

List of referrals by user (backend)	Story ID : 1
<p>Description:</p> <p>As a user, I want to get the list of users who have been referred by a user.</p> <p>So that when a user logs into the new invites screen, they can see their invites and their statuses.</p>	
<p>Acceptance Criteria:</p> <ul style="list-style-type: none"> • Create the general structure of the API responsible for managing the referred user's functionalities.(ReferralsController) • Create the first API endpoint(referrals) that retrieves all users referred by a specific user. Select only their name and invitation status. Consider invitations with a status other than "Pending". • Method: GET, which receives the ID of the user logged into the application. • Response (200 OK) <p>Example:</p> <pre>[{ "inviteeName": "Cristian Campos", "status": "Completed" }, { "inviteeName": "Kattia Sanchez", "status": "Completed" }]</pre> <ul style="list-style-type: none"> • If no invitations exist, return 404. Response (404 Not Found): <pre>{ "message": "No referrals found for this user." }</pre> <p>If the user does not exist: (404 Not Found):</p> <pre>{ "message": "User not found.." }</pre>	

If there is an unexpected problem with the database or the query execution: 500
Internal Server Error

Technical Notes:

- Authentication is not verified: it is assumed that the application sends a valid user GUID.

List of referrals by user (Frontend)

Story ID : 2

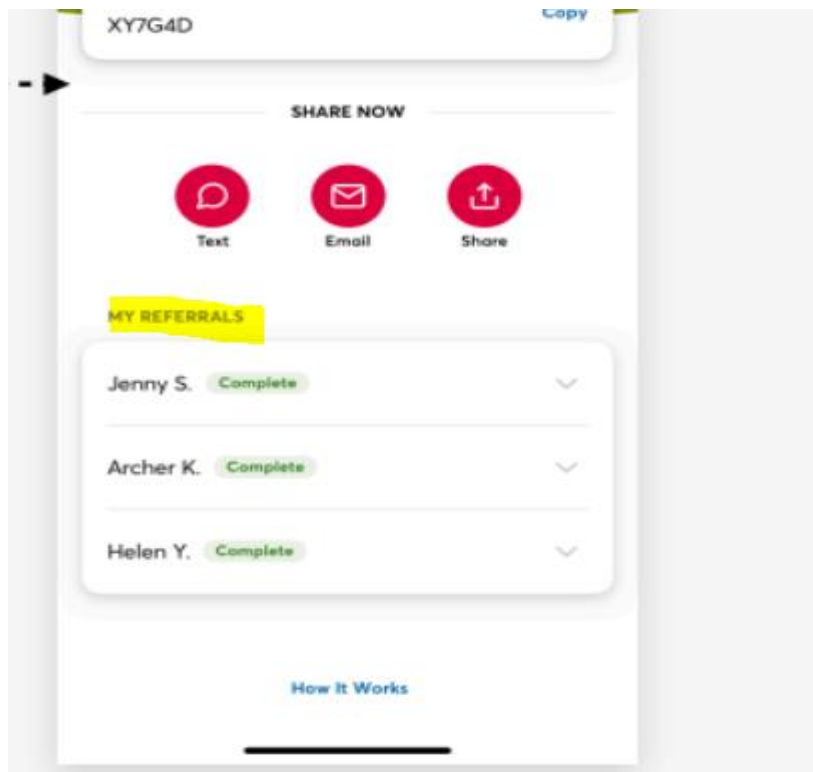
Description:

As a user, I want to get the list of users who have been referred by a user.

So that when a user logs into the new invites screen, they can see their invites and their statuses.

Acceptance Criteria:

- Display the list of people in the MyReferrals section of the app, showing their invite name and status.

