

1. Handover	2
1.1 App Instructions	3
1.2 What has been delivered	5
1.3 Third Party Services	8
1.3.1 Auth0 (Authentication)	9
1.3.2 AWS (Storage)	11
1.3.3 DigitalOcean (Deployment)	14
1.3.4 EmailJS>Email Sender)	17
1.3.5 GitHub (Codebase)	20
1.3.6 TinyMCE (RichText Editor)	21
1.4 Handover - Requirements	23
1.4.1 Handover - Background	24
1.4.2 Handover - Project Scope	26
1.4.3 Handover - Motivational Modelling	28
1.4.4 Handover - Personas	29
1.4.5 Handover - User Stories & Acceptance Criteria	34
1.4.5.1 Handover - User Stories for MM Project	35
1.4.5.2 Handover - User Stories for QR and Animation Projects	37
1.4.5.3 Handover - Acceptance Criteria for MM Project	38
1.4.5.4 Handover - Acceptance Criteria for QR and Animation Projects	39
1.5 Handover - Design	40
1.5.1 Handover - API Endpoints	41
1.5.2 Handover - Architectural Diagram	57
1.5.3 Handover - Database Diagram	58
1.5.4 Handover - Frontend Structure	59
1.5.5 Handover - Mood board and Logo	60
1.5.6 Handover - Prototypes	63
1.5.6.1 Handover - Admin Platform Prototype	64
1.5.6.2 Handover - Glass Flowers Mobile App Prototype	65
1.6 Handover - Testing Plan	66

Handover

This folder is setup to document what Estelle needs to know prior to the end-of-year handover. It will include instructions on how to use the app, next steps for the project, what aspects are missing, instructions on how to use 3rd party services, login details and other relevant information.

App Instructions

Instructions - Sam/My Tien:

Sign In to Admin Platform

- Go to the Admin Platform at <https://mezzuzotproject.com/admin>
- Username: mezzuzotproject2021@gmail.com
- Password: Mezzuzotproject2021

Add Flower/Mezzuzah Story

Flower Story:

- Log in to the Admin Platform at <https://mezzuzotproject.com/admin>
- Click on the "Add Flower" button on the top right corner
- Fill in the details
- Click the "Create flower story" button at the bottom of the page

Mezzuzah Story:

- Log in to the Admin Platform at <https://mezzuzotproject.com/admin>
- Navigate to the "Missing Mezzuzot" tab by clicking "Missing Mezzuzot" from the list on the left
- Click on the "Add a Mezzuzot" button on the top right corner
- Fill in the details
- Click the "Create mezzuzah story" button at the bottom of the page

View Flower/Mezzuzah Story (Public)

From the Admin Platform:

- Log in to the Admin Platform at <https://mezzuzotproject.com/admin>
- Click on the "Edit" button on a flower that you wish to view
- Scan the QR code on the bottom right corner with your phone
- If you are creating a new flower and would like to see the QR code, click "Create flower story" first, it will take you back to the Admin Platform page, where you can choose the flower you have just created to view the QR code

From the public facing website:

- Go to the public facing site at <https://mezzuzotproject.com>
- Click on a Flower you wish to view

Edit Flower/Mezzuzah Story

- Log in to the Admin Platform at <https://mezzuzotproject.com/admin>
- Click on the "Edit" button on a flower that you wish to edit
- Enter the new details
- After you've finished, click the "Update Flower Story" button at the bottom of the page
- For editing Mezzuzah story, first navigate to the "Missing Mezzuzot" by clicking "Missing Mezzuzot" from the list on the left, then follow the similar steps

Delete Flower/Mezzuzah Story

- Log in to the Admin Platform at <https://mezzuzotproject.com/admin>
- Click on the "Edit" button on a flower that you wish to edit
- Click on the "Delete Flower Story" button at the bottom of the page
- When prompted with "Are you sure?", you can choose to "Delete" or "Cancel"
- For deleting Mezzuzah story, first navigate to the "Missing Mezzuzot" tab by clicking "Missing Mezzuzot" from the list on the left, then follow the similar steps

View Animations

- Go to the public facing site at <https://mezzuzotproject.com>
- Click on the 'Animations' tab on the header bar
- Click on the animation that you wish to view

Add Animation

- Go to the Admin Platform at <https://mezzuzotproject.com/admin>
- Click on the 'Animations' tab on the left hand navigation list
- Click 'Add Animation' in the top right corner
- Click 'Upload Animation'
- Select the video file that you would like to upload
- Give the animation a title and description
- Click 'Confirm' (Note that it may take a couple of minutes to upload depending on the size of the video)

Edit Animation

- Go to the Admin Platform at <https://mezzuzotproject.com/admin>
- Click on the 'Animations' tab on the left hand navigation list
- Click 'Edit' on the animation that you would like to change
- Edit the title and/or the description
- Click 'Confirm'

Delete Animation

- Go to the Admin Platform at <https://mezzuzotproject.com/admin>
- Click on the 'Animations' tab on the left hand navigation list
- Click 'Delete' on the animation that you would like to remove
- Click 'Delete' again on the pop-up that appears

Edit About Page

- Go to the Admin Platform at <https://mezzuzotproject.com/admin>
- Click on the 'About Us' tab on the left hand navigation list
- Click 'Edit' on the 'Glass Flower' item
- Amend the title and description.
- Add any media objects (images, videos, audios)
- Click 'Save' in the top right hand corner

Use Contact Page

- Go to the public facing site at <https://mezzuzotproject.com>
- Click on the 'Contact' tab on the header bar
- Enter name, email and message
- Click 'Send Message'

What has been delivered

What has been completed:

Admin Platform:

Requirement	Description
Deployment	<ul style="list-style-type: none">The admin platform has been deployed and is able to be accessed from:<ul style="list-style-type: none">mezzuzotproject.com
Authentication	<ul style="list-style-type: none">The admin platform is secure and the user is able to login/logout without difficultyLogin Details:<ul style="list-style-type: none">Email: mezzuzotproject2021@gmail.comPassword: Mezzuzotproject2021
Flower Story	<ul style="list-style-type: none">User is able to add/edit/delete flower stories. A flower story includes the following fields:<ul style="list-style-type: none">Family NameCountry (Should be extended to include the ability to add a new country)Family MembersBrief DescriptionDetailed Description (incl. ability to format)Media<ul style="list-style-type: none">ImagesVideosAudiosOnce a flower story has been created, a QR code is generated which when scanned takes the user to the corresponding public flower story page.
Mezzuzah Story	<ul style="list-style-type: none">User is able to add/edit/delete mezzuzah stories. A mezzuzah story includes the following fields:<ul style="list-style-type: none">Family NameFamily MembersBrief DescriptionDetailed Description (incl. ability to format)Media<ul style="list-style-type: none">ImagesVideosAudiosOnce a mezzuzah story has been created, a QR code is generated which when scanned takes the user to the corresponding public flower story page
Animations	<ul style="list-style-type: none">User is able to add/edit/delete animations from the platform. Animations have the following fields:<ul style="list-style-type: none">Animation TitleAnimation Description (multi-lined so that both a caption and artist attribution can be included)
About Us	<ul style="list-style-type: none">User is able to update the 'About' information for each of the following pages:<ul style="list-style-type: none">Glass FlowersMissing MezzuzotEstelle Rozinski Projects (Overall)Each 'About' information includes the following fields:<ul style="list-style-type: none">TitleDescriptionMedia<ul style="list-style-type: none">ImagesVideosAudiosPLEASE NOTE THAT THIS FUNCTIONALITY IS INCOMPLETE.<ul style="list-style-type: none">To edit the public-facing 'About' page, please edit the 'Glass Flower' About section on the admin platform. The others have not been properly linked yet.

Public Facing Platform:

Requirement	Description
Deployment	<ul style="list-style-type: none"> The public-facing platform has been deployed at the following link: <ul style="list-style-type: none"> https://mezzuzotproject.com/
Flower Stories	<ul style="list-style-type: none"> Exhibition visitor is able to scan QR code which directs them to the corresponding flower story page. Visitor can also view a list of all of the uploaded flower stories User is able to search flower stories based on name and country (Need to add 'Search by Address' functionality).
Mezzuzah Stories	<ul style="list-style-type: none"> Exhibition visitor is able to scan QR code which directs them to the corresponding mezzuzah story page. Visitor can view a list of all the uploaded mezzuzah stories User is able to search mezzuzah stories based on name (Need to add 'Search by Address' functionality).
Animations	<ul style="list-style-type: none"> Visitor is able to view a list of all the uploaded animations When they click on an individual animation, they are able to view it. Recommended animations are also shown on this page.
About Us	<ul style="list-style-type: none"> Visitor is able to read the 'About Us' information entered on the admin platform (Please look at the About Us section for the admin platform above - this functionality is incomplete).
Contact Us	<ul style="list-style-type: none"> Visitor is able to Contact Estelle through an email portal. User is able to include their name, their email and their message. Email is currently sent to: <ul style="list-style-type: none"> missingmezuzots@gmail.com Email can be updated by following the steps here: EmailJS>Email Sender

What still needs to be done (Next Steps):

Admin Platform:

Requirement	Description
Design	<ul style="list-style-type: none"> Text of links in admin sidebar should all be lowercase Default font of the site should be changed to 'Open Sans'
Flower/Mezzuzah Stories	<ul style="list-style-type: none"> When adding family members, adding a comma "," should separate each individual member into their own string. The final object should be an array of strings. Add sections for 'What happened next?' and 'Where are they now?' Need to implement function to add more countries that stories can be assigned to
Animations	<ul style="list-style-type: none"> Need to add ability to set a custom thumbnail for each animation
Sign-in	<ul style="list-style-type: none"> Auth0 logo will need to be updated once the 'Estelle Rozinski Projects` logo has been created

Public Facing Platform:

Requirement	Description
Domain Name	<ul style="list-style-type: none"> Needs to be changed to 'rozinskiprojects.com' (cost ~\$20 per year)

Design	<ul style="list-style-type: none"> ■ Logo to be updated to 'Estelle Rozinski Projects' ■ Default font of the site should be changed to 'Open Sans' ■ Colour scheme to be updated to reflect Estelle's preferences. This palette may fit: <div data-bbox="368 249 943 523"> <p>#FF0D0D #FF4E11 #FF8E15 #FAB733 #ACB334 #69B34C</p> </div> <p><small>Red-Orange-Green Gradient Color Scheme - by SchemeColor.com</small></p> <ul style="list-style-type: none"> ■ Header Bar should be taller, with a warmer red overlay ■ Header Bar names should be updated: <ul style="list-style-type: none"> ■ Glass Flowers Through the Glass Flower ■ Animations And he taught the canaries to sing... (possibly shortened to 'Canaries')
Animations	<ul style="list-style-type: none"> ■ Include space for an intro video/animation to introduce the animations section as a whole ■ Animation thumbnails need to be displayed (once this functionality has been implemented on the admin platform)
Flower /Mezzuzah Stories	<ul style="list-style-type: none"> ■ Update design to include About section as well as 'What happened next?'/Where are they now? sections ■ Need to add 'Search by Address' functionality to the search bars
About Sections	<ul style="list-style-type: none"> ■ Each section of the website needs its 'About' section to be linked with the admin platform. See 'About Us' item for the admin platform in the 'What has been completed' section for how it works currently. ■ The About sections need to be split up into: <ul style="list-style-type: none"> ■ About the Project ■ About the Artist ■ About the Curator
'Support Us' Link	<ul style="list-style-type: none"> ■ Need to add an item on the header bar called 'Support Us' which will link to an external site that Estelle will set up.

Missing Mezzuzot:

The Mezzuzot Project still needs to be properly implemented with a light-up display and physical mezzuzot. More details about this component can be found at: [Handover - Requirements](#).

Third Party Services

Auth0 (Authentication)

In this project, we implement Authentication by Auth0.

1. Auth0 introduction and management platform

- <https://auth0.com/>

2. Login details

- Email: MissingMezuzots@gmail.com
- Password: MissingMezuzots2021

3. Functionality

3.1 Customize the style of Login Box

- The user can click "Try it out" to have a look at the current Login box
- The user can click "Customize" to modify the style, including Company Logo, Primary Color and Page Background Color.

