**Functional Design Document**

**Group: 1**

**Members: Egidijus Ukrinas**

**Rafael Tavares**

**Class: DHI1V.Sq**

**Teacher: CBR23**

**Table of Contents**

[1. Introduction 3](#_Toc163147715)

[2. Business goals 4](#_Toc163147716)

[3. Context and data subjects 5](#_Toc163147717)

[Stakeholder analysis 5](#_Toc163147718)

[Users 6](#_Toc163147719)

[System context diagram 8](#_Toc163147720)

[4. Requirements 9](#_Toc163147721)

[Use case diagrams 9](#_Toc163147722)

[User and system requirements 10](#_Toc163147723)

[Quality requirements 15](#_Toc163147724)

[5. Wireframes 16](#_Toc163147725)

[Login Wireframe 16](#_Toc163147726)

[Account Creation Wireframe 17](#_Toc163147727)

[Account viewing, editing and deletion Wireframe 17](#_Toc163147728)

[Administrator Wireframes 18](#_Toc163147729)

[Accountant Wireframes 20](#_Toc163147730)

[6. Conclusion 21](#_Toc163147731)

[Appendix A: The Interview 22](#_Toc163147732)

[Appendix B: Tests 28](#_Toc163147733)

[Traceability Matrix 28](#_Toc163147734)

[Test cases 29](#_Toc163147735)

# Introduction

The Functional Design Document (FDD) acts as a detailed blueprint of the functional aspects of the system developed for Run for Life. Its purpose is to offer clarity on how the system will meet requirements and fulfill the stakeholders' needs. By defining functional specifications, features, and interactions, the document ensures alignment among project team members and stakeholders concerning the intended functionality of the solution. The document follows a structured approach: It begins by introducing the business goals and then conducts a thorough stakeholder analysis. Following this, it outlines the user profiles and their roles within the system. A context diagram is provided to illustrate the system's interactions with external entities. This sets the stage for a detailed use case diagram that shows how users interact with the system. User and quality requirements are also specified with all the wireframes. The document ends with several tests made in the system.

The project was made to create a robust system to handle registrations, donations, and event coordination for Run for Life, an organization dedicated to hosting races for charitable causes. This system aims to streamline processes for both accountants and administrators, ensuring seamless file processing without data loss for runners and accurate donation processing for sponsors, while also facilitating efficient event management for the organization.

The new application will cover registration management, donation processing, and event coordination functionalities. The FDD details specific features, requirements, and workflows essential for achieving project objectives effectively.

# Business goals

For Run for Life, who want to get their website files to be automatically processed,

the application CSV Cruncher (CC) will process and categorize the files that the website provides,

Unlike now where the files are manually getting processed and getting mishandled and prone to result in errors, CC will process the files with more accuracy and be less prone to errors.

For Run for Life, who want to keep track of the runners and the registrations from sponsors for the charities. The application CSV Cruncher (CC) will fetch and store the information about each race and sponsor registrations in their correct format. Unlike now the files are getting lost or entered incorrectly, the system will store and not lose track of the sponsor registrations or runner information.

# Context and data subjects

## Stakeholder analysis

**The director** holds a high role in the project's strategic direction and decision-making. Their leadership is crucial for ensuring that the project aligns with Run for Life's goals and objectives. The success or failure of the project directly impacts the organization. By providing strategic guidance and overall support, the Director influences the project's trajectory and fosters alignment with the organization's mission.

**The runners** are individuals who participate in the races organized by Run for Life. They want the registration for races to be seamless. Runners are impacted by errors that may occur with manual processing of their information, so the project ensures that it does not happen.

**The sponsors** are organizations or individuals who provide financial donations to charities. They want their donations to be sent to the charities. Sponsors are impacted by their registrations going missing, so the project makes sure that their donations are accounted for in the system.

**The development team** are the ones who are responsible for designing and building the application for the project. The development team ensures that the application meets the functional requirements outlined in the document. They also shape the application's architecture, functionality, and performance.

**The charity**, a key stakeholder, relies on donations from Run for Life races to support their initiatives. Their function involves receiving donations. They're affected by the project's outcomes, as errors in donation processing can hinder their income.

**The Run for Life employees**, crucial stakeholders, manage daily operations including event planning and registration. Their function supports project execution. They're directly impacted by project outcomes, where a well-functioning system streamlines their tasks. Their input guides project development, ensuring systems align with operational needs and workflows effectively.

## Users

**Sergio Perez, Admin**

Sergio Perez is a dedicated administrator in his mid-30s, entrusted with the responsibility of managing the details of runners, sponsors, and event information at Run for Life. Over the years, Sergio has encountered numerous challenges in his role, particularly concerning the manual entry of data which often led to inaccuracies and the appearing risk of data loss.

**Goals:** Sergio´s primary goal is to ensure the efficiency and accurate management of all runners, sponsors, and events information within the organization. He aims to mitigate the challenges posed by manual data entry, striving for a streamlined process that minimizes errors and enhances overall operational efficiency.

**Scenario:** With the recent implementation of a new system, Sergio's role as an administrator has undergone a significant transformation. The updated system introduces advanced features that streamline the data input process, significantly reducing the likelihood of inaccuracies and data loss. Furthermore, robust measures for data management and editing have been integrated, offering new capabilities for handling runner information, and addressing concerns surrounding data reliability.