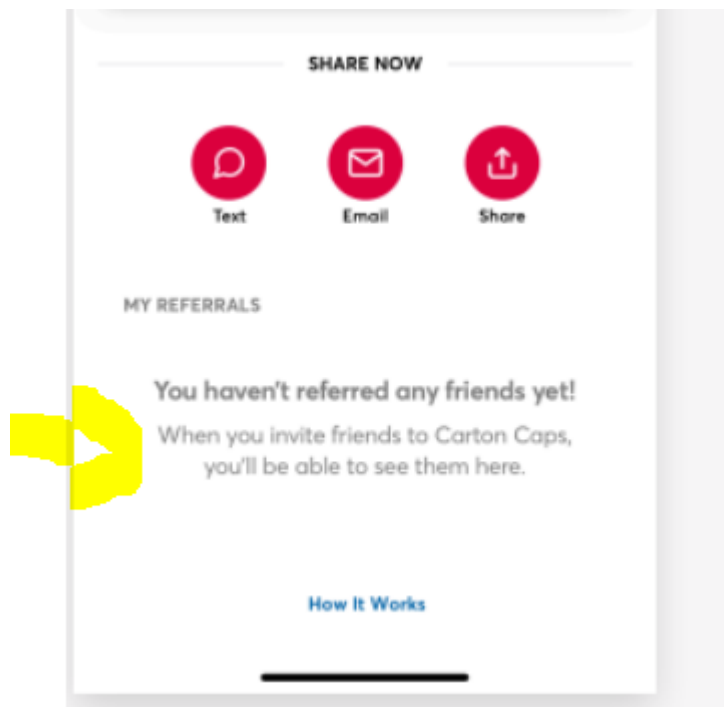


- Use the API developed in User Story #1. (referrals). "Get method", send the user ID as part of the path route. /api/Referrals/{referrerUserId}/referrals

- Endpoint response example

```
[  
  {  
    "inviteeName": "Cristian Campos",  
    "status": "Completed"  
  },  
  {  
    "inviteeName": "Kattia Sanchez",  
    "status": "Completed"  
  }  
]
```

- If there is no data, display the message:



Create invitation and generate referral link (backend)	Story ID : 3
<p>Description:</p> <p>As a user, I want to send an invitation to a friend using the Email, SMS, or Share option.</p> <p>So that the other person can receive the link to install the app.</p>	
<p>Acceptance Criteria:</p> <ul style="list-style-type: none"> • Create a new endpoint for the referrals API (CreateReferral), which should generate a unique referral code for the referral table, construct an invitation link, and save the record in the Referral table with an initial status of Pending. • Use the POST method, which receives an object with the user's ID as a parameter. • if successful: Response (200 OK): <pre> { "referralCode": "XZ7TE4S", "linkGenerated": "https://api.caps.com/referral/redirect?code=XZ7TE4S" } </pre> <p>If the user does not exist: (404 Not Found):</p> <pre> { "message": "User not found.." } </pre> <p>400 – Bad Request:</p> <ul style="list-style-type: none"> - The base link required to build the invitation URL is empty Message: "Base link is required." - The base link is not a valid absolute URL Message: "Invalid base link URL." - The referral code base (random character pool) is empty Message: "Referral code base is required." - The userId is empty Message: " UserId cannot be empty.." <p>If there is an unexpected problem with the database when saving the invitation or generating the referral code: 500 Internal Serve Error</p> 	

- Business Rules / Validations:
 - Generate a unique code for the referral table.
 - Save the invitation with a Pending status.
 - Build the link by concatenating the base link + ?code={referralCode}. Base link represents the redirect endpoint that we will develop and will be specified in user story #5
 - Validate that the generated code does not already exist.
 - Validate that the links are valid.
 - Validate that the user exists and is valid.

Technical Notes:

- The code is generated using a random string based on characters defined in the configuration file (appsettings.json).
- Authentication is not verified: it is assumed that the application sends a valid user GUID.

