**Test Plan**: **Campaign Scheduling Feature**

**Prepared by:**

**Sunil Kumar M S**

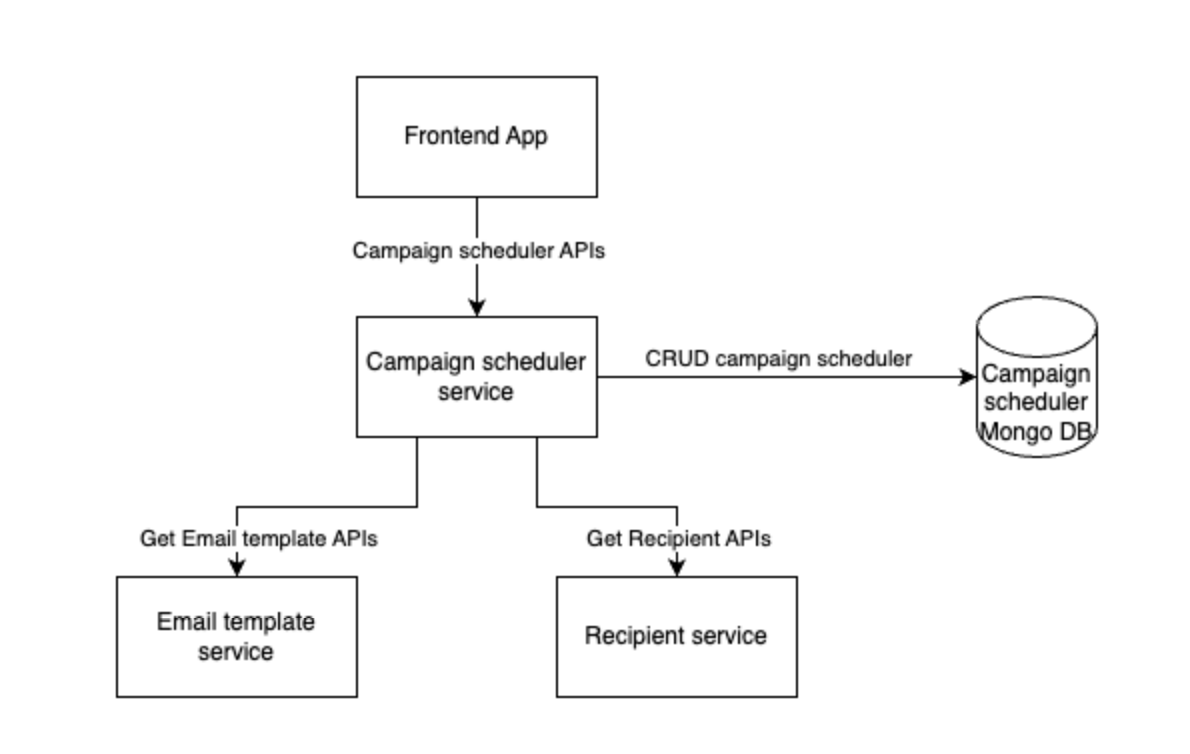
1. **Introduction**

**Purpose**: This test plan outlines the approach and strategies for testing the Campaign Scheduling feature of the Inflection application.

**Scope**: The test plan covers end-to-end and integration testing of the Campaign Scheduling feature, including the creation, scheduling, and execution of email campaigns.

**Note : This is not the final document . It will be updated iteratively as per the scope .**

**HLD**



2. **Test Objectives**

* Verify the functionality of the Campaign Scheduling feature, ensuring that users can create, schedule, and manage email campaigns.
* Validate the integration between the Campaign Scheduling service and the Recipient Management and Email Template services.
* Ensure the overall reliability, performance, and scalability of the Campaign Scheduling feature.

3. **Test Methodology**

**Test Approach**: The test plan will include both end-to-end and integration testing.

**Test Techniques**: The test cases will cover functional, integration, and non-functional (e.g., performance, security) aspects of the Campaign Scheduling feature.

**Test Levels**: The testing will be conducted at the following levels:

**Unit Testing**: Performed by the development team to ensure the correctness of individual components.

**Integration Testing**: Performed to validate the integration between the Campaign Scheduling service and the Recipient Management and Email Template services.

**End-to-End Testing**: Performed to verify the overall functionality of the Campaign Scheduling feature, including the end-to-end flow.

**Non-Functional Testing**: Performed to assess the performance, security, and scalability of the Campaign Scheduling feature.

4. **Test Environment**

**Hardware and Software**: The test environment will be set up using the provided Docker Compose files, as mentioned in the assignment.

**Dependencies**: The test environment will include the necessary dependencies, such as the Recipient Management and Email Template services, as well as any other required components.