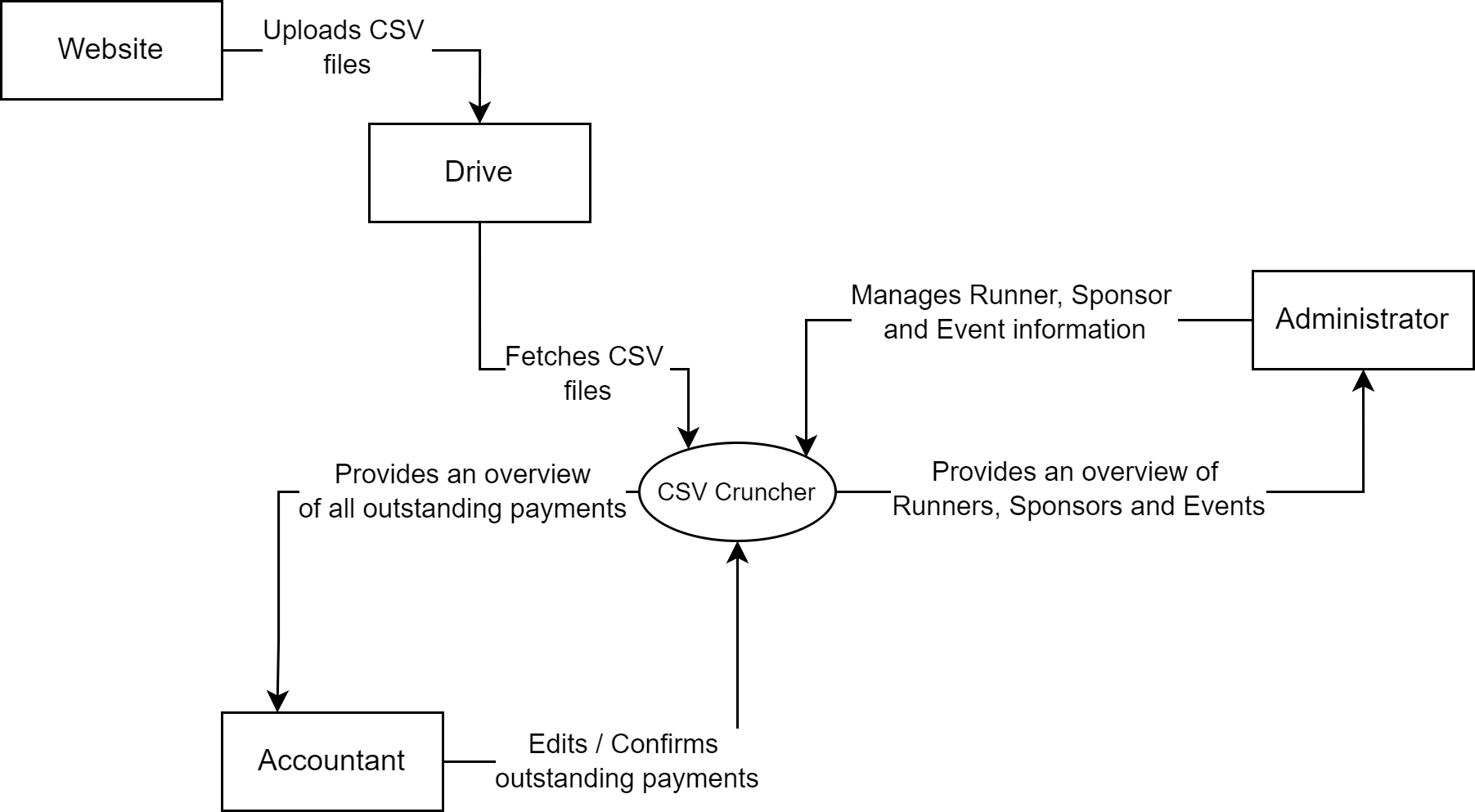
**Lewis Hamilton, Accountant**

Lewis Hamilton is a dedicated accountant in his late 30s, responsible for managing the financial aspects of Run for Life. Despite facing distinct financial tracking challenges in the past due to the limitations of the outdated methods of data handling, Lewis has remained committed to ensuring the financial integrity of the organization's sponsored runs.

**Goals:** Lewis's primary goal is to streamline the financial tracking process and enhance the accuracy of financial records related to runner registrations, sponsorships, and donations. He aims to overcome the challenges posed by the previous method of financial management, striving for a more efficient and error-free accounting system.

**Scenario:** With the recent implementation of a new system, Lewis's role as an accountant has experienced a significant transformation. The updated system introduces advanced financial tracking tools such as filters and organized data, enabling Lewis to effectively monitor outstanding payments and better track financial transactions related to runner registrations, sponsorships, and donations.

## System context diagram



In the diagram the main entities include the Website, Drive, Administrator and Accountant, all of which are either connected directly or indirectly to the CSV Cruncher application.

The Website, an existing system owned by Run for Life, its purpose for the application is to upload CSV files to the Drive, from which CSV Cruncher fetches those CSV files.

Administrators are employees of Run for Life; they use the application to manage Runner, Sponsor and Event information. The application provides them with an interface to view and edit their details and add new Runners, Sponsors and Events.

Similarly, the Accountant, also an employee of Run for Life, uses the application to confirm and edit outstanding payments from Runners and Sponsors. The application presents an overview of all payments from Runners and Sponsors, where the accountant can edit and confirm them.

# Requirements

## Use case diagrams

A diagram of a diagram

Description automatically generated

The first case diagram from CC (CSV Cruncher) represents how the accountant uses our system. In this scenario, the accountant views outstanding details regarding runners and sponsors. Also, they verify payments from runners and sponsors and can change any information regarding outstanding payments.

