

**Tobor Inc.**

App Automation: Register, Deliver and Report

Detailed Process Description

Version 1.10

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Date Issued | Version | Description | Author |
| *16.06.2020* | *1.0* | *Initialize* | *James Harper* |
| *17.06.2020* | *1.01* | *Updates* | *James Harper* |
| *18.06.2020*  *21.06.2020*  *23.06.2020*  *28.06.2020* | *1.02*  *1.03*  *1.10*  *1.2* | *Updates*  *Updates*  *Finished alpha file. Ready to be sent for basic review with some properties needing finished*  *Finalized document* | *James Harper*  *James Harper*  *James Harper*  *James Harper* |

Contributors

The content of this document has been authored with the combined input of the following group of key individuals.

|  |  |  |
| --- | --- | --- |
| Name | Role | Area |
| *James Harper* | *RPA Developer* | *Consulting* |

Business Sign-off

The following table contains the people required to sign-off and/or review this document and those that require the document for information only.

|  |  |  |
| --- | --- | --- |
| Name | Department | Responsibility |
| *David Bradbury* | *Management Director* | *Sign Off* |
| *Roberto Fernandez* | *Backend Application Manager* | *Review* |
| *Chris Lucas* | *Consultant Project Liaison* | *Information* |

Document Classification

|  |  |
| --- | --- |
| Classification | *Person Confidential* |
| Definition | *Program is to be kept within the domain of Roberto Fernandez* |
| Context | *Customer details are to be kept within the program itself and not taken outside of the scope where possible. Knowledge of the coding itself is to be kept with the Application manager* |
|  |  |

Contents

[1 Introduction 4](#_Toc26352448)

[2 Manual Process 4](#_Toc26352449)

[2.1 Overview 4](#_Toc26352450)

[2.2 Detailed Process Flow 4](#_Toc26352451)

[3 Automation Proposal 4](#_Toc26352452)

[3.1 Overview 4](#_Toc26352453)

[3.2 Automated Process Flow 4](#_Toc26352454)

[3.3 Target Systems & User Requirements 4](#_Toc26352455)

[3.4 Impacted Business Areas 5](#_Toc26352456)

[3.5 Workload 5](#_Toc26352457)

[3.6 Operational Constraints 5](#_Toc26352458)

[3.7 Delivery 5](#_Toc26352459)

[3.8 Contact List 5](#_Toc26352460)

[4 Automation Details 6](#_Toc26352461)

[4.1 Automation Walkthrough 6](#_Toc26352462)

[4.1.1 *First robot action* 6](#_Toc26352463)

[4.1.2 *second robot action* 6](#_Toc26352464)

[4.1.3 *third robot action etc.* 6](#_Toc26352465)

[4.2 Reporting 6](#_Toc26352466)

[4.2.1 Business Exceptions 6](#_Toc26352467)

[4.2.2 System Exceptions 6](#_Toc26352468)

[4.2.3 Performance 7](#_Toc26352469)

[4.2.4 Triggers 8](#_Toc26352470)

# 1 Introduction

*Tobor Inc. has requested consultation with their app automation. This process involves taking in registering details from a customer to create a presence on their software.*

*This then goes onto expand with taking in data from a selection of websites that the user selects via a pre-defined list of hobbies, e.g. sports and hobbies. Then the automation goes onto tidy up the data and send it to the user.*

*Lastly, the user options they requested are tallied down and noted as means of data collection for Tobor Inc. to help with further development.*

