

Videoteka

We Don't Byte komanda

Produktą kūrė:

Tautvydas Mačiulis

Julius Rasimas





Aurimas Arlauskas

Simonas Vingis

Povilas Stašys

1. Produkto vizija

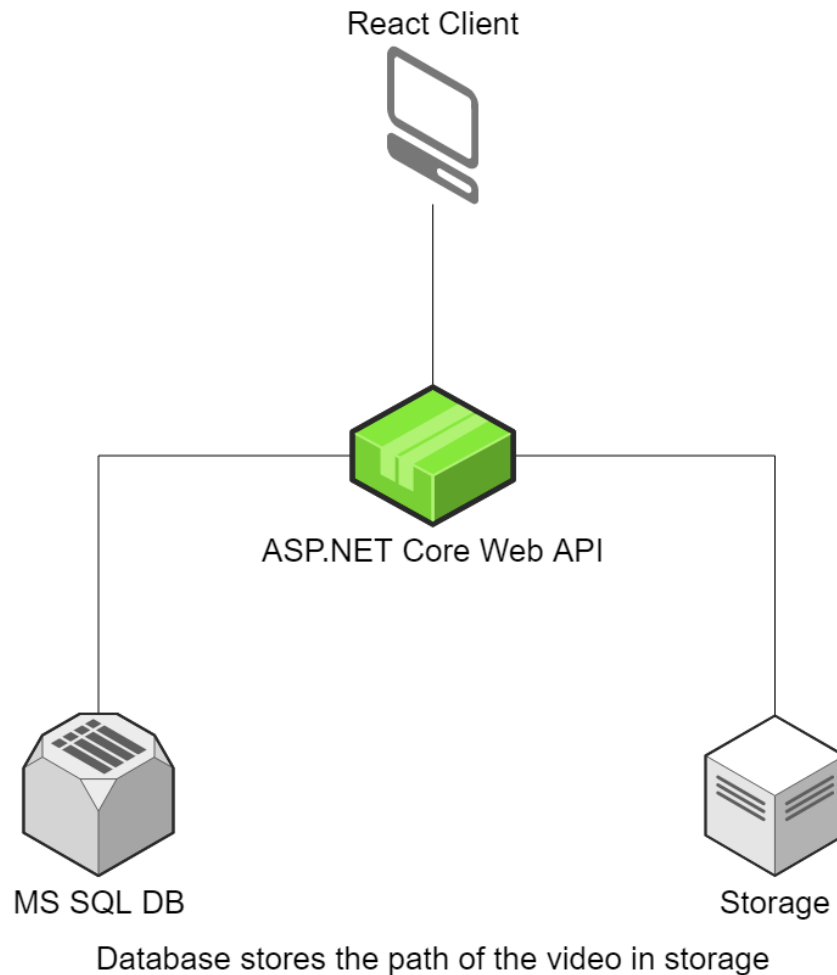
pav. 1 Produkto vizija

<div> Vision</div> <p>This product offers a simple, easily accessible and convenient cloud storage solution for safe keeping one's videos.</p>		
<div> Target Group</div> <ul style="list-style-type: none">• Hobbyist filmmakers• An average person, who just wants to offload his videos from local storage to cloud	<div> Needs</div> <p>Once a person moves his videos to our product, he can then easily reach them from almost every device, which is connected to the Internet</p> <p>People who like to film their life's moments usually face with limited storage problems on their drives, so it would be great for them to offload some of the videos to cloud.</p>	<div> Product</div> <p>This product is a simple video library solution, which stores user's videos on cloud storage.</p> <p>This product stands out from others due to its convenience and ease of access.</p>

2. Produkto koncepcijos diagrama

Produkto įgyvendinimui pasirinkome naudoti React ir ASP.NET Core karkasus. React klientas kreipiasi į ASP.NET Core Web API, kuris naudodamasis EF Core karkasu bendrauja su MS SQL duombaze. Duomenų bazė saugo nuorodas (paths) į video failus saugykloje (storage).

pav. 2 Koncepcijos diagrama



3. Sistemas personas

Martin

Profile: Martin is a financial analyst who works a typical 9-5 job at a financial firm. Since childhood he likes to spend his free time taking videos with his camera, either with phone's or digital one. Martin enjoys filming his best moments during his travels, but also, he likes to sometimes capture heart-warming moments with his friends or any other interesting episodes from his day-to-day life.

Pain Points:

- *Since Martin captures so much video content almost every day, he needs to find a solution which could help him with storage problems. He has a limited amount of storage on his laptop and external drives.*
- *Martin sometimes wants to show one of his videos to his friends, but not always he has his laptop or drive nearby.*

Goals:

From these pain points we can conclude two main goals for Martin:

- Cloud storage for his videos
- Easily accessible and convenient online video library

Profile Attributes

Age: 27

Hobbies: Photography, filmmaking.

Job: Financial Analyst

Greta

Profile: Greta is an aspiring animator, who enjoys drawing and expressing herself through various means. As a freelance animator, her daily tasks at her job include creating short, animated videos for her clients.

Pain Points:

- *Since Greta creates lots of animated short videos, she wants to have a single place where she could conveniently store them.*
- *Greta has to constantly show the results of her creative work to her clients, so it would be great if she could have a convenient way to share them quickly.*

Goals:

From these pain points we can conclude two main goals for Greta:

- Cloud storage for her animated short videos, where she could conveniently manage them.
- A way to easily share her work to clients by sending a shareable link.

Profile Attributes

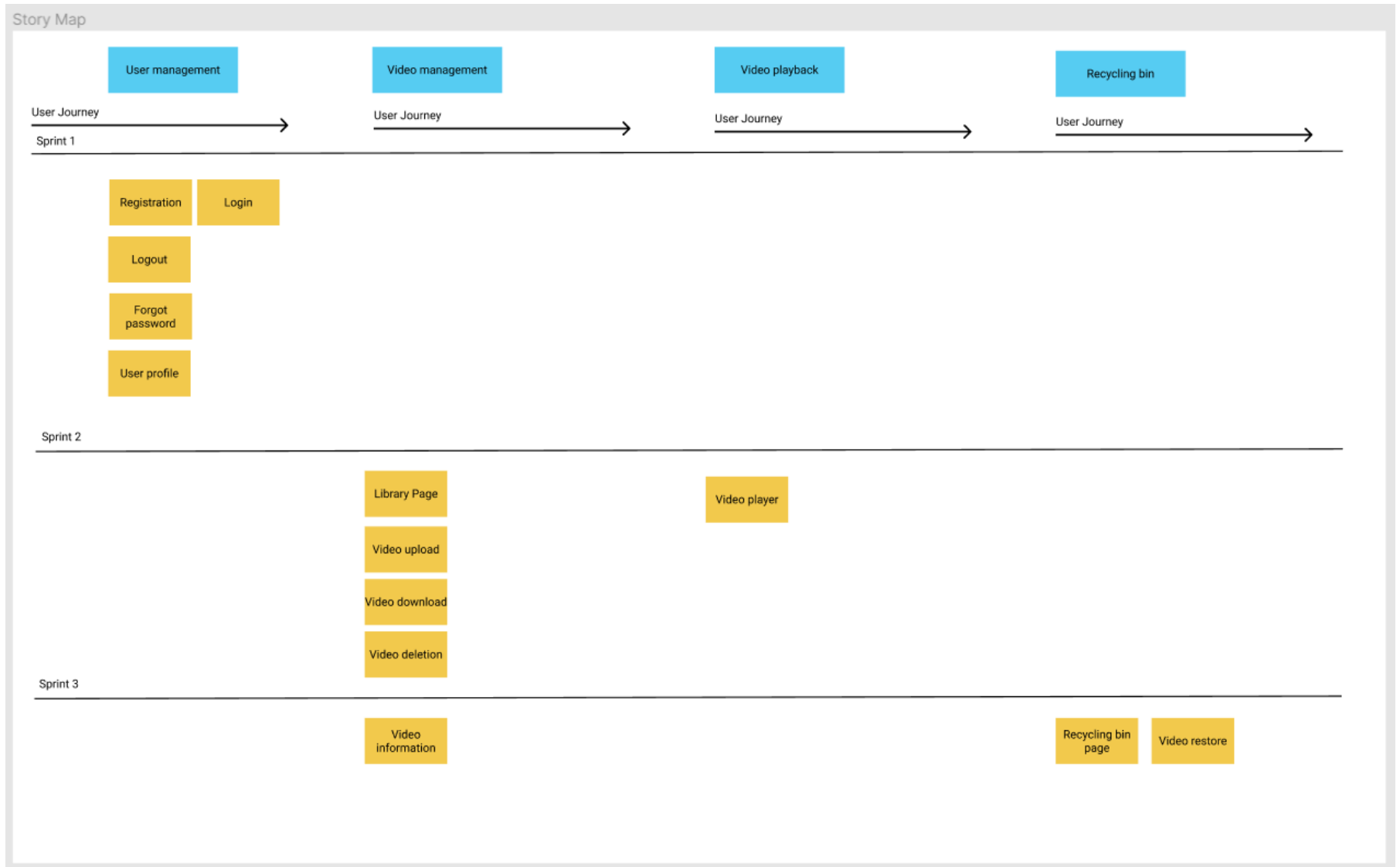
Age: 25

Hobbies: Drawing, hiking.

Job: Freelance animator

4. Story Map

pav. 3 Story Map



5. Prioretizuotas Backlog

Kadangi mūsų komanda darbą pradėjo anksčiau nei sužinojome, kad reikės pateikti pradinio prioretizuoto backlog screenshot, todėl apačioje esantis screenshot vaizduoja ne visą prioretizuotą backlog, t.y. tik tą backlog kuris buvo sprint 2 metu, tad į jį neįeina visi stories, kuriuos buvome suplanavę atlikti. Pateiktame backlog'e trūksta tik tų stories, kuriuos įgyvendinome sprint 1 metu: Login, Registration, Logout, User Profile. Taip yra todėl kad Jira, užbaigiant sprintą, automatiškai ištrina iš backlog tuos stories, kurie buvo praeitam sprinte, tad apačioje pavaizduotas backlog yra nepilnas.

pav. 4 Backlog antro sprinto metu

▼ WDB Sprint 2 4 May – 18 May (7 issues)		0 5 31	Complete sprint	...
WDB-4	Video library	8	DONE	
WDB-5	Video upload	8	DONE	
WDB-8	Video player	8	DONE	
WDB-6	Video deletion	5	DONE	
WDB-94	Video download	5	CODE REVIEW	
WDB-04	Forgot password	2	DONE	
WDB-50	Technical tasks		IN PROGRESS	
+ Create issue				

▼ Backlog (4 issues)		0 0 0	Create sprint	
WDB-21	Recycling bin			
WDB-9	Video information			
WDB-29	Navigation			
WDB-90	Reports and presentation			
+ Create issue				

Tam, kad galėtume pateikti visų suplanuotų stories sąrašą, Jira įrankyje įgalinome Issue Navigator funkciją, kuri leidžia peržiūrėti visų (taip pat ir jau užbaigtų) sprint stories.

