NAVEEN JINDAL SCHOOL OF MANAGEMENT – UT DALLAS

GENEROCITY

System Analysis and Project Management

Prepared By,

Rashmi Doddasomanahally Rajanna **(rxd210025)**

Contents

EXECUTIVE SUMMARY

The modern world has become smaller and people are becoming more responsible. This trend is manifested through the appearance of numerous non-profit organizations that help underprivileged people or simply people in need. It is noteworthy that 67% of reporting non-profit organizations accept online donations as per 2021 statistics data in the nonprofitsource.com and this increases the need for a comprehensive and reliable data management system.

“GeneroCity” is a mobile-based application that provides accessibility to the network of charities where users have the privilege to select the charity of interest to make a donation. This system has four modules namely, Donor, Charities, Requestor, and Volunteer. Users are allowed to perform several different actions on their registered profile such as access and offer donations, access donation routes, volunteer, request items, and perform administrative tasks. Requestors can register and raise requests for items in need under a preferred charity in the network, GeneroCity sends a notification of the request that is raised to various registered donors. The donors have an option to make monetary donations and/or goods such as kids’ clothing, accessories, furniture, and toys. The goals include establishing an online payment system, establishing a login and registration page, scheduling appointments, allowing users to cancel appointments, and allowing them to access financing and other donations.

The volunteer can sign-up for donation pickup/delivery by selecting the schedule available in the GeneroCity interface according to their convenience and get notified about route and vehicle assignment as a pickup confirmation. GeneroCity manages appointment reminders and activity notifications to update its users with the latest events including notifying donors of items requested, and notifying requestors of donated items available if the users have enabled push notification.

The main concern of this project is to improve the efficiency and effectiveness of the whole system by creating a single point of interaction among the donor, requestor, volunteer, and charity.

Table of Contents

PROBLEM STATEMENT

PROBLEMS:

1. Charity organizations currently have owned websites where donors can opt-in for email notifications of clothing, toy, and home pickup donations.
2. These e-newsletters are the only method of communication between charities and potential donors.
    a. Potential donors often miss the newsletter because of email inbox clutter
    b. There is no reminder system in place for scheduled donation pick-up
3. Charities may not have the capital to spend on updated websites for schedule systems
4. Unreached market for potential donors due to inconvenient process or they are unaware of community charities
5. Charity organizations lack the manpower to support more frequent deliveries

OBJECTIVES:

1. Design a system that will create a network for the community - charities, donors, requesters, and volunteers to connect. Each of these users will have separate accounts within the system.
    a. Charity: front end will have information about the network of charities; the back end will allow charities to manage schedules, outgoing requests, and incoming donations
    b. Volunteer: ad hoc user, can sign up for pick-ups/deliveries with charities, and manage documentation for donations
    c. Donors: have inventory options for items they want to give to a charity, can manage scheduled pick-ups/deliveries, can manage documentation for donations, send physical and monetary donations to the charity of their choice, and be able to manage tax documentation

   d. Requestor: view the available donations of the charities, can request items, create a profile of items in need, and receive notifications of items in need once posted in the database

2. New system would connect charities, volunteers, requestors, and donors based on geographical location, and donors will be able to schedule pick up or drop off based on the charity's schedule.
3. System will provide push notifications of upcoming deliveries and their route ETA.
4. System will document/send invoicing once a transaction is completed for donors and volunteers (can be accessed within the profile)
5. The system will have a marketplace feature for verified people in need (requestors) looking for specific types of donations (ex: girl/boy clothing, shoes, etc.) and the donor can fulfill those requests.

SCOPE:

1. Estimated cost is approximately $75,000 for the entire system.
   a. Website, app, and branding (reliability 3 9's) 99.9% reliable, 99.999999999%, performance, usability UI Design (FURPS+)
   b. Employees
   c. Insurance
2. The development and operations require resources for developing the app, website, and administration.
3. Time needed for completion is estimated to be 8 months for Minimum Viable Product (MVP).
4. The system requires database servers (GoogleCloud, Azure, or AWS)
   a. Choice of server and tier of company will dictate the initial and maintenance/fixed costs