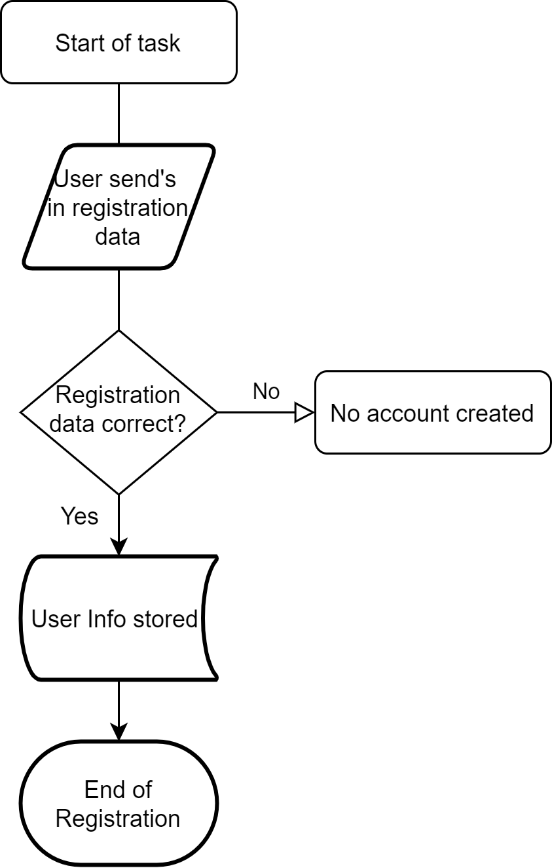
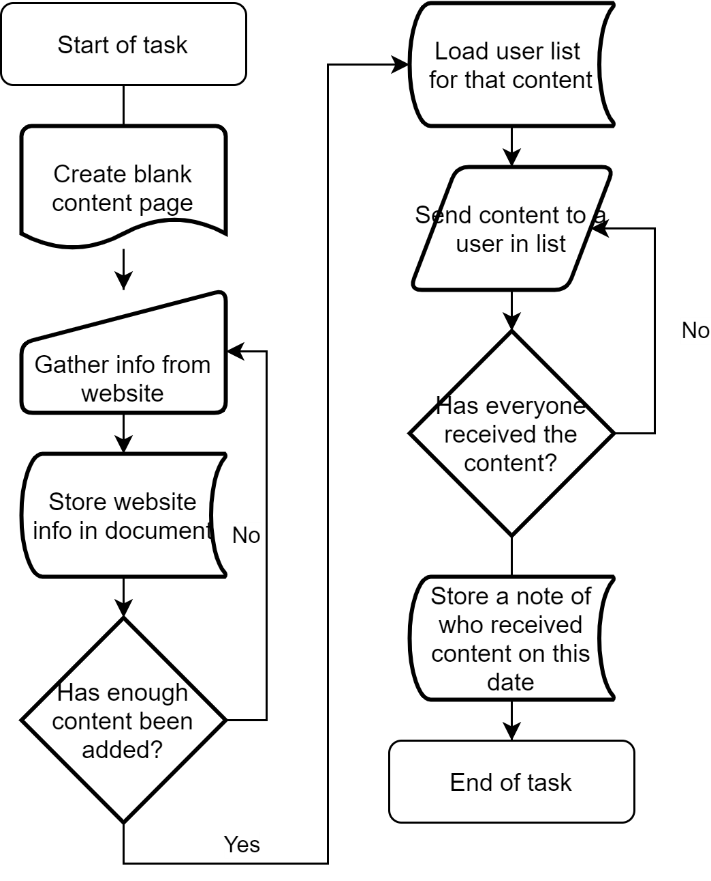
# 2 Manual Process

## 2.1 Overview

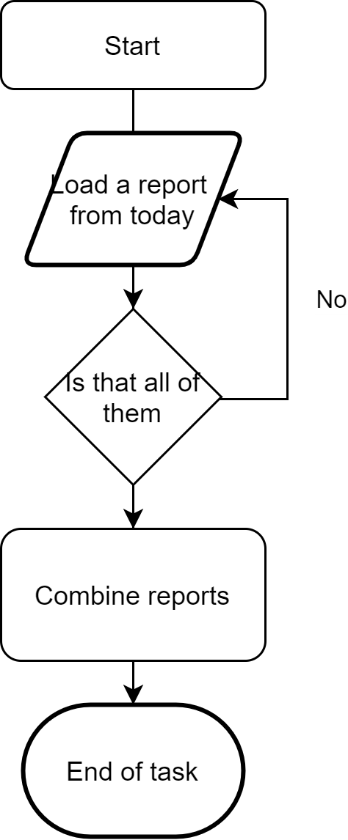
* *User sends their registration details to the company email. Details being the user’s first name, last name, phone number, address, email address, the content they request and the frequency it gets delivered.*
* *Details are then stored by the company, usually locally, with the option of changing/removing them.*
* *Content is taken from three different website options (Sport, Tech, Hobbies) daily. Content being described as a brief description of what’s going on.*
* *A report is taken before the content is sent off, noting who will receive that aggregated data and what the day is.*
* *These reports are then compiled together into a larger report of what was sent out on that day. This is stored locally on a company computer.*

## 2.2 Detailed Process Flow

Registration: Content Delivery:

Report Collation:



# 3 Automation Proposal

## 3.1 Overview

*The automation proposed would take care of the registering, content aggregation, report filing and content delivery. There is to be an overarching automation aspect that happens at half eleven in the morning triggering all the processes contained. If any submissions are to be made after this deadline, they will need to be held until the next day.*