The screenshot shows the 'Getting Started' section of the Auth0 dashboard. On the left, there's a sidebar with various navigation items like 'Getting Started', 'Activity', 'Applications', 'Authentication', etc. The 'Authentication' item is currently selected. The main area displays four cards: 'Try your Login box' (with a 'Customize' button), 'Invite your team members' (with a 'Invite Members' button), 'Integrate Auth0 into your application' (with a 'Create Application' button), and 'Add a social login provider' (with a 'Add Social Connections' button). A blue rectangle highlights the 'Customize' button in the first card.

3.2 Authentication Configuration

- By clicking the "Applications" – "Applications" button (The parts drawn by the blue rectangle) in the side navigation bar, the user will be redirected to the "Applications" page. "Mezzuzot Project" is the application used in this project.

The screenshot shows the 'Applications' page of the Auth0 dashboard. The sidebar has the same structure as before, with 'Authentication' selected. The main area lists applications: 'Default App' (Client ID: 1UC7zu13VWRKxy2uJnJ560M9DUEy0, N1m) and 'Mezzuzot Project' (Client ID: 8E3TUTEZ5X09uIxPxy4yft7oEY4xNj). A blue rectangle highlights the 'Mezzuzot Project' entry.

- the user will be redirected to the configuration page by clicking the "Mezzuzot Project". Then he/she can configure the authentication on this page.
- Some important configuration items are shown in the areas drawn by the rectangle in the figure below. Please refer to the site notes for their functions.

Application URIs

Application Login URI
https://myapp.org/login

In some scenarios, Auth0 will need to redirect to your application's login page. This URL needs to point to a route in your application that should redirect to your tenant's /authorize endpoint. Learn more.

Allowed Callback URLs

- https://mezzuzotproject.com/admin
- https://mezzuzotproject.com
- https://www.mezzuzotproject.com/admin/manageFlowers
- https://www.mezzuzotproject.com/admin/http://localhost:3000/admin

After the user authenticates we will only call back to any of these URLs. You can specify multiple valid URLs by comma-separating them (typically to handle different environments like QA or testing). Make sure to specify the protocol (`https://`) otherwise the callback may fail in some cases. With the exception of custom URI schemes for native clients, all callbacks should use protocol `https://`. You can use Organization URL parameters in these URLs.

Allowed Logout URLs

- http://mezzuzotproject.com/logout
- https://mezzuzotproject.com/admin/logout
- https://www.mezzuzotproject.com/admin/logout

A set of URLs that are valid to redirect to after logout from Auth0. After a user logs out from Auth0 you can redirect them with the `returnTo` query parameter. The URL that you use in `returnTo` must be listed here. You can specify multiple valid URLs by comma-separating them. You can use the star symbol as a wildcard for subdomains (`*.google.com`). Query strings and hash information are not taken into account when validating these URLs. Read more about this at <https://auth0.com/docs/logout>.

Allowed Web Origins

- http://mezzuzotproject.com
- https://www.mezzuzotproject.com
- http://localhost:3000
- http://localhost:3000/login
- http://localhost:3000/admin

Comma-separated list of allowed origins for use with CORS or JSONP

3.3 Manage Users/Accounts

a. By clicking the "User Management" – "Users" button (The parts drawn by the blue rectangle) in the side navigation bar, the user will be redirected to the "Users" page, where he/she can check all active accounts and create new accounts.

Users

An easy to use UI to help administrators manage user identities including password resets, creating and provisioning, blocking and deleting users. [Learn more →](#)

+ Create User

Name	Connection	Logins	Latest Login
missingmezzuzots@google.com	Username-Password-Authent...	50	2 days ago
mezzuzotproject2021@gmail.com	Username-Password-Authent...	39	4 days ago
eric@student.unimelb.edu.au	Username-Password-Authent...	2	2 months ago
yayaom1@student.unimelb.edu.au	Username-Password-Authent...	10	2 months ago

b. The user can click the three dots behind the account to manage the account or click the "Create User" button to create a new account.

AWS (Storage)

Overview

We are using Amazon Web Services (AWS) for the storage of data for the app. Specifically, the app uses **DynamoDB** (storing of textual/JSON data) and **S3** (storing of media objects).

This is a **paid service** (monthly cost ~\$15 AUD).

Payment details pending switch to Estelle's preferred payment method.

Access details

Email	mezzuzotproject.eric@gmail.com
Password	M3zzuz0t

Implementation details:

- Currently, the API keys for the app to communicate with AWS are hard-coded into the app (*not secure*). **It will be necessary to migrate to:**
 - (Backend) Storing the API keys in the environment variables
 - (Frontend) Retrieving the S3 Access Key by making an authenticated call to the Backend

Instructions:

Viewing DynamoDB tables/items:

The screenshot shows the AWS DynamoDB console. The left sidebar has a 'Tables' link under the 'Dashboard' section, which is highlighted with a red arrow. The main dashboard displays sections for 'Alarms (0)', 'DAX clusters (0)', and 'Create resources'. A blue banner at the top states: 'The new DynamoDB console is now complete, and becomes your default experience. Following the preview phase in which we analyzed and incorporated your feedback, we have completed the new DynamoDB console, making it even easier for you to manage your data and resources. Let us know what you think. You can still choose to return to the previous console from the navigation pane.'

The new DynamoDB console is now complete, and becomes your default experience

Following the preview phase in which we analyzed and incorporated your feedback, we have completed the new DynamoDB console, making it even easier for you to manage your data and resources. Let us know what you think. You can still choose to return to the previous console from the navigation pane.

DynamoDB

- Dashboard
- Tables**
- Items New
- PartiQL editor New
- Backups
- Exports to S3 New
- Reserved capacity

DAX

- Clusters
- Subnet groups
- Parameter groups
- Events

Tell us what you think

Return to the previous console experience

Feedback English (US) ▾

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

Name	Status	Partition key	Sort key	Indexes	Read capacity mode	Write capacity mode
About	Active	id (String)	-	0	Provisioned (1)	Provisioned (1)
FlowerStory	Active	id (String)	-	0	Provisioned (1)	Provisioned (1)
Media	Active	id (String)	-	0	Provisioned (1)	Provisioned (1)
MezzuzahStory	Active	id (String)	-	0	Provisioned (1)	Provisioned (1)

The new DynamoDB console is now complete, and becomes your default experience

Following the preview phase in which we analyzed and incorporated your feedback, we have completed the new DynamoDB console, making it even easier for you to manage your data and resources. Let us know what you think. You can still choose to return to the previous console from the navigation pane.

DynamoDB

- Dashboard
- Tables**
- Items New
- PartiQL editor New
- Backups
- Exports to S3 New
- Reserved capacity

DAX

- Clusters
- Subnet groups
- Parameter groups
- Events

Tell us what you think

Return to the previous console experience

Feedback English (US) ▾

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

FlowerStory

General information

Partition key id (String)	Sort key -	Capacity mode Provisioned	Table status Active No active alarms
Indexes 0 globals, 0 locals	DynamoDB stream Disabled	Point-in-time recovery (PITR) Disabled	Time to Live (TTL) Info Disabled
Replication Regions 0 Regions	Encryption Owned by Amazon	Date created May 16, 2021, 21:59:21 (UTC+10:00)	

Amazon Resource Name (ARN)
arn:aws:dynamodb:ap-southeast-2:850181362316:table/FlowerStory

View S3 Bucket:

S Services ▾

Amazon S3

Buckets

- Access Points
- Object Lambda Access Points
- Multi-Region Access Points
- Batch Operations
- Access analyzer for S3

Block Public Access settings for this account

Storage Lens

- Dashboards
- AWS Organizations settings

Feature spotlight 3

AWS Marketplace for S3

Feedback English (US) ▾

We're continuing to improve the S3 console to make it faster and easier to use. If you have feedback on the updated experience, choose Provide feedback.

Easily move terabytes of data into S3 with AWS DataSync - an online data migration service. Learn More

Amazon S3

Account snapshot

Storage lens provides visibility into storage usage and activity trends. Learn more

View Storage Lens dashboard

Buckets (1) Info

Buckets are containers for data stored in S3. Learn more

Name	AWS Region	Access	Creation date
mezzuzot-media	Asia Pacific (Sydney) ap-southeast-2	Public	May 6, 2021, 11:29:03 (UTC+10:00)



© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

S Services ▾

Amazon S3

Buckets

- Access Points
- Object Lambda Access Points
- Multi-Region Access Points
- Batch Operations
- Access analyzer for S3

Block Public Access settings for this account

Storage Lens

- Dashboards
- AWS Organizations settings

Feature spotlight 3

AWS Marketplace for S3

Feedback English (US) ▾

We're continuing to improve the S3 console to make it faster and easier to use. If you have feedback on the updated experience, choose Provide feedback.

Amazon S3 > mezzuzot-media

mezzuzot-media Info

Publicly accessible

Objects (105)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Name	Type	Last modified	Size	Storage class
0293ac64-0903-4776-b109-20e8a3996c1.jpeg	jpeg	September 28, 2021, 14:22:48 (UTC+10:00)	308.3 KB	Standard
04b9551a-7006-4f04-aa53-c66ca9e361ab.mp4	mp4	October 5, 2021, 21:29:01 (UTC+11:00)	1.6 MB	Standard
		October 17, 2021, 20:43:29		

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

DigitalOcean (Deployment)

Overview

We are using DigitalOcean to deploy both the frontend and backend of the app.

This is a [paid service](#) (monthly cost ~\$25 AUD).

Payment details pending switch to Estelle's preferred payment method.

Access details

Email	mezzuzotproject2021@gmail.com
Password	missingmezzuzot

Deployment Instructions

Backend

Containerize the app using Docker

In Terminal:

1. Ensure [doctl](#) is installed and authenticated with an API token .
2. Use [the registry login command](#) to authenticate Docker with your registry:

```
doctl registry login
```

3. Use docker to build an image of the app:

```
docker build . -t backend
```

4. Use the docker tag command to tag your image with the fully qualified destination path:

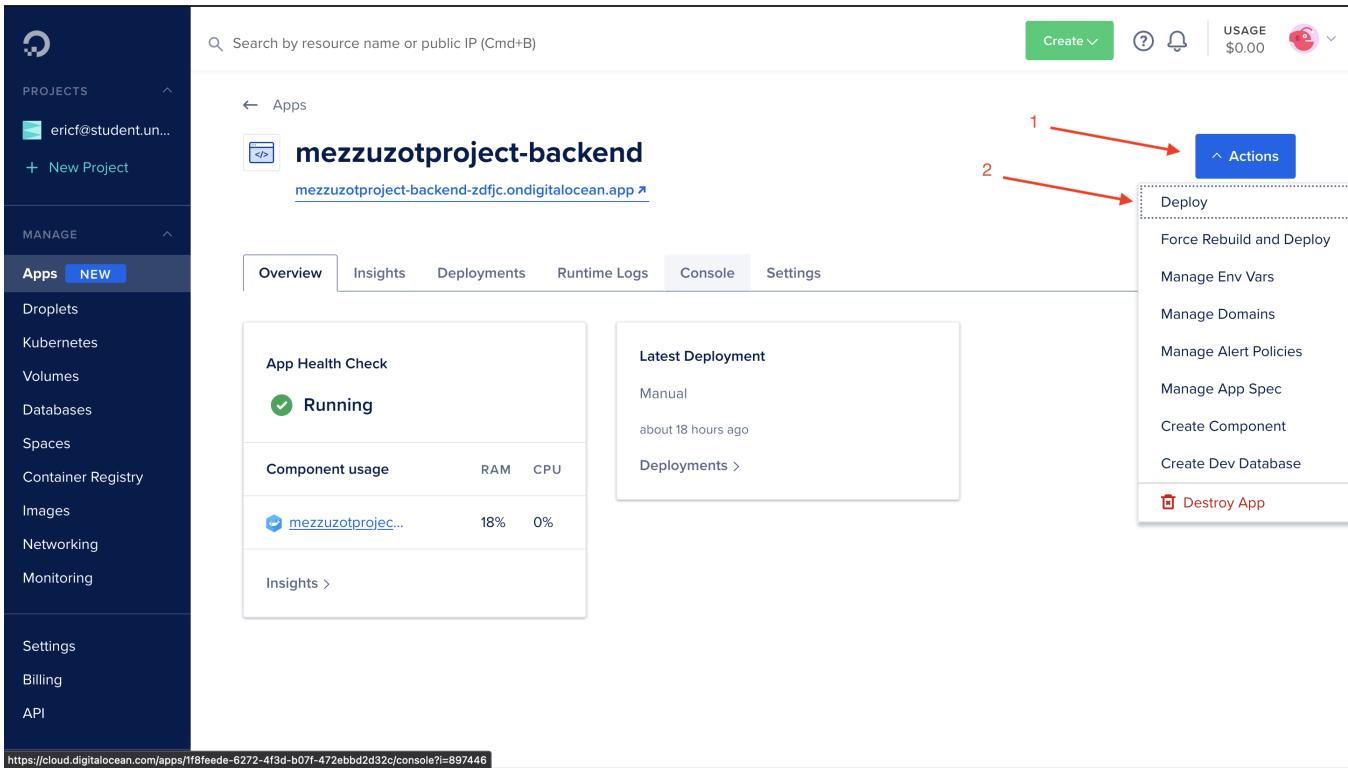
```
docker tag backend:latest registry.digitalocean.com/mezzuzotproject/backend
```

