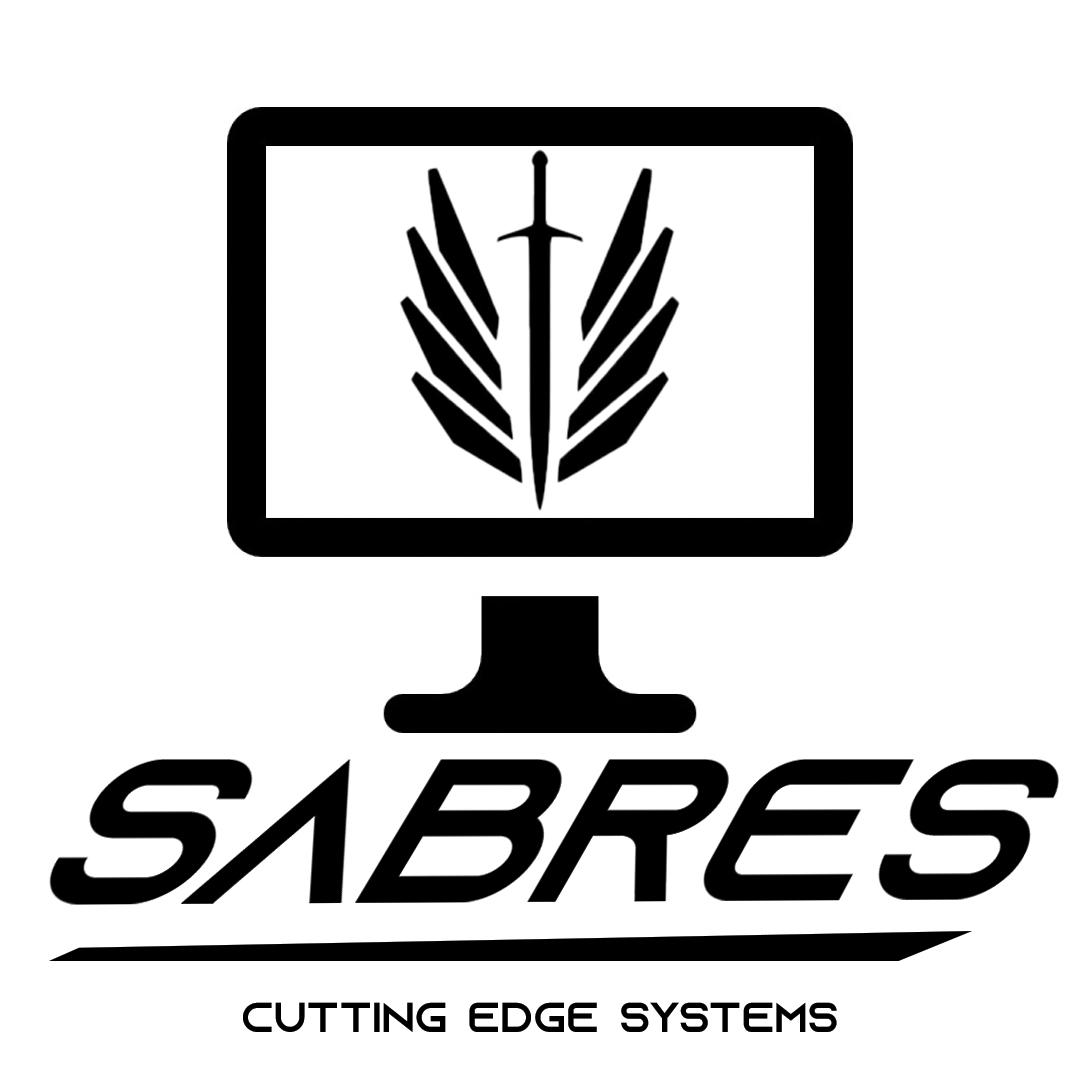
**miLESTONE 2**



sabres

1825462 - Mologadi Chuene

1612235 - Timothy Mooney

1916999 - Shuiab Booley

1834316 - Rahube Seabi

1811176 - Nastassja Nydoo

## **1. Executive SUmmary**

After extensive analysis and design, we have determined the following for the Moneta System.

Donation, donor, application, applicant and admin are the classes applicable in our system which are represented in the form of an ERD (Entity Relationship Diagram). The ERD represents these classes as tables. It shows the relationships between these tables and the respective attributes (data stored) within each table. This diagram is a key tool for the database designer as it lays the foundation of the database structure and thus the entire system.