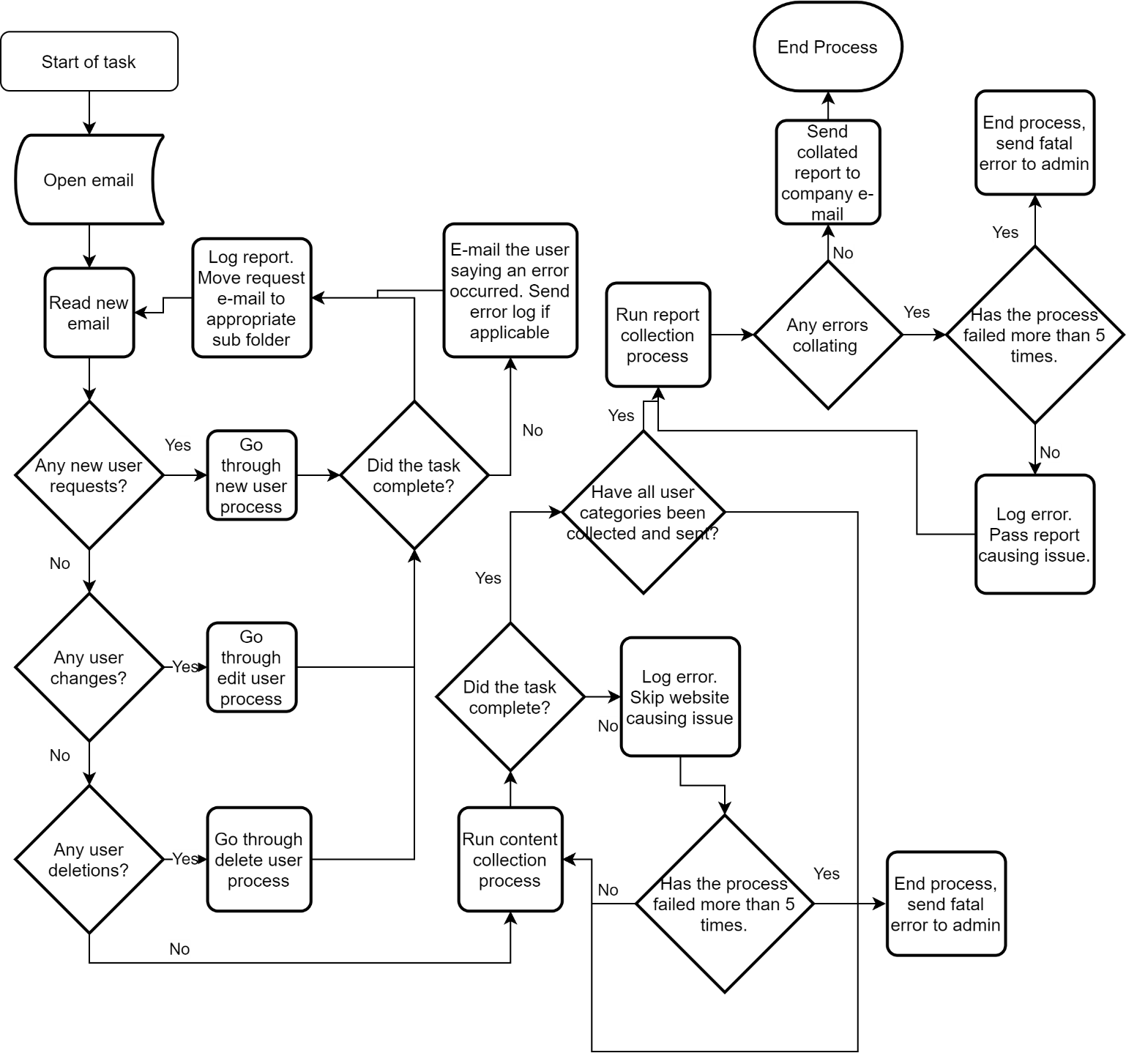
*Registering would occur via the user sending their details, as described in the manual process, to the company email. In this case we’re using a new outlook e-mail to simulate that as accessing a company email isn’t possible. The Automation involved will then log these details down and send them to the UIPath Orchestrator as an individual object with its own unique identifier. If any of these users send an incomplete dataset, or there are other problems an email will be sent back to the user stating as such, while also logging a fatal error with UIPath. A follow up e-mail can then be sent at the user’s discretion if they want their detail altered or removed with sending a subject of ‘CHANGE’ or ‘DELETE’.*