Pastaba: šis Issue Navigator funkcijos pateiktas sąrašas **nėra prioretizuotas**, tad tai neatspindi mūsų komandos suplanuoto backlog prieš pradėdant pirmąjį sprintą, nors į jį įeina visi user stories.

pav. 5 Pilnas user stories sąrašas (neprioretizuotas)

Type	Key	Summary	Assignee	Reporter	P	Status
■	WDB-94	Video download	Unassigned	TM Tautvydas Mačiulis	↑	CODE REVIEW ▾
■	WDB-90	Reports and presentation	Unassigned	TM Tautvydas Mačiulis	↑	TO DO ▾
■	WDB-81	Forgot password	Unassigned	TM Tautvydas Mačiulis	↑	DONE ✓
■	WDB-50	Technical tasks	Unassigned	AA Aurimas Arlauskas	↑	IN PROGRESS ▾
■	WDB-29	Navigation	Unassigned	TM Tautvydas Mačiulis	↑	IN PROGRESS ▾
■	WDB-21	Recycling bin	Unassigned	TM Tautvydas Mačiulis	↑	IN PROGRESS ▾
■	WDB-12	User profile	Unassigned	TM Tautvydas Mačiulis	↑	DONE ✓
■	WDB-9	Video information	Unassigned	TM Tautvydas Mačiulis	↑	TO DO ▾
■	WDB-8	Video player	Unassigned	TM Tautvydas Mačiulis	↑	DONE ✓
■	WDB-7	Logout	Unassigned	TM Tautvydas Mačiulis	↑	DONE ✓
■	WDB-6	Video deletion	Unassigned	TM Tautvydas Mačiulis	↑	DONE ✓
■	WDB-5	Video upload	Unassigned	TM Tautvydas Mačiulis	↑	DONE ✓
■	WDB-4	Video library	Unassigned	TM Tautvydas Mačiulis	↑	DONE ✓
■	WDB-3	Registration	Unassigned	TM Tautvydas Mačiulis	↑	DONE ✓
■	WDB-1	Login	Unassigned	TM Tautvydas Mačiulis	↑	DONE ✓

6. Wireframes/Prototipas

Tam, kad sukurti wireframes ir galų gale prototipą, naudojome Figma įrankį. Sukurtus wireframes eksportavome į Zeplin įrankį, kurio pagalba galėjome patogiai prisegti wireframe prie atitinkamo user story Jira įrankyje. Ne visi stories turi savo wireframe. Taip yra dėl to, kad atitinkami wireframes apima daugiau nei vieną story.

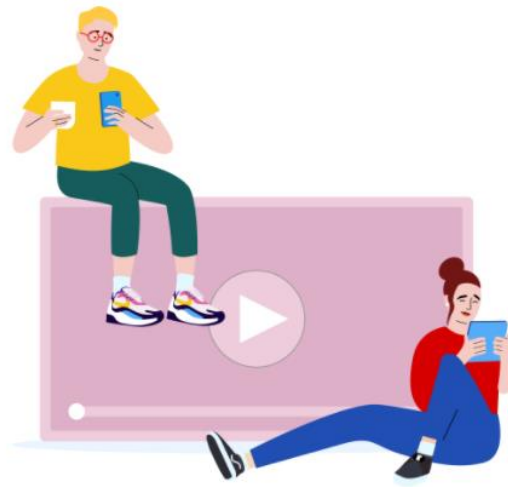
Prototipas yra pasiekiamas per šią nuorodą:

<https://www.figma.com/file/J8VIYdKcOAOAssEL1OPBkd/Prototype?node-id=0%3A1>

Pastaba: Tam, kad paleisti prototipą reikia paspausti Present mygtuką viršutiniame kairiame kampe. Patartina atsisiųsti Figma programą ir per ją pasileisti prototipą, nes kitu atveju, leidžiant per internetinę naršyklę, gali puslapis lėtai veikti.

1. Login user story wireframe

pav. 6 Login wireframe

A wireframe diagram of a login form. At the top is an orange square icon with a white play button. Below it is the word "Login" in a bold, black font. The form contains two input fields: "Email Address*" with the text "juris.jurgaitis@gmail.com" and "Password*" with a toggle eye icon. Below the password field is a link "Forgot your password?". There is a checkbox labeled "Remember me?". At the bottom are two buttons: a purple "LOGIN" button and a white "REGISTER" button with a purple border.

Login

Email Address*
juris.jurgaitis@gmail.com

Password*

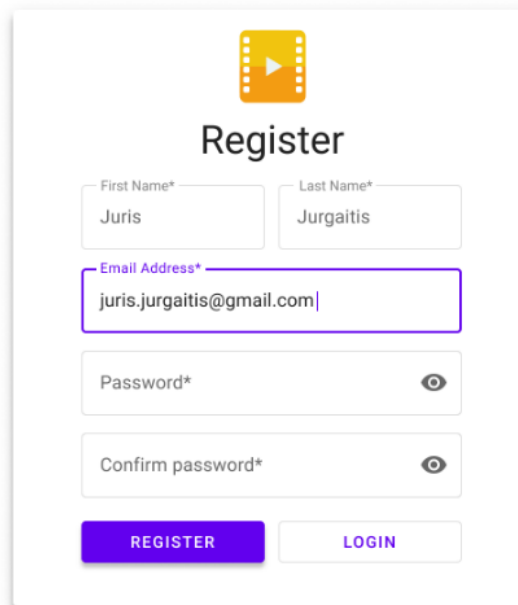
[Forgot your password?](#)

☐ Remember me?


LOGIN REGISTER

2. Registration user story wireframes

pav. 7 Registration page wireframe



The wireframe shows a registration form with a yellow play button icon at the top. Below the icon is the title "Register". The form contains five input fields: "First Name*" with the value "Juris", "Last Name*" with the value "Jurgaitis", "Email Address*" with the value "juris.jurgaitis@gmail.com", "Password*", and "Confirm password*". Each password field has an eye icon for toggling visibility. At the bottom are two buttons: a blue "REGISTER" button and a white "LOGIN" button.





Register

First Name*
Juris

Last Name*
Jurgaitis

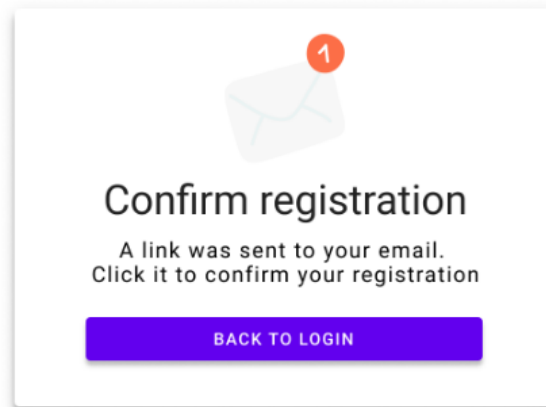
Email Address*
juris.jurgaitis@gmail.com

Password* 

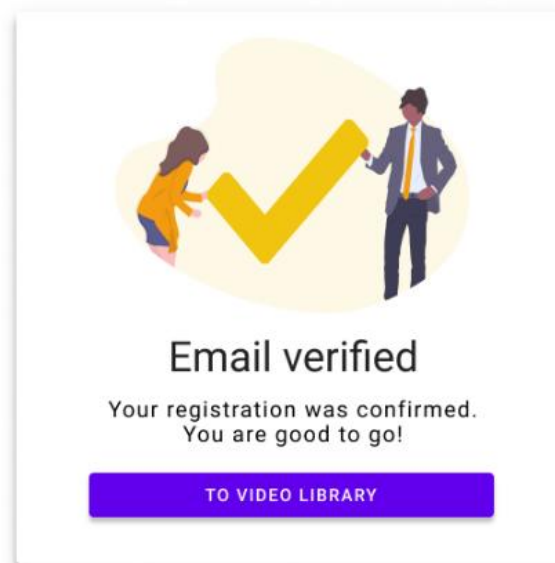
Confirm password* 

REGISTER LOGIN

pav. 8 Confirm registration page wireframe

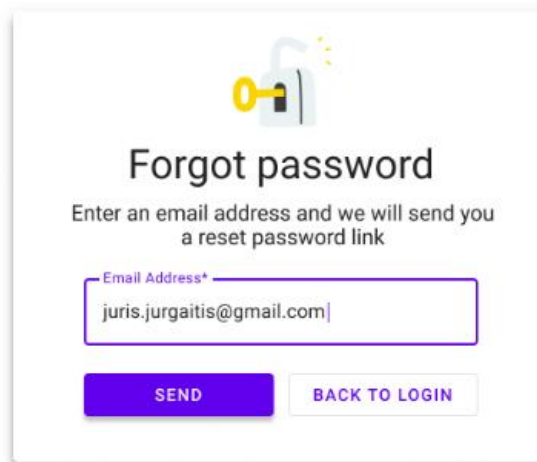


pav. 9 Email verified page wireframe




3. Forgot Password user story wireframes

pav. 10 Forgot password page wireframe



The wireframe shows a centered white box with a light gray border. At the top is an icon of a yellow key and a blue padlock. Below the icon is the title "Forgot password" in a large, bold, black font. Under the title is the instruction "Enter an email address and we will send you a reset password link" in a smaller black font. Below this is a text input field with a purple border. The placeholder text "Email Address*" is at the top left of the field, and the email address "juris.jurgaitis@gmail.com" is entered. At the bottom of the box are two buttons: a solid purple button labeled "SEND" and a white button with a purple border labeled "BACK TO LOGIN".



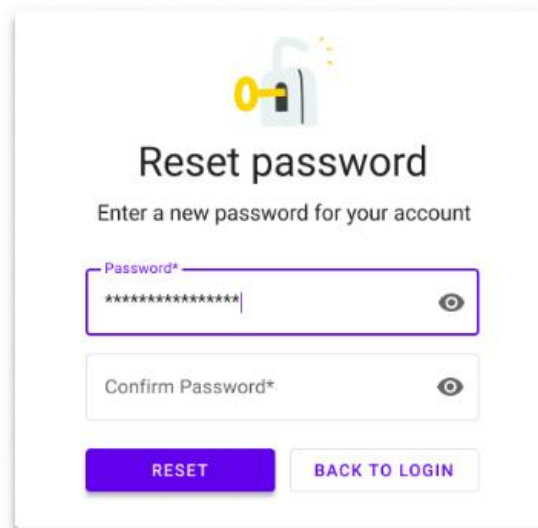
Forgot password

Enter an email address and we will send you a reset password link


Email Address*

[SEND](#) [BACK TO LOGIN](#)

pav. 11 Reset password page wireframe



A wireframe for a password reset page. At the top is an icon of an open padlock with a yellow key. Below it is the title "Reset password" in a large, bold font. Under the title is the instruction "Enter a new password for your account". There are two input fields: the first is labeled "Password*" and contains ten asterisks; the second is labeled "Confirm Password*" and is empty. Both fields have a toggle icon on the right. At the bottom are two buttons: a blue "RESET" button and a white "BACK TO LOGIN" button with a blue border.