Multiple use cases were derived from the class diagram from the previous milestone, these same use cases have undergone revision and have changed accordingly due to more knowledge and experience being obtained by all team members. We believe that is much more functional as there is better flow of data between the user and the system. The use cases have been summarized using a use case set, which outlines the main functionalities of the system and the interactions between them.

The “Create Application”, “Update Applicant” and “Create Donation” use cases have been expanded upon using a use case description. A use case description goes in depth with regards to the actual processing of the data, with this we have also included a sequence diagram for each previously mentioned use case. It visually represents all the processing done by each respective use case thus also outlining the flow of logical activities. The sequence diagram can be passed to the programmer directly to translate it into code during implementation without any need for further clarification due to its extensive detailing.

We have included story boards for the “Update Applicant” and “Create Donation” use cases. Story boards show what the possible user interface would look like. It physically tells a story of the experience the user would go through. This will be given to the UX designer to construct.

Three reporting tables and prototypes have been included for “Applicants who have been in the “Red” status for sixty (60) days or more”, “Donors that have dropped to a certain percentage of their available donation balance” and “Donors whose funding balance have not changed for 2 months”. We strongly believe that with these reports the company would benefit greatly from. It will allow for quick and easy decision making with regarding applicants who’ve not been approved for funding for at least two months thus decreasing the pool for applications which effectively leads to the fact that not many students are able to have access to the benefits of the system which is the opposite of what is intended. For donors whom are in danger of depleting their available funds which results in the system having a lack of money available to be donated thus decreasing the systems sustainability. For the recipients to view which donors are the least active regarding donations i.e. donors who have funds but whom are not donating their money or whom have donated very little.

A holistic understanding is shown by the use of all models. All assumptions are clearly stated of which we at a high detailed level, express how we believe the system works. We believe that these designs will meet the requirements of the proposed system once implemented.

## **2. Revised Analysis**

*2.1 Revised Use Case Set*

|  |  |  |
| --- | --- | --- |
| Entity | Use Case | Related Use Case |
| Applicant | Create applicant |  |
|  | View applicant |  |
|  | Update applicant | View applicant (includes) |
|  | Delete applicant |  |
| Admin | Create admin |  |
|  | View admin |  |
|  | Update admin | View admin (includes) |
|  | Delete admin |  |
| Donor | Create donor |  |
|  | View donor |  |
|  | Update donor | View donor (includes) |
|  | Delete donor |  |
| Application | Create application | View applicant (includes) |
|  | View application |  |
|  | Update application | View application (includes) |
|  | Delete application |  |
| Donation | Create donation | View application (includes); View donor (includes); Update application (includes); Update donor (includes); View applicant (includes) |
|  | View donation |  |
|  | Update donation | View donation (includes) |
|  | Delete donation |  |
| Report | Generate Report |  |

*2.2 Revised Update Applicant Use Case*

|  |  |  |
| --- | --- | --- |
| Use case name | Update Applicant | |
| Actors | Applicant- Primary actor; Administrator- Primary actor | |
| Pre-Conditions | Applicant must exist in the APPLICANT table. Data stored with attributes: applicant\_IDNo, applicant\_firstName, applicant\_lastName, applicant\_dob, applicant\_cell, applicant\_email, applicant\_address, applicant\_university, applicant\_degree, applicant\_studentNo and applicant\_status | |
| Post-Conditions | if user = “Administrator”, either applicant\_status has been recorded as “Green” in the APPLICANT table or the old data in the APPLICANT table has been updated to the new data per request by the applicant.  Changes made in Applicant data must be stored in the APPLICANT table under either of the following attributes: applicant\_IDNo, applicant\_firstName, applicant\_lastName, applicant\_dob, applicant\_cell, applicant\_email, applicant\_address, applicant\_university, applicant\_degree, applicant\_studentNo or applicant\_status  Applicant data must be displayed  Confirmation message will be displayed | |
| Flow of Activities | Actor  1. Request to update applicant  2. Enter applicant\_IDNo  3. Enter new applicant data  4. Confirms/cancels applicant update | System  if actor is admin   * 1. Prompt for applicant\_IDNo   2.1 Use applicant\_IDNo to invoke view applicant  2.2 Display all applicant data  2.3. Prompt to enter new data: applicant\_firstName, applicant\_lastName, applicant\_IDNo, applicant\_dob, applicant\_cell, applicant\_email, applicant\_address, applicant\_university, applicant\_degree, applicant\_studentNo, applicant\_status  If actor is applicant   * 1. Use applicant\_IDNo to invoke view applicant   2. Display all applicant data, excluding applicant\_status   3. Prompt to enter new data: applicant\_firstName, applicant\_lastName, applicant\_IDNo, applicant\_dob, applicant\_cell, applicant\_email, applicant\_address, applicant\_university, applicant\_degree, applicant\_studentNo   3.1. Prompt for confirmation of applicant update  if confirmation = “yes”  4.1 Record new data in the APPLICANT table using applicant\_firstName, applicant\_lastName, applicant\_IDNo, applicant\_dob, applicant\_cell, applicant\_email, applicant\_address, applicant\_university, applicant\_degree, applicant\_studentNo, applicant\_status  4.2 Display confirmation of applicant update |