*The second automation proposes that select websites be used for the content aggregation. For this, each hobby will have a list of websites it needs to access for it’s data set. Every day the automation will go to the websites given and collect snippets of data, a title and first paragraph for example, depending on the website. The automation will then take the snippets and format it to look more presentable to the user, then send the final product in an email. Once sent a local report will be generated to say who this was sent to, and on what date. This will then be repeated for every hobby class for that day.*

*Lastly, the content reports are collected for that day, collated together and sent to the company email, or the QA email in this case. These are marked with how many people ordered that topic for that day.*

## 3.2 Automated Process Flow

*Main Process:*

**

## 3.3 Target Systems & User Requirements

| Name | Description | User Permissions/Access |
| --- | --- | --- |
| *MS Outlook* | *Email inbox simulating the company box* | *Company Inbox required: TroborUser@Outlook.co.uk* |
| *Gmail* | *Email inbox for the user* | *User Inbox required: waddupitsthatrobot@gmail.com* |

## 3.4 Impacted Business Areas

* *Sales*
* *IT*
* *Management*
* *Marketing*

## 3.5 Workload

|  |  |
| --- | --- |
| *Avg no. of Login Requests per week* | *5* |
| *How many people do this process per day?* | *1* |

***Automating the steps below will realise an average time saving of 530 minutes (9.18 hrs) per day for the total process:***

* *New registrations often took a max of 15 minutes per user, with changes taking around the same time. Estimate is based off 10 new users per day.*
* *Aggregation of the content can take a max of 2 hours.*
* *To send this all off is then another 10 minutes.*
* *Reporting was only done when time allowed, with no estimate of how long it taken. However, manually collating the reports should only take around 10 minutes, with an extra 5 per report. With an established base of 50 users a day this would go to 250 minutes.*

## 3.6 Operational Constraints

* *Website maintenance*
* *Email system maintenance*
* *Coding errors*

## 3.7 Delivery

*This is to be delivered on the twenty sixth of June*

## 3.8 Contact List

*David Bradbury – Managing Director*

*Roberto Fernandez – Backend Application Manager and Project PM*

*Chris Lucas – Consultant Liaison*

# 4 Automation Details

## 4.1 Automation Walkthrough

There is a single robot running the entire process chain. As such the following steps will be labelled with process actions instead of robot actions for user readability.

### 4.1.1 *First Process action (main)*

* Error count is set up. Initialised as zero.
* Flow decision of if a config file was detected.
  + If no, an error is reported to Orchestrator.
  + If yes, the file is loaded.
* Stage count for main process is initialised and set to 0.
* Report structure is initialised for C.R.U.D, the content, and the users to receive it.
* First flow switch detects the current stage.
* Stage 0 begins with a log will being sent to Orchestrator, detailing that the company email has been loaded from the config, and the emails contained within are being loaded into UIPath.
* A variable detailing how many emails have been changed is set up. This is initialised at 0.
* A flow decision detects if any new un-read emails have been received.
  + If there’s none detected, a log is sent to Orchestrator as a warning detailing that no new emails were picked up.
  + If any new emails are detected a for loop will begin with process 4.1.2 and will repeat until no new emails are detected. Emails changed is increased by one for every email changed/added/deleted.
* The stage is set to stage plus one.
* Return to first flow switch in main, this then detects the new stage.
* A log is sent of how many emails were changed.
* Stage 1 begins with a log being sent off declaring as such.
* Process 4.1.3 is invoked.
* Stage is set to stage plus one.
* This then returns to the flow switch.
* The third stage then fires. This stage is set to the default stage, as such if anything other than zero or one is detected this stage will fire.
* Process 4.1.4 is then invoked within a try catch statement.
  + If this fails a fatal error is sent to Orchestrator detailing as such, as well as the error detected.
* The tallied report is then sent off to the “ReportReciever” that is contained inside of the config file.
* A report is then sent off to Orchestrator detailing that the program has finished, with the total amount of errors.