5. Use the docker push command to upload your image:

```
docker push registry.digitalocean.com/mezzuzotproject/backend
```

Deploy app in DigitalOcean console

6. Navigate to the mezzuzotproject-backend App
7. Click on Actions in the upper right corner
8. Click on Deploy



Frontend

In Terminal:

1. Use npm to create a production build of the React app

```
npm run build
```

2. (Do steps 1-2 in Backend deployment if not yet done)

3. Use docker to build an image of the app:

```
docker build . -t frontend
```

4. Use the docker tag command to tag your image with the fully qualified destination path:

```
docker tag frontend:latest registry.digitalocean.com/mezzuzotproject/frontend
```

5. Use the docker push command to upload your image:

```
docker push registry.digitalocean.com/mezzuzotproject/frontend
```

Deploy app in DigitalOcean console

6. Navigate to the mezzuzotproject-frontend App

7. (Same as steps 7-8 in Backend)

Garbage Collection

From time to time, you will need to initiate manual garbage collection in the DigitalOcean Container Registry to ensure the storage limit is not exceeded. This is done through the DigitalOcean console:

The screenshot shows the DigitalOcean Container Registry interface for the project "mezzuzotproject". In the top right corner, there are buttons for "Create", "?", and a bell icon, along with usage information showing \$0.00. Below the header, there's a search bar and a "Actions" button.

The main area displays storage usage at 33% (1.66 GB / 5 GB) and a "Garbage collection" section indicating 1 GB of unused registry data. A red arrow points from the text "Unused registry data" to the "Empty garbage" link below it.

Under the "Repositories" tab, two images are listed: "backend" (1 Tag, recent tag: latest, 12 days ago) and "frontend" (1 Tag, recent tag: latest, 7 days ago). Below the repository list are links for "Push images using Docker CLI" and "Using images with Kubernetes workloads".

The left sidebar includes sections for "PROJECTS", "MANAGE" (with "Apps" selected), "Container Registry" (with "Images" selected), and various management options like "Settings", "Billing", and "API".

This screenshot shows the "Run garbage collection" dialog box overlaid on the main interface. The dialog explains what garbage collection does and provides options for space freed (1 GB) and manifest deletion. It also includes a note about the registry being put into read-only mode during the process. A red arrow points to the "Empty Garbage" button at the bottom right of the dialog.

The background shows the same "Garbage collection" section and repository list as the first screenshot, with the "History" and "Settings" tabs also visible. The left sidebar remains the same.

EmailJS(Email Sender)

In this project, we implement the Email Communication function on the "Contact Us" page by EmailJS.

1. EmailJS introduction

- <https://www.emailjs.com/>

2. EmailJS management platform

- <https://dashboard.emailjs.com/sign-in>

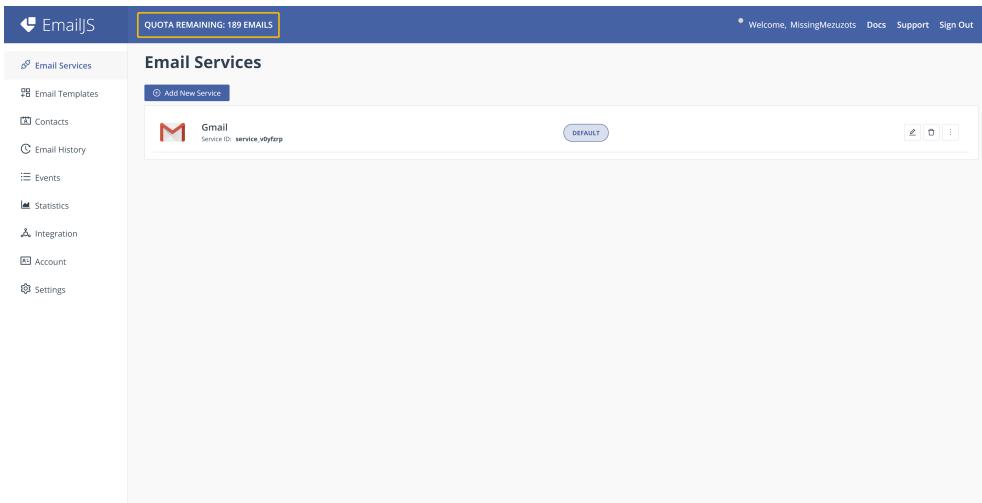
3. Login details

- Email: MissingMezuzots@gmail.com
- Password: Mezuzots

4. Functionality

4.1 Usage and quota remaining

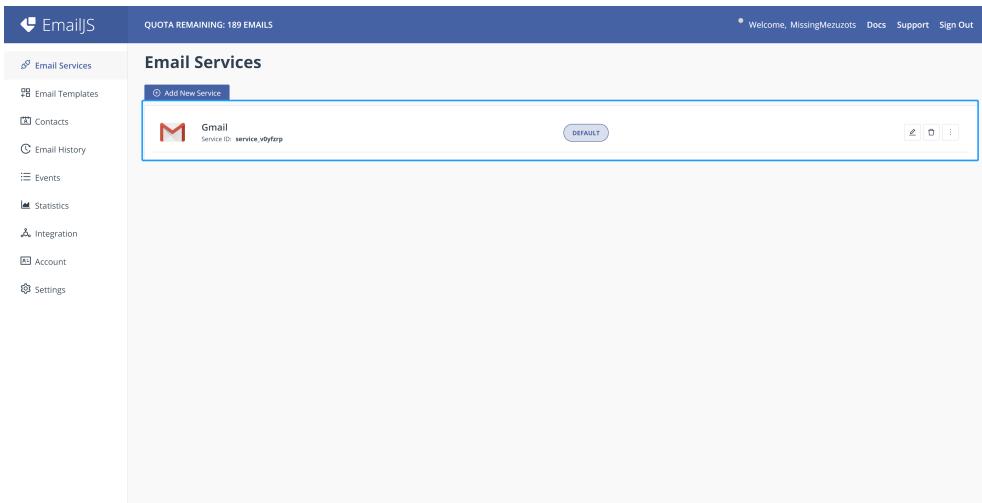
- a. For the current free account, the monthly quota is 200 emails.
- b. The user can see the latest quota remaining at the top of the page.



The screenshot shows the EmailJS management platform dashboard. At the top, there is a blue header bar with the EmailJS logo on the left, a 'QUOTA REMAINING: 189 EMAILS' badge in the center, and a 'Welcome, MissingMezuzots' message along with 'Docs', 'Support', and 'Sign Out' links on the right. Below the header is a sidebar on the left containing links for 'Email Services', 'Email Templates', 'Contacts', 'Email History', 'Events', 'Statistics', 'Integration', 'Account', and 'Settings'. The main content area is titled 'Email Services' and contains a section for 'Gmail' with a 'Service ID: service_vdyfnp' label. There are edit, delete, and info icons for this entry. A 'DEFAULT' button is also present. The overall interface is clean and modern.

4.2 Update the bound email address

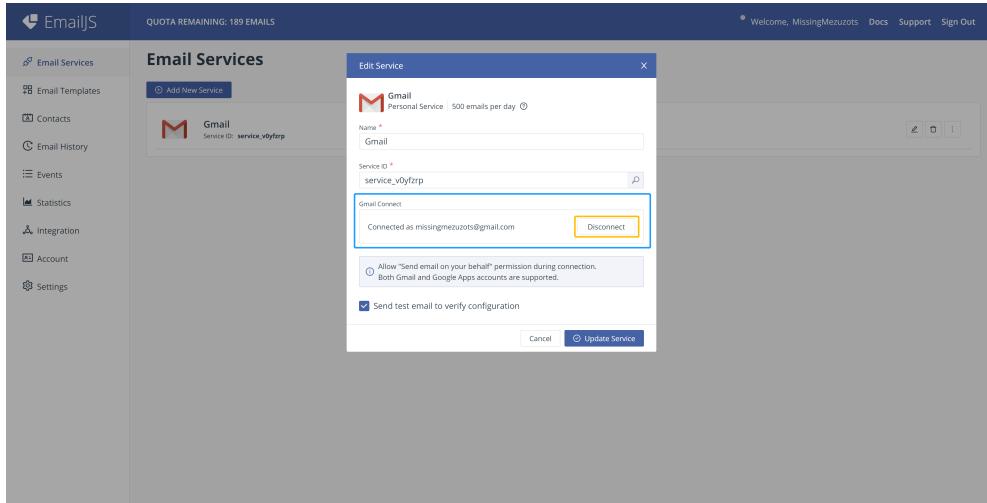
- a. Click the button drawn by the blue rectangle.



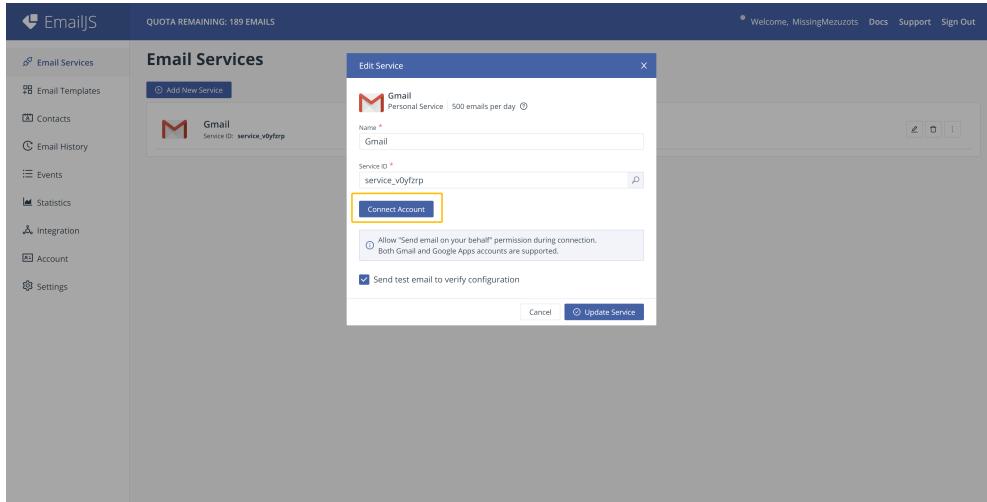
This screenshot is identical to the one above, showing the EmailJS management platform dashboard. The 'Edit' icon next to the 'Gmail' service entry in the 'Email Services' list has been highlighted with a blue rectangle, indicating where the user should click to update the bound email address.

b. The user can see the current connected email: "missingmezuzots@gmail.com" as shown in the figure below.