The second case diagram from CC (CSV Cruncher) represents how administrators will use our system. In this scenario, administrators will have the ability to add new runners, sponsors, or events, change any details regarding them, and access their information. They can also review proposed events and delete them if they are deemed unsuitable or have already been added to the system. The administrator can also export BIB numbers as a PDF for an event.

There is also an administrator super user, which, besides having the same capabilities as a normal administrator, also has access to create new accountant or administrator accounts.

## User and system requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature: CSV file processing** | | **Prio** | **Source** |
| **U1** | As a user, I want the application to process the CSV files, so that I would not need to manually do them | Must | Letter |
| **S1.1** | The system should be able to process CSV files | Must | Letter |
| **S1.2** | The system should automatically fetch CSV files from the drive | Must | Interview |
| **S1.3** | The system should delete the CSV files that have already been fetched from the drive | Must | Interview |
| **S1.4** | The system should upload the processed CSV files to the drive | Must | Interview |
| **S1.5** | The system should be able to have a button to tell the system to fetch any new CSV files from the drive | Must | Interview |

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature: Runner, Sponsor and Event information viewing and editing** | | **Prio** | **Source** |
| **U2** | As an administrator, I want to be able to view Runner, Sponsor and Event information, so I can maintain oversight and ensure accountability. | Must | Letter |
| **S2.1** | The system should be able to list all the Runners, Sponsors and Events | Must | Letter |
| **S2.2** | The system should be able to filter Runners, Sponsors or Events | Should | Interview |
| **S2.3** | The system should be able to search for specific Runner, Sponsor or Event | Should | U2 |
| **S2.4** | The system should be able to have a scroll bar to navigate through the list, if there are too many rows to show on the screen | Must | U2 |
| **S2.5** | The system should be able to change the checkbox of a specific Runner or Sponsor to green to indicate their outstanding payment completion | Could | U2 |
| **S2.6** | The system should show the current time | Could | U2 |
| **S2.7** | The system should have an option menu to navigate through different screens | Must | U2 |
| **S2.8** | The system should show their BIB number if they paid, name, birthday, phone number, email, a checkmark if they have paid their outstanding payment and which event, they are attending for runners | Must | U2 |
| **S2.9** | The system should show the sponsor’s name, contact name, phone number, email, a checkmark if they have paid their outstanding payment and which event, they are donating towards for sponsors | Must | U2 |
| **S2.10** | The system should show the name, location, current registered runner count, date, enter cost, how much has been donated by sponsors and runners for events | Must | U2 |
| **U3** | As an administrator, I want to be able to edit Runner, Sponsor and Event information so that I can update any changes if needed | Must | Letter |
| **S3.1** | The system should be able to edit Runner, Sponsor and Event information | Must | Letter |

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature: Adding new Runners, Sponsors or Events** | | **Prio** | **Source** |
| **U4** | As an administrator I want to be able to add new Runners, Sponsors or Events, so that I could keep up to date with any changes that may occur and expand the list of events | Must | Interview |
| **S4.1** | The system should be able add new Runners, Sponsors or Events | Must | U4 |
| **S4.2** | The system should be able close the new pop-up window | Must | U4 |
| **S4.3** | The system should indicate which fields are necessary to fill in | Should | U4 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature: Outstanding payment reviewing and editing** | | **Prio** | **Source** |
| **U5** | As an accountant, I want to view outstanding payments to gain a comprehensive overview and apply filters for precise analysis | Must | Interview |
| **S5.1** | The system should be able to list all the outstanding payments | Must | U5 |
| **S5.2** | The system should be able to filter outstanding payments | Should | U5 |
| **S5.3** | The system should be able to search for specific information of the outstanding payments | Should | U5 |
| **S5.4** | The system should be able to have a scroll bar to navigate through the list of outstanding payments, if there are too many rows to show on the screen | Must | U5 |
| **S5.5** | The system should be able to change the row of a specific outstanding payment to green to indicate completed payment | Could | U5 |
| **S5.6** | The system should show the current time | Could | U5 |
| **S5.7** | The system should have an option menu to navigate through different screens | Must | U5 |
| **S5.8** | The system should show how much the outstanding amount is owed, how much has been paid, name, contact number, date, reference number and which event the outstanding payment is for | Must | U5 |
| **U6** | As an accountant I want to edit outstanding payment information so that I could keep up to date with any changes that may occur | Must | Interview |
| **S6.1** | The system should be able to change outstanding payment information | Must | Interview |

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature: Login Screen** | | **Prio** | **Source** |
| **U7** | As an application user I want to be able to login, so I could only see specific functions that are needed for me. | Must | Interview |
| **S7.1** | The system should be able to log users in with their credentials | Must | U7 |
| **S7.2** | The system should be able to log users out | Must | U7 |
| **S7.3** | The system should have a forgot password button to reset password | Must | Advice |