### 4.1.2 *second PRocess action (Register)*

* Variables Emailbody, UserOption, Sender, Bodysplit and Index are initialised.
  + EmailBody being the loaded emails’ plaintext content.
  + UserOption being the loaded emails subject line.
  + Sender being the laoded emails’ sender.
  + Bodysplit being the EmailBody string, but split up with all empty spaces removed, and separated into a string with delimiter of a tab space.
* First flow switch of the CRUD method. This detects the UserOption.
* If “DELETE” is detected:
  + A method is invoked which goes into the local excel file detailed in the config. This then detects the user’s row by their unique ID and deletes it from the system.
  + If this fails at any point, a log is sent off to Orchestrator with the user’s email and the exception that occurred.
* If “UPDATE” is detected:
  + A method is invoked which goes into the local excel file detailed in the config. This then detects the user’s row by their unique ID and updates it witjh the information given.
  + If this fails at any point, a log is sent off to Orchestrator with the user’s email and the exception that occurred.
* If “CREATE” is detected:
  + A TimeRightNow, and a UniqueID variable are set up. The TimeRightNow variable being the time in seconds from epoch time. The unique ID is then set to this value, rounded to no decimal places.
  + A method is then invoked adding the user to the local excel file, with all the details given in the email body.
  + If this was successful, an email is sent back to the user, and the email moved to the create folder inside the company email.
  + If this wasn’t successful the user is then sent a similar email but informing them the company will be in touch soon regarding the error. A log is placed with Orchestrator as to why this failed.
* If any other subject is detected:
  + A log will be sent to Orchestrator detailing what was sent, with an email sent to the user informing them of the issue and that someone will be in touch.
  + The email received from the user is then placed in the errors folder for investigation.

### 4.1.3 *third Process action (Content Aggregation)*

* *The variables DayName and DayDate are set up. These are the day’s name and the date respectively.*
* A method is then invoked retrieving the users form the local excel file.
* The users are then sorted itno if they should be receiving content today.
* The ones that are receiving content are uploaded to Orchestrator into a queue designated for weekday or daily usage.
* A new stage is initialised inside this method. It is set to 0.
* A flow switch is used to filter which content is to be aggregated.
  + If 0 is detected, the Tech content is aggregated from three different websites and formatted into a user-friendly version. The content collected is then added to the content collection report as a new row.
  + If 1 is detected, the Hobby content is aggregated from three different websites, with one focusing on tv news, one on music and the last on general entertainment. Again, this is then formatted into a user-friendly version. The content collected is then added to the content collection report as a new row.
  + If 2 is detected, the Sports content is collected. One website collects what matches are on for that day, the next detects any cricket news, the last one detects any UFC news. Again, this is then formatted into a user-friendly version. The content collected is then added to the content collection report as a new row.
  + If anything other than 0, 1, or 2 is detected a log is sent to Orchestrator detailing as such. This process then ends.
* A Day detector then identifies which Orchestrator queue is used.
* Another flow detector is then used to identify a user on the queue identified. If there are no more users, the process ends.
* A flow switch is then used to detects which content is to be sent. The process is the exact same no matter what is detected except the ContentToSend variable is set to the respective content detected.
* The content is then formatted into an email to be sent to the user.
* The content is then sent off. If this fails a log is sent to Orchestrator detailing as such, alongside the exception.
* The user’s transaction status within the Orchestrator queue is then set to processed.
* The user is then added to the content report, detailing the content they’ve received.
* This process then loops back to the flow detector that identifies if there are any users left to have their content sent out.

### 4.1.4 *Forth Process action (Report Collation)*

* *A Word file is opened within the asset location specified within the config. The date is then appended as the save name for the file.*
* *Three text strings are appended.*
  + *The first detailing the amount of total errors up until that point.*
  + The second, third and forth simply being bookmarks for the reports to be inserted at.
* The user report, content collected report and users who received content reports are then inserted into the word document.
* The file is then saved, and exported as a .pdf file.

## 4.2 Reporting

### 4.2.1 Business Exceptions

|  |  |
| --- | --- |
| Exception | Solution |
|  |  |

### 4.2.2 System Exceptions

|  |  |
| --- | --- |
| Exception | Solution |
| *Files not being detected.* | *Change location in the config file.* |
| *Website content changing selector anchors* | *A new website will have ot be chosen.* |

A performance report will be emailed to *Tobor Inc.*  each time the process runs (showing worked cases, exceptions and a cumulative processing log)

### 4.2.3 Performance

Once the processes have successfully completed a performance report and processing log will be emailed to *Roberto Fernandez* as an excel file.

**Performance Report**

This will contain all exceptions (business and system) and successes for the automated Process, based on the last automation execution completion (i.e. based on the last time the process ran). These will all be logged with Orchestrator.

**Processing Log**

This will show cumulative successes from the automated Process:

**CredBest**

**CredBest**

### 4.2.4 Triggers

*The robot will be triggered at half eleven every morning. Then, depending on user preference, the aggregation and delivery method will fire every day or only every weekend.*