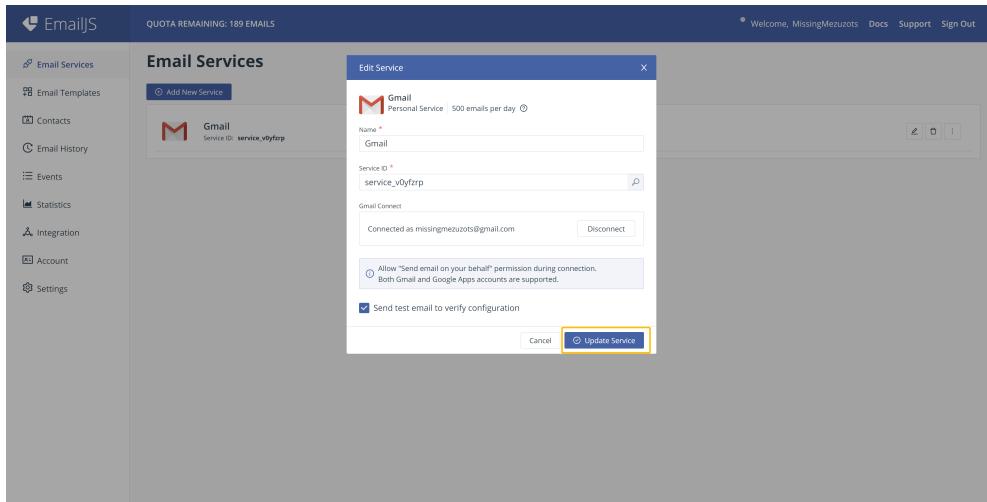
c. If the user wants to modify the connected email. Click the "Disconnect" button.



d. Then, the user clicks on the "Connect" button and fills in the Gmail account information in the new pop-up window.



e. After the user finishes all modifications, clicks on the "Update Service" button to save.



4.3 Modify Email Template

- The user can modify the email template as well.
- The user clicks the "Email Templates" button on the side bar to redirect to the "Email Templates" page.
- The user can create a new template by clicking the "Create New Template" button and modifying an existing template by clicking the template name.

- The user can update the template. The important parameters are introduced below.

- `{{from_name}}`, `{{to_name}}`, `{{message}}`, `{{from_email}}`, `{{reply_to}}`: The parameter passed from the code.
- Other contents are plain strings.
- After finishing, the user clicks the "Save" button to save the modifications.

GitHub (Codebase)

GitHub Organisation set up at:

- <https://github.com/Mezzuzot-Project>

Individual Repositories:

- Glass Flowers & Admin Platform: <https://github.com/Mezzuzot-Project/Glass-Flowers-and-Admin-Platform>
- Backend: <https://github.com/Mezzuzot-Project/Backend>

Access details

Email	erozinski@gmail.com
Password	Mezzuzotproject2021

TinyMCE (RichText Editor)

Overview

TinyMCE is a free rich text editor provider used in our code.

Link to documentation: <https://www.tiny.cloud/docs/>

Account login

Email: MissingMezuzots@gmail.com

Password: MissingMezuzots2021

Key Usages

To initialise a new RichText Editor, add the API key associated with the account to when initialising the Editor. Instructions are in the dashboard when logging in to <https://www.tiny.cloud>

The screenshot shows the 'Dashboard' page of the TinyMCE account. On the left sidebar, under 'MY ACCOUNT', the 'Approved Domains' section is highlighted. The main content area displays the 'Your Tiny API Key' section, which includes a unique API key: d496jo1ebny2d6uzgktaguluekt5obp1dm9rpe0e1ul3w1j6. A 'COPY' button is available to copy this key. Below this, there's a 'Setup Guide' section with tabs for 'TinyMCE' and 'TinyMCE + Collaboration'. Under 'TinyMCE Installation', there's a code snippet for integrating TinyMCE into an HTML document, with a 'COPY' button for it as well. The footer contains links for 'Privacy Policy', 'Terms of Use', and social media icons for Facebook, Twitter, and LinkedIn.

You can go to the "Approved Domains" section from the side bar

The screenshot shows the 'Dashboard' page of the TinyMCE account. The 'Approved Domains' section is highlighted in the sidebar. The main content area is currently empty, showing a placeholder message: 'No domains have been added yet. Click here to add a domain.' Below this, there are sections for 'JWT Keys' and 'Settings'.

Currently, it includes localhost and <https://mezzuzotproject.com/>. You can add any other domain or change an existing one in case the domain name for the project changes.

Approved Domains

Domain List

TinyMCE will only be loaded on your approved root domains. You can add multiple domains at the same time by separating them with a comma.

Domain name

Add

Approved Domains

- localhost
- mezzuzotproject.com **(x)**

Handover - Requirements

Handover - Background

Version	Published	Changed By	Comment
CURRENT (v. 1)	19 Oct, 2021 10:29	 Sam Ross	

Overview

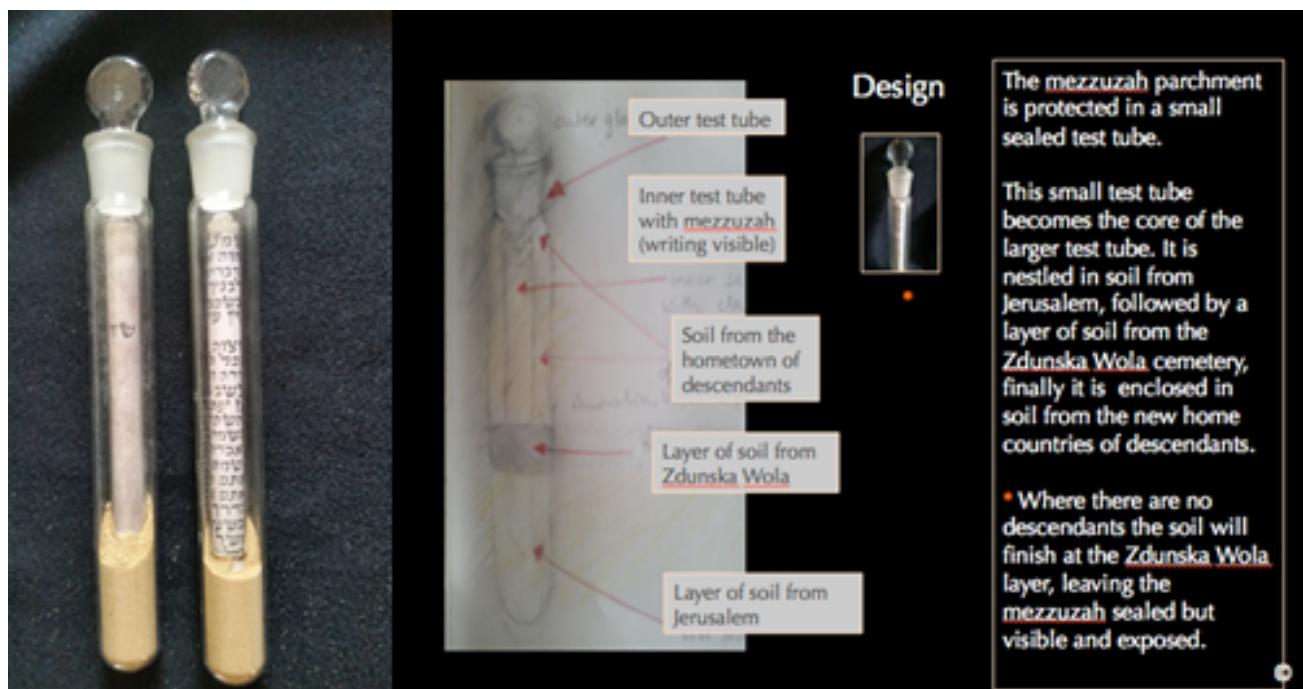
The Missing Mezzuzot of Zdunska Wola is an interactive exhibition/installation of 3,000 'mezzuzot' created by Estelle Rozinski as a metaphor and a celebration of each of the Jewish families living in Zdunska Wola, Poland before 1939. Zdunska Wola is the birthplace of Estelle's mother's family.

In addition to the Missing Mezzuzot exhibition, Estelle is in the process of creating an exhibition of glass flowers, to be displayed in the Melbourne Jewish Holocaust Centre, with the aim of educating the public about the vibrant and normal lives of pre-Holocaust Jewish families. Each glass flower will serve as a memorial to a particular Jewish family, and be accompanied by a QR code, that when scanned, should direct the scanner to a page with informative content related to that family. The content may include text, images, audio files and videos.

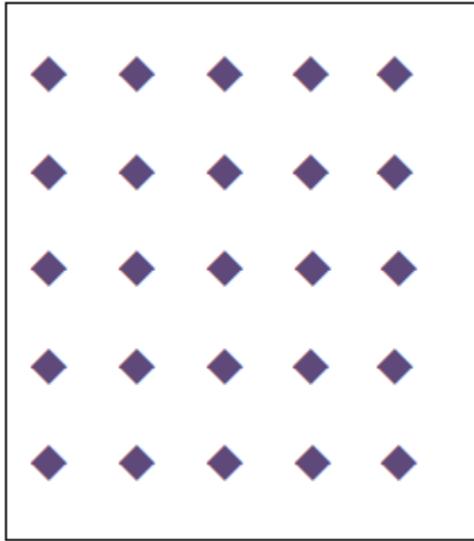
Furthermore, Estelle is currently working with Australian animators to create three short animations retelling stories of the Jewish people before WW2. Two of three animations have been completed and the third is a work in progress.

Missing Mezzuzot exhibition

A *mezzuzah* (plural: '*mezzuzot*') is a small decorative box affixed to the doorpost of a Jewish home that encases the Shema, the most important prayer in Judaism. This holy prayer is inscribed on parchment in Hebrew, serving as a reminder to enter and leave the home in peace and to maintain the sanctity of the home.



Each mezzuzah is mounted on a sheet of Perspex (20 cm x 30 cm).
The name of the family is engraved on a small brass plaque that sits below each mezzuzah.
The Perspex sheets line 4 wall panels.
The mezzuzot are organised in a configuration of 5 across x 5 down.



An iPad or tablet mirrors the installation, with an onscreen icon corresponding to each mezzuzah.
A tablet is placed beside or at the base of each of the four mezzuzah panels.

When an icon on the tablet is touched, the corresponding mezzuzah on the installation board will illuminate, showing a particular family's current location or tragic fate. Simultaneously, vignettes, video clips, stories, music, anecdotes and/or photographs of each chosen family will open on the tablet, enabling the audience to access a family's story.

An accompanying catalogue, entitled 'Looking for Zdunska Wola' features a foreword by curator Estelle Rozinski, curatorial essays, family images, archival documents, stories and statements.

Handover - Project Scope

Version	Published	Changed By	Comment
CURRENT (v. 1)	19 Oct, 2021 10:29	 Sam Ross	

Background:

The Missing Mezzuzot of Zdunska Wola is an interactive exhibition/installation of 3,000 'mezzuzot' created as a metaphor and a celebration of each of the Jewish families living in Zdunska Wola, Poland before 1939. Zdunska Wola is the birthplace of Estelle's mother's family.

A *mezzuzah* (plural: '*mezzuzot*') is a small decorative box affixed to the doorpost of a Jewish home that encases the Shema, the most important prayer in Judaism. This holy prayer is inscribed on parchment in Hebrew, serving as a reminder to enter and leave the home in peace and to maintain the sanctity of the home.

In addition to the Missing Mezzuzot exhibition, Estelle is in the process of creating an exhibition of glass flowers, to be displayed in the Melbourne Jewish Holocaust Centre, with the aim of educating the public about the vibrant and normal lives of pre-Holocaust Jewish families. Each glass flower will serve as a memorial to a particular Jewish family, and be accompanied by a QR code, that when scanned, should direct the scanner to a page with informative content related to that family. The content may include text, images, audio files and videos.

Furthermore, Estelle is currently working with Australian animators to create three short animations retelling stories of the Jewish people before WW2. Two of three animations have been completed and the third is a work in progress.

(see [Handover - Background](#) for more detailed background)

Project Motivation:

Currently, the digital component of the Missing Mezzuzot exhibition is not fit for purpose. The information that is displayed on the tablets is stored internally in four SD cards (one for each tablet) and there is currently no central database. As Estelle herself notes, the current setup is very inflexible and inextensible, frustratingly difficult to maintain, and not scalable.