*2.3 Revised Create Application Use Case*

|  |  |  |
| --- | --- | --- |
| Use case name | Create Application | |
| Actors | Admin (Primary)  Applicant (Primary) | |
| Pre-Conditions | Applicant must exist in APPLICANT table  applicant\_status must be “Green” | |
| Post-Conditions | application\_status would be “Unfunded”  Application would be created in APPLICATION table using application\_ID, applicant\_IDNo, admin\_ID, donation\_ID, application\_fundType, application\_requiredAmount, application\_date, application\_status, application\_fundedAmount  Confirmation of application created  Donation\_ID = null  Application fund amount = 0 | |
| Flow of Activities | Actor  1. Request to create application  2a. Enter applicant\_IDNo  3 Select application\_fundType  4 Enter Amount  5 Confirms/cancels | System  If actor is “Admin”:  1.1a Prompt for applicant\_IDNo  2.1 Invoke View applicant to view applicant\_status  if applicant\_status = “green”  2.2 Prompt for application\_fundType  3.1 Prompt for required amount  4.1 Prompt for confirmation of Application  if confirmation = “yes”  5.1 Create Application  application\_status = “unfunded”  application\_fundedAmount = entered amount  5.2 Display confirmation |

*2.4 Revised Create Donation Use Case*

|  |  |  |
| --- | --- | --- |
| Use case name | Create Donation | |
| Actors | Admin (primary) | |
| Pre-Conditions | Applicant must exist in the APPLICANT table; Application must exist in the APPLICATION table and Donor must exist in the DONOR table. Donor must have enough funds to donate. For an application to have a donation made against it, it must have a status of either “Unfunded’ or “Partially funded”. | |
| Post-Conditions | ● New Donation recorded in the DONATION table using: donation\_ID, application\_ID, donor\_ID, donation\_fundedAmount, donation\_date  ● Confirmation of donation creation | |
| Flow of Activities | Actor  1. Request to Create Donation  2. Enter applicant\_firstName, applicant\_studentNo  3. Select Application  4. Enter donor\_cell, donor\_firstName, donor\_orgType  5. Enter donation\_fundedAmount | System  1.1 Prompt for applicant\_firstName, applicant\_studentNo  2.1. Use applicant\_firstName and applicant\_studentNo to search and invoke View Applicant in order to find applicant\_IDNo  2.2. Use applicant\_IDNo to search and invoke View Application to display application\_status, application\_fundType and application\_ID where application\_status is not “Funded”  3.1. Prompt for donor\_cell and donor\_firstName and donor\_orgType  4.1. Use donor\_cell, donor\_firstName, donor\_orgType to search and invoke View Donor to find donor\_ID, donor\_cell, donor\_firstName, donor\_lastName, donor\_orgType, donor\_fundBalance, donor\_maxAmount  If donor\_fundBalance NOT = 0  4.2. Prompt to enter donation data: donation\_fundedAmount    5.1. Generate donation\_ID  5.2. Create new donation in the DONATION table using donation\_fundedAmount, donation\_ID, donor\_ID, application\_ID, donation\_date, admin\_ID  5.3. Use application\_ID to invoke Update Application: application\_fundedAmount  5.4. Use donor\_ID to invoke Update Donor: donor\_fundBalance  5.5. Display confirmation of Donation Create |