Create invitation and generate referral link (frontend)	Story ID : 4
<p>Description:</p> <p>As a user, I want to send an invitation to a friend using the Email, SMS, or Share option.</p> <p>So that the other person can receive the link to install the app.</p>	
<p>Acceptance Criteria:</p> <ul style="list-style-type: none"> • Consume the API developed in user story #3. (/api/Referrals/create) • The API is a POST request, and you must send an object containing the logged-in user's ID in the body of the request. <pre> { "referrerUserId": "5fbf7d9b-2d26-49cf-8941-d94863c5104b" } </pre> <ul style="list-style-type: none"> • The API returns the following object: <pre> { "linkGenerated": " https://localhost:7154/api/ReferralsVisit/redirect?code=QQDO4KJ" "referralCode": "XZ7TE4S", "message": "Referral created successfully." } </pre>	

```
}
```

If the user does not exist: (404 Not Found):

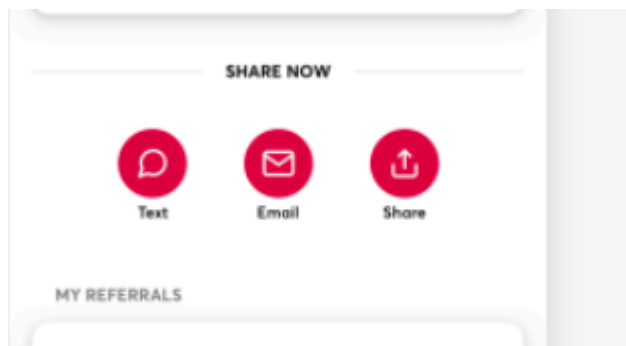
```
{  
  "message": "User not found.."  
}
```

400 – Bad Request:

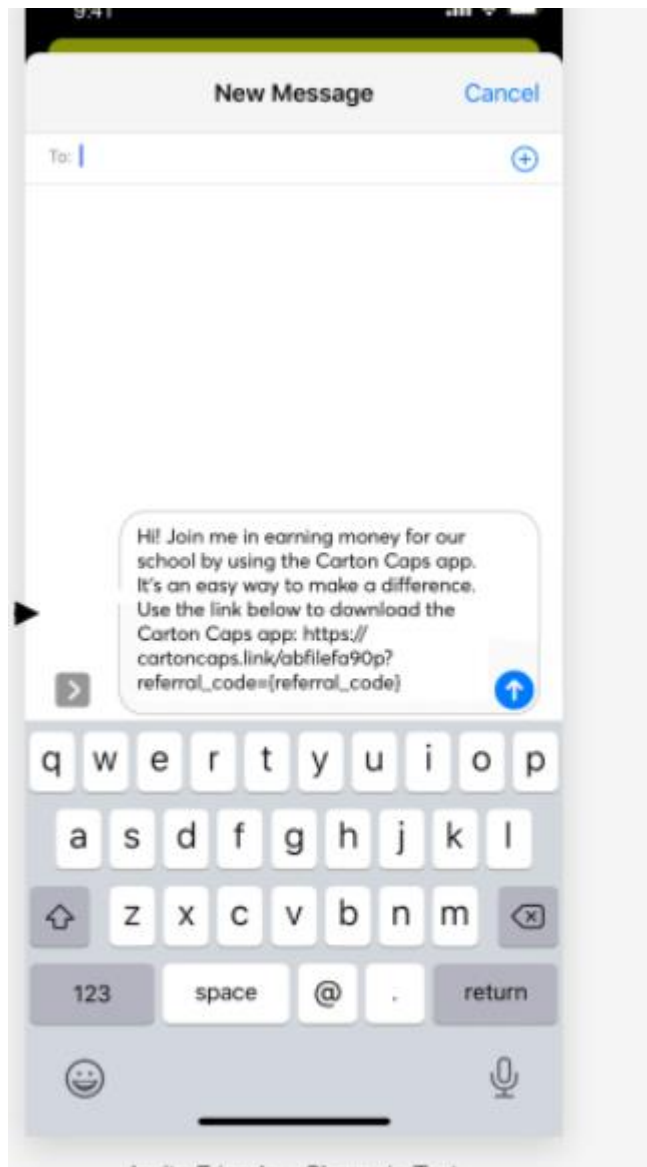
- The base link required to build the invitation URL is empty
Message: "Base link is required."
- The base link is not a valid absolute URL
Message: "Invalid base link URL."
- The referral code base (random character pool) is empty
Message: "Referral code base is required."
- The userId is empty
Message: " UserId cannot be empty.."