BUSINESS PROCESS MODEL

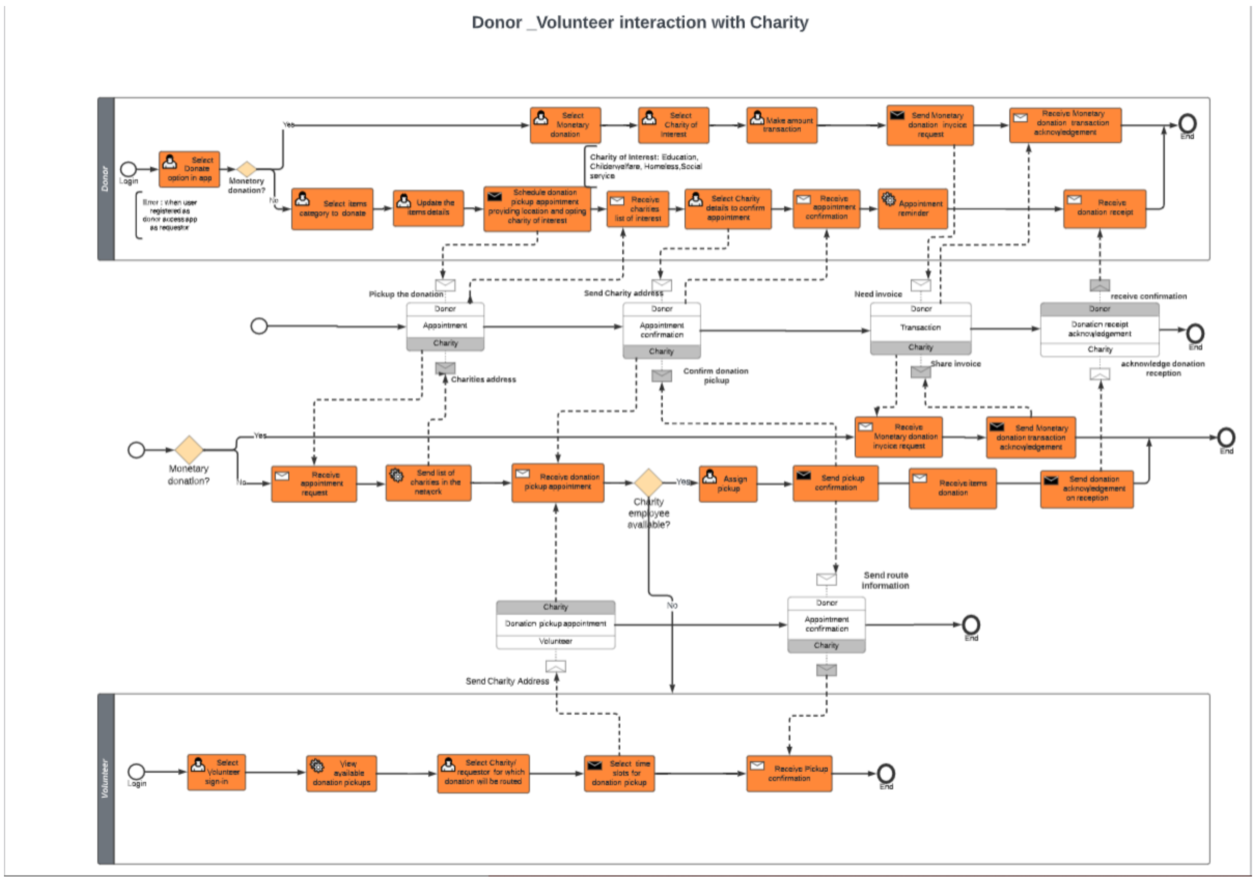
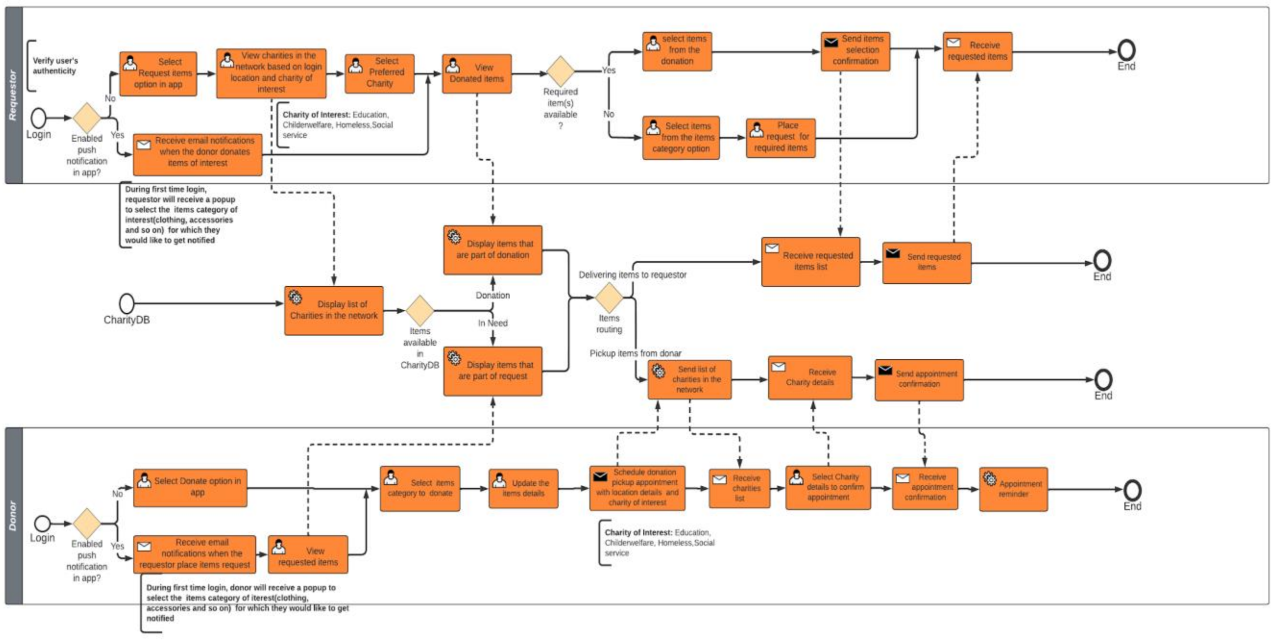
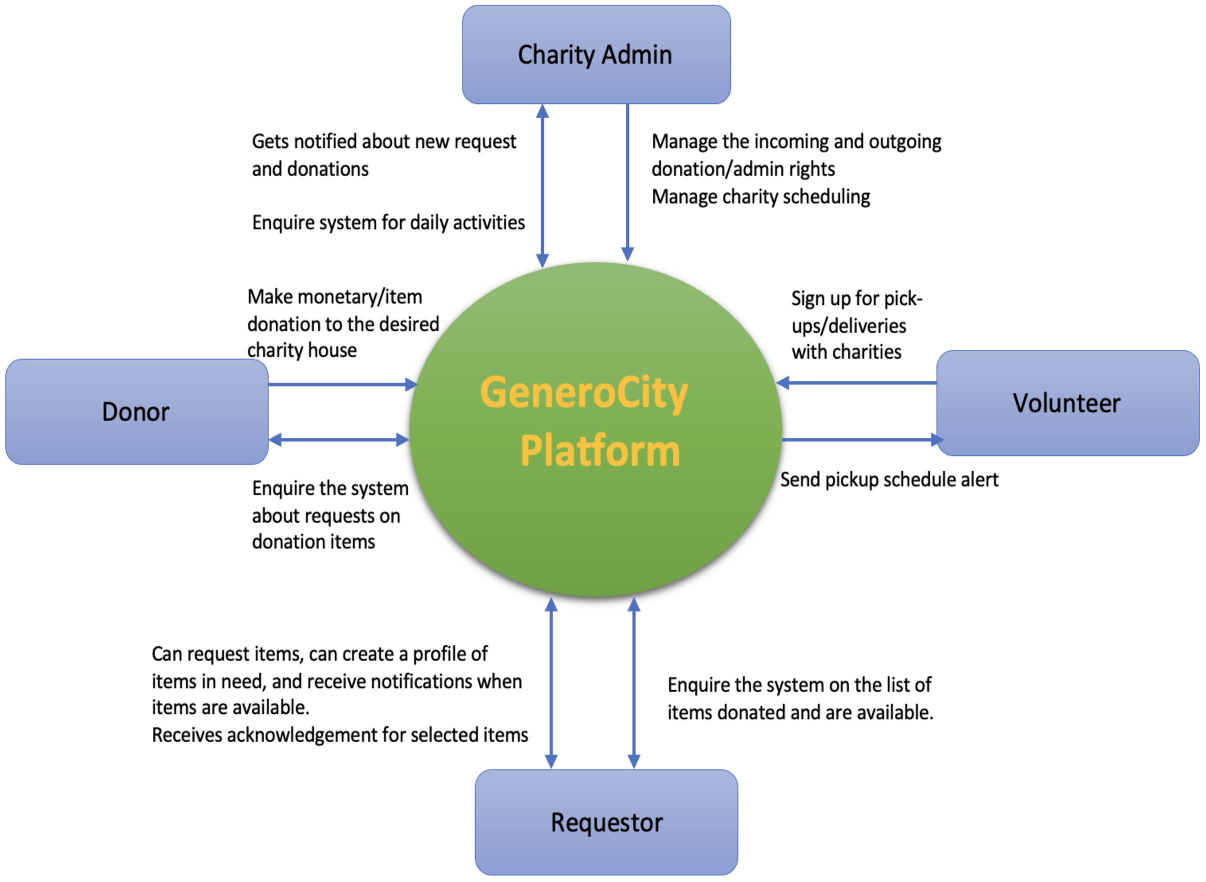