## **3. Database Design**

*3.1 Entity Relationship Diagram* A screenshot of a cell phone

Description automatically generated

*3.2 Data Dictionary Extract*

**Table: APPLICANT**

|  |  |
| --- | --- |
| Scope | Data about applicants ***who complete applications***, including their ID number, admin id number, first name, last name, date of birth, cell number, email, address, university, degree, student numbers and statuses. Attributes stored in the Applicant table |
| ***Attribute Listing:*** |  |
| applicant\_IDNo PK | Description: A unique applicant identification number given by South African government  Type: 13-digit integer  Format: #############  Note: This is the primary key, no non-numeric value |
| admin\_ID FK | Description: A unique admin identification number given by #FundMe  Type: 7-digit integer  Format: #######  Note: This is a foreign key (linked to the “Admin” table), no non-numeric values |
| applicant\_firstName | Description: The name given by the applicant  Type: 50-character string  Format: N/A  Note: Can contain letters and special characters |
| applicant\_lastName | Description: Applicant family name  Type: 50-character string  Format: N/A  Note: Can contain letters and special characters |
| applicant\_dob | Description: The date the applicant was born  Type: Datetime  Format: DD/MM/YYYY  Note: N/A |
| applicant\_cell | Description: A unique number to contact applicant  Type: 10-digit integer  Format: ### #### ###  Note: No non-numeric values |
| applicant\_email | Description: A unique email address to contact applicant  Type: 50-character string  Format: N/A  Note: Can contain numbers, letters and special characters |
| applicant\_address | Description: A multi value attribute indicating the residence of applicant  Type: 50-character string  Format: Building Number, Street Number, Street Name, Suburb, City, Postal Code  Note: Can contain numbers, letters and special characters |
| applicant\_university | Description: The university that the applicant is registered for  Type: 50-character string  Format: N/A  Note: Can contain numbers, letters and special characters |
| applicant\_degree | Description: The degree that the applicant is registered for  Type: 50-character string  Format: N/A  Note: Can contain numbers, letters and special characters |
| applicant\_studentNo | Description: A unique number given to the applicant by the given university  Type: 10-character string  Format: ##########  Note: No non-numeric values |
| applicant\_status | Description: Indicates the applicant’s funding suitability  Type: Boolean  Format: T/F  Note: N/A |

**Table: DONOR**

|  |  |
| --- | --- |
| Scope | Data about donors, including their donor id, admin id, donor first name, donor last name, donor org type, donor cell number, donor email and donor address. Attributes stored in Donor table |
| ***Attribute Listing:*** | |
| donor\_ID PK | Description: A unique donor identification number assigned by the Moneta system  Type: 6-digit integer  Format: ######  Note: This is a primary key, no non-numeric values |
| admin\_ID FK | Description: A unique admin identification number given by #FundMe  Type: 7-digit integer  Format: #######  Note: This is a foreign key (linked to the “Admin” table), no non-numeric values |
| donor\_firstName | Description: The name given by the Donor  Type: 50-character string  Format: N/A  Note: Can contain letters and special characters |
| donor\_lastName | Description: Donor family name  Type: 50-character string  Format: N/A  Note: Can contain letters and special characters |
| donor\_orgType | Description: Indicates whether Donor is an individual or an organisation  Type: 30-Character String  Format: ##########################  Note: Can contain letters and special characters |
| donor\_cell | Description: A unique number to contact the donor  Type: 10-digit integer  Format: ### #### ###  Note: No non-numeric values |
| donor\_email | Description: A unique email address to contact the donor  Type: 50-character string  Format: N/A  Note: Can contain numbers, letters and special characters |
| donor\_address | Description: A multi value attribute indicating the residence of donor  Type: 50-character string  Format: Building Number, Street Number, Street Name, Suburb, City, Postal Code  Note: Can contain numbers, letters and special characters |
| donor\_maxAmount | Description: The maximum amount a donor is willing to donate.  Type: Double  Format: Currency  Note: No non-numeric values |
| donor\_fundBalance | Description: The amount remaining of the donor\_maxAmount  Type: Double  Format: Currency  Note: No non-numeric values |