Reset password

Enter a new password for your account

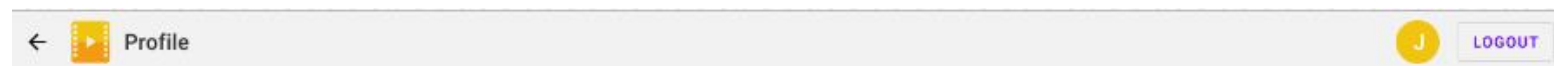
Password*


Confirm Password*

[RESET](#) [BACK TO LOGIN](#)

4. User profile story wireframe

pav. 12 User profile page wireframe



 **Juris Jurgaitis**

Email Address

Current Password

New Password

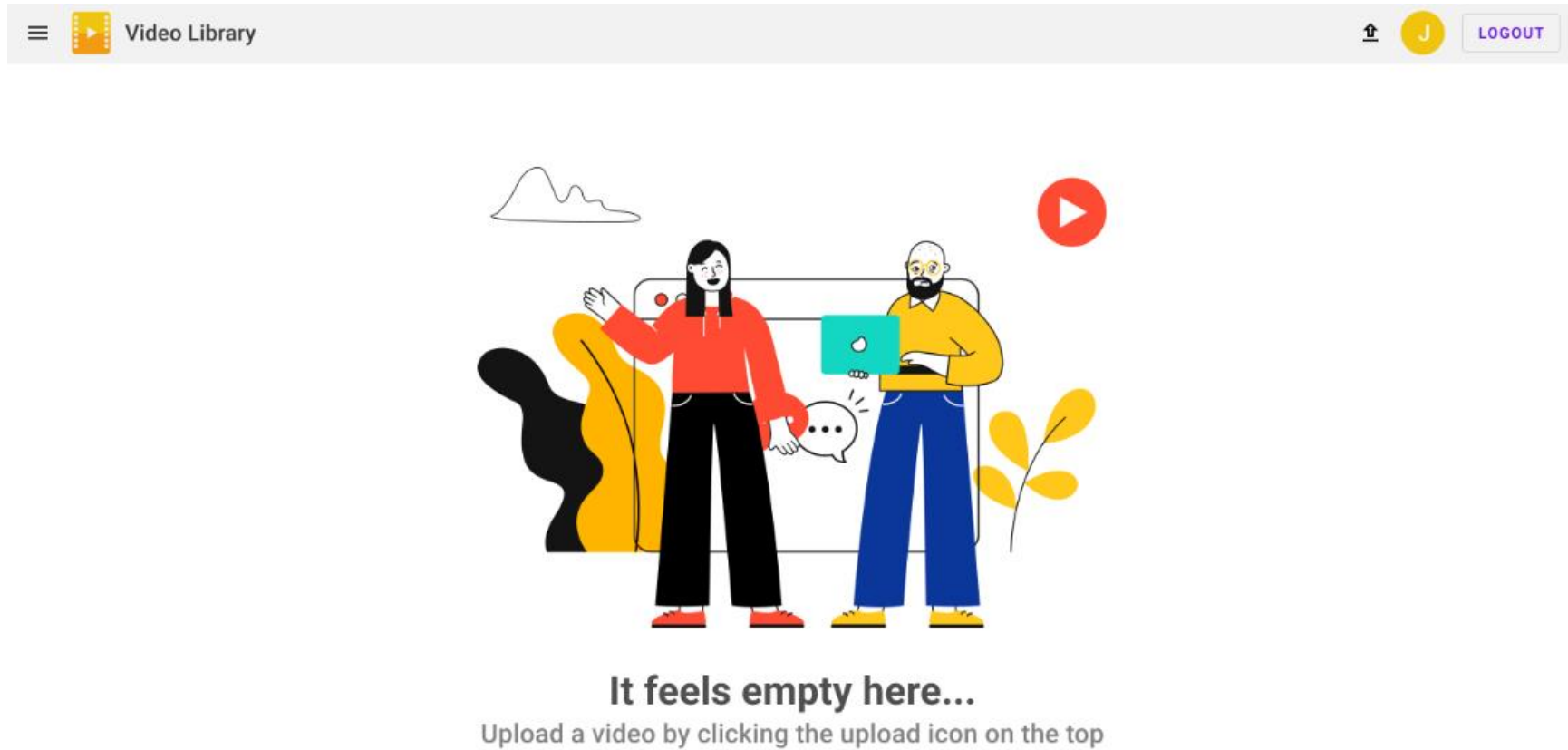
SAVE CHANGES

Changes saved successfully

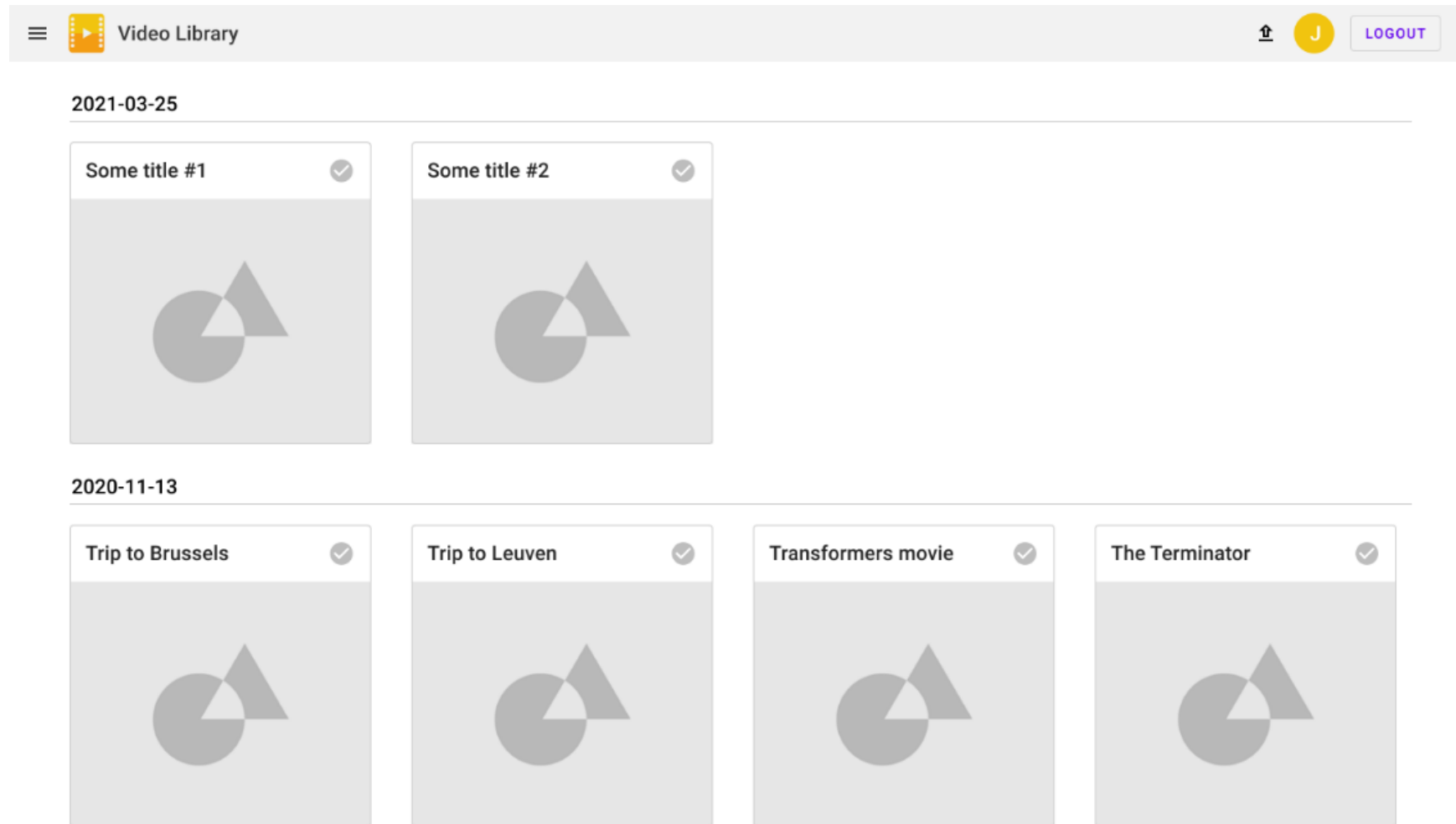
Something wrong happened :/ Changes were not saved

5. Video library story wireframes

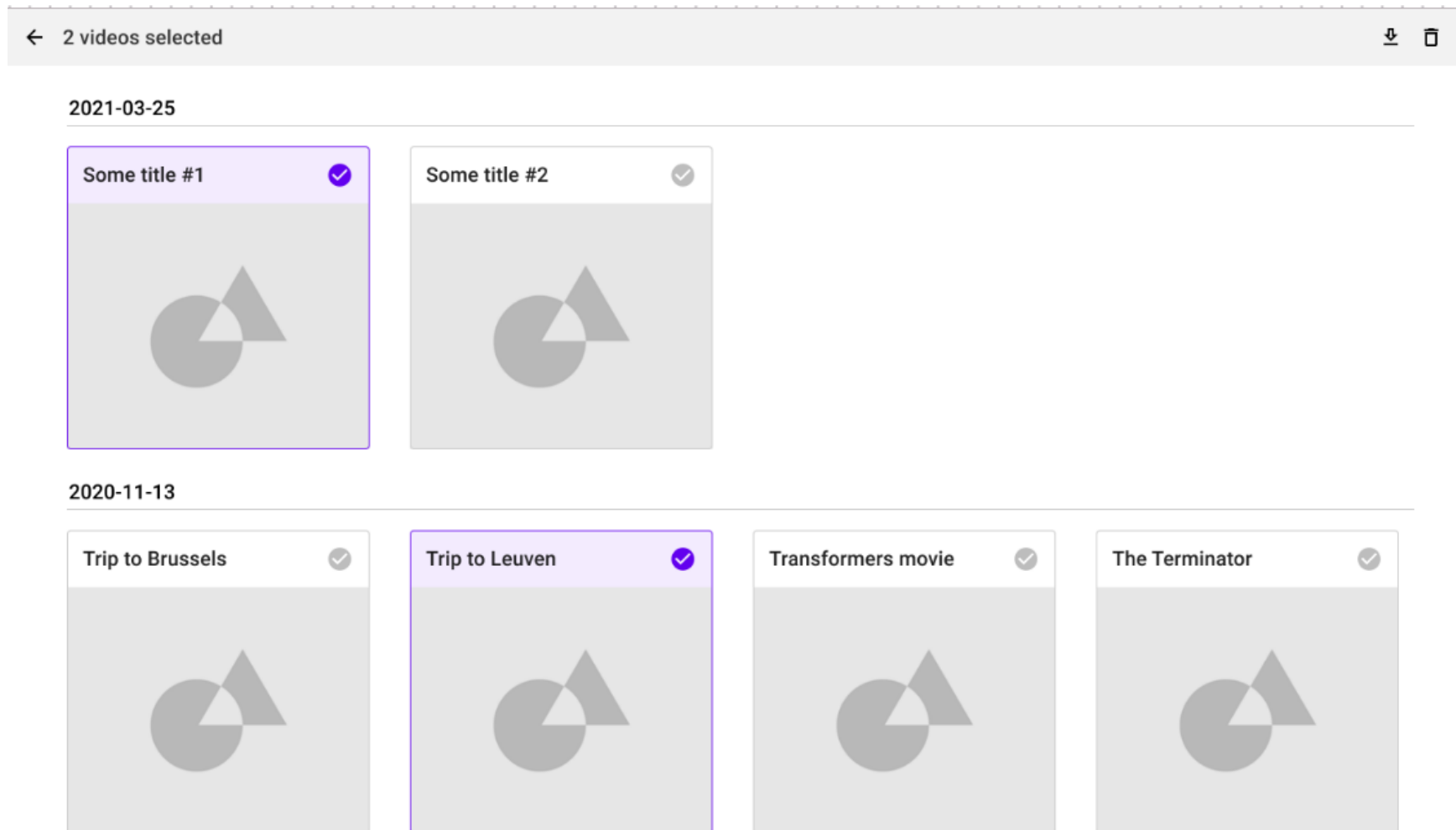
pav. 13 Empty video library page wireframe