Donor _Volunteer interaction with Charity
Donor
Login
Select Donate option in app
Monetory donation?
Yes
No
Error : When user registered as donor access app as requestor
Select Monetary donation
Select Charity of Interest
Make amount transaction
Send Monetary donation invoice request
Receive Monetary donation transaction acknowledgement
End
Charity of Interest: Education, Childerwelfare, Homeless,Social service
Select items category to donate
Update the items details
Schedule donation pickup appointment providing location and opting charity of interest
Receive charities list of interest
Select Charity details to confirm appointment
Receive appointment confirmation
Appointment reminder
Receive donation receipt
Pickup the donation
Donor
Appointment
Charity
Charities address
Send Charity address
Appointment confirmation
Confirm donation pickup
Need invoice
Transaction
Share invoice
receive confirmation
Donation receipt acknowledgement
acknowledge donation reception
Monetary donation?
Receive appointment request
Send list of charities in the network
Receive donation pickup appointment
Charity employee available?
Assign pickup
Send pickup confirmation
Receive items donation
Send donation acknowledgement on reception
Receive Monetary donation invoice request
Send Monetary donation transaction acknowledgement
Send route information
Donation pickup appointment
Volunteer
Send Charity Address
Volunteer
Select Volunteer sign-in
View available donation pickups
Select Charity/ requestor for which donation will be routed
Select time slots for donation pickup
Receive Pickup confirmation

Table of Contents

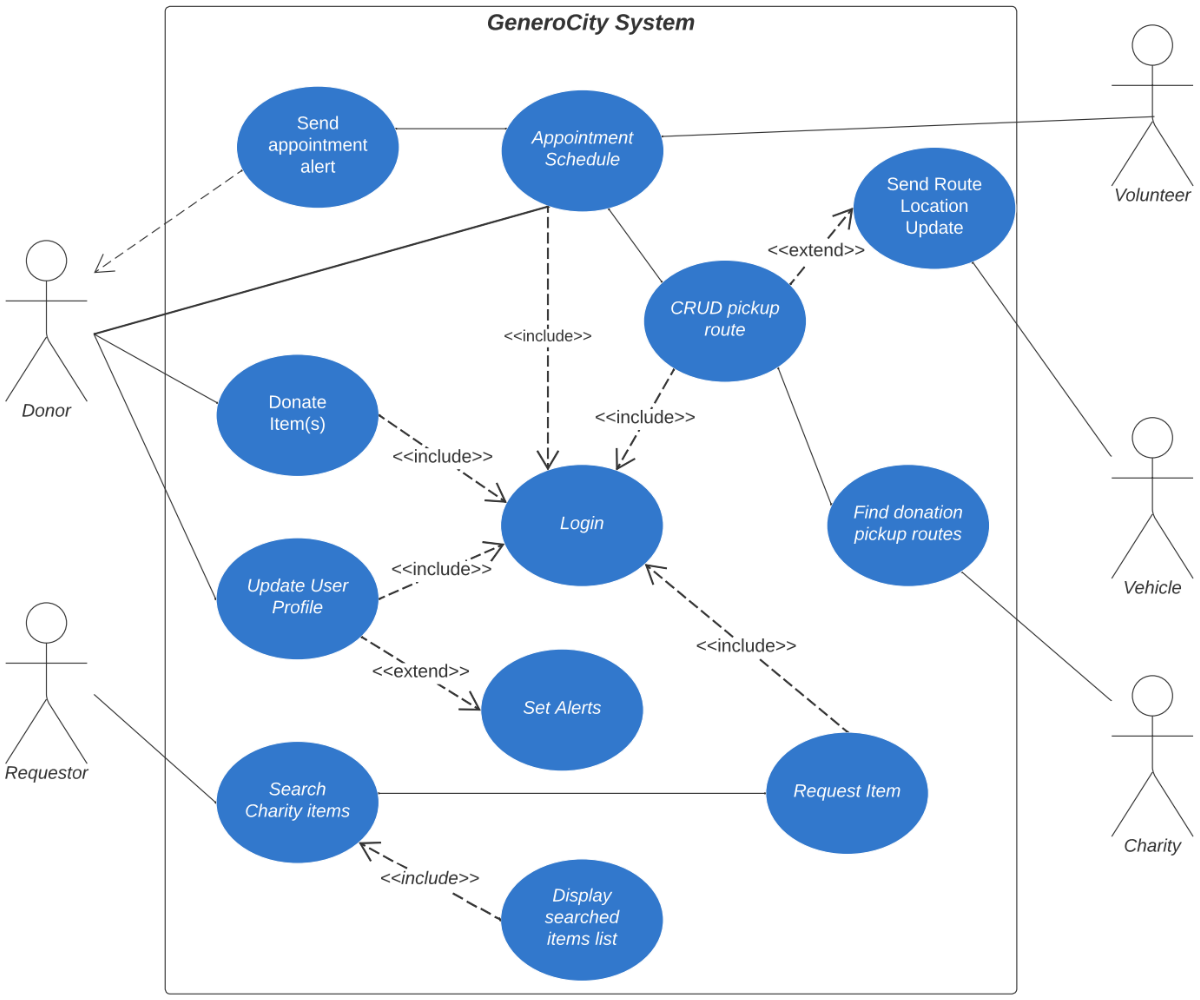
Requestor_Donor interaction with Charity