**Table: APPLICATION**

|  |  |
| --- | --- |
| Scope | Data about applications belonging to the applicant, including their application Id number, applicant Id number, admin number, donation id, application fund type, application required amount, application date, application date, application status and application funded amount. Attributes stored in Application table. |
| ***Attribute Listing:*** | |
| **application\_ID PK** | Description: A unique identification number assigned after registration by Moneta system (do we give the number)  Type: 10-digit integer  Format: ##########  Note: This is a primary key, no non-numeric values |
| **applicant\_IDNo FK** | Description: A unique applicant identification number given by South African government  Type: 13-digit integer  Format: #############  Note: This is a foreign key (linked to the “Applicant” table), no non-numeric values |
| **admin\_ID FK** | Description: A unique admin identification number given by #FundMe  Type: 7-digit integer  Format: #######  Note: This is a foreign key (linked to the “Admin” table), no non-numeric values |
| **donation\_ID FK** | Description: A unique donation identification number assigned by Moneta system (do we give the number)  Type: 8-digit integer  Format: ########  Note: This is a foreign key (linked to the “Admin” table), no non-numeric values |
| **application\_fundType** | Description: Indicates the type of funding the application needs  Type: 30-character string  Format: N/A  Note: The fund types such as tuition, accommodation, telecommunication costs, historical debt relief, study materials and living expense, would be listed and be required to be checked if applicable. |
| **application\_requiredAmount** | Description: Indicates the amount needed for each selected fund type  Type: Double  Format: Currency  Note: No non-numeric values |
| **application\_date** | Description: The Date of which the application was made  Type: Datetime  Format: DD/MM/YYYY  Note: N/A |
| **application\_status** | Description: Indicates the application’s funding status  Type: String  Format: N/A  Note: Applications will be set to either “Unfunded”, “Partially funded” or “Funded”. |
| **application\_fundedAmount** | Description: The amount that has been funded through the donation/s  Type: Double  Format: Currency  Note: No non-numeric values |

**Table: DONATION**

|  |  |
| --- | --- |
| Scope | Data about donations belonging to an application, including their donation id, application id, donor id, donation funded amount and donation date. Attributes stored in the Donation Table |
| ***Attribute Listing:*** | |
| donation\_ID PK | Description: A unique donation identification number.  Type: 10-digit integer  Format: ##########  Note: This is a primary key, no non-numeric values |
| application\_ID FK | Description: A unique identification number assigned after registration  Type: 10-digit integer  Format: ##########  Note: This is a foreign key (linked to the “Application” table), no non-numeric values |
| donor\_ID FK | Description: A unique donor identification number  Type: 10-digit integer  Format: ########  Note: This is a foreign key (linked to the “Donor” table), no non-numeric values |
| admin\_ID FK | Description: A unique admin identification number given by #FundMe  Type: 7-digit integer  Format: #######  Note: This is a foreign key (linked to the “Admin” table), no non-numeric values |
| donation\_fundedAmount | Description: The amount donated by the donor  Type: Double  Format: Currency  Note: No non-numeric values |
| donation\_date | Description: The date the donation was made by the Donor  Type: Datetime  Format: DD/MM/YYYY  Note: N/A |