pav. 14 Video library page with videos uploaded wireframe

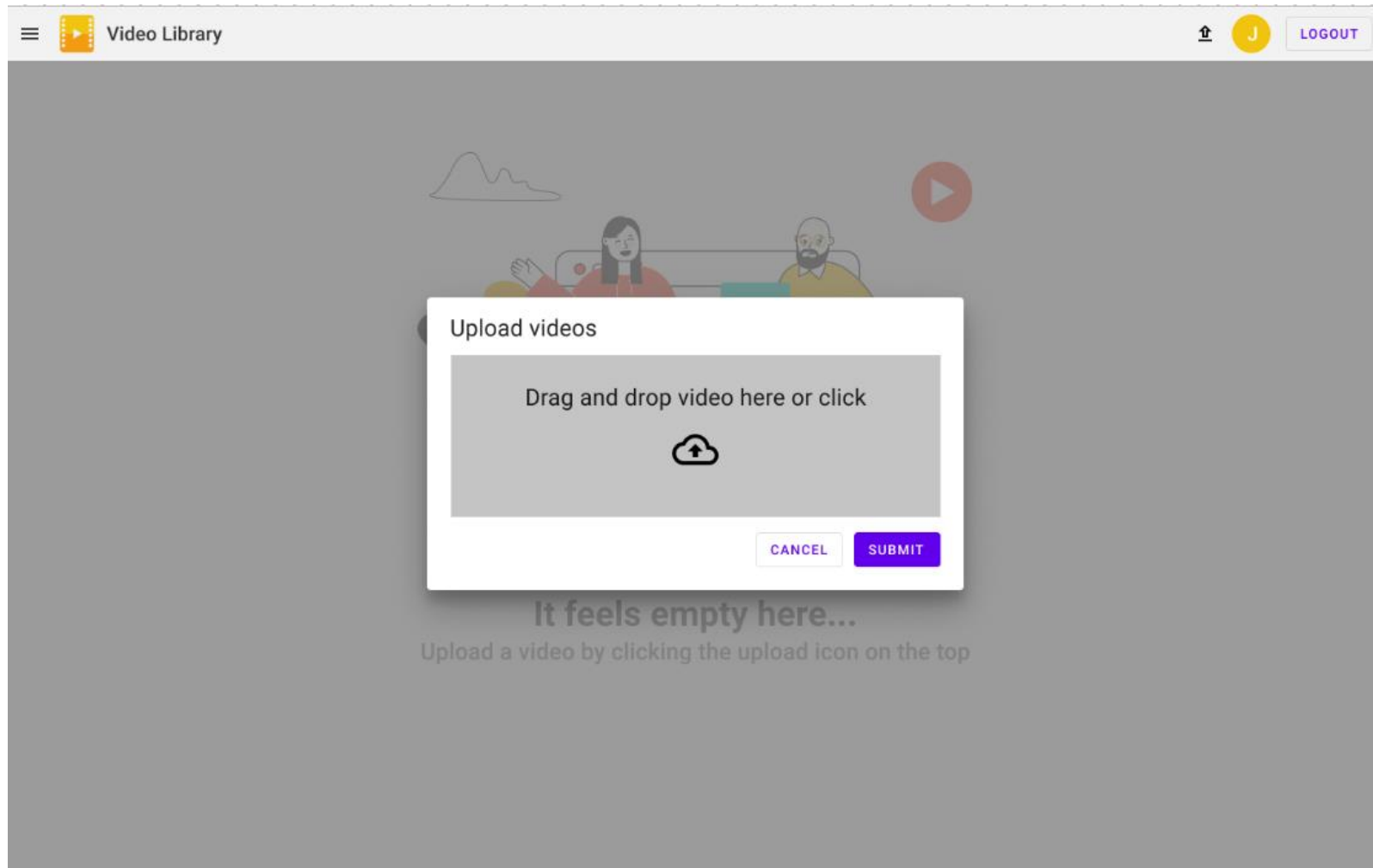


pav. 15 Video library page with videos selected wireframe

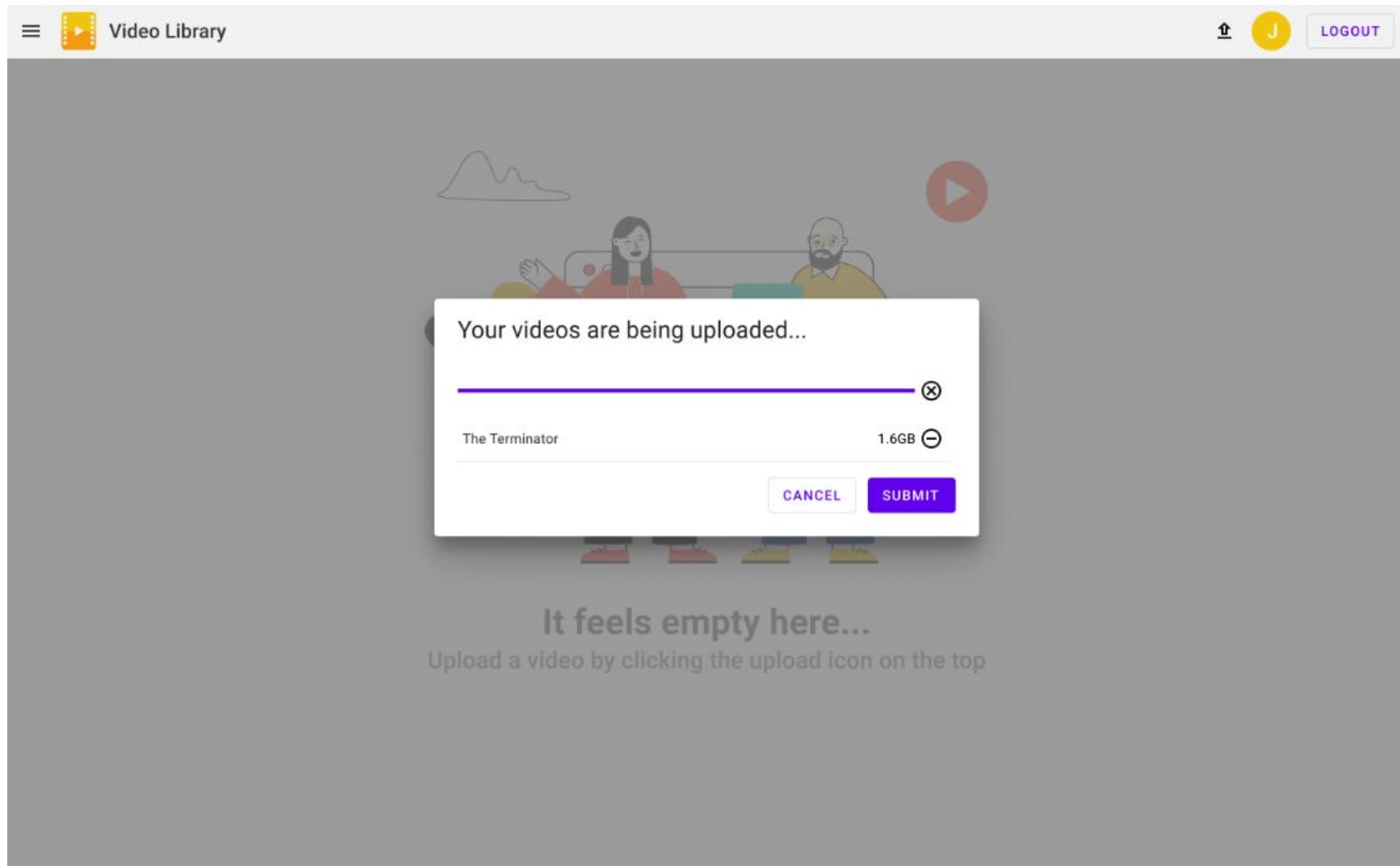


6. Video upload story wireframes

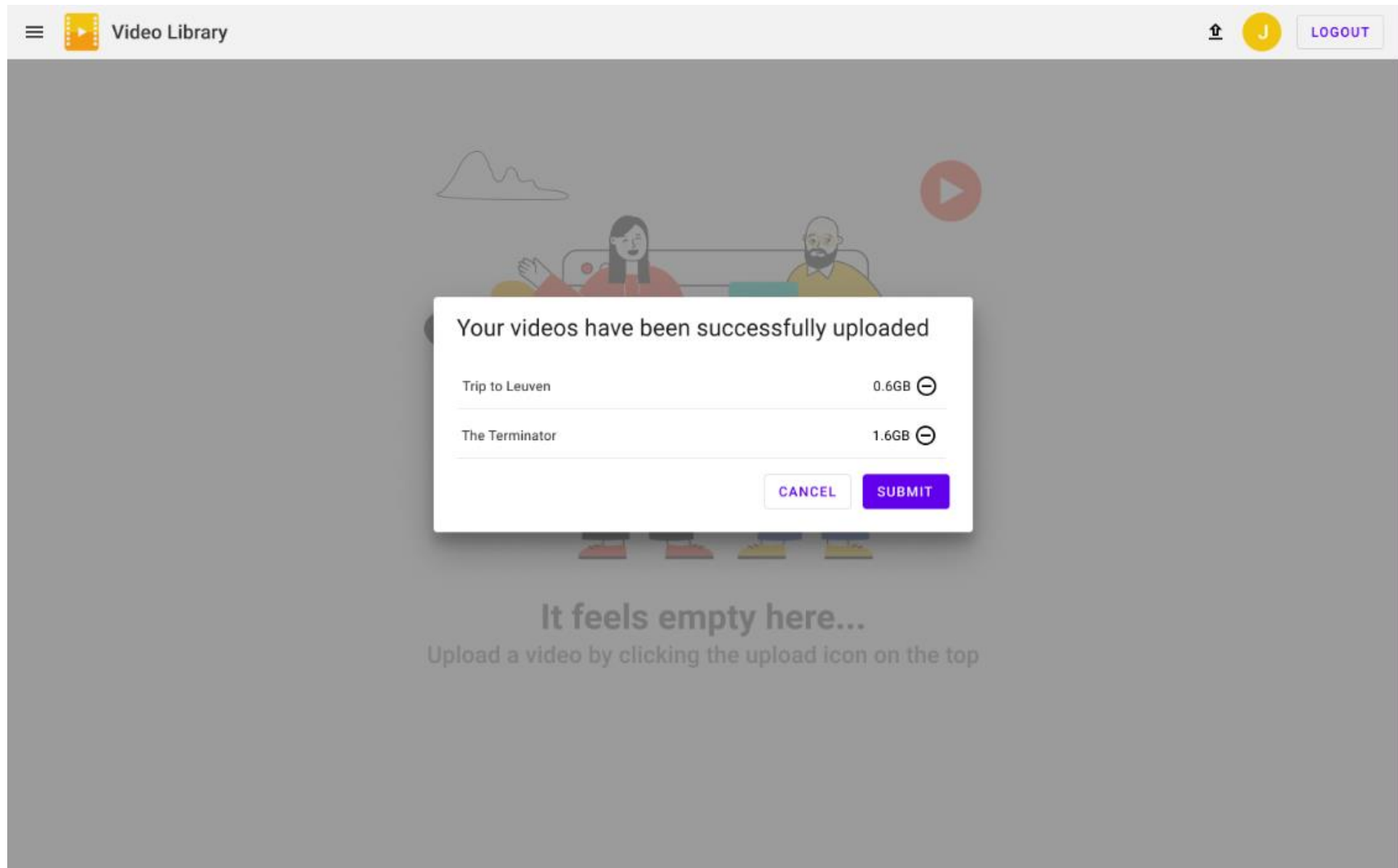
pav. 16 Upload modal with no videos



pav. 17 Upload modal with videos being uploaded

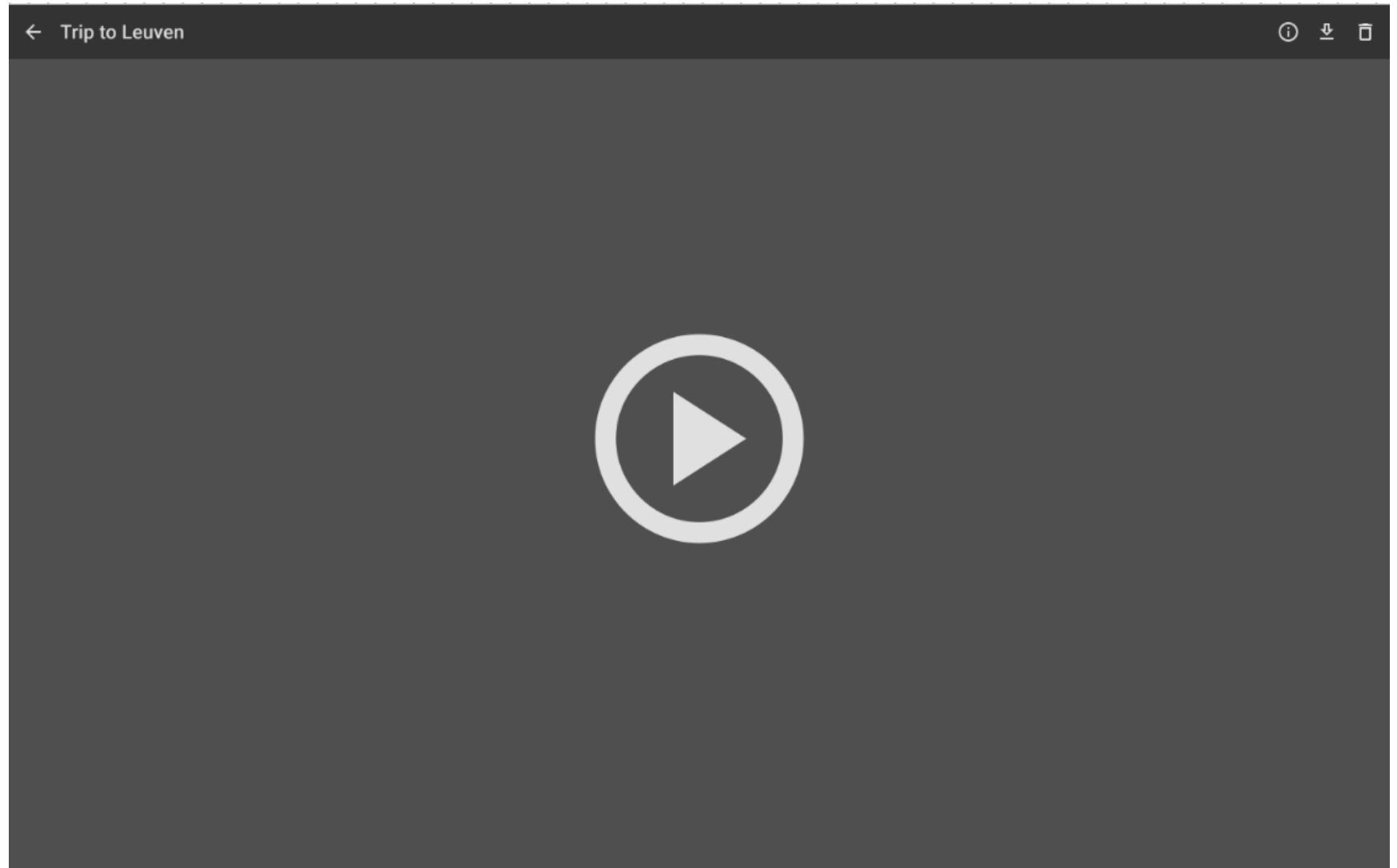


pav. 18 Upload modal with videos upoaded



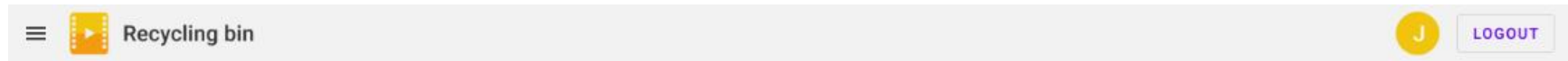
7. Video player story wireframe

pav. 19 Video player wireframe