Requestor
Verify user's authenticity
Login
Enabled push notification in app?
No
Yes
Select Request items option in app
View charities in the network based on login location and charity of interest
Select Preferred Charity
View Donated items
Receive email notifications when the donor donates items of interest
Charity of Interest: Education, Childerwelfare, Homeless,Social service
Required item(s) available ?
Yes
No
select items from the donation
Send items selection confirmation
Receive requested items
End
Select items from the items category option
Place request for required items
During first time login, requestor will receive a popup to select the items category of interest(clothing, accessories and so on) for which they would like to get notified
CharityDB
Display list of Charities in the network
Items available in CharityDB
Donation
In Need
Display items that are part of donation
Display items that are part of request
Delivering items to requestor
Items routing
Pickup items from donar
Receive requested items list
Send requested items
End
Send list of charities in the network
Receive Charity details
Send appointment confirmation
End
Donor
Login
Enabled push notification in app?
No
Yes
Select Donate option in app
Receive email notifications when the requestor place items request
View requested items
Select items category to donate
Update the items details
Schedule donation pickup appointment with location details and charity of interest
Receive charities list
Select Charity details to confirm appointment
Receive appointment confirmation
Appointment reminder
End
Charity of Interest: Education, Childerwelfare, Homeless,Social service
During first time login, donor will receive a popup to select the items category of iterest(clothing, accessories and so on) for which they would like to get notified

CONTEXT DIAGRAM


Charity Admin
Gets notified about new request and donations
Enquire system for daily activities
Manage the incoming and outgoing donation/admin rights
Manage charity scheduling
Make monetary/item donation to the desired charity house
Donor
Enquire the system about requests on donation items
GeneroCity Platform
Sign up for pick-ups/deliveries with charities
Volunteer
Send pickup schedule alert
Can request items, can create a profile of items in need, and receive notifications when items are available.
Receives acknowledgement for selected items
Enquire the system on the list of items donated and are available.
Requestor

USE CASE DIAGRAM



USE CASE DESCRIPTIONS

Use Case 1: User Login

Use Case Name: User Login	**ID:** 1	**Importance:** High
Primary Actor: Requestor, Donor, Volunteer	**Use Case Type:** Essential, Detail	
Stakeholders and Interests: Requestor, Donor, Volunteer		
Brief Description: Login to Application		
Trigger: [App launch on media] User opens app. **Type:** External		
Normal Flow of Events: 1. User enters UserId and Password to login to the GeneroCity 2. User submits credentials 3. System verifies credentials and grant access		
Exceptional Flow of Events: 1. When user enters wrong user and password 2. Authentication failed, display invalid username and/or password (error screen) 3. Prompt reset		

Use Case 2: Get Charity List

Use Case Name: Get Charity List	**ID:** 2	**Importance:** High
Primary Actor: Requestor, Donor	**Use Case Type:** Essential, Detail	
Stakeholders and Interests: Requestor, Donor		
Brief Description: View list of charities and select charity of interest.		
Trigger: [Drop Down Menu] Selects charity from drop-down menu **Type:** External		
Normal Flow of Events: 1. User Login to GeneroCity app 2. User selects charity drop-down menu 3. System displays list of charities available 4. User selects a charity		
Exceptional Flow: Not Applicable		

Use Case 3: Choose Donation Type

Use Case Name: Choose Donation Type	**ID:** 3	**Importance:** High
Primary Actor: Donor	**Use Case Type:** Essential, Detail	
Stakeholders and Interests: Donor, Charity		
Brief Description: Donor selects the type of donation submission		
Trigger: [Donate Button] Donor selects donation type: monetary or items **Type:** External		
Normal Flow of Events: 1. Donor selects 'Donate' button **2.** Select donation type: Monetary or Item		
Exceptional Flow: Not Applicable		

Use Case 4: Get Category

Use Case Name: Get Category	**ID:** 4	**Importance:** High
Primary Actor: Requestor, Donor	**Use Case Type:** Essential, Detail	
Stakeholders and Interests: Requestor, Donor		
Brief Description: Select a category from option		
Trigger: [Select List of Categories] User request query of items from a selected category **Type:** External		
Normal Flow of Events: 1. User click on category button 2. GeneroCity displays available categorylist 3. User selects a preferred category type 4. User select items under a selected category		
Exceptional Flow: Not Applicable		

Use Case 5: Search For Item

Use Case Name: Search for Item	**ID:** 5	**Importance:** High
Primary Actor: Requestor, Donor	**Use Case Type:** Essential, Detail	
Stakeholders and Interests: Requestor, Donor		
Brief Description: How to search for an item in inventory for 1, many or all charities available to the app.		
Trigger: [Search Button] User accesses application and searches using the search bar within the app by entering a search keyword. **Type:** External		
Normal Flow of Events: 1. User logins to GeneroCity 2. Authentication credentials retrieved 3. User select Charity(s) to search (all or selected) inventory 4. User selects the Category button to display category types 5. User searches for items in the Item 6. List of items that closely match the keyword populate in order of relevance		
Exceptional Flow: **1.** Search item returned 'not found'		

Use Case 6: Add Item to Cart

Use Case Name: Add Item to Cart	**ID:** 6	**Importance:** High
Primary Actor: Requestor	**Use Case Type:** Essential, Detail	
Stakeholders and Interests: Requestor		
Brief Description: Request item and submit interest		
Trigger: [Add Item] Choose the add item to cart button. **Type:** External		
Normal Flow of Events: 1. Select Item from Item list returned via Use Case "Get Items" 2. Choose "add item"		
Exceptional Flow: Not applicable		