## **4. Systems Design**

*4.1 Sequence Diagram: Update Applicant*

A screenshot of a cell phone

Description automatically generated

*4.2 Sequence Diagram: Create Application*

A screenshot of a cell phone

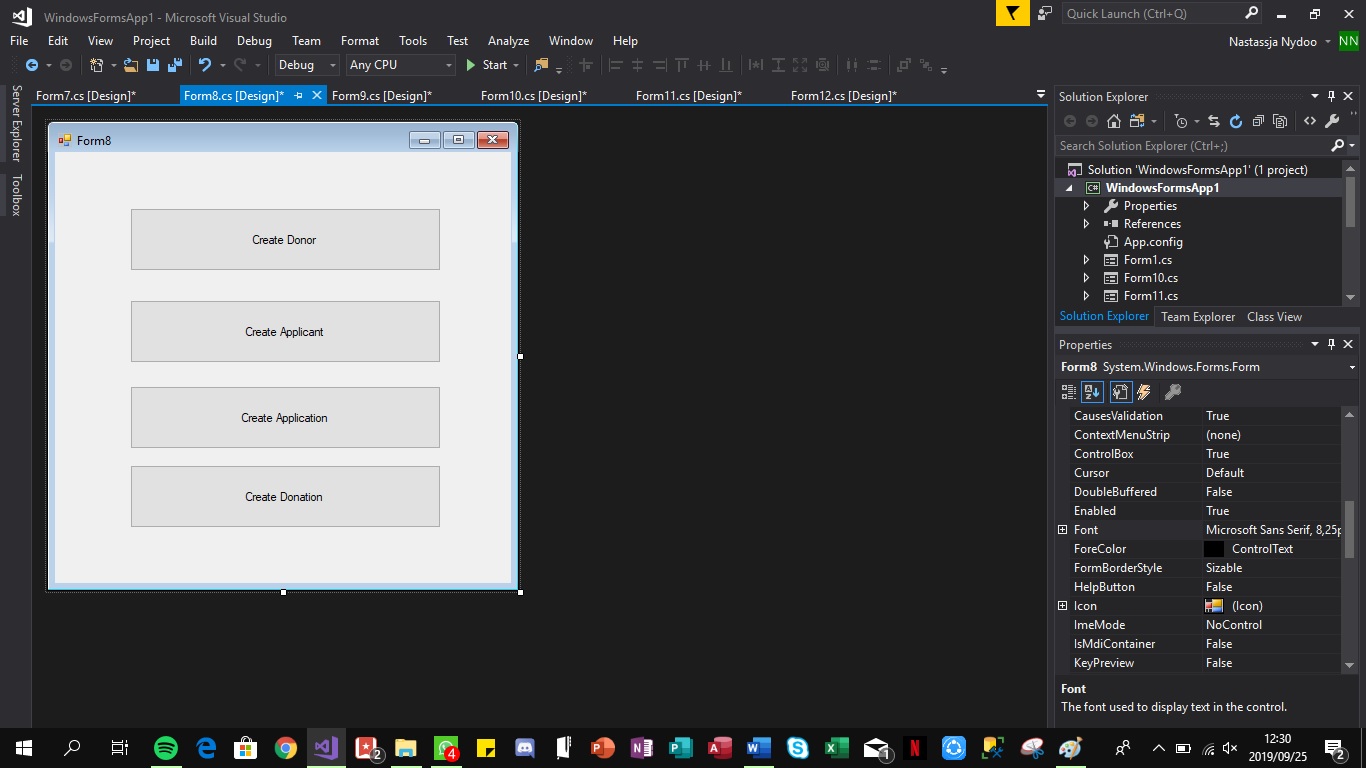
Description automatically generated

*4.3 Sequence Diagram: Create Donation*

A screenshot of a cell phone

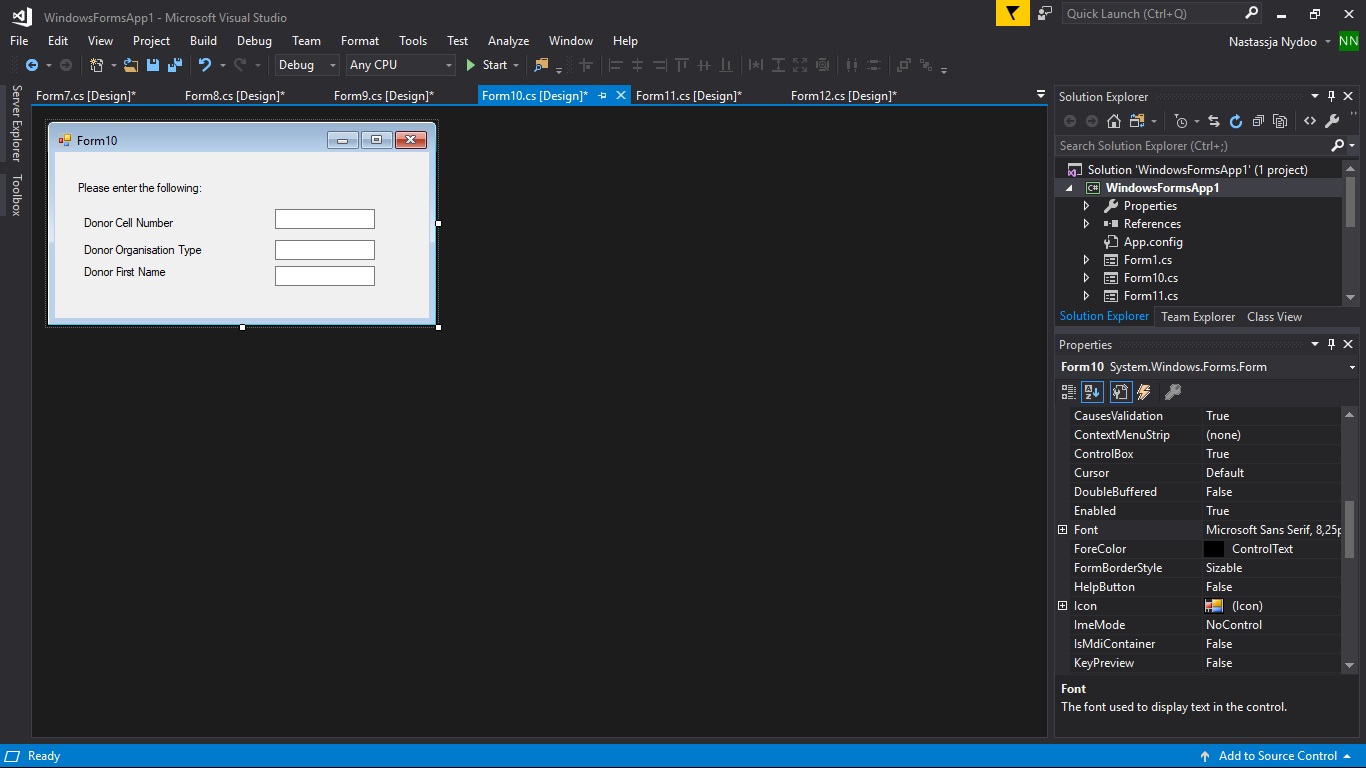
Description automatically generated

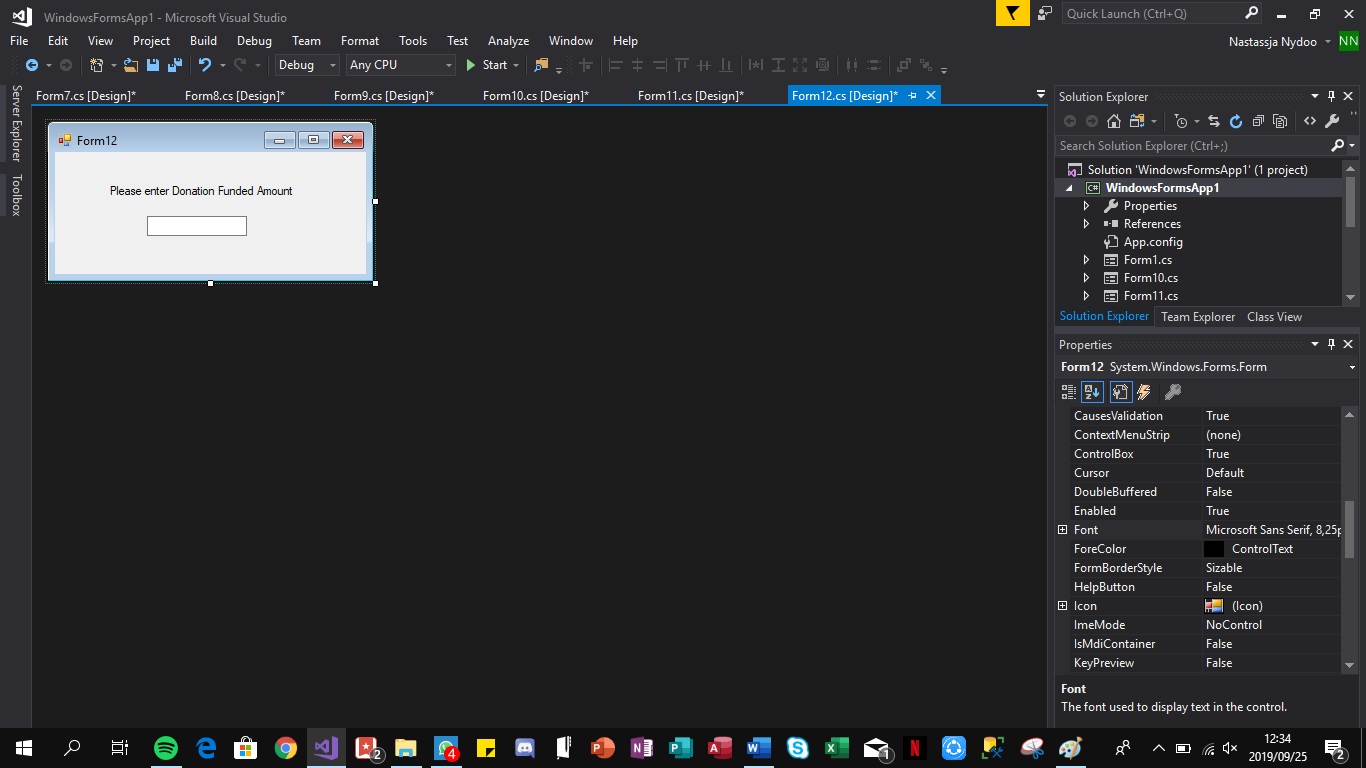
*4.4 Storyboard: Create Donation*

 A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated 

 A screenshot of a computer

Description automatically generated

*4.5 Storyboard: Create Application*

If the user is Admin

A screenshot of a computer

Description automatically generated A screenshot of a computer

Description automatically generated

Or if Applicant

A screenshot of a computer

Description automatically generated A screenshot of a cell phone

Description automatically generated A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

## **5. Report Design**

*5.1 Report on applicants who have remained in “Red” status for sixty (60) days or more*

|  |  |
| --- | --- |
| Report name: | Applicants who have remained in the “Red” status for sixty (60) days or more. |
| Report type: | Electronic |
| Report format: | Pie Chart |
| Report recipients: | Shareholders/ stakeholders |
| Report frequency: | Bimonthly |
| Report justification/ rationale: | This provides insight to the recipients on how many applicants are operating with their system. It enables constrictive questioning in order to improve the company’s chances of success. Based on the results the reports provide, certain questions such as, should there perhaps be less restrictive rules for an applicant to qualify to create applications; are the required documentation too difficult to obtain if so why, how could we make it easier for people to obtain such documents and so forth. |