Estelle hopes the exhibition can in future be displayed in both Jewish focused museums such as the Jewish Museum of Australia, as well as in general museum/archives, such as at UNSW. She also hopes to be able to add families to the Mezzuzot collection over time, and also add other informative and memorial content to the site.

This includes the informational aspect of her Glass Flowers exhibit, and the short animations created by the animators that tell stories of the Jewish people from before WW2.

Realising her vision requires a robust, flexible and extensible digital Missing Mezzuzot platform.

Scope:

- Create a digital QR code based Missing Mezzuzot platform and accompanying database
- Create QR code system for Estelle's Glass Flowers exhibit, that will link up with pages in the Missing Mezzuzot platform that display information relevant to a particular family.
- Accommodate the storage and display (on the Missing Mezzuzot platform) of short animations from animators

Functional Requirements

- Store all information and artefacts (eg. photos, videos, recordings) corresponding to each Zdunska Wola Jewish family
- Scan physical QR codes accompanying the Missing Mezzuzot exhibit
- Display information on each Zdunska Wola Jewish family when the corresponding QR code is scanned
- Display other related content and artifacts
- Store information and artefacts (eg. photos, videos, recordings) corresponding to each Jewish family included in the Glass Flowers exhibit.
- Scan physical QR codes accompanying the Glass Flowers exhibit
- Display the corresponding information when a QR code is scanned
- Store the animations created by the animators
- Display the animations created by the animators

Non-functional Requirements

- Accessibility
- Extensibility
- Maintainability
- Security
- Scalability
- Look and feel

Emotional Requirements

It is hoped that users of the Missing Mezzuzot platform (regardless of which of the three channels they access it from) will feel:

- **Moved** by the human stories behind each mezzuzah.
- **Surprised/shocked** at the sheer magnitude of the profound loss
- **Enlightened** by the educational and historical content, the dialogue it encourages, and the lessons it provides for people in the present
- **Connected** to the enduring beauty and legacy of the ZW Jewish Community

Out of scope:

- Integration of Missing Mezzuzot digital exhibition with physical lighting display

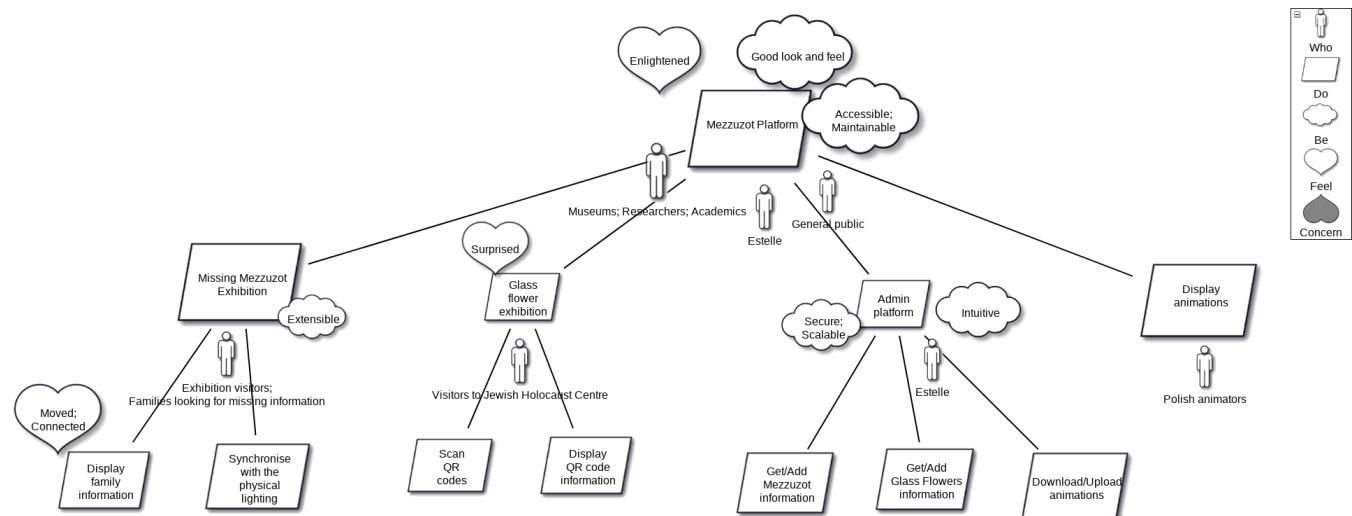
Handover - Motivational Modelling

Version	Published	Changed By	Comment
CURRENT (v. 1)	19 Oct, 2021 10:29	 Sam Ross	

Do/Be/Feel table

Who	Do	Be	Feel
Estelle	Interact with tablet	Adding of information needs to be accessible to those without much technical knowledge	Moved
Visitors to the exhibition	Read QR codes	Extensible (e.g. adding animations to the Missing Mezzuzot exhibition)	Surprised - emotionally, as well as at the breadth and vibrance of community that was destroyed
Museums/researchers /academics	Redirect to a website	Must be maintainable without too much technical knowledge	Enlightened
Families looking for missing information	Retrieve information from a database	Database needs to be secure	Connected
Visitors to Krakow	Add information	Scalable	

Goal model



Modifications discussed after Supervisor Meeting on 31 March:

- Add Polish Animators as stakeholder
- Change "Database Setup" to "Admin platform"
- Replace the "Store" Dos with a single "Enter information/data" Do under "Admin Platform"
 - That "Enter information/data" Do can branch out to "Enter Mezzuzot information", "Enter Glass Flowers information" and "Upload animations"
- Change "display" to "view"
- Add "intuitive" as a Feel to "Admin platform"

Handover - Personas

Version	Published	Changed By	Comment
CURRENT (v. 1)	19 Oct, 2021 10:29	 Sam Ross	

1. Persona for Ms. Estelle Rozinski

NAME	TYPE
Estelle Rozinski	Project Founder And Originator
	Background <p>The project commemorates the vital Jewish community of Zdunska Wola in Poland that was destroyed in the Holocaust. The originator's mother was a member of that community. The Missing Mezzuzot exhibit remembers that community by allowing visitors to click, using a tablet, on an artistic rendition of a mezuzah, something exemplifying a Jewish home, to learn of a Jewish family from Zdunska Wola.</p> <p>At present, the Missing Mezzuzot exhibit is on display at the Muzeum Historii Miasta Zduńska Wola ul. Złotnickiego 7. The digital component of the Missing Mezzuzot exhibition is not fit for purpose and needs to be upgraded. Furthermore, the exhibit should be made generalisable to other communities and be able to be deployed in other museums.</p>
Demographic <p> Female 50 years</p> <p> Melbourne, Victoria</p> <p>Married</p>	Goals <ul style="list-style-type: none"> • Build the system that can support interaction with visitors and can run on tablets. • Have confidence in the reliability, extensibility, and portability of this system. • Be able to store and browse multimedia data --- text, pictures, and videos on this platform. • Encourage the visitors with an attractive UI. • Be able to input new data to the platform. The operation should be user-friendly and easy to use. • Be able to modify / export existing data from the platform. The operation should be user-friendly and easy to use.
Motivations <p>Ms. Estelle Rozinski wants a better digital component of the Missing Mezzuzot exhibition, which should be reliable, extensible, and portable. Besides, it also should be attractive enough to catch more visitors' attention.</p>	Frustrations <ul style="list-style-type: none"> • The current physical platform has poor portability and extensibility as the information that is displayed on the tablets is stored internally in four SD cards (one for each tablet) and there is currently no central database, which is hard to update the data and re-use. • Currently, the project lacks an attractive e-Portal (such as a website), so visitors can only learn about the project in the museum, but not online.

UXPRESSIA

This map was built in uxpressia.com

2. Persona for Researchers or Academics



NAME

Nicholas Keller

TYPE

Researchers Or Academics**Bio**

Nicholas Keller is a 35-year-old researcher from Melbourne, Victoria, who is currently studying pre-war European Jewish culture. He hears about the Mission Mezzuzot project and shows a strong interest. The stories about the community of Jewish life are valuable and attractive for him. He has been to the museum where the project is exhibited many times, but he is still not satisfied with the data obtained and is eager for more stories about Jewish families in this project.

Demographic

Male _____ 35 years

Melbourne, Victoria

Married _____

Researcher of Jewish Culture

Goals

- Be able to get more digital data online about the Mission Mezzuzot project for research.
- Be able to select and export the relevant data himself. Besides, the operation of retrieving the data should be user-friendly.
- Be able to browse diverse forms of data, not only long text but also corresponding pictures or videos.

Motivations

Nicholas Keller does want an online platform where he can browser, select and export the data he wants himself, which is much more convenient for him.

Frustrations

- Most of the data Nicholas got before was about the Jewish community as a whole, rather than focusing on each Jewish family.
- Most of the contents of some websites about Jewish Mezzuzot culture are long texts, lacking expressiveness and attraction.
- The information displayed in the museum lacks the digital version. Therefore, Nicholas had to go to the museum and record the data in person, which is not convenient.

UXPRESSIA

This map was built in uxpressia.com

3. Persona for visitors to MM, families looking for missing information

NAME	TYPE
Adam Nowak	Visitors To MM, Families Looking For Information
	Background <p>Adam was born in Poland but was raised in Melbourne because his family moved to Australia when he was 6. He is not so familiar with his family history because he has been educated in Australia and lived in Melbourne for over 40 years. Though his family respects the Jewish culture, there are not many Jewish customs left in his family because the community they live in is a cultural melting pot.</p> <p>Now Adam is the dad of a 7-year-old boy, which reminds him of his childhood in Poland. He missed his friends and the happy times when he was a kid, and he wants to connect with the Jewish community and pass on traditional cultural values to his child. Now, Adam has been collecting information about his family, and he wants to fill the stories with some missing information. However, he met difficulties because the data he could find is not detailed enough and he is not an expert on using modern technologies.</p>
Demographic Male 47 years Melbourne, Victoria, Australia Married Jew looks for his root 7-year-old boy's dad	Goals <ul style="list-style-type: none"> Adam wants to find some details from other families in Poland, where his family originated, to fill the margin on his family's story, luckily may find clues of his neighbours. Adam has a non-Jewish cultural grow-up background, so he wants to feel connected to the Jewish culture and establish connections with people sharing the same culture. Adam wants to know other families' stories to feel more involved in the Jewish community and enable his son exposed to the Jewish culture. Adam would love to share his own family stories with others.
Motivations Adam wants to access a system where presents Interesting and beautiful stories of Jewish families. He would love to visit an exhibition where focus more on the individuals' or families' stories than just statistics. By reading these stories, he hopes he can find some clues about his childhood and pass some Jewish culture and customs to his son. He hopes the system to be easy-to-use and full of fun.	Frustrations <ul style="list-style-type: none"> The exhibitions he went to before are always concentrating on the Jewish group, not individuals or families, which makes it hard to retrieve useful information to fill his own family's history or interest his son. Adam tries to find his neighbours when he was a kid, but he could not remember details and could not find useful information online. Both Adam and his son were not very comfortable with modern technologies, complicated operations would bore them. Furthermore, his son is not patient when reading plain stories with massive words. It is hard for the overseas-born younger generation Jews to feel involved in and build a connection with their own culture, he has met trouble when trying to pass culture to his son because of the cultural assimilation of Jews.

UXPRESSIA

This map was built in upressoia.com

4. Persona for visitors to Krakow

NAME

Louise Bailey

TYPE

**Visitors To
Krakow**

Background

Louise has grown up in neighborhoods where an abundance of different cultures operate in harmony. She is interested in all kinds of cultures and wants to experience them in person. Louise chose Krakow as the first stop on a tour to Europe for multiple reasons. Firstly she was touched by the history and beautiful story of the Krakow Jewish Quarter. The main reason is that Krakow is one of the most historical and oldest cities in Europe. Besides, she would love to experience an exotic life and know the local lives are different from hers.

She is not an active person, but she is a curious visitor who pays attention to everything in a foreign city and hopes to know history, people, and local life in-depth. She is a sensitive person, and she regards feeling and proper imagination as the most important and intuitive way to experience and she enjoys her imagination brought by reading.