**Test Data**: Appropriate test data, including recipient lists, email templates, and campaign details, will be prepared and maintained for testing purposes.

5. Execution plan  
**End-to-End Testing**

**Aim:** To ensure the end-to-end functionality of the Campaign Scheduling feature.

**Tasks:**

* Create test cases for scheduling email campaigns with specific dates and times.
* Verify the accuracy and efficiency of sending large volumes of emails.
* Test the selection of recipient lists and email templates for scheduled campaigns.
* Validate the ability to edit and cancel scheduled campaigns.

**Execution:** Utilize Docker compose files and access services at provided URLs for API contracts.

**Output:** Detailed end-to-end test plan covering all functional aspects of the Campaign Scheduling feature.

**Integration Testing**

**Aim:** To test the integration of the Campaign Scheduling service with other relevant services.

**Tasks:**

* Create test cases for integrating campaign scheduling service with recipient management and email template services.
* Verify the functionality and performance of the integrated services.
* Validate the base URL for the mock APIs of email and recipient services as environment variables.

**Execution:** Utilize Docker compose files for integration testing.

**Output:** Detailed test plan focusing on the integration of the Campaign Scheduling service with other services.

Test cases are documented at the end of the doc

6. **Test Execution and Reporting**

**Test Execution:** The test cases will be executed using the provided scripts or tools, as mentioned in the assignment.

**Test Reporting:** The test results will be documented, and any defects or issues will be tracked and reported using a suitable bug tracking system.

7. **Risk Management**

**Potential Risks:** Identify and analyze potential risks that may impact the testing process, such as data integrity issues, service unavailability, or performance bottlenecks.

**Risk Mitigation Strategies:** Develop strategies to mitigate the identified risks, such as implementing data validation checks, monitoring service health, or conducting load testing.

8**. Conclusion**

This test plan outlines the approach and strategies for testing the Campaign Scheduling feature of the Inflection application, covering both end-to-end and integration testing.

The test plan will be reviewed and updated as necessary throughout the testing process to ensure the quality and reliability of the Campaign Scheduling feature.

5. **Test Cases**

**End to End Test Cases**

Note: Canceling campaign api is not implemented yet . So excluding it for now and will added when it is available

**Pre -requisite** : Both e2e & integration dockers should be running and live.

**Test Case 1: Campaign Flow**

**Objective:** Validate the end-to-end flow of creating, updating, associating recipients and email templates, and canceling a campaign.

**Steps:**

* Create a new campaign with a scheduled send time and date.
* Retrieve the details of the created campaign to verify successful creation.
* Select a recipient list and associate it with the campaign.
* Choose an email template and apply it to the campaign.
* Update the name of the campaign.
* Verify that the campaign details reflect the updated name.

**Test Case 2: Email Template Retrieval**

**Objective:** Ensure the ability to retrieve and view email templates.

**Steps:**

* Fetch a list of available email templates from the Email Template Service API.
* Check the details of a specific email template to ensure its content and structure.

**Test Case 3: Recipient Management**

**Objective**: Verify the functionality of retrieving recipient lists and individual recipients.

**Steps**:

* Retrieve a list of recipients from the Recipient Service API.
* Select an individual recipient from the list to view their details.

**Integration Test Cases**

**Pre - requisite** : Both e2e & integration dockers should be running and live.  
 Email template service and Recipient service are downstream services for Campaign service where downstream services were called to check the availability of template & recipient by their provided id .

**Test Case 1: verify the campaign creation with empty emailTemplateId**

**Steps:**

* Create a new campaign with empty email template id

**Expected result :** campaign creation api should throw template id not be null/blank error.

**Test Case 2: verify the campaign creation with null emailTemplateId**

**Steps:**

* Create a new campaign with null email template id

**Expected result :** campaign creation api should throw template id not be null/blank error.

**Test Case 3: verify the campaign creation with invalid emailTemplateId**

**Steps:**

* Create a new campaign with invalid email template id

**Expected result :** campaign creation api should throw template id error.

**Test Case 4: verify the campaign creation with empty recipientId**

**Steps:**

* Create a new campaign with empty recipientId

**Expected result :** campaign creation api should throw recipientId not be null/blank error.

**Test Case 5: verify the campaign creation with null** recipientId

**Steps:**

* Create a new campaign with null email recipientId

**Expected result :** campaign creation api should throw recipientId not be null/blank error.

**Test Case 6: verify the campaign creation with invalid emailTemplateId**

**Steps:**

* Create a new campaign with invalid email recipientId

**Expected result :** campaign creation api should throw invalid recipientId error.