| Decision(s) made as a result of the report: | If there is a significant number of applicants being in the “Red” status for sixty days or more, it is thereby prudent to find out its leading factor which will then further be analyzed to see if the company can amend the gap.  If there is no significant number of applicants who are in the “Red” status for sixty days or more, the company could investigate the leading factor behind its success as part of evaluation and for any future references especially with regards to understanding their clients better. |

*5.2 Report on donors that have dropped to a certain percentage of their available donation balance*

|  |  |
| --- | --- |
| Report name: | Donors that have dropped to a certain percentage of their available donation balance |
| Report type: | Electronic |
| Report format: | Bar Graph |
| Report recipients: | Shareholders/ stakeholders |
| Report frequency: | Semiannually |
| Report justification/ rationale: | By being able to view who is below 20% of their available donation balance, it enables the recipients to roughly gather how much of available funds are in the system and if it is sustainable. By being aware of the estimated available funds it would lead to ensuring that majority of their donors do not remain in or below the 2% as this could have adverse effects in the future of the company’s successfulness. |
| Decision(s) made as a result of the report: | Addressing the respective donor whose available donation balance dropped to 2% by informing the donor as to update their donation funding amount or it may result in the remaining balance being forfeited to the system after another consecutive month of the donor remaining or dropping even further than the 20%. By ensuring that majority of the system’s donors do not remain in or below 20%, it ensures more continuous sustainability in the successfulness of applications being funded. |