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature: Account Creation** | | **Prio** | **Source** |
| **U8** | As an administrator super user, I want to be able to create new accounts, so in case there are new employees that will use the system I could give them access to our system | Must | Interview |
| **S8.1** | The system should be able to create new accounts | Must | U8 |
| **S8.2** | The system should be able to set an expiration date for the account if necessary | Must | Interview |
| **S8.3** | The system should only allow super users to create new accounts | Must | Advice |
| **S8.4** | The system should have a drop-down list of what type the new account is | Must | U8 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature: Event suggestion viewing and removing** | | **Prio** | **Source** |
| **U9** | As an administrator I want to be able to view suggested events and remove them, so that I could review and decide their suitability. If an event is not suitable, I want to be able to remove it | Must | Letter |
| **S9.1** | The system should be able to list all the suggested events | Must | U9 |
| **S9.2** | The system should be able to remove suggested events | Must | U9 |
| **S9.3** | The system should be able to have a scroll bar to navigate through the list of suggested events, if there are too many to show on the screen | Must | U9 |
| **S9.4** | The system should be able to expand the row, based on how much text needs to be shown | Must | U9 |
| **S9.5** | The system should be able show the ID, date, user, name and idea for suggested events | Must | U9 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature: BIB number exporting in PDF format** | | **Prio** | **Source** |
| **U10** | As an administrator I want to be able to export a PDF with the required BIB numbers for an event, so that I could print them out | Must | Interview |
| **S10.1** | The system should be able to generate bib numbers for runners who have paid | Could | U10 |
| **S10.2** | The system should generate additional bib numbers based on the administrators need | Could | Interview |

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature: Account viewing, editing, and deleting** | | **Prio** | **Source** |
| **U11** | As a super administrator I want to be able to view, edit and delete accounts, so that in case somebody wants their password changed, or I need to delete an account I would be able to do it | Must | Interview |
| **S11.1** | The system should be able to list all the accounts in the system | Must | U11 |
| **S11.2** | The system should be able to delete accounts | Should | U11 |
| **S11.3** | The system should be able to edit account information like name, email, or account type | Should | U11 |
| **S11.4** | The system should be able to have a scroll bar to navigate through the list of accounts, if there are too many rows to show on the screen | Must | U11 |
| **S11.5** | The system should display the name, email, account type, creation date, expiration date if the account is temporary for each account | Must | U11 |
| **S11.6** | The system should only allow super users to manage accounts | Must | Advice |

Letter – Client letter (Reference 1. LETTER Run for Life Foundation)

Interview – Interview with the client (Reference 2. Interview-06-03-2024)

Advice – Advice to the client

U – User requirement

S – System requirement

Q – Quality requirement

## Quality requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **1. Availability** | | **Prio** | **Source** |
| **Q1.1** | The system should only be accessible at the office | Must | Interview |

|  |  |  |  |
| --- | --- | --- | --- |
| **2. Integrity** | | **Prio** | **Source** |
| **Q2.1** | The system checks CSV input data to see if it is in the correct format | Should | Letter |

|  |  |  |  |
| --- | --- | --- | --- |
| **3. Performance** | | **Prio** | **Source** |
| **Q3.1** | The system should have a response time of 2 seconds at least 95% of the time | Could | Advice |

|  |  |  |  |
| --- | --- | --- | --- |
| **4. Robustness** | | **Prio** | **Source** |
| **Q4.1** | The system must prevent the input of data that duplicates existing entries | Must | Advice |
| **Q4.2** | The system ensures accurate entry of outstanding payment or payment status information. | Should | Advice |

|  |  |  |  |
| --- | --- | --- | --- |
| **5. Security** | | **Prio** | **Source** |
| **Q5.1** | The system should have a login system | Could | Interview |
| **Q5.2** | The system should only be accessible at the office | Must | Interview |

|  |  |  |  |
| --- | --- | --- | --- |
| **6. Usability** | | **Prio** | **Source** |
| **Q6.1** | The system should only show the relevant options that the user requires | Could | Interview |

# 

# 5. Wireframes

## Login Wireframe

A screenshot of a login screen

Description automatically generated

Here the users will be able to log in only in the space of the office. The user will open the app and insert their credentials, after that they will click on log in and if the account is valid, they will enter their account. If needed the user is asked for their username to recover their password.

Accountants would then go to frame [Accountant.](#_Accountant_Wireframe_1)Admin staff would go to frame [Administrator.](#_Administrator_Wireframe)

## Account Creation Wireframe

A screenshot of a computer

Description automatically generated

Here the super administrator will be able to create a new account. The account can have an expiration date, for example to a part time helper. In addition, when the account is being created, a drop-down menu will appear to choose the role for the new account. Then the email of the account holder will be entered, and they will get an email to set their own password.