If there is an unexpected problem with the database when saving the invitation or generating the referral code: 500 Internal Serve Error

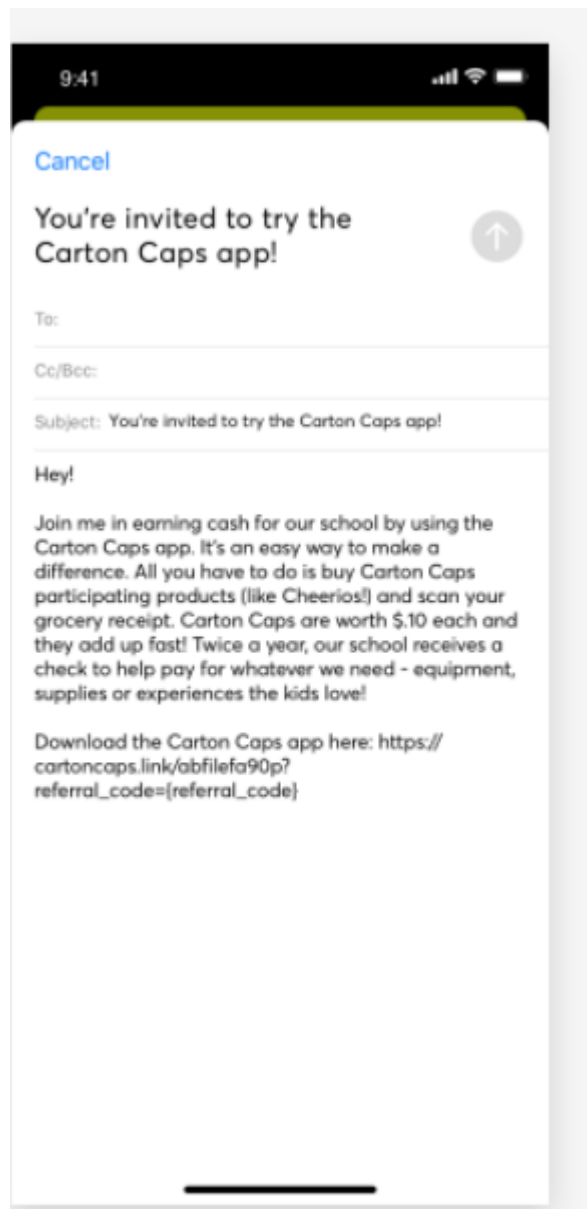
- The screen should display the email, SMS, and share options. Open these options by sending the link returned by the API.