Demographic

Female 37 years
 Miami, United States
Single
Love and respect cultures
History lover

Goals

- Louise wants to feel the ancient cultural atmosphere of Krakow. Learn some of Krakow's local information and experience local life. Hopes to know some background knowledge about Krakow to make the history vivid.
- Louise looks forward to knowing some individuals' stories or families' histories in Krakow without too much communication with people and record some beautiful stories. An ideal way for her is to retrieve information directly on her phone.
- Louise wants to learn more about Jewish culture, getting different cultures and perspectives on her own life.

Motivations

Louise is attracted by the historical Krakow and wants to know more about its local life. She hopes to read stories about individuals' backgrounds, families' history in Krakow. Particularly, she is interested in Jewish culture and wants to know more about it by knowing the local lives. It would be better if she can feel the humanity of the city without communicating with others.

Frustrations

- Louise is monolingual so she has concerns about the language barrier and cross-language communication in Krakow.
- Past experience shows that the exhibitions are more concentrated on educating people with the same culture than spreading culture to the public. Lack of Jewish cultural background and knowledge makes her worried about whether she can benefit from the visit.
- Louise is quite hard to please because she has visited plenty of museums and exhibitions, she found them all much of a muchness. She is frustrated when she found the exhibitions turn individuals into a trivial part of the statistics.
- The past experience is not satisfying for Louise is also because the demonstrations are mostly general stories and data of the group, however, she expects to see individuals' stories and families' histories under the cultures.

UXPRESSIA

This map was built in uxpressia.com

Handover - User Stories & Acceptance Criteria

- [Handover - User Stories for MM Project](#)
- [Handover - User Stories for QR and Animation Projects](#)

Handover - User Stories for MM Project

Version	Published	Changed By	Comment
CURRENT (v. 1)	19 Oct, 2021 10:29	 Sam Ross	

User Story ID	Story description			Priority	Acceptance Criteria ID
	As	I want to	So that		
1	Adapt to computers and tablets				
1.2	a visitor	access the information by using tablets	I can interact with the exhibits in the museum and get to know details	high	
1.3	a researcher	browse all exhibit information online	I can learn and quote the information of Jewish culture and history	low	
1.4					
2	Interactive UI				
2.1	a visitor	scan the QR codes	I can know the background information behind each mezzuzah	high	
2.2	a visitor	click on the pictures to view the background information	I can read the corresponding story of each exhibit while watching the exhibition	high	
2.3	a researcher	see all the information being sorted by different tags	I can quickly find the information that I interested in	medium	
2.4	a researcher	see the relationship between all the exhibits and stories	I can better study the history and culture behind the exhibition	low	
3	Retrieve, edit or delete information				
3.1	Estelle	store all the information in a reliable database	I can easily store all the information online	high	
3.2	Estelle	access and edit all the information	I can manage and update the stories while necessary	high	
3.3	a researcher	get some of the source data	I can make use of the data in my research	low	
4	Maintain and extend the system				
4.1	a software programmer	make the code clear and understandable	I can easily maintain the system and fix bugs in the future	high	
4.2	a software programmer	the framework of the system to be extendable	I can easily add new features in the future	high	
4.3					

Version	Published	Changed By	Comment
CURRENT (v. 1)	19 Oct, 2021 10:29	 Sam Ross	

User Story ID	Story description			Priority	Acceptance Criteria ID
	As	I want to	So that		
1	Adapt to computers and tablets				
1.2	a visitor	access the information by using tablets	I can interact with the exhibits in the museum and get to know details	high	
1.3	a researcher	browse all exhibit information online	I can learn and quote the information of Jewish culture and history	low	
1.4					
2	Interactive UI				

2.1	a visitor	scan the QR codes	I can know the background information behind each mezzuza	high	
2.2	a visitor	click on the pictures to view the background information	I can read the corresponding story of each exhibit while watching the exhibition	high	
2.3	a researcher	see all the information being sorted by different tags	I can quickly find the information that I interested in	medium	
2.4	a researcher	see the relationship between all the exhibits and stories	I can better study the history and culture behind the exhibition	low	
3	Retrieve, edit or delete information				
3.1	Estelle	store all the information in a reliable database	I can easily store all the information online	high	
3.2	Estelle	access and edit all the information	I can manage and update the stories while necessary	high	
3.3	a researcher	get some of the source data	I can make use of the data in my research	low	
4	Maintain and extend the system				
4.1	a software programmer	make the code clear and understandable	I can easily maintain the system and fix bugs in the future	high	
4.2	a software programmer	the framework of the system to be extendable	I can easily add new features in the future	high	
4.3					

Handover - User Stories for QR and Animation Projects

Version	Published	Changed By	Comment				
CURRENT (v. 1)	19 Oct, 2021 10:29	 Sam Ross					
Story ID	User Story			Priority	Acceptance Criteria		
	As..	I want to...	So that...		Given that...	When...	Then...
1. QR code system for Estelle's Krakow Glass Flowers							
1.1	a visitor	scan the QR codes on the glass flowers	I can learn about the story associated with that flower / learn about the family that has been commemorated.	Must have			
1.2	Estelle	add/delete data for a new/removed glass flower	I can easily extend the exhibition, and increase the wealth of knowledge associated with the project.	Should have			
1.3	Estelle	modify information of the existing flower	I can improve content if I receive feedback from the Jewish community.	Should have			
1.4	Estelle	update the flowers and families information in just a few intuitive steps	I can manage the data without much technical knowledge.	Could have			
1.5	a visitor	view the glass flower stories without scanning a QR code	I can learn more about Jewish/Polish history from the comfort of my own home.	Should have			
1.6	a visitor	be able to easily hear the audio from both the audio files and the videos that are displayed with a story	I can experience the stories of Jewish families in full.	Should have			
2. Display short animations							
2.1	a visitor	watch the short animations from the Polish animators	I can learn the history and stories of the Jewish people before World War II	Must have			
2.2	a visitor	easily access the animations from the Missing Mezzuzot system	I can understand aspects of Jewish/Polish history beyond the Mezzuzot/glass flower stories	Must have			
2.3	Estelle	add/modify/delete animations for display	I can enrich the content with more animations	Should have			

Handover - Acceptance Criteria for MM Project

Handover - Acceptance Criteria for QR and Animation Projects

Handover - Design

Handover - API Endpoints

Version	Published	Changed By	Comment
CURRENT (v. 1)	19 Oct, 2021 10:32	 Sam Ross	

Classification	Type	URL	Description	Format
Glass Flower	GET	api/flower/{\$flower_story_id}	Returns the flower story of a certain \$flower_story_id	<p>The returned data format after calling this API:</p> <pre>{ "message": { "flowerType": "Australia", "familyMembers": "dsd", "videos": [{ "bucket": "mezzuzot-media", "location": "https://mezzuzot-media.s3-ap-southeast-2.amazonaws.com/6371fd3a-3223-4ac4-a9c6-75e0420b8bd6.mp4", "key": "6371fd3a-3223-4ac4-a9c6-75e0420b8bd6.mp4", "status": 204 }], "audios": [], "briefDescription": "ddf", "images": [{ "bucket": "mezzuzot-media", "location": "https://mezzuzot-media.s3-ap-southeast-2.amazonaws.com/d0aa8bed-dee7-4f88-bd5d-4b765b86922b.png", "key": "d0aa8bed-dee7-4f88-bd5d-4b765b86922b.png", "status": 204 }, { "bucket": "mezzuzot-media", "location": "https://mezzuzot-media.s3-ap-southeast-2.amazonaws.com/5f229e3b-6763-4df8-alde-2a61a5d64e93.jpeg", "key": "5f229e3b-6763-4df8-alde-2a61a5d64e93.jpeg", "status": 204 }, { "bucket": "mezzuzot-media", "location": "https://mezzuzot-media.s3-ap-southeast-2.amazonaws.com/e25f3d4b-ebc8-49a2-86ff-c7e6c5322d9d.jpeg", "key": "e25f3d4b-ebc8-49a2-86ff-c7e6c5322d9d.jpeg", "status": 204 }, { "bucket": "mezzuzot-media", "location": "https://mezzuzot-media.s3-ap-southeast-2.amazonaws.com/c5cdbfff-7d10-4f33-a12b-169a58daa77c.png", "key": "c5cdbfff-7d10-4f33-a12b-169a58daa77c.png", "status": 204 }] } }</pre>

```
{  
    "bucket": "mezzuzot-media",  
    "location": "https://mezzuzot-media.  
s3-ap-southeast-2.amazonaws.com/ecc5a8b7-c9fb-4aae-  
92b7-be53e458575b.png",  
    "key": "ecc5a8b7-c9fb-4aae-92b7-  
be53e458575b.png",  
    "status": 204  
}  
],  
"address": "df",  
"detailedDescription": "dffdfdf",  
"link": "http://localhost:8080/flowers  
/87d29e8e-4aa4-4a25-8dab-457f9b83e0ae",  
"id": "87d29e8e-4aa4-4a25-8dab-457f9b83e0ae",  
"familyName": "Test!"  
}  
}
```

	GET	/api/flower	Returns a list of all flower stories	The returned data format after calling this API:
				<pre>{ "message": [{ "flowerType": "Australia", "familyMembers": "Teting From Ma", "videos": [{ "bucket": "mezzuzot-media", "location": "https://mezzuzot-media.s3-ap-southeast-2.amazonaws.com/6371fd3a-3223-4ac4-a9c6-75e0420b8bd6.mp4", "key": "6371fd3a-3223-4ac4-a9c6-75e0420b8bd6.mp4", "status": 204 }, ...], "audios": [{ "bucket": "mezzuzot-media", "location": "https://mezzuzot-media.s3-ap-southeast-2.amazonaws.com/6371fd3a-3223-4ac4-a9c6-75e0420b8bd6.mp4", "key": "6371fd3a-3223-4ac4-a9c6-75e0420b8bd6.mp4", "status": 204 }, ...], "briefDescription": "Teting From Ma", "images": [{ "bucket": "mezzuzot-media", "location": "https://mezzuzot-media.s3-ap-southeast-2.amazonaws.com/d0aa8bed-dee7-4f88-bd5d-4b765b86922b.png", "key": "d0aa8bed-dee7-4f88-bd5d-4b765b86922b.png", "status": 204 }, ...], "address": "Teting From Ma", "detailedDescription": "Teting From Ma", "link": "http://localhost:8080/flowers/87d29e8e-4aa4-4a25-8dab-457f9b83e0ae", "id": "eff30fb-3349-4195-b51f-510d981f1cb6", "familyName": "Teting From Ma" }, ...] }</pre>

	GET	/api/about/gf	Returns the 'About' information for Glass Flowers	The returned data format after calling this API:
				<pre>{ "id": "gf", "category": "gf" "updatedAt": 1628665974000, "title": "The Title", "description": "a rich text description", "audios" : this.audios, "images": [{ "key": "image_key_1", "bucket": "about media", "caption": "some one play soccer", "location": "www.s3.com/imagekey1" }, ...], "audios": [{ "key": "audio_key_1", "bucket": "about media", "caption": "some one play soccer", "location": "www.s3.com/audiokey1" }, ...], "videos": [{ "key": "video_key_1", "bucket": "about media", "caption": "some one play soccer", "location": "www.s3.com/videokey1" }, ...] }</pre> <p>NOTE:</p> <ol style="list-style-type: none"> 1. The 'id' and 'category' fields both refer to the same thing, and can take one of three values - 'overall', 'gf' or 'mm' - referring to the site (overall, glass flowers, or missing mezzuzot) which the about information is associated with 2. 'timeLastModified' is a UNIX timestamp (milliseconds since Epoch) representing the time the About information was last edited.