Use Case 7: Checkout Cart (Request Items)

Use Case Name: Checkout Cart (Request Items)	**ID:** 7	**Importance:** High
Primary Actor: Requestor	**Use Case Type:** Essential, Detail	
Stakeholders and Interests: Requestor		
Brief Description: Request item and submit interest		
Trigger: [Submit] Choose submit in the checkout **Type:** External		
Normal Flow of Events: 1. Confirm Cart Items/Edit Cart Items 2. Select "Checkout" to display the cart contents 3. Get confirmation of request		
Exceptional Flow: Not Applicable		

Use Case 8: Make Appointment

Use Case Name: Make Appointment	**ID:** 8	**Importance:** High
Primary Actor: Donor	**Use Case Type:** Essential, Detail	
Stakeholders and Interests: Donor		
Brief Description: Donor will schedule appointment with charity for item donation.		
Trigger: [Select check box]Makes appointment **Type:** External		
Normal Flow of Events: 1. Donor picks a charity 2. Donor gets charity appointment schedule 3. Donor selects an appointment 4. Confirmation Notification is sent 5. Donor gets an appointment reminder		
Exceptional Flow: 1. Route is canceled by charity 2. Donor rescheduling appointment		

Table of Contents

Use Case 9: Get Invoice

Use Case Name: Get Invoice	**ID:** 9	**Importance:** High
Primary Actor: Donor	**Use Case Type:** Essential, Detail	
Stakeholders and Interests: Donor		
Brief Description: Donor receives an invoice for donated items		
Trigger: Donor [selects request invoice] within GeneroCity app and downloads. **Type:** External		
Normal Flow of Events: 1. Donor login 2. Select donor profile 3. Select donations 4. Download invoice for selected donation		
Exceptional Flow: 1. Invoice not available 2. Donor request invoice with charity		

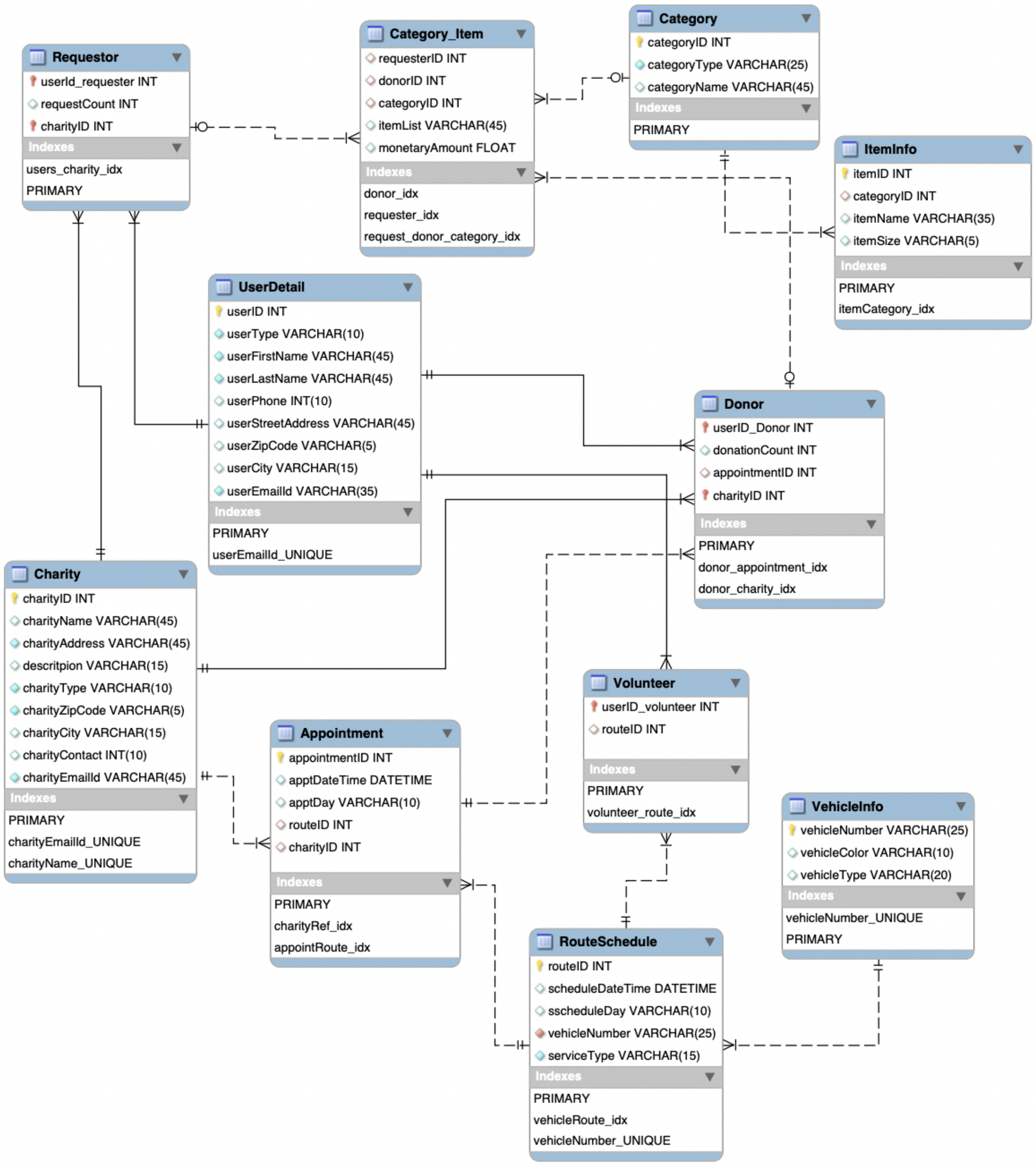
Use Case 10: CRUD Schedule Routes – Charity/Employee

Use Case Name: CRUD Schedule Routes – Charity/Employee	**ID:** 10	**Importance:** High
Primary Actor: Charity Admin	**Use Case Type:** Essential, Detail	
Stakeholders and Interests: Charity Admin		
Brief Description: Charity adds routes and create appointment availability list		
Trigger: [Confirm] Charity adds a donation pickup route **Type:** External		
Normal Flow of Events: 1. Log in as charity admin 2. Schedule route to CRUD 3. Make the change for route 4. Confirm route schedule		
Exceptional Flow: Not Applicable		