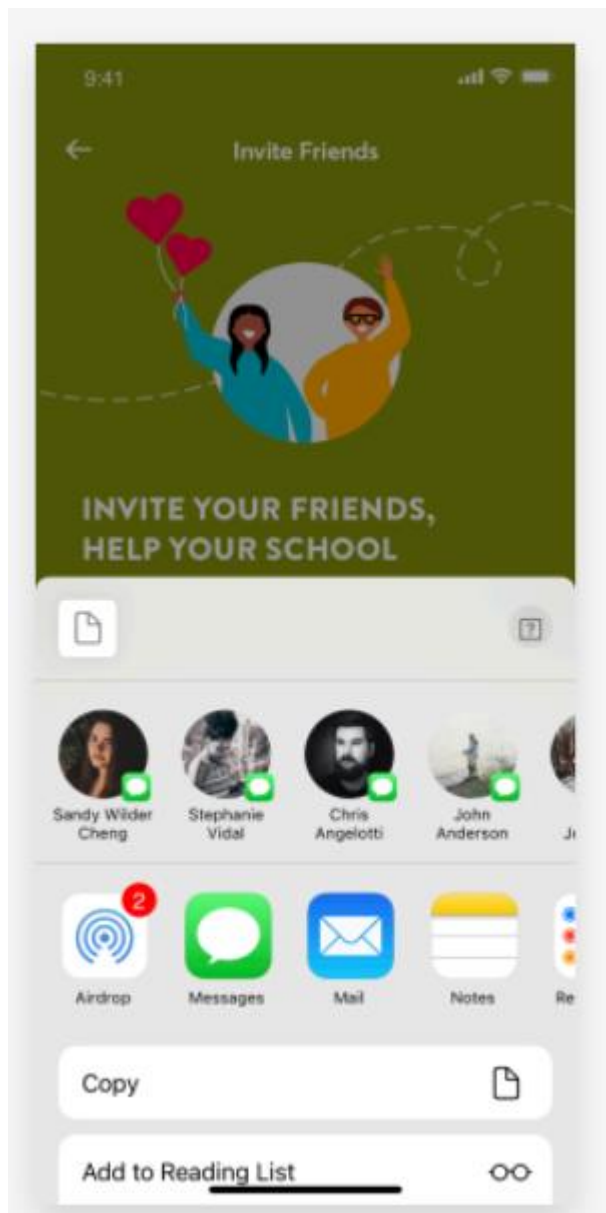
SMS :



Email:



Share:



Redirect Endpoint and Click Tracking to Mitigate Abuse (backend)	Story ID : 5
<p>Description:</p> <p>As a user, I want an endpoint to run when the invited friend clicks on the sent link, to manage and track the clicks. I also want to redirect them to the app store to download the app.</p> <p>So that we can track each click on the referral link and mitigate potential abuse.</p>	
<p>Acceptance Criteria:</p> <ul style="list-style-type: none"> • Create the general structure of the API responsible for managing visits or clicks that are executed on friend invitation links(ReferralsVisitController). • Create the first API endpoint (CreateReferralVisit) to record visits or clicks on the referral link so that we can prevent abuse, and return the URL that redirects the referred friend to the store to download the app. • Note for the challenge environment: The redirect endpoint (/api/ReferralsVisit/redirect) returns an HTTP 200 response, but Swagger displays the message: Failed to fetch. Possible Reasons: - CORS - Network Failure - URL scheme must be "http" or "https" for CORS request. This happens because the redirect URL is a simulated link (configured in appsettings.json) that represents the App Store download page, and browsers block redirects with external URLs when called from Swagger. For challenge purposes, a dedicated endpoint was implemented to allow testers to review all click records and verify the correct behavior of the referral visit logic(/api/ReferralsVisit/referralVisitHistory). If the referral code does not exist: (404 Not Found): <pre>{ "message": "Referral code not found." }</pre> • Business Rules / Validations: <ul style="list-style-type: none"> - Do not register the link click if the referral record is already in the Completed state. - For the purposes of this challenge, new user registration is not considered. However, as part of the analysis, it is inferred that in this registration, or when the friend registers, in addition to registering the new user, the relationship between that new user and the referring user will be stored in a field in the users table. 	

Additionally, this process must update the referral record and set it to Completed.

- The number of clicks made by invited friends must be monitored. A time interval must be set to allow the system to register each new click made by the same friend. For the purposes of this challenge, a 1-minute interval was set for display in the tests.

Note: When a click is rejected by **Business rules**, the service logs the reason instead of storing the event. This is only for challenge review; as a future improvement, these rejected click attempts should be persisted in a dedicated database table for auditing and abuse-prevention purposes

Technical Notes:

- Authentication is not verified: it is assumed that the application sends a valid user GUID.