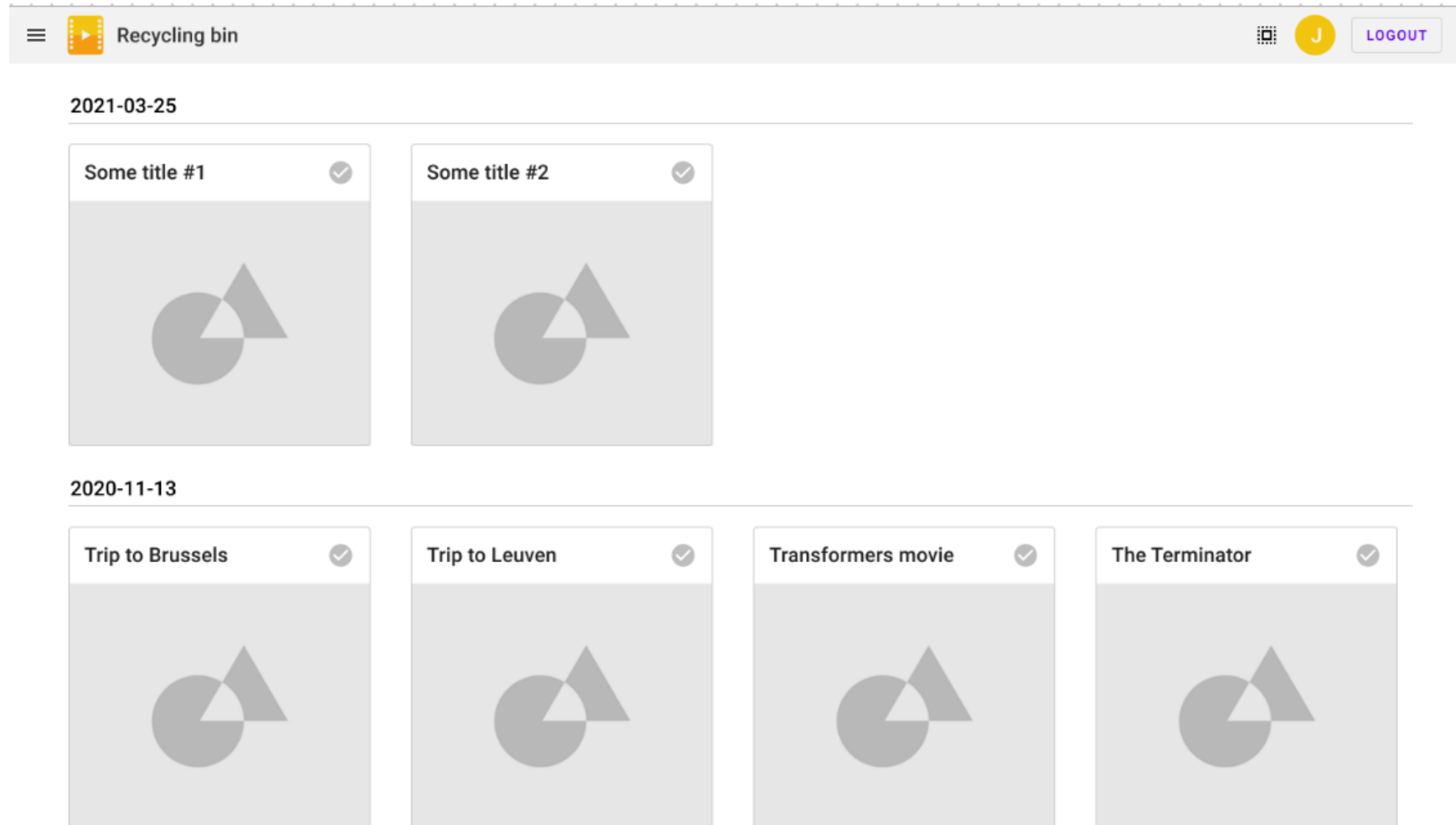
8. Recycling bin wireframes

pav. 20 Empty recycling bin page wireframe

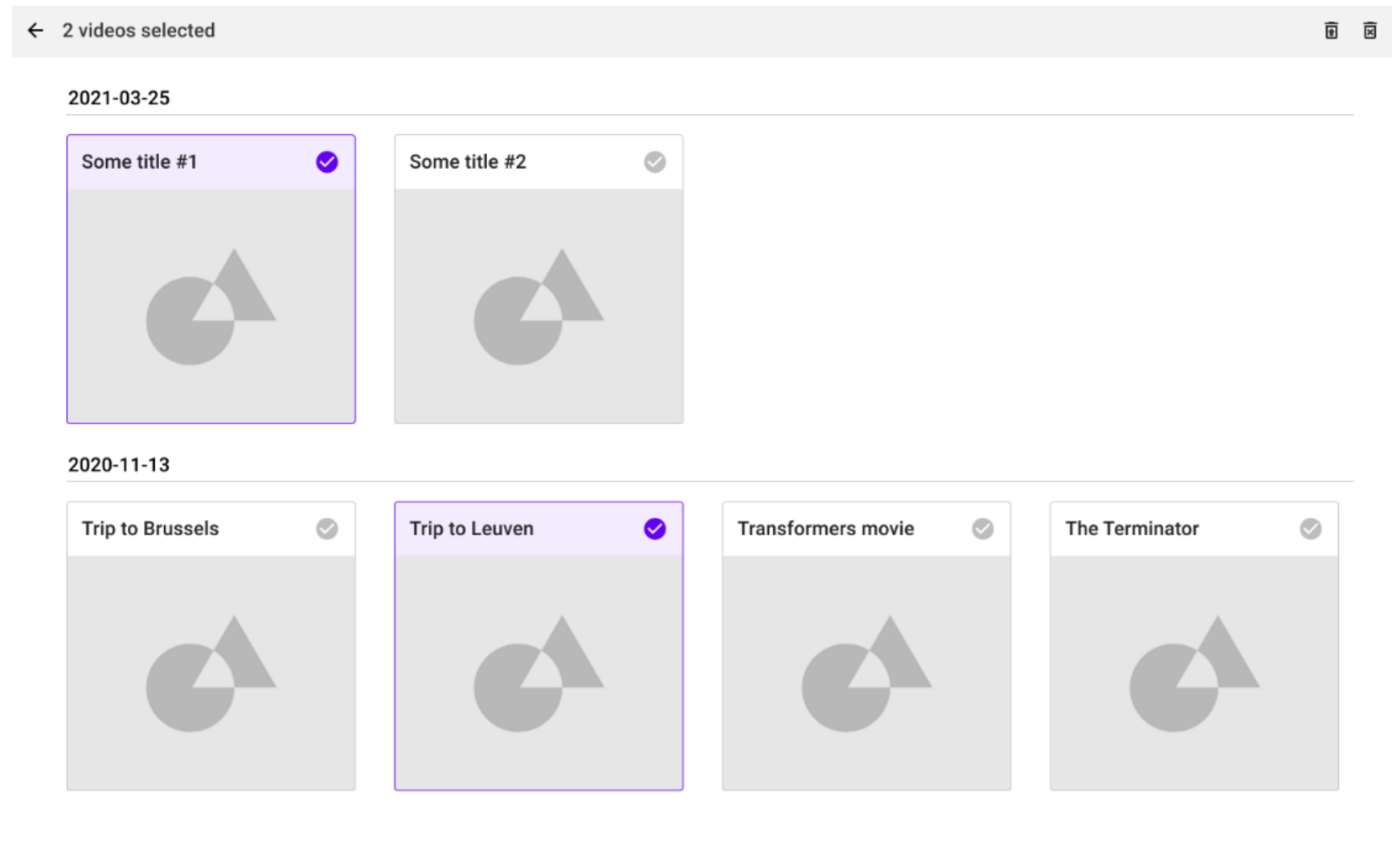


Recycling bin is empty

pav. 21 Recycling bin with deleted videos page wireframe

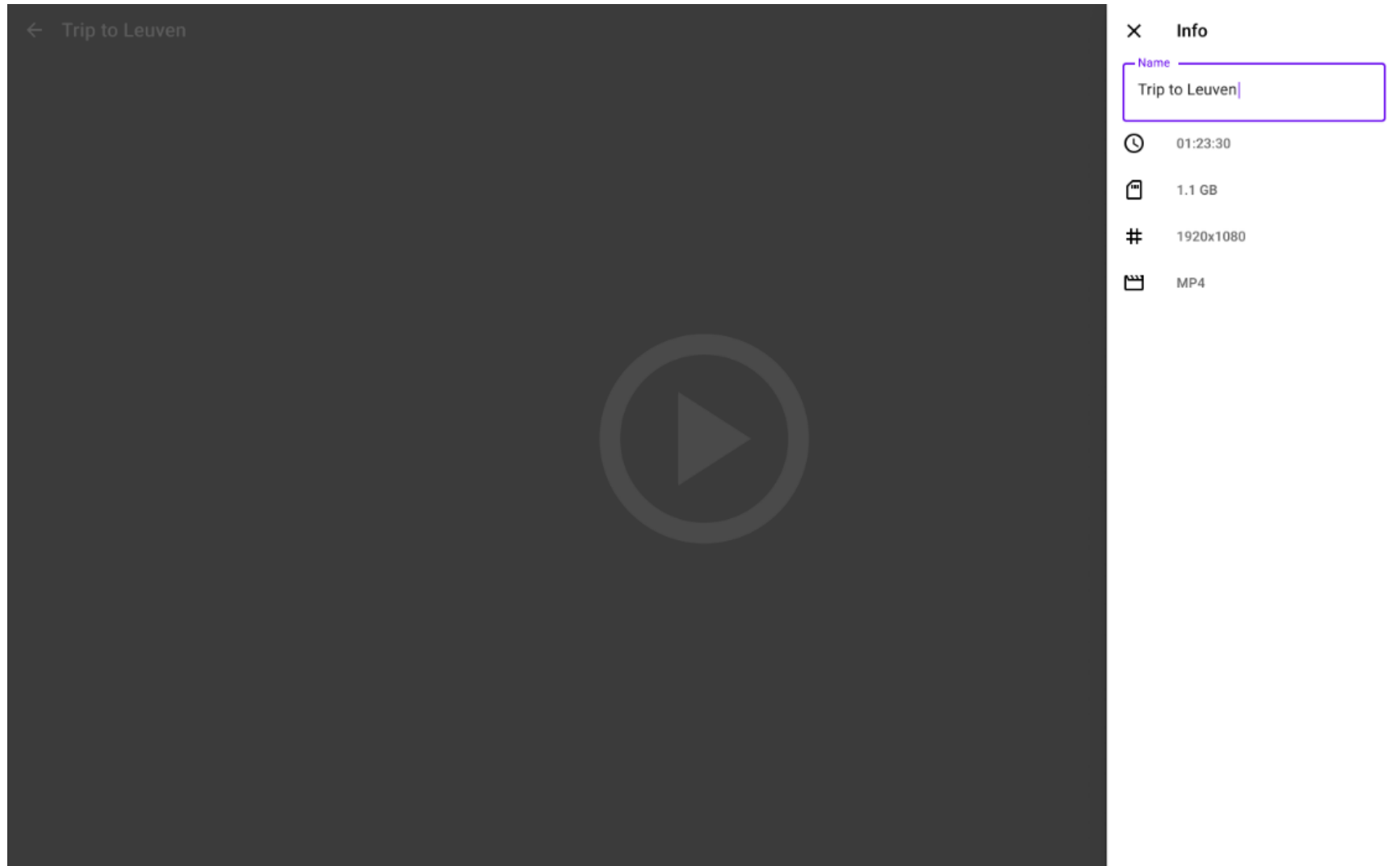


pav. 22 Recycling bin page with videos selected wireframe



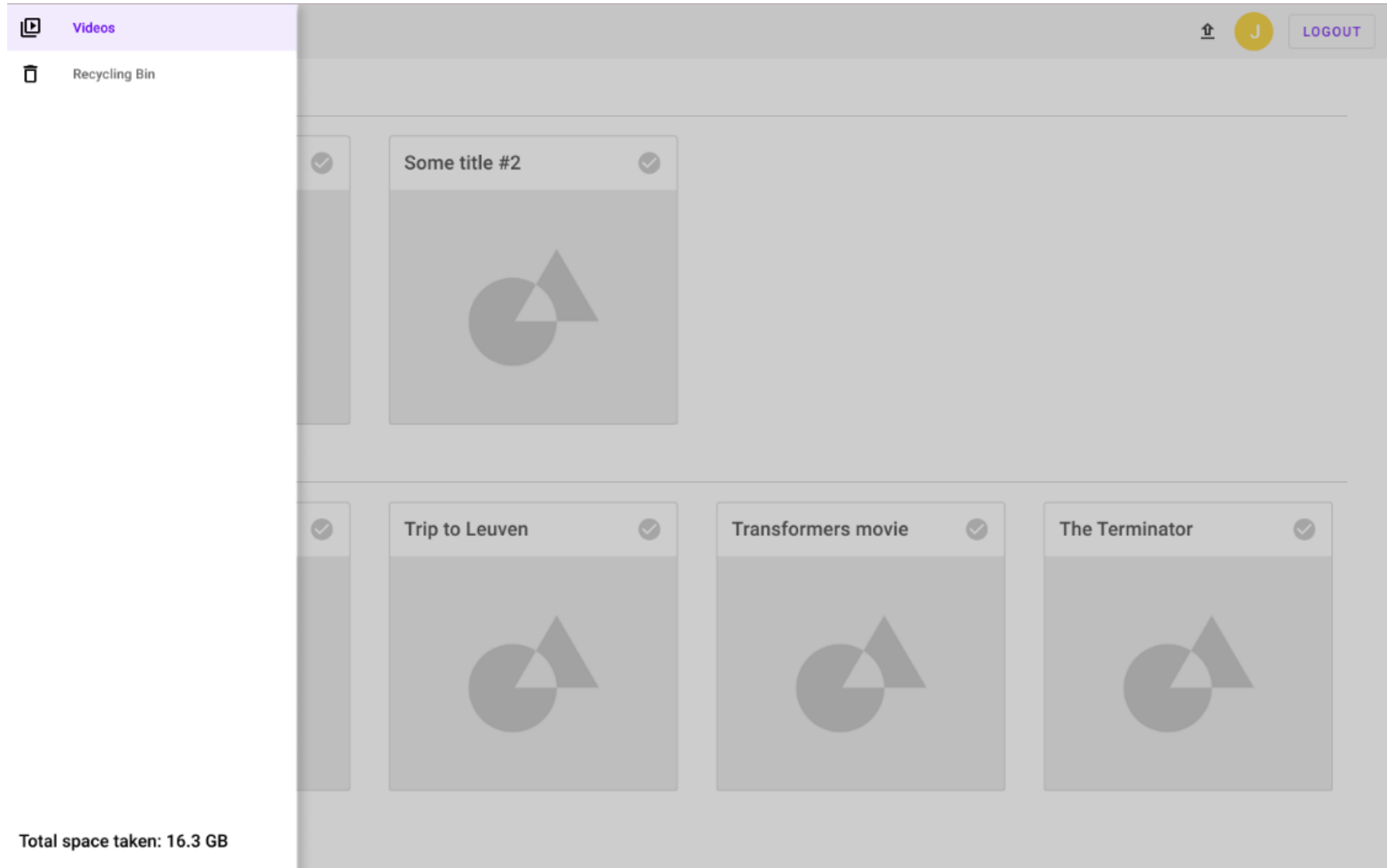
9. Video information story wireframe

pav. 23 Video information sidebar wireframe



10. Navigation story wireframe

pav. 24 Navigation drawer wireframe



7. Aprašyti user stories ir jų acceptance kriterijai

Kiekvieną user story aprašydavome naudodamiesi tokiu šablonu: „As a user I want to <do some action>, so I could <get some benefit/do something else>. Aprašę user story, Acceptance kriterijus surašydavome į sąrašą papunkčiui. Taip pat, prie kiekvieno Jira user story prisegdavome iliustracijų ir ikonų priedus bei pridėdavome nuorodą į atitinkamus wireframes Zeplin įrankyje. Galiausiai sprintų planavimo metu, kiekvieną story įvertindavome story points ir išskaidydavome į frontend ir backend užduotis, tad kiekvienas story turėdavo po atitinkamą kiekį sub-tasks.