Use Case 11: Get Volunteer Schedule

Use Case Name: Get Volunteer Schedule	**ID:** 11	**Importance:** High
Primary Actor: Volunteer	**Use Case Type:** Essential, Detail	
Stakeholders and Interests: Volunteer		
Brief Description: Volunteer selects schedule and views assigned route.		
Trigger: [View Route Schedule] Open profile and select 'Route Schedule' **Type:** External		
Normal Flow of Events: 1. Volunteer logins to GeneroCity app 2. Selects a Charity from list of charities 3. Selects View Route Schedule 4. Volunteer Selects available route 5. Volunteer receives route assignment confirmation		
Exceptional Flow: Not Applicable		

DATABASE DESIGN



CONSTRAINTS:

UserDetail (userID, userType, userFirstName, userLastName, userPhone, userStreetAddress, userZipCode, userCity,userEailId**)**

userID is PK should be non-null and unique.

Requestor(userId_requester, requestCount, charityID**)**

userId_requester is FK referenced from the UserDetail (userID) table.

charityID is FK, should be non-null and should exist in the Charity table.

Donor(userID_Donor, donationCount, appointmentID, charityID**)**

userID_Donor is FK referenced from the UserDetail (userID) table.

appointmentID is FK, should be non-null and should exist in the Appointment table.

charityID is FK, should be non-null and should exist in the Charity table.

Volunteer(userID_volunteer, routeID**)**

userID_volunteer is FK referenced from the UserDetail (userID) table.

routeID is FK, should be non-null and should exist in the RouteSchedule table.

Charity(charityID, charityName, charityAddress, description, charityType, charityZipCode, charityCity, charityContact, charityEmailId**)**

charityID is PK, should be non-null and unique.

Category_Item(requesterID, donorID, categoryID, itemList, monetaryAmount**)**

requesterID is FK referenced from the Requestor table.

donorID is FK referenced from the Donor table.

categoryID is FK referenced from the Category table.

Category(categoryID, categoryType, categoryName**)**

categoryID is PK, should be non-null and unique.

ItemInfo(itemID, categoryID, itemName, itemSize**)**

itemID is PK, should be non-null and unique.

categoryID is FK, should be non-null and should exist in the Category table.

Table of Contents

Appointment(appointmentID, apptDateTime, apptDay, routeID, charityID**)**

appointmentID is PK, and should be non-null and unique.

routeID is FK, should be non-null and should exist in the RouteSchedule table.

charityID is FK, should be non-null and should exist in the Charity table.

RouteSchedule(routeID, scheduleDateTime, scheduleDay, vehicleNumber, serviceType**)**

routeID is PK, and should be non-null and unique.

vehicleNumber is FK, and should be non-null and should exist in the VehicleInfo table.

VehicleInfo(vehicleNumber, vehicleColor vehicleType**)**

vehicleNumber is PK, should be non-null and unique.

Table of Contents

DATA DICTIONARY

User Preference = Notification Alerts + Default Charity + (profile type)

UserDetail = UserID + (userType) + userFirstName + userLastName + userPhone + userStreetAddress + userZipCode + userCity + userEmail

userType = [charity Admin | volunteer | donor | Requestor]

Login Info = Username + Password

Username = Email Profile

Password = data element

Donor = userId_Donor + donationCount + appointmentID + charityID

Requestor = RequestorID + categoryID + {itemList}

Add Item to Cart = {category + item} + charity + User

Category = categoryID + categoryType

categoryType = [Monetary | Item]

Item = ItemID + CategoryID + (itemName) + (itemDescription) + (itemSize)

Charity = charityId + charityName + charityAddress + (description) + charityType + charityZipCode + charityCity + charityContact + charityEmail

Appointment = AppointmentID + apptDateTime + apptDay

Volunteer = userId + routeId + pickUpDay + pickUpDateTime

RouteSchedule =routeNumber + scheduleDateTime + scheduleDay + serviceType

serviceType = [Delivery |pickUp]

VehicleInfo = VehicleNumber + {vehicle color} + (vehicleType)

CLASS DIAGRAM

Baseline Business Scenario considered: **'**Charity' entity is deciding and preparing 'RouteSchedule' for their respective charities which will then be displayed to 'Donor' to pick 'Appointment' of their choice and 'Volunteer' to select 'RouteSchedule' from the provided list by the charities.

- WITHOUT METHODS:

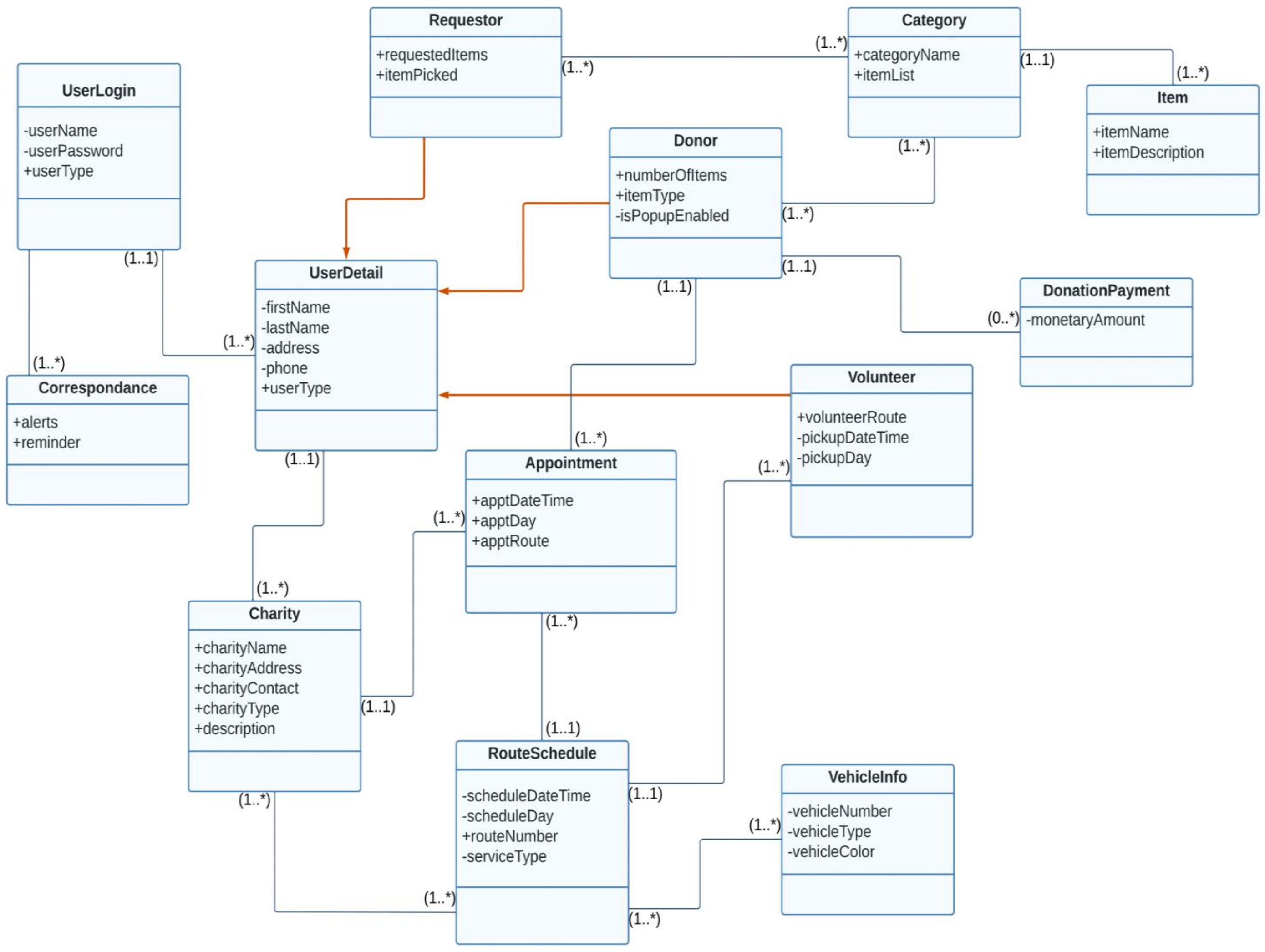
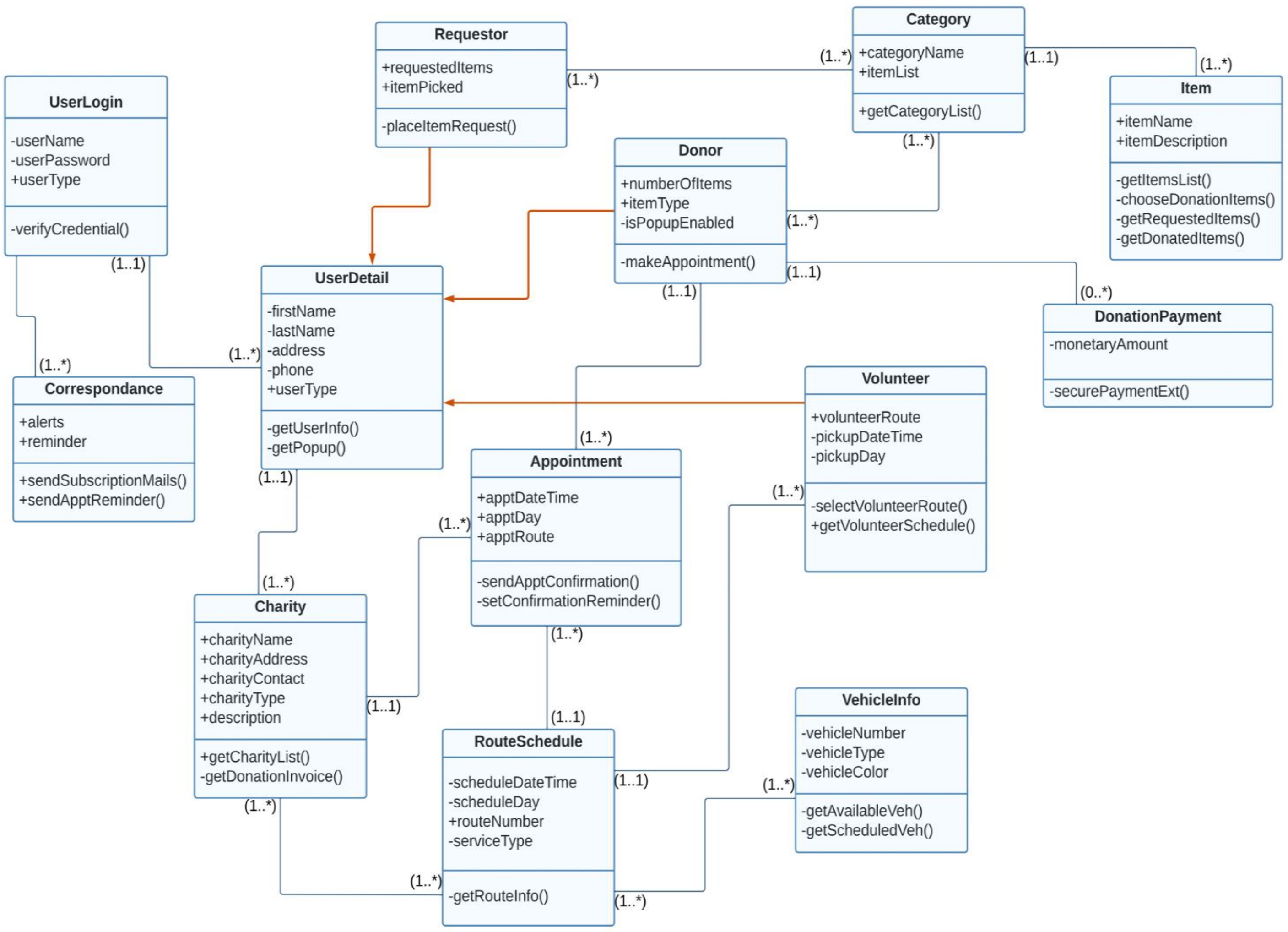