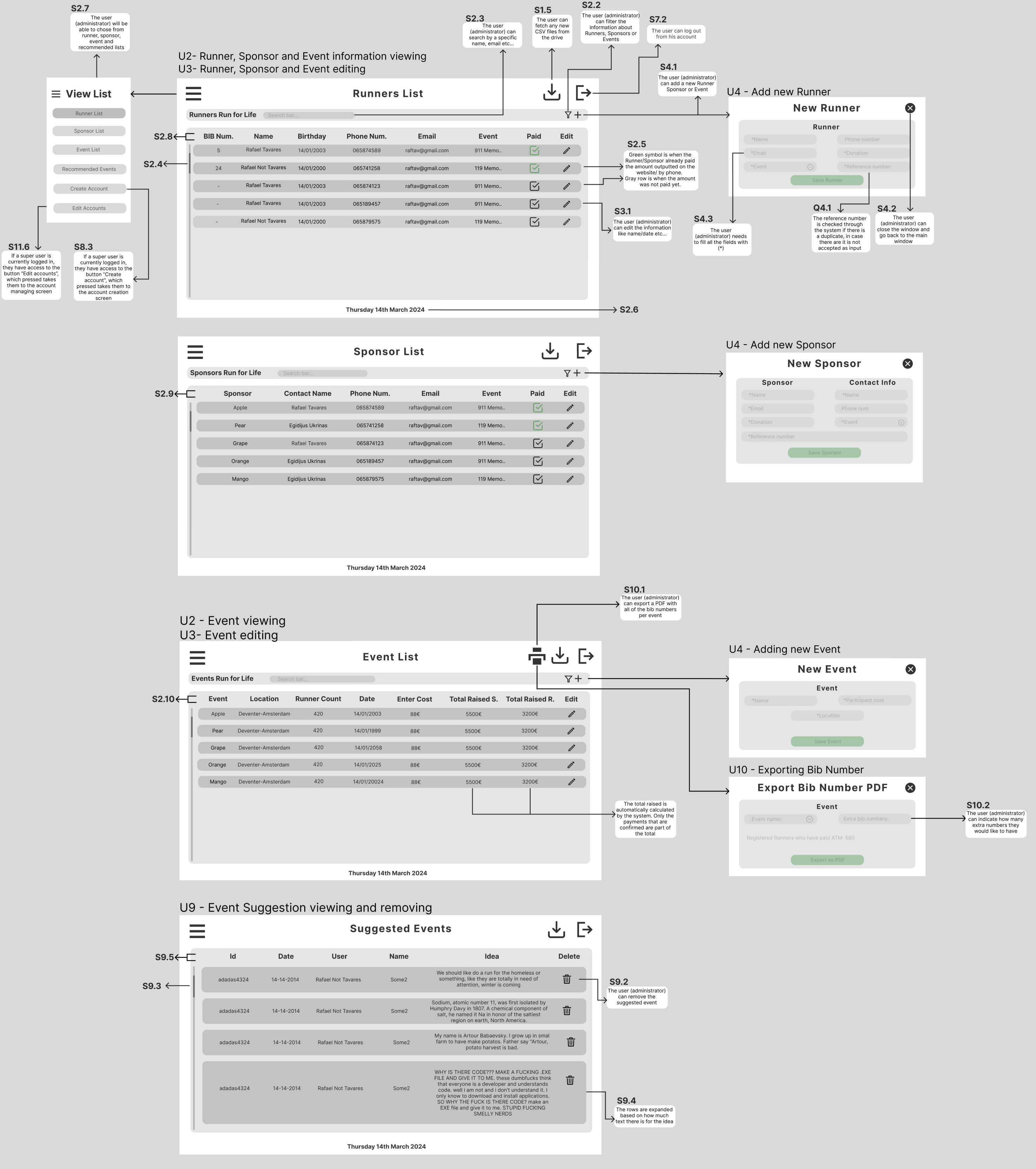
## Account viewing, editing and deletion Wireframe

A screenshot of a computer

Description automatically generated

Here the super administrator will be able to edit or delete an account. The account can be deleted or can be edited (username, email or expiration date).

## Administrator Wireframes



Wireframe 1 (Would be default when accountant logs in) and Wireframe 2 will appear when the user chose on the options menu. Here in these 2 wireframes, the user that will be using them will be the administrator. The administrator, as mentioned, will be mainly able to see all the data from runners and sponsors. The administrator can log off in the top-right corner.

The administrator can edit any runner or sponsor information with the pencil tool, that is located on the right of each line and more importantly, he can create a sponsor or a runner with the plus button.

Also, all the information can be filtered by using the filter button or can be searched (ex: by a specific name or event). Additionally, all the payments that were confirmed by the accountant will have a green check box for the administrator to see.   
A new account can be created by pressing the create account and after that a new window will open [Account Create](#_Account_Creation_Wireframe). Only the super administrator can create new accounts.

Wireframe 3 and Wireframe 4 will appear when the user chooses on the options menu.

Here in these 2 wireframes, the user that will be using them will be the administrator. The administrator, as mentioned, will be mainly able to see all the data about the events and the recommended events from the users. The administrator can log off in the top-right corner.

The administrator can edit any event information with the pencil tool, that is located on the right of each line and more importantly, he can create a new event with the plus button. All the recommendations from the users can be deleted when the user needs to.

When seeing an event, the administrator can click the print button to print the bib numbers to an event.

Also, all the information can be filtered by using the filter button or can be searched (ex: by a specific name or event) and at any time if needed the accountant can fetch manually the most recent data to the system.

## Accountant Wireframes

A screenshot of a computer

Description automatically generated

Wireframe 1 (Would be default when accountant logs in) and Wireframe 2 will appear when the user chose on the options menu. Here in these 2 wireframes, the user that will be using them will be the accountant.

The accountant, as mentioned, will be able to see all the data from the payments. The accountant can log off in the top-right corner.

The accountant can edit any payment, and more importantly, he can validate a payment and confirm that a sponsor or a runner paid by changing the amount paid. He can do all of that with the pencil tool, and with that, we will be able to edit any information.

Also, all the payments can be filtered by using the filter bar or can be searched (ex: by a specific name or event). Additionally, all the payments that were confirmed by the accountant will turn green. The accountants can also fetch manually the most recent data to the system.

# 6. Conclusion

In conclusion, the Functional Design Document (FDD) serves as a detailed blueprint outlining the functional aspects of the system developed for Run for Life. The primary goal of this project is to create a robust software solution that efficiently handles registrations, donations, and event coordination for charitable races organized by Run for Life.

The system automates the processing of CSV files either from the drive or based on user preference. Upon logging in, users are directed to their workspace, which can be customized through the options menu. Accountants possess the capability to view, modify, and validate payments from sponsors or runners. Additionally, they can utilize filtering and search functions to locate specific information as required.