Be to, projekto metu turėjome dar du papildomus Jira issues: Technical tasks ir Reports and presentation. Technical task issue migruodavo iš vieno sprinto į kitą, nes kiekvieno sprinto metu vis atsirasdavo kokių nors techninių užduočių, kaip tarkim patobulinti (refactor) kokį nors komponentą ir pan. Reports and presentation issue buvo susijęs su ataskaitų rašymu ir pan.

pav. 26 Registration user story

Registration

Attach

Add a child issue

Link issue

▼

Description • Unsaved changes

As a user I want to register, so I could login to the system later on.

Acceptance criteria:


1. All fields should have proper regex and should indicate the user, when the entered details are incorrect. **Password requirements:** at least 8 symbols with at least one capital letter and at least one number.
2. Registration form should include these fields: First name, Last name, Email, Password, Confirm Password.
3. Registration form should include Back to Login button.
4. On successful registration an email should be sent to the user, which includes a link. Also user should be redirected to confirm email page, which includes a back to login button.
5. On click of the confirmation link, user should be logged in and directed to email verified page. Email verified page should include go to video library button.

Registration page: Zeplin - Projects


Confirm email page: Zeplin - Projects

Email verified page: Zeplin - Projects

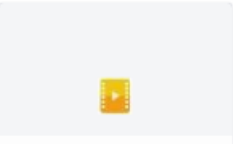
Attachments (3)



ill-confirmation.svg
29 Apr 2021, 04:35 PM



ill-mail.svg
29 Mar 2021, 05:43 PM



product-icon.svg
25 Mar 2021, 06:44 PM

pav. 25 Registration user story sub-tasks


Child issues


100% Done


WDB-28	Setup API call for "Register" button	AA	DONE
WDB-85	Create email verified page	AA	DONE
WDB-34	Create endpoint for email verification	AA	DONE
WDB-27	Create endpoint for registration	AA	DONE
WDB-84	Create confirm email page	TM	DONE
WDB-59	Setup EmailSender	SV	DONE
WDB-18	Setup registration form	TM	DONE

pav. 28 Login user story

Login

 Attach

 Add a child issue

 Link issue



Description


As a user I want to login to the system, so I could access all of it's features.


Acceptance criteria:

1. Login page should have button for registration
2. On successful login, user should be directed to main page.
3. Login form should include fields for email, password.
4. All fields should have proper regex and should indicate the user, when the entered details are incorrect.
5. Login form should include checkbox "Remember me?", which should save login details.
6. On successful login response should include JWT token.





Login page:  [Zeplin - Projects](#)

Attachments (2)


ill-video-time.svg
25 Mar 2021, 05:28 PM


product-icon.svg
25 Mar 2021, 09:02 AM

pav. 27 Login user story sub-tasks

 WDB-13	Setup login page Login API call	AA	DONE ▼
 WDB-76	Setup controller for Login API call	AA	DONE ▼
 WDB-11	Setup login page form	TM	DONE ▼
 WDB-10	Setup login page	J	DONE ▼

pav. 29 Logout user story and its sub-tasks

Logout

 Attach  Add a child issue  Link issue 

Description

As a user I want to logout, so I could end my session.





Acceptance criteria:

1. When user is logged in, then there should be a Logout button on the appbar.
2. On Logout button click, then the cookies should be cleared and user should be directed to Login page.

Logout button in library page: <https://zpl.io/V0vKAdO>

Child issues

... +

<div></div>		100% Done
	WDBB-17 Logout button clears localStorage and refreshes the page	 DONE ✓
	WDBB-14 Create a logout button and App bar	 DONE ✓

pav. 30 User profile story and its sub-tasks

User profile

 Attach  Add a child issue  Link issue 

Description

As a user I want to view my profile, so I could see profile details and change some of them.

Acceptance criteria:

1. Profile page should include these fields: full name, password, confirm password.
2. Password and confirm password fields should be editable.
3. There should be Save changes button, which on click should update the profile details and user should be notified by a snackbar.

 Zeplin - Projects

Child issues

... +

100% Done

 W000-00 Add snackbars to inform users if the changes were saved successfully or not.	 DONE 
 W000-01 Implement "Save" button functionality	 DONE 
 W000-02 Setup user profile page	 DONE 

pav. 32 Forgot password user story

Forgot password

Attach

Add a child issue

Link issue

▼

Description

As a user I want to reset my password, so I could access my account again when I forget my password.

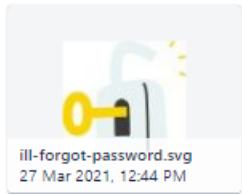
Acceptance criteria:

- 1. Login form should include button "Forgot your password?", which directs user to forgot password form.
- 2. When user enters his email and clicks button send, system should send to the provided email a link, which on click would redirect user to reset password page.
- 3. Reset password form should have same regex as registration form's password fields.
- 4. On successful reset, user should be directed logged in and redirected to library page.

Forgot password page: Zeplin - Projects

Reset password page: Zeplin - Projects

Attachments (1)



pav. 31 Forgot password user story sub-tasks

Child issues		... +	
		100% Done	
WDB-09	Create reset password page	J	DONE▼
WDB-08	Setup reset password endpoint	J	DONE▼
WDB-06	Create forgot password page	J	DONE▼
WDB-07	Setup forgot password endpoint	J	DONE▼
WDB-15	Setup login page "Forgot password?" button redirection	J	DONE▼

pav. 34 Video library user story

Video library

Attach

Add a child issue

Link issue

Description

As a user I want to access library page, so I could see all my videos.

Acceptance criteria:

- 1. If there are no videos, then empty page should be shown.
- 2. If there are videos, then they should be sorted by upload date.
- 3. Video card should show a frame from the video. If frame is not accessible, then it should show placeholder image.
- 4. Each video card should be selectable. Multiple selection should be possible.
- 5. It should be possible to select all videos of particular date.
- 6. When video card is selected, then it should be possible to download or delete selected videos.(At the moment show only the icons on appbar, these functionalities will be implemented in other stories).
- 7. On click of video card, video player should be opened.(Implemented in Video Player story)


Empty library page: <https://zpl.io/V0vKAdO>


Library page: <https://zpl.io/2v3DXM5>

Library page with videos selected: <https://zpl.io/a34mGNY>






Attachments (6)


product-icon.svg
25 Mar 2021, 07:00 PM


arrow-back.svg
25 Mar 2021, 06:58 PM




upload.svg
25 Mar 2021, 06:53 PM

pav. 33 Video library user story sub-tasks

 WDB-109	Create endpoint to fetch all user's videos(not files, only entities)	AA	DONE
 WDB-110	Create API call in frontend to fetch all user's videos	AA	DONE
 WDB-108	Sort video cards by upload date	J	DONE
 WDB-60	Setup video cards, should be filled with a screenshot from the video and have the title ...	J	DONE
 WDB-95	Create empty library page	J	DONE

pav. 36 Video upload user story

Video upload


 Attach  Add a child issue  Link issue 


Description • Unsaved changes


As a user I want to upload a video from my local storage, so it would be accessible in video library.

Acceptance criteria:

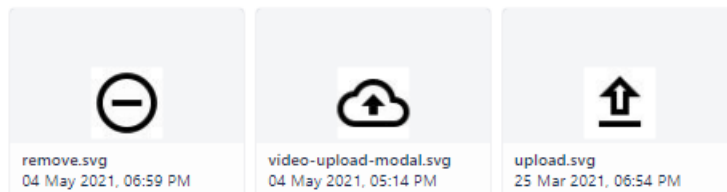
1. Upload process should begin when Upload button on the appbar is clicked.
2. When upload button is clicked, then file upload modal should be opened.
3. It should be possible to drag a file from local storage to modal.
4. Also, it should be possible to open local file explorer where user can manually select which file to upload.
5. Modal should include Submit button, which finalizes the upload process and refreshes the page. Also, it should include cancel button.
6. On successful/failed upload, user should be notified about it.
7. Modal should reject files which are not of video type.

Upload modal with no videos:  [Zeplin - Projects](#)

Upload modal upload in progress:  [Zeplin - Projects](#)

Upload modal with videos uploaded:  [Zeplin - Projects](#)

Attachments (3)







pav. 35 Upload user story sub-tasks

Child issues

... +

100% Done

	WDB-98 Create storage/folder structure for videos	AA	DONE ✓
	WDB-103 Create upload endpoint	AA	DONE ✓
	WDB-104 Make Upload API call in frontend	TM	DONE ✓
	WDB-32 Create upload modal	TM	DONE ✓

pav. 37 Video download user story and sub-tasks

Video download



Attach



Add a child issue



Link issue



Description

As a user I want to download videos, so I could have it in my local storage

Acceptance criteria:

1. It should be possible to download selected videos from library page
2. It should be possible to download multiple videos, when more than one is selected in library page.
3. It should be possible to download video from video player.

Child issues



100% Done



WDB-106 Add download button in appbar where necessary



DONE ✓



WDB-97 Create download flow



DONE ✓



WDB-101 Add informational snackbars/loading bars



DONE ✓



WDB-96 Create Download endpoint



DONE ✓

pav. 39 Video deletion user story

Video deletion

 Attach

 Add a child issue

 Link issue



Description


As a user I want to delete my videos, so I could free up my storage in the video library.

Acceptance criteria:


- 1. When video card in the library page is selected, then it should be possible to delete the video by clicking Delete button on appbar.
- 2. Also it should be possible to delete a video from video player page.
- 3. When Delete button is clicked, then a dialog should appear, which asks the user to confirm the deletion of video.


