, dispozitive mobile sau tablete (considerând preferințele utilizatorului).

Funcționalități (minime)

Ca profesor pot defini o activitate la o anumită dată, cu o descriere și un cod unic de acces la activitate. Activitatea poate fi accesată pentru o durată prestabilită de timp. **1p**

Ca student pot introduce un cod pentru a participa la o activitate definită. Codul este valabil pentru durata activității. **1p**

Ca student care a accesat o activitate am acces la o interfața împarțită în 4 cadrane fiecare cu un emoticon (smiley face, frowny face, surprised face, confused face). **1p**

În orice moment pot apăsa un emoticon pentru a reacționa la activate. Ca student pot să adaug oricâte instanțe de feedback. **1p**

Ca profesor pot vedea feedback-ul continuu cu momentele de timp asociate. Feedback-ul este anonim. **1p**

Feedback-ul este accesibil atât în timpul activității cât și ulterior. **1p**

Continuous feedback application

Objective

Developing a web application which enables users to provide continuous feedback on an activity

Description

The application must enable users to provide continuous feedback on a course or tutorial.

The application is built on a Single Page Application architecture and is accessible from the browser on a desktop, mobile device or tablet (depending on user preference).

(Minimal) functionality

As a professor, I can define an activity at a particular date, with a description and a unique access code for the activity. The activity is accessible for a set period of time. **1p**

As a student, I can input a code to participate in a defined activity. The code can be used for the duration of the activity. **1p**

As a student who has accessed an activity, I have access to an interface split into 4 rectangles each containing an emoticon (smiley face, frowny face, surprised face, confused face). **1p**

At any time I can press an emoticon to react to the activity. As a student, I can add an unlimited number of feedback instances. **1p**

As a professor, I can see the continuous feedback stream, with each feedback instance associated with the time it was generated at. I can only see the feedback as anonymous. **1p**

For myself, feedback is available both during and after the activity. **1p**

Template general

[Obiectiv General]

Realizarea unei aplicații pe una dintre temele specificate, cu back-end RESTful care accesează date stocate într-o bază relațională pe baza unui API de persistenţă și date expuse de un serviciu extern și frontend SPA realizat cu un framework bazat pe componente.

[Limitări tehnologice]

* Front-end-ul trebuie realizat cu ajutorul unui framework bazat pe componente (React.js, care este acoperit în curs, sau Angular 2+, Vue.js)
* Back-end-ul trebuie să aibă o interfață REST și să fie realizat în node.js
* Stocarea se va face peste o bază relațională și accesul la baza se va face prin intermediul unui ORM
* Codul trebuie versionat într-un repository git, cu commit incrementale cu descrieri clare

[Stil și calitatea codului]

* Aplicație reală, coerentă din punct de vedere al logicii de business
* Codul trebuie să fie bine organizat, numele variabilelor trebuie să fie sugestive (și trebuie să se utilizeze un standard de numire oricare ar fi el e.g. camel case), codul trebuie să fie indentat pentru a fi ușor citibil
* Codul trebuie documentat cu comentarii la fiecare clasa, funcție etc.
* Aplicațiile care nu funcționeaza nu primesc punctaj. Se poate însă demonstra doar funcționarea back-end-ului sau a front-end-ului
* Opțional: test coverage

[Livrabile parțiale] - 3 etape (livrare se face introducând un link la un repository într-un google form; cadrul didactic coordonator va fi invitat ca un contribuitor la repository) - nelivrarea la o etapă intermediară reduce punctajul maxim cu 10% (i.e. dacă punctajul maxim este de 5 puncte din nota finală livrarea direct la final implică un punctaj maxim de 4 puncte)

* Specificații detaliate, planul de proiect, prezența unui proiect în git - 21.11.2021
* Serviciu RESTful funcțional în repository + instrucţiuni de rulare -  05.12.2021
* Aplicața completă - se livrează în ultimul seminariu (demo) - ultimul seminariu

General template

[General objective]

Developing an application on one of the specified topics, with a RESTful backend which accessed data stored in a relational database through a persistence API as well as data from an external service and a SPA front-end built with a component-based framework.

[Technological constraints]

* Front-end must be built with a component-based framework(React.js, which is covered in the course, or Angular 2+, Vue.js)
* The back end must have a REST interface and must be implemented in node.js
* Storage must be over a relational database and access to the database must be done via an ORM
* Code must be versioned in a git repository with incremental commits with clear descriptions

[Code quality and style]

* Real application, coherent from a business logic standpoint
* The code must be well organized, variable names should be suggestive of their purpose and use a naming standard (e.g. camel case), and the code should be indented for readability
* The code must be documented with comments for each class, function etc.
* Non-working applications receive no points. However, the functionality of the front-end or back-end can be demonstrated separately
* Opțional: test coverage

[Partial deliverables] - 3 stages (delivery is done by supplying a link to a repository through a google form; the coordinating university staff member will be invited as a contributor to the repository) - non-delivery of an intermediary stage reduces the maximum receivable points by 10% (i.e. if the maximum grade for the project is 5 points, delivering it directly at the end means the maximum points receivable is 4)

* Detailed specifications, project plan, presence of a git project - 21.11.2021
* Functional RESTful service present in the repository + instructions to run said service -  05.12.2021
* Complete application - delivered in the last tutorial (demo) - last tutorial