Administrators, including the super administrator, have access to and can modify information related to runners, sponsors, and events. They also have the privilege to create new events, register runners, and manage recommendations from website users. Workspace customization options are available for administrators via the options menu. Aside from the normal administrator, a super administrator holds the authority to create new accounts. They possess all the capabilities of regular administrators, allowing them to efficiently manage and oversee the system's operations.

The FDD provides clarity on how the system will meet requirements and fulfill stakeholders' needs by detailing specific features, requirements, and workflows essential for achieving project objectives effectively. By making things simpler for both accountants and administrators, the system makes sure that files are handled smoothly without any information getting lost for the runners, and donations are processed correctly for sponsors and runners. Additionally, it helps Run for Life manage their events more efficiently, which ultimately helps them in their mission of supporting charitable causes.

# Appendix A: The Interview

The client was presented with a series of queries, referred to as "questions" and their corresponding responses, termed "client responses". In certain instances, these questions encompassed sub-questions. And the transcription of the interview is as follows:

**Question:** How are we defining system users? Who will be using the system?

**Client Response:** The primary users of our system are exclusively our employees. Runners, event organizers, and those proposing events primarily interact with our website or contact our office through phone communication to initiate processes. The current users accessing and managing tasks within the system include my administrative staff, our accountant, and our recruiters.

They are the individuals responsible for handling CSV files and, currently, the intricate pay sheets that manage various aspects of our operations. The system is designed to address and resolve the challenges faced by these internal users, rather than being intended for public or runner-facing functionalities.

**Question:** The current problem is the processing of manual files. Apart from that, what other problems should the system solve? What is the most important thing that the system should do?

**Client Response:** Okay, so besides addressing the issue of manual file processing, what other problems should the system solve? What is the most crucial aspect that needs improvement? Currently, the primary challenge with manual capturing is the uncertainty about whether sponsors have paid or not.

Tracking this is difficult, even with the best efforts of our accountant. The main problem we want the system to address is the ability to determine whether sponsors have indeed paid the indicated amount for a particular event.

Another concern is that runners are assigned bib numbers only after payment. While some runners pay online, others pay later, and the bib numbers should be allocated accordingly. Automating this process would be beneficial.

Currently, we manage this by creating different spreadsheets for each event, listing all the sponsors. It becomes challenging for our accountant to match the incoming funds to the respective sponsors when reconciling bank statements.

**Sub-question:** So, is this the most important thing?

I would emphasize that this is crucial for the role of my accountant. The accountant has a significant responsibility in this regard. While I don't want the system to have direct access to my bank account, I do want to facilitate my accountant's job by making it easier for them to match the incoming funds to specific sponsors for a particular event. This, I believe, is the most important improvement needed.

**Question:** Are the sponsors and the runners stored in the same files that the website provides?

**Client Response:** They are stored in separate files, but both are on a shared drive. The website provides CSV files, and we have a shared drive set up internally in the office to manage these files. This shared drive consists of three separate files. The first file contains information about sponsors, which includes individuals or organizations indicating their sponsorship for a specific event.

The second file consists of CSV data for individuals who have registered to run in the event. These are distinct from the sponsors. The third file is for individuals proposing new events or charities for us to organize runs. Users visiting the website can suggest events, specifying the location and the associated charity.

We review these proposals and decide whether to support the event or not. These three CSV files are placed in the shared drive.

Currently, we input all our data into a spreadsheet on a different shared drive. This arrangement is convenient for us because it is responsive and has built-in backup features. We plan to continue with this structure. After capturing the CSV files into the spreadsheet, we typically delete the CSV files to keep the system organized.

**Question:** Where will the processed data be output?

**Client Response:** We currently have backups set up on our mapped drive. Specifically, we use one mapped drive to retrieve the CSV from the website. Subsequently, we utilize a second mapped drive to store our spreadsheets. I suggest utilizing the same folder location for the output, as it is automatically backed up on a server with security measures and backup policies in place.

Additionally, for operational purposes, we print out lists on the day of the event. These include details of registered runners, those who have paid, and those who haven't. Runners who have paid already receive their bib numbers, which are printed along with some extras. If individuals register on the event day, they will be assigned a different bib number.

Regarding data output, I prefer the data to be stored on a drive. However, I also want the ability to print these lists for on-the-day assistance, ensuring accurate distribution of bib numbers to the correct runners.

**Question:** I would like to know, for those sponsors, are they usually entities or companies, or are they just individuals?

**Client Response:** Both. Sometimes, no matter if it's an organization or a company, we still expect them to provide a contact person. When an organization registers as a sponsor, having just an organizational name is tricky because then, when you reach out, they might be unaware of the specifics. You can't merely contact a company; you need to know who within the organization is handling the sponsorship.

Therefore, even if it's an organization, they need to designate a contact person. Sponsors can be individuals or organizations, but in the case of organizations, there must be an individual responsible for the sponsorship within the organization. It's crucial to have a point of contact within the company for effective communication.

**Sub-question:** And what do sponsors get in return?

That's a good question. What happens is, we consciously print out a list of all sponsors, but we record the list only on the day of the event. We highlight the biggest sponsors, varying based on the situation. We would like to know the range of individuals or entities sponsored. During the event, we express our gratitude over the microphone, award prizes, and sometimes print banners.