Pages where deletion is possible:

 [Zeplin - Projects](#)

 [Zeplin - Projects](#)

Attachments (2)














arrow-back.svg
25 Mar 2021, 06:59 PM


delete.svg
25 Mar 2021, 06:56 PM

pav. 38 Video deletion user story sub-tasks

Child issues

100% Done

 WDB-50	Make API call for video deletion		DONE 
 WDB-42	Create delete confirmation dialog		DONE 
 WDB-46	Add delete button in appbar where necessary		DONE 
 WDB-49	Create endpoint for video deletion		DONE 

pav. 40 Video player user story and its sub-tasks

Video player

Attach

Add a child issue

Link issue

▼

Description

As a user I want to open my videos in the system, so I could watch them.


Acceptance criteria:


- 1. On video card click in the library page, a video player should be opened.
- 2. In video player, it should be possible to play, delete, download the video and it should be possible to view information of it.(Implemented in other story)
- 3. On video deletion, it should be moved to recycling bin and user should be directed to library page. Snackbar should also show up, notifying the user whether the deletion was successful.


 Zeplin - Projects


Attachments (4)

... +


arrow-back.svg
25 Mar 2021, 06:58 PM


delete.svg
25 Mar 2021, 06:58 PM



download.svg
25 Mar 2021, 06:58 PM


info.svg
25 Mar 2021, 06:58 PM

Child issues


... +

100% Done

 WDB-105 Find video streaming solution from storage/backend to frontend

AA

DONE ▼

 WDB-65 Create a page for video viewing

AA

DONE ▼

pav. 42 Recycling bin user story

Recycling bin

Attach

Add a child issue

Link issue

Description • Unsaved changes

As a user I want to have a recycling bin page, so I could restore my deleted files or delete them forever.

Acceptance criteria:


- 1. If there are no videos recycling bin page, then empty page should be shown.
- 2. If there are videos, then they should be sorted by delete or upload date(Feel free to choose which way you think is the best).
- 3. It should be possible to select all videos.
- 4. On video card selection, it should be possible to restore the video or videos, or to delete it/them forever.
- 5. When user clicks delete forever, then confirmation dialog should be shown.
- 6. If videos stay for 30 days or longer in the recycling bin, then they should be deleted forever automatically.

Empty recycling bin page: <https://zpl.io/bzQ4DKA>

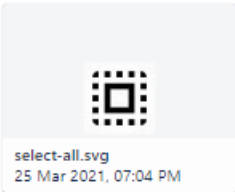
Recycling bin page: <https://zpl.io/VxAE5Nx>

Recycling bin page with videos selected: <https://zpl.io/addZ8Xe>

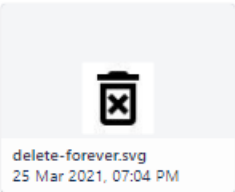
Attachments (4)



ill-recycling-bin.svg
16 May 2021, 03:27 PM



select-all.svg
25 Mar 2021, 07:04 PM



delete-forever.svg
25 Mar 2021, 07:04 PM

pav. 41 Recycling bin user story sub-tasks

Child issues

33% Done

WDB-113	Get all user's deleted videos endpoint	P	DONE
WDB-43	Restore videos endpoint	P	DONE
WDB-40	Create empty recycling bin page	TM	DONE
WDB-48	Implement auto deletion after 30 days(Possibly a service worker)		TO DO
WDB-112	Create recycling bin page with videos	J	IN PROGRESS
WDB-41	Select all videos functionality in frontend		TO DO
WDB-45	Reuse DeleteConfirmationDialog for delete forever		TO DO
WDB-114	Reuse previous delete endpoint for delete forever in Recycling bin		TO DO
WDB-127	Make API calls in Recycling bin page		TO DO

pav. 43 Video information user story and its sub-tasks

Video information

 Attach  Add a child issue  Link issue 

Description



As a user I want to view video information, so I could see video's details and edit some of it.


Acceptance criteria:


1. Video information side bar should include fields like: title, duration, size, resolution, format.
2. Title field should be editable. It should automatically save changes on each edit.


 [Zeplin - Projects](#)


Attachments (5)


 


format-type.svg
25 Mar 2021, 07:02 PM




resolution.svg
25 Mar 2021, 07:02 PM


storage.svg
25 Mar 2021, 07:02 PM














close.svg
25 Mar 2021, 07:01 PM


info.svg
25 Mar 2021, 07:01 PM

Child issues


 


0% Done

 WDB-117	Expand video entity with necessary properties(duration, resolution, etc.)	 CODE REVIEW 
 WDB-118	Information sidebar with editable title field	 CODE REVIEW 
 WDB-133	Extract information from video	 CODE REVIEW 
 WDB-136	Connecting frontend and backend	 CODE REVIEW 

pav. 44 Navigation user story and its sub-tasks

Navigation

 Attach

 Add a child issue

 Link issue



Description

As a user I want to navigate between pages, so I could reach either recycling bin or library page.

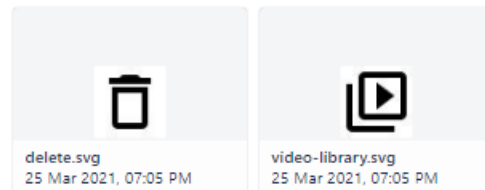
Acceptance criteria:

1. Navigation drawer should include two tabs: Videos, Recycling Bin.
2. Navigation drawer should also show how much storage is taken.

 Zeplin - Projects





Attachments (2)

... +



Child issues

... +

<div></div> 100% Done	
 WDB-132 Make API call in LibraryPage and RecyclingBinPage	AA DONE ▾
 WDB-131 Create endpoint to fetch user's taken space amount(marked for deletion videos as well)	AA DONE ▾
 WDB-122 Show user's storage amount in Navigation Drawer	AA DONE ▾
 WDB-119 Navigation drawer component	TM DONE ▾

pav. 45 Technical tasks issue and its sub-tasks

Technical tasks

Attach

Add a child issue

Link issue

Description

Add a description...

Child issues

94% Done

WDB-135	Create Axios interceptor for handling 401 responses from API	J	DONE
WDB-134	Harden stream endpoint	J	DONE
WDB-129	Refactor folder creation in backend for video storage	TM	DONE
WDB-128	Refactor upload modal to enable multiple files upload	TM	DONE
WDB-111	Split back-end to business logic and data access layer	SV	DONE
WDB-56	Setup email sender	SV	DONE
WDB-130	Logging	SV	IN PROGRESS
WDB-61	Configure authentication in backend	SV	DONE
WDB-51	Create and setup react router	J	DONE
WDB-82	Setup Docker containers for development	J	DONE
WDB-62	Configure password encryption	J	DONE

pav. 46 Reports and presentation issue and its sub-tasks

Projects /  We Don't Byte /  Add epic /  WDB-90

Reports and presentation

 Attach

 Add a child issue

 Link issue



Description

Add a description...

Child issues

... +

0% Done

 WDB-91 Write technical report	 TO DO ▾
 WDB-92 Write TSPI report	 IN PROGRESS ▾
 WDB-125 Write document for BA	 IN PROGRESS ▾
 WDB-93 Create presentation	 TO DO ▾

8.Jira sprint board struktūra ir užduočių būsenos kaita

Mūsų komanda nusprendė, kad Jira sprint board turėtų susidėti iš keturių stulpelių: To Do, In Progress, Code Review ir Done. Kiekvienas story ir jų sub-task'ai projekto eigoje migruodavo iš To Do iki Done, priklausomai nuo to kokioje stadijoje jie yra:

pav. 47 Jira sprint board

Projects / We Don't Byte

WDB Sprint 2

⚡ ☆ ⌚ 5 days remaining Complete sprint ⋮

🔍 TM J AA SV | Label ▾ GROUP BY Subtask ▾

TO DO 19 ISSUES	IN PROGRESS	CODE REVIEW 2 ISSUES	DONE 20 ISSUES ✓
<div>➤ WDB-4 Video library 6 issues IN PROGRESS</div> <div>➤ WDB-5 Video upload 6 issues IN PROGRESS</div> <div>➤ WDB-8 Video player 2 issues TO DO</div> <div>➤ WDB-6 Video deletion 4 issues TO DO</div> <div>➤ WDB-94 Video download 4 issues TO DO</div> <div>➤ WDB-81 Forgot password 5 issues DONE</div> <div>➤ WDB-50 Technical tasks 14 issues IN PROGRESS</div> <div>➤ Issues without Subtask no issues</div>			

pav. 48 Jira sprint board su išskleistu user story

Projects / We Don't Byte

WDB Sprint 2

TM

J

AA

SV

Label

⚡

☆

🕒 2 days remaining

Complete sprint

TM

J

AA

SV

Label

GROUP BY Subtask

TO DO 13 ISSUES

IN PROGRESS 1 ISSUE

CODE REVIEW 3 ISSUES

DONE 22 ISSUES

+

▼

WDB-5

Video upload

4 issues

IN PROGRESS

Make Upload API call in frontend

Frontend

WDB-104

TM

Create upload endpoint

Backend

WDB-103

AA

Create storage/folder structure for videos

Backend

WDB-98

AA

Create upload modal

Frontend

WDB-32

✓

TM

💡 Quickstart

Po kiekvieno sprinto dalis backlog stories būdavo automatiškai ištrinami užbaigiant sprintą. Tuomet mūsų komanda kito sprinto planavimo metu, dalį likusių backlog stories perkeldavo į kitą sprintą:

pav. 49 Backlog antro sprinto metu

▼ WDB Sprint 2 4 May – 18 May (7 issues)

0 5 31 Complete sprint ...

WDB-4 Video library	8	DONE	
WDB-5 Video upload	8	DONE	
WDB-8 Video player	8	DONE	
WDB-6 Video deletion	5	DONE	
WDB-94 Video download	5	CODE REVIEW	
WDB-8+ Forgot password	2	DONE	
WDB-50 Technical tasks		IN PROGRESS	

+ Create issue

▼ Backlog (4 issues)

0 0 0 Create sprint

WDB-21 Recycling bin			
WDB-9 Video information			
WDB-29 Navigation			
WDB-90 Reports and presentation			

+ Create issue

pav. 50 Backlog trečio sprinto metu

Projects / We Don't Byte

Backlog



Epic ▾

▼ WDB Sprint 3 18 May – 24 May (5 issues)

0 13 2 Complete sprint ...

 WDB-21 Recycling bin	 5 IN PROGRESS ▾ 
 WDB-9 Video information	 3 IN PROGRESS ▾ 
 WDB-29 Navigation	 2 DONE ▾ 
 WDB-90 Reports and presentation	 5 IN PROGRESS ▾ 
 WDB-50 Technical tasks	 IN PROGRESS ▾ 

+ Create issue

▼ Backlog (0 issues)

0 0 0 Create sprint

Your backlog is empty.

+ Create issue