*5.3 Report on donors whose funding balance have not changed for 2 months.*

|  |  |
| --- | --- |
| Report name: | Report on donors whose funding balance have not changed for 2 months. |
| Report type: | Electronic |
| Report format: | Column Chart |
| Report recipients: | Shareholders/ stakeholders |
| Report frequency: | Bimonthly |
| Report justification/ rationale: | By being able to view each donors’ available balance from the beginning of the first month to the end of the second month, it enables the recipients to clearly see who their most frequent donors are, i.e. having the ability to easily view and comprehend which donors are donating money to applications. It will be assumed that the active donors will have an expected drop in their balance. More importantly and something more noteworthy, is that it enables the recipients to also see who their least active donors are, i.e. the donors who are in the system but whom have not or whom have barely donated any money to applications thus their donation balance will most likely not have dropped or at the very least have a slight difference in their donation balance. By making a decision as a result of this report to encourage more donors to donate their money, it will adversely increase the chances of applications being funded which then increases the company’s success rate of funding students in financial need. This is advantageous for the company as it gives them more credibility to more potential donors. |
| Decision(s) made as a result of the report: | By knowing which donors are not active in the system, the company could find the reasoning behind it in order to take action to change that. Such action could be by creating an incentive for donors to encourage them to donate their money this could be done using a reward system if required. |