	GET	/api/video/flower	Returns a list of all videos that are included in a flower story, along with the associated flower story id	The returned data format after calling this API: <pre>{ "message": [{ "flowerStoryId": "eff30bfb-3349-4195-b51f-510d981f1cb6", "s3": { "bucket": "mezzuzot-media", "location": "https://mezzuzot-media.s3-ap-southeast-2.amazonaws.com/6371fd3a-3223-4ac4-a9c6-75e0420b8bd6.mp4", "key": "6371fd3a-3223-4ac4-a9c6-75e0420b8bd6.mp4", "status": 204 } }, ...] }</pre>
--	-----	-------------------	---	---

Admin Platform	Flower	GET	/api/flower	Returns a list of all flower stories	The returned data format after calling this API:
					<pre>{ "message": [{ "flowerType": "Australia", "familyMembers": "Teting From Ma", "videos": [{ "bucket": "mezzuzot-media", "location": "https://mezzuzot-media.s3-ap-southeast-2.amazonaws.com/6371fd3a-3223-4ac4-a9c6-75e0420b8bd6.mp4", "key": "6371fd3a-3223-4ac4-a9c6-75e0420b8bd6.mp4", "status": 204 }, ...], "audios": [{ "bucket": "mezzuzot-media", "location": "https://mezzuzot-media.s3-ap-southeast-2.amazonaws.com/6371fd3a-3223-4ac4-a9c6-75e0420b8bd6.mp4", "key": "6371fd3a-3223-4ac4-a9c6-75e0420b8bd6.mp4", "status": 204 }, ...], "briefDescription": "Teting From Ma", "images": [{ "bucket": "mezzuzot-media", "location": "https://mezzuzot-media.s3-ap-southeast-2.amazonaws.com/d0aa8bed-dee7-4f88-bd5d-4b765b86922b.png", "key": "d0aa8bed-dee7-4f88-bd5d-4b765b86922b.png", "status": 204 }, ...], "address": "Teting From Ma", "detailedDescription": "Teting From Ma", "link": "http://localhost:8080/flowers/87d29e8e-4aa4-4a25-8dab-457f9b83e0ae", "id": "eff30fb-3349-4195-b51f-510d981f1cb6", "familyName": "Teting From Ma" }, ...] }</pre>
		GET	/api/admin/flower?id={\$#flower_story_id\$}	Returns the flower story of a certain \$flower_story_id\$	The returned data format after calling this API:
					<pre>{ "id": "flowerId", "familyName": "FN", ... }</pre>

```
        "address": "address",
        "familyMembers": [
            "member1",
            "member2",
            ...
            "memberN"
        ],
        "briefDescription": "BD",
        "detailedDescription": "DB",
        "flowerType": "FT",
        "url": "url",
        "images": [
            {
                "key": "image_key_1",
                "bucket": "mezzuzah media",
                "caption": "some one play soccer",
                "location": "www.s3.com/imagekey1"
            },
            {
                "key": "image_key_2",
                "bucket": "mezzuzah media",
                "caption": "some one play soccer",
                "location": "www.s3.com/imagekey2"
            },
            ...
        ],
        "audios": [
            {
                "key": "audio_key_1",
                "bucket": "mezzuzah media",
                "caption": "some one play soccer",
                "location": "www.s3.com/audiokey1"
            },
            {
                "key": "audio_key_2",
                "bucket": "mezzuzah media",
                "caption": "some one play soccer",
                "location": "www.s3.com/audiokey2"
            },
            ...
        ],
        "videos": [
            {
                "key": "video_key_1",
                "bucket": "mezzuzah media",
                "caption": "some one play soccer",
                "location": "www.s3.com/videokey1"
            },
            ...
        ]
    }
}
```

			<pre>{ "key": "video_key_2", "bucket": "mezzuzah media", "caption": "some one play soccer", "location": "www.s3.com/videokey2" }, ...</pre>
POST	/api/admin/flower	<p>1. Add a new flower story OR 2. Update the flower story (including adding /removing image, audio or video)</p>	<p>NOTE: 1. If the operation is adding a new flower story, leave ''id' as empty. If the operation is updating an existing flower story, please set ''id' as the ID you would like to update.</p> <p>2. Please add all images, audio, and videos every time you call this API.</p> <p>POST argument format:</p> <pre>{ "id": "flowerId", "familyName": "FN", "address": "address", "familyMembers": ["member1", "member2", ... "memberN"], "briefDescription": "BD", "detailedDescription": "DB", "flowerType": "FT", "url": "url", "images": [{ "key": "image_key_1", "bucket": "mezzuzah media", "caption": "some one play soccer", "location": "www.s3.com/imagekey1" }, { "key": "image_key_2", "bucket": "mezzuzah media", "caption": "some one play soccer", "location": "www.s3.com/imagekey2" }, ...], "audios": [{ "key": "audio_key_1", "bucket": "mezzuzah media", "caption": "some one play soccer", ...</pre>

```
"location":
```

```
"www.s3.com/audiokey1"
},
{
    "key": "audio_key_2",
    "bucket": "mezzuzah media",
    "caption": "some one play soccer",
    "location": "www.s3.com/audiokey2"
},
...
],
"videos": [
{
    "key": "video_key_1",
    "bucket": "mezzuzah media",
    "caption": "some one play soccer",
    "location": "www.s3.com/videokey1"
},
{
    "key": "video_key_2",
    "bucket": "mezzuzah media",
    "caption": "some one play soccer",
    "location": "www.s3.com/videokey2"
},
...
]
```

The returned data format after calling this API:

```
If the call executed failed:
```

```
    "Internal error: Failed to add flower" for Add
    "Internal error: Failed to update flower with
id\=' + id" for update
```

POST /api/admin/flower/delete

Delete the flower story whose id={
\$id}

NOTE: This operation will delete
the whole flower story,
including images, audios, and
videos.

POST argument format:

```
{
    "id": "flowerId"
}
```

The returned data format after calling this API:

```
If the call executed successfully:
```

```
    "Successful"
```

```
If the call executed failed:
```

```
    "Internal error: Failed to delete flower with
id\=' + id"
```

mezzuzah	GET	/api/mezzuzah	Returns a list of all mezzuzah stories	The returned data format after calling this API: <pre>[{ "id": "mezzuzahId", "address": "address", "familyName": "FN" }, { "id": "mezzuzahId", "address": "address", "familyName": "FN" }, ...]</pre>
	GET	/api/admin/mezzuzah?id={\$mezzuzah_id}	Returns the mezzuzah story of a certain <i>\$mezzuzah_id</i>	The returned data format after calling this API: <pre>{ "id": "mezzuzahId", "familyName": "FN", "address": "address", "familyMembers": ["member1", "member2", ..., "memberN"], "briefDescription": "BD", "detailedDescription": "DB", "url": "url", "images": [{ "key": "image_key_1", "bucket": "mezzuzah media", "caption": "some one play soccer", "location": "www.s3.com/imagekey1" }, { "key": "image_key_2", "bucket": "mezzuzah media", "caption": "some one play soccer", "location": "www.s3.com/imagekey2" }, ...], "audios": [{ "key": "audio_key_1", "bucket": "mezzuzah media", "caption": "some one play soccer", "location": "www.s3.com/audiokey1" }] }</pre>

```

        } ,
        {
          "key" :
          "audio_key_2",
          "bucket" :
          "mezzuzah media",
          "caption" :
          "some one play soccer",
          "location" :
          "www.s3.com/audiokey2"
        } ,
        ...
      ],
      "videos": [
        {
          "key" :
          "video_key_1",
          "bucket" :
          "mezzuzah media",
          "caption" :
          "some one play soccer",
          "location" :
          "www.s3.com/videokey1"
        } ,
        {
          "key" :
          "video_key_2",
          "bucket" :
          "mezzuzah media",
          "caption" :
          "some one play soccer",
          "location" :
          "www.s3.com/videokey2"
        } ,
        ...
      ]
    }
  
```

POST	/api/admin/mezzuzah	<p>1. Add a new mezzuzah story OR 2. Update the mezzuzah story (including adding /removing image, audio or video)</p> <p>NOTE: 1. If the operation is adding a new mezzuzah story, leave '<code>id</code>' as empty. If the operation is updating an existing mezzuzah story, please set '<code>id</code>' as the ID you would like to update.</p> <p>2. Please add all images, audio, and videos every time you call this API.</p> <p>POST argument format:</p> <pre>{ "id": "mezzuzahId", "familyName": "FN", "address": "address", "familyMembers": ["member1", "member2", ... "memberN"], "briefDescription": "BD", "detailedDescription": "DB", "url": "url", "images": [{ "key" : "image_key_1", "bucket" : "mezzuzah media", "caption" : "some one play soccer", "location" : "www.s3.com/imagekey1" } , ...] }</pre>

```
        "location":  
        "www.s3.com/imagekey1"  
    },  
    {  
        "key":  
        "image_key_2",  
        "bucket":  
        "mezzuzah media",  
        "caption":  
        "some one play soccer",  
        "location":  
        "www.s3.com/imagekey2"  
    },  
    ...  
],  
    "audios": [  
    {  
        "key":  
        "audio_key_1",  
        "bucket":  
        "mezzuzah media",  
        "caption":  
        "some one play soccer",  
        "location":  
        "www.s3.com/audiokey1"  
    },  
    {  
        "key":  
        "audio_key_2",  
        "bucket":  
        "mezzuzah media",  
        "caption":  
        "some one play soccer",  
        "location":  
        "www.s3.com/audiokey2"  
    },  
    ...  
],  
    "videos": [  
    {  
        "key":  
        "video_key_1",  
        "bucket":  
        "mezzuzah media",  
        "caption":  
        "some one play soccer",  
        "location":  
        "www.s3.com/videokey1"  
    },  
    {  
        "key":  
        "video_key_2",  
        "bucket":  
        "mezzuzah media",  
        "caption":  
        "some one play soccer",  
        "location":  
        "www.s3.com/videokey2"  
    },  
    ...  
]  
}
```

The returned data format after calling this API:

			<pre>If the call executed failed: "Internal error: Failed to add mezzuzah" for Add "Internal error: Failed to update mezzuzah with id\=' + id" for update</pre>
POST	/api/admin/mezzuzah /delete	<p>Delete the mezzuzah story whose id={\$id}</p> <p>NOTE: This operation will delete the whole mezzuzah story, including images, audios, and videos.</p>	<p>POST argument format:</p> <pre>{ "id": "mezzuzahId" }</pre> <p>The returned data format after calling this API:</p> <pre>If the call executed successfully: "Successful" If the call executed failed: "Internal error: Failed to delete mezzuzah with id\=' + id"</pre>

About	POST	/api/admin/about	<p>1. Add new About information OR 2. Update the existing About information</p> <p>NOTE: 1. The 'id' and 'category' fields both refer to the same thing, and can take one of three values - 'overall', 'gf' or 'mm' - referring to the site (overall, glass flowers, or missing mezzuzot) which the about information is associated with</p> <p>2. If the operation is adding new About information, leave 'id' as empty. If the operation is updating existing About information, please set 'id' as current 'id'</p> <p>3. If the about information contains images, audio or videos, please include them every time you call this API.</p> <p>POST argument format:</p> <pre>{ "id": "gf", "category": "gf" "title": "The Title", "description": "a rich text description", "audios" : this.audios, "images": [{ "key": "image_key_1", "bucket": "about media", "caption": "some one play soccer", "location": "www.s3.com/imagekey1" }, ...], "audios": [{ "key": "audio_key_1", "bucket": "about media", "caption": "some one play soccer", "location": "www.s3.com/audiokey1" }, ...], "videos": [{ "key": "video_key_1", "bucket": "about media", "caption": "some one play soccer", "location": "www.s3.com/videokey1" }, ...] }</pre> <p>The returned data format after calling this API:</p> <pre>If the call executed failed: "Internal error: Failed to add About information" for Add "Internal error: Failed to update About information for <type>" for update</pre>
-------	------	------------------	---

	Media	POST	/api/admin/media	<p>1. Add a new media entry OR 2. Update the media entry 3. Currently used for animations only 4. Includes Video, Audio, Images (one per entry)</p> <p>NOTE: 1. If the operation is adding a new media entry, leave 'id' as empty. If the operation is updating an existing media entry, please set 'id' as the ID you would like to update.</p> <p>2. Categories is always "animation" currently</p> <p>3. fileType is one of 3 values: "video", "image" and "audio"</p> <p>POST argument format:</p> <pre>{ "id": id, "fileType": "video", "category": "animation", "caption": "caption of file", "s3": { "bucket": "missing mezzuzot", "key": "example_video.mp4", "location": "www.s3.com/videokey1" } }</pre>
Missing Mezzuzah		GET	/api/mezzuzah?id={\$mezzuzah_id}	Returns the mezzuzah story of a certain \$mezzuzah_id