Keeping track of sponsorships is not for my administrative staff to know for this event. We identify the sponsors, possibly ordering them from the best to the lowest, making it easier for us to acknowledge them based on their contributions. The level of acknowledgment depends on the event and the audience's engagement. It's always nice to express gratitude, and the system helps us filter sponsors based on their financial contributions.

**Sub-question:** So, the system can also show, and filter sponsors based on their financial contribution, right?

Yeah, I would say probably according to the event savings. Now, for this event, we identify who the sponsors are and potentially order them based on financial contributions.

Another noteworthy aspect is sometimes sponsors do not fulfill their pledged amount. They might say they will sponsor a certain amount but have not paid it yet. They claim their accountant will reconcile it with the bank statement.

We may consider following up with sponsors who have outstanding payments from previous events. It is a nice-to-have rather than a necessity now addressing it can enhance our sponsorship management. If someone has outstanding money from a previous event, thanking them again at another event when they still owe us money may not be ideal. That is something we consider a nice-to-have feature.

**Question:** What kind of information is stored in the files other than the runners and the sponsors?

**Client Response:** Well, for the sponsors, it is not only the organization, as discussed, but also the contact person's details. We typically prefer an email and a phone number, but having one of them is acceptable. For the runners, it is similar, with a requirement for a name and contact details.

Currently, we are facing challenges in matching the money paid to the respective payers. So, having a unique reference number for a sponsorship offer would be beneficial. This unique identifier would be helpful when sponsors make payments, allowing us to match more efficiently. I am also considering if our website generates a unique number when a sponsorship offer is made. Moving on to those organizing runs or events, we store the location, the charity associated, and the contact details of the organizers.

**Sub-question:** About the runners, is it stored if they have already paid online or are they going to pay in person?

That information is stored separately as a distinct field or entry. We track whether runners have paid online or not. If they have not paid online, they can make payments on the event day, with a reference or another form of payment.

We allocate numbers based on payment status, but it is crucial to know the total number of registered runners. We print numbers for both those who have paid and those registered to pay on the day, including some extra for potential walk-ins. For instance, if there are 50 people registered and 40 have paid, we print 15 additional numbers. While only 15 are assigned, we print 55 numbers for those registered, including an extra five for potential on-the-day payments.

**Question:** Do runners get their BIB number even when they did not pay?

**Client Response:** The BIB number is only assigned when a runner has paid. So, for example 50 runners registered and only 40 paid. I print out 50 BIB numbers, but only assign 40 of them to the runners who paid, then I assign to who ever pays to participate regardless of if they have registered before. Roughly 10% extra BIB numbers would be great to have been made.

(Not the main point of the system)

**Question:** Do you need a manual editing capabilities in the system?

**Client Response:** Yes, both runners and sponsors need to be able to be edited, by out employees. For example, a runner might enter incorrect information or not be participating anymore, so we must be able to remove that runner from the system. Or a sponsor changing their contact person.

**Question:** Is there any existing system or tools besides the shared drive that the new system will need to integrate with?

Client Response: Regarding that, you need to get the CVSs from the shared folder, and I suggest storing the all the data in the other shared folder, but I don’t want our system to access the bank information, because I don’t want that level of security or that responsibility. My accountant will manually do it all. The accountant should be able to look up who that money is from or what it is for and then be able to tick it.

**Question:** What kind of protection of the data do you want? Are logins or 2-step authentication required?

**Client Response:** My employees use it; I would like them to only use it on-site. We do have a VPN access to our office, to access it from home. I don’t want it to be accessible outside of the office and I would find 2-step verification annoying and pointless. Most security is in the VPN we have. We do want login capabilities, because different employees need different access to information and have different responsibilities. So, the accountant should only be able to mark companies or runners that they have paid. It’s nice to have when recruiter finds a sponsor, that it would store that information that the recruiter found that sponsor in the system.

**Question:** What is the timeline and budget for the project?

**Client Response:** Irrelevant information for the scope of this course.

**Question:** How many users can we expect to use the system?

**Client Response:** We might expand, but currently we have an accountant, one recruiter, part-time recruiter and one who is an administrator. Currently we are a small company 7-8 employees, but we are looking into expanding. So, there are 3 roles accountant, recruiter, administrator. Most important is the accountant role, the other two don’t have clear responsibilities in their roles. Make the accountants job as easy as possible. We might get temporary volunteers, so we can make them temporary accounts. Biggest case 25 workers in total.

**Question:** Will runner information be recorded, even when they haven’t paid yet?

**Client Response:** The information comes from the CSV file, so that information needs to be kept, because they can pay at the event. After the run, if they haven’t paid, we can remove them, but we want to keep the runners who have paid, maybe also add their run time or if they did not finish the race.

**Question:** Will the company allow refunds?

**Client Response:** We don’t often get that because it’s a charity, all the proceeds go to the charity. If somebody hasn’t ran after paying, it’s considered a donation. If somebody requests one, we are obliged to refund it, but my accountant will deal with that. The system should be able to edit that person who has run, that he has not paid anymore.