Table of Contents

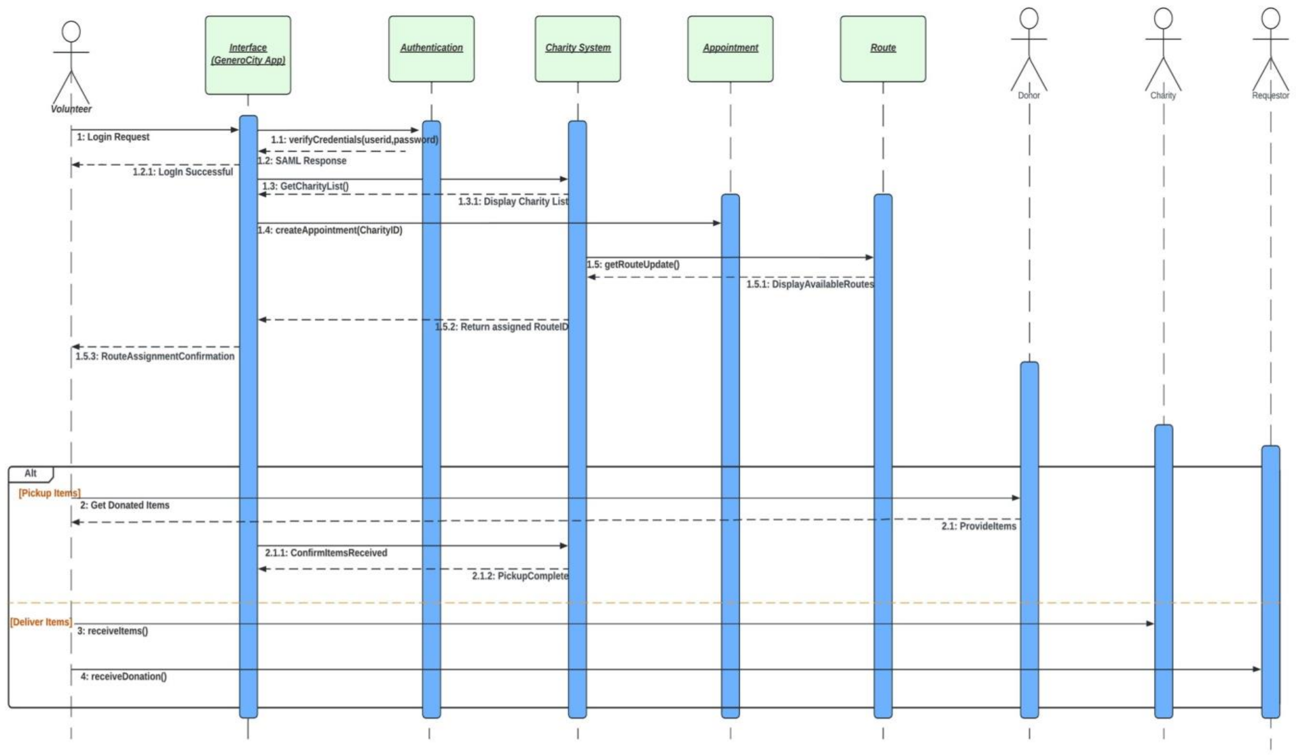
- WITH METHODS:



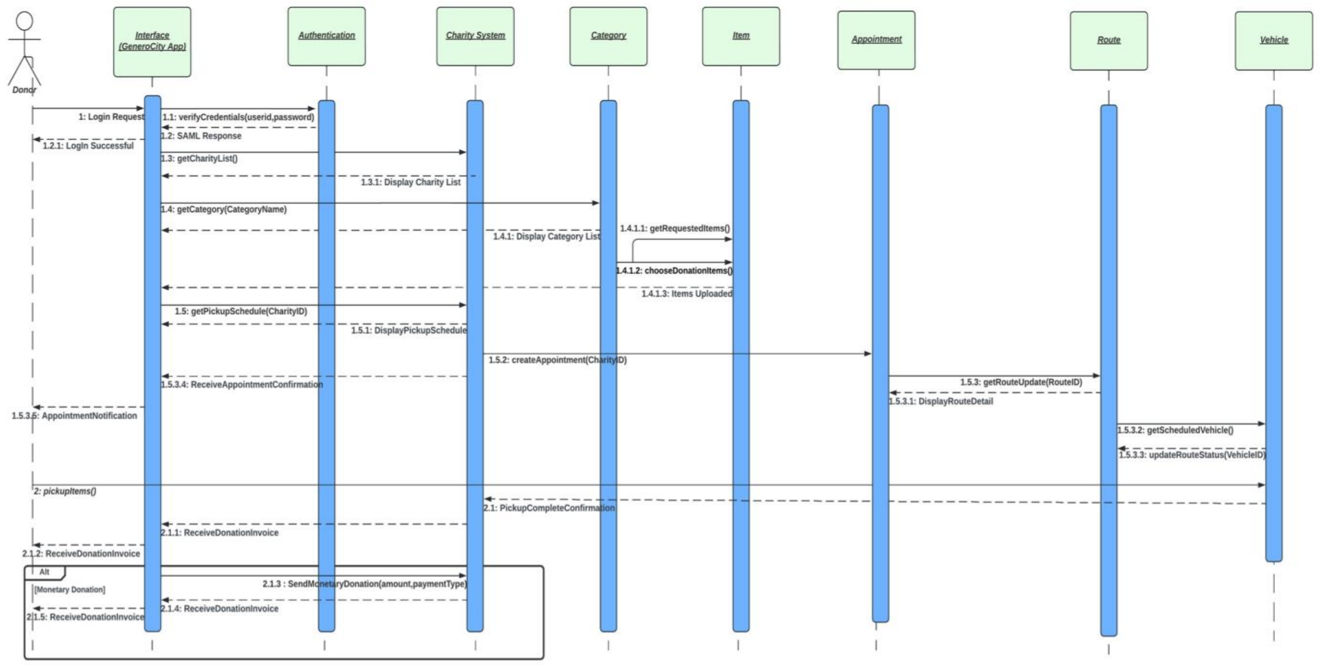
[Table of Contents](#)

SEQUENCE DIAGRAMS