			<pre> "some one play soccer", "location": "www.s3.com/audiokey1" }, { "key": "audio_key_2", "bucket": "mezzuzah media", "caption": "some one play soccer", "location": "www.s3.com/audiokey2" }, ...], "videos": [{ "key": "video_key_1", "bucket": "mezzuzah media", "caption": "some one play soccer", "location": "www.s3.com/videokey1" }, { "key": "video_key_2", "bucket": "mezzuzah media", "caption": "some one play soccer", "location": "www.s3.com/videokey2" }, ...] } </pre>	
Animations	GET	/api/{\$media_id}	Returns the media story of a certain \$media_id	The returned data format after calling this API: <pre> { "id": id, "fileType": "video", "category": "animation", "caption": "caption of file", "s3": { "bucket": "missing mezzuzot", "key": "example_video.mp4", "location": "www.s3.com/videokey1" } } </pre>

Handover - Architectural Diagram

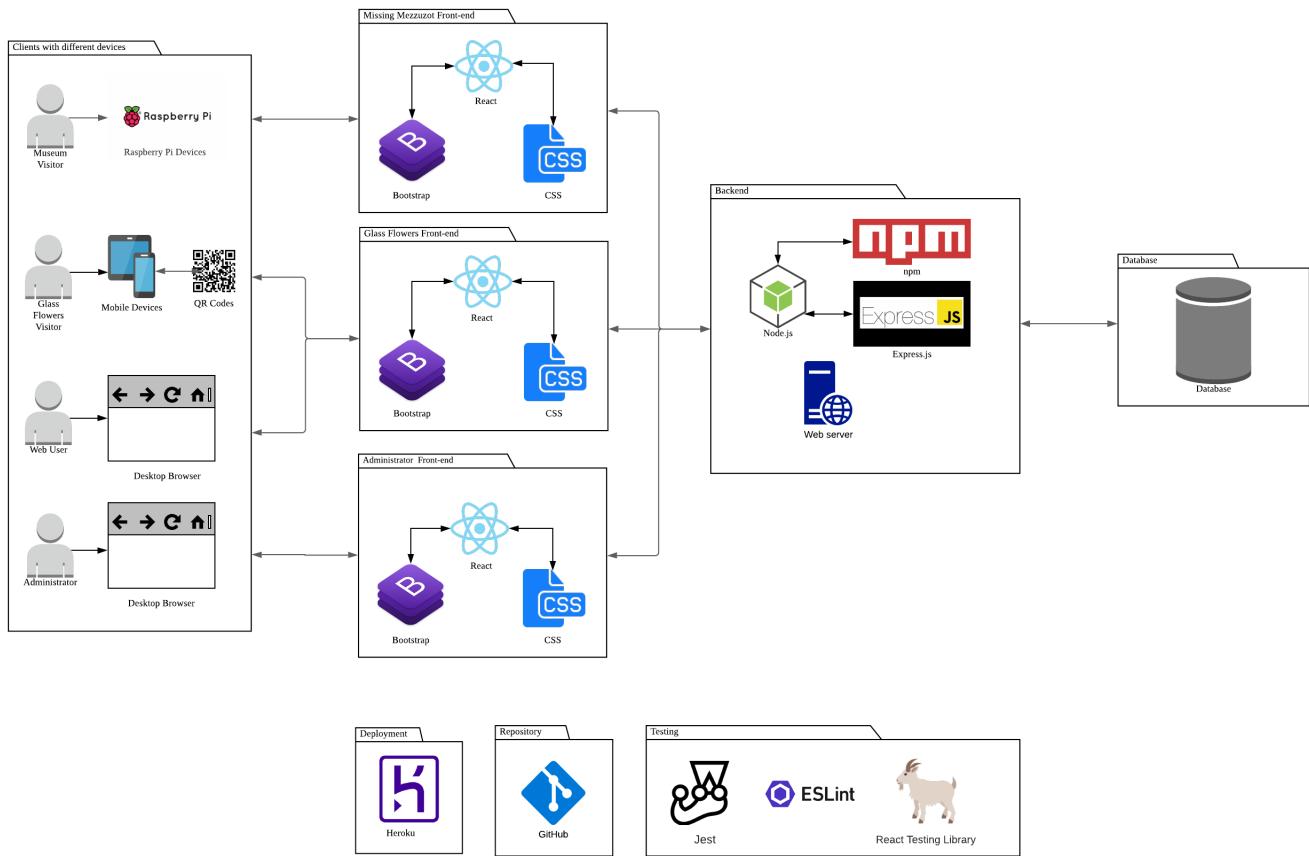
Version	Published	Changed By	Comment
CURRENT (v. 1)	19 Oct, 2021 10:32	 Sam Ross	

High level components of the system:

- Raspberry Pi devices
- Missing Mezzuzot iPad interface
- Glass Flowers web interface
- QR Codes
- Administrator interface
- Database

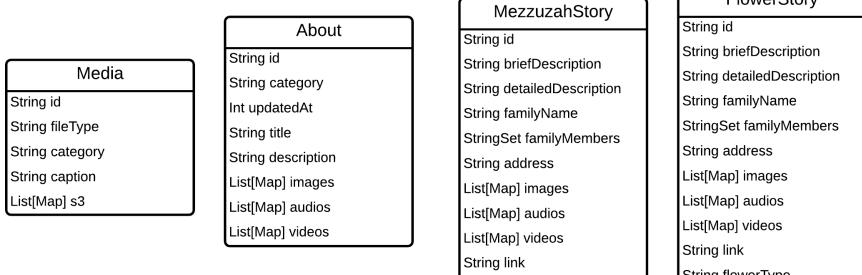
Architecture Diagram

Link: https://lucid.app/lucidchart/invitations/accept/inv_74d1cf6-2902-40c4-8a55-7992d6d2bc91?viewport_loc=-11%2C-11%2C1579%2C917%2C0_0



Handover - Database Diagram

Version	Published	Changed By	Comment
CURRENT (v. 1)	19 Oct, 2021 10:32	 Sam Ross	



Example of format of storing image/audio/video data:

```

{
  ...
  "Images": [
    {
      "key": "imagekey1",
      "bucket": "mezzuzot media",
      "caption": "someone play soccer",
      "location": "www.s3.com/imagekey1"
    },
    {
      "key": "imagekey2",
      "bucket": "rf",
      "caption": "rf",
      "location": "rf"
    }
  ],
  "Videos": [
    {
      "key": "videokey1",
      "bucket": "rf",
      "caption": "rf",
      "location": "rf"
    },
    {
      "key": "videokey2",
      "bucket": "rf",
      "caption": "rf",
      "location": "rf"
    }
  ],
  "Audios": [
    {
      "key": "audiokey1",
      "bucket": "rf",
      "caption": "rf",
      "location": "rf"
    },
    {
      "key": "audiokey2",
      "bucket": "rf",
      "caption": "rf",
      "location": "rf"
    }
  ]
}

```

Latest changes are marked in red.

Invitation link to database diagram on Lucidchart: https://lucid.app/lucidchart/invitations/accept/inv_4798c096-81fa-45e5-836e-b037e2c53b23

Handover - Frontend Structure

1. Commands

```
npm install
```

command for installing the project

```
npm run start
```

command for loading the project, if compile successfully, can preview the webpage at <http://localhost:3000>

2. File Structure

- public
 - static
 - logo // save the logo or icon
 - images // save the images or video
- src
 - mocks // mock data
 - components // the reusable components
 - mixins // the controller for screen adapter
 - pages // different page
 - theme // UI settings
 - utils // helper functions
 - routes // router for navigation
- Styles
 - global.css // global css setting

3. Code structure

use AddFlower.js as example

Need to `export default AddFlower;` at the end to the file

The AddFlower is defined as `const AddFlower = () => {}`

Can define the helper functions or call the api get before the return;

All the html code would inside the return call.

4. Resource

We choose the material-UI for our project, so the following resource would be helpful:

1. The guidance for material-UI: <https://material-ui.com/getting-started/installation/>
2. Example for the material UI project: <https://github.com/devias-io/material-kit-react>

Handover - Mood board and Logo

Version	Published	Changed By	Comment
CURRENT (v. 1)	19 Oct, 2021 10:32	 Sam Ross	

Mood Board v1.0





Logo v1.0



Missing Mezzuzot

Handover - Prototypes

Version	Published	Changed By	Comment
CURRENT (v. 1)	19 Oct, 2021 10:32	 Sam Ross	

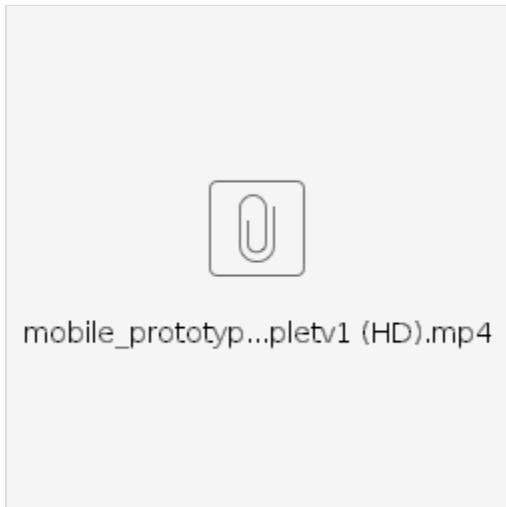
Prototype for Admin system: <https://jenpop.axshare.com/>

Handover - Admin Platform Prototype

<https://jenpop.axshare.com>

Handover - Glass Flowers Mobile App Prototype

V1 (prepared by [Yiran Wei](#))



Handover - Testing Plan

Version	Published	Changed By	Comment
CURRENT (v. 1)	29 Oct, 2021 12:17	 Eric Fan	
Testing Level Who Method Process			
Lint Testing	Developers	ESLint	<ul style="list-style-type: none"> Setup the rule for coding format IDE will automatically check the code and find the error every time commit Developer fix it and re-commit the code If pass the code then push it to git
Unit Testing	Developers	Jest and Enzyme	<p>Front-end</p> <ul style="list-style-type: none"> If the Component includes some functions that are complicated and error-prone Write the unit test <ul style="list-style-type: none"> Behavioral: Simulating the behavior of a user and testing the expected change. Component-based: Test directly the toggle method of the component and see if it does what we expect it to do. This can be helpful when the toggle function is more complicated and we want to test different scenarios. The git pipeline will run all the unit tests for each push <p>Back-end</p> <ul style="list-style-type: none"> Perform unit tests once a single function is finished. Generate the unit test to test <ul style="list-style-type: none"> The correct API Endpoints, The correct SQL call statements, The correct results after executing SQL call statements. The git pipeline will run all the unit tests for each push
Integration Testing	Tester	Jest and Enzyme	<ul style="list-style-type: none"> Take unit tested components and build a program structure by joining the components while testing to find errors associated with interfaces between components Each build combines and tests one subset of the modules, packages, or objects of the system Typical build strategies then try and integrate all those modules with no dependencies first, the modules that depend on these second, and so on The git pipeline will run all the Integration tests for each update for develop branch
System Testing (End-to-end testing)	Developers, testers	<ul style="list-style-type: none"> System Functional Test System Performance Test 	<p>System Functional Test</p> <ul style="list-style-type: none"> Test the entire system against the functional requirements and other external sources that determine requirements such as the user manual <p>System Performance Test</p> <ul style="list-style-type: none"> Test the non-functional requirements of the system and record the actual behavior of the software, for example: <ul style="list-style-type: none"> the load that the system response times usability (a survey) The System testing should be held before each release
Acceptance Testing	Developers, testers and clients	Acceptance criteria	<ul style="list-style-type: none"> Use the Acceptance Criteria as the foundation, build our own Acceptance Testing set Meeting with client and approve each acceptance criteria after each release