# Appendix B: Tests

## Traceability Matrix

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | T1.1 Close application | T1.2 Fetch new files | T1.3 Edit outstanding information | T1.4 Search for outstanding information | T1.5 Filter outstanding information | T1.6 Edit outstanding paid to negative | T1.7 Attempt to Edit Without Selecting a Row |
| U1 CSV file processing |  | x |  |  |  |  |  |
| S1.5 Button for fetching CSV files |  | x |  |  |  |  |  |
| U5 Displaying outstanding payments |  |  | x | x | x | x | x |
| S5.1 List all the outstanding payments |  |  | x | x |  | x |  |
| S5.2 Filter outstanding payments |  |  |  |  | x |  |  |
| S5.3 Search specific outstanding payments |  |  |  | x |  |  |  |
| S5.5 Row changes to green indicating payment made |  |  | x |  |  | x |  |
| U6 Editing outstanding payments |  |  | x |  |  | x |  |
| S6.1 Changing outstanding payments |  |  | x |  |  | x |  |
| S7.2 Logging out | x |  |  |  |  |  |  |
| Q4.2 Accurate entry of payment information |  |  |  |  |  | x |  |

## Test cases

|  |  |  |
| --- | --- | --- |
| TC 1.1 | | |
| Name | Close application | |
| Description | The user will press the logout button and the app will close. | |
| Requirements | S7.2 | |
| Preconditions | The tester needs to be logged in to their account. | |
| Step | Action | Expected results |
| 1 | User clicks on the logout button | The application initiates the logout process. |
| 2 | User waits for the application to close | The application closes automatically without any additional confirmation. |
| Postconditions | The application is closed, and the user is logged out of their account. | |

|  |  |  |
| --- | --- | --- |
| TC 1.2 | | |
| Name | Fetch new files | |
| Description | The user fetches new files that may have been uploaded by the website | |
| Requirements | U1, S1.5 | |
| Preconditions | The user is logged into the application | |
| Step | Action | Expected results |
| 1 | User clicks on the button labeled “Fetch” | New rows are added in the table below |
| Postconditions | The table is updated with new information | |

|  |  |  |
| --- | --- | --- |
| TC 1.3 | | |
| Name | Edit outstanding information | |
| Description | The user will select a row to and then edits the information in a popup window | |
| Requirements | U5, S5.1, S5.5, U6, S6.1 | |
| Preconditions | The user needs to be logged in their account and data must exist to be edit | |
| Step | Action | Expected results |
| 1 | User clicks on the second row to select it. | The application highlights the selected row indicating it's selected. |
| 2 | User clicks on the edit button. | The application opens a popup window displaying the existing information from the selected row. |
| 3 | User changes payment to "250” in the paid text field | The amount is changed to 250 in the paid text field. |
| 4 | User clicks on the ok button. | The information is saved and updated in the table, and the color of the row is changed to green. |
| Postconditions | The payment amount for the selected row is updated to 250 in the system and color of the row turns green. | |

|  |  |  |
| --- | --- | --- |
| TC 1.4 | | |
| Name | Search for outstanding information | |
| Description | Regular scenario for searching of outstanding information by the user | |
| Requirements | U5, S5.1, S5.3 | |
| Preconditions | The user is logged into the application | |
| Step | Action | Expected results |
| 1 | User left clicks on the search bar that has “search …” inside it | The application focuses on the search bar and lets user enter text |
| 2 | User types in “567” into the search bar | The application searches for any name, email or reference number that has “567” in it and displays the rows that have that contain the search input |
| Postconditions | The table now displays the rows that have “567” in its reference number | |

|  |  |  |
| --- | --- | --- |
| TC 1.5 | | |
| Name | Filter outstanding information | |
| Description | Regular scenario for filtering of outstanding information by the user | |
| Requirements | U5, S5.2 | |
| Preconditions | The user is logged into the application | |
| Step | Action | Expected results |
| 1 | The user clicks on the filter button: | 2 check boxes and a choice box appear on the left side of the filter button |
| 2 | The user unmarks the check box on the left of the label “Not Paid” | The application only displays rows in the table that have paid equal or more than the outstanding amount |
| 3 | The user marks the check box on the left of the label “Not Paid” | The application displays all rows that are available |
| 4 | The user clicks on the choice box that has “ALL” inside it | The application will display all available events that can be filtered |
| 5 | The user clicks on “RunForYaLife” | The application only lists rows that are part of the “RunForYaLife” event |
| Postconditions | The table now displays the rows that are part of the “RunForYaLife” event | |

|  |  |  |
| --- | --- | --- |
| TC 1.6 | | |
| Name | Edit outstanding paid to negative | |
| Description | The user will select a row and then try to insert invalid data for the paid text field. | |
| Requirements | U5, S5.1, S5.5, U6, S6.1, Q4.2 | |
| Preconditions | The user will select a row to and then edits the information in a popup window. | |
| Step | Action | Expected results |
| 1 | User clicks on the second row to select it. | The application highlights the selected row indicating it's selected. |
| 2 | User clicks on the edit button. | The application opens a popup window displaying the existing information from the selected row. |
| 3 | User changes payment to "-50" in the paid text field. | The amount is changed to "-50” in the paid text field. |
| 4 | User clicks on the ok button. | The application displays a message indicating that the user cannot change the payment to a negative number. |
| 5 | User clicks on the cancel button | The application closes the popup window |
| Postconditions | The payment information remains unchanged in the system. | |

|  |  |  |
| --- | --- | --- |
| TC 1.7 | | |
| Name | Attempt to Edit Without Selecting a Row | |
| Description | The user attempts to edit information without selecting a row. | |
| Requirements | U5 | |
| Preconditions | The user needs to be logged in their account and data must exist to be edited. | |
| Step | Action | Expected results |
| 1 | User clicks the edit button without selecting a row. | The application displays a popup window with a message stating, "You have to select an item". |
| 2 | User clicks the OK button in the pop-up window | The pop-up window closes and allows further interaction with the application |
| Postconditions | The application remains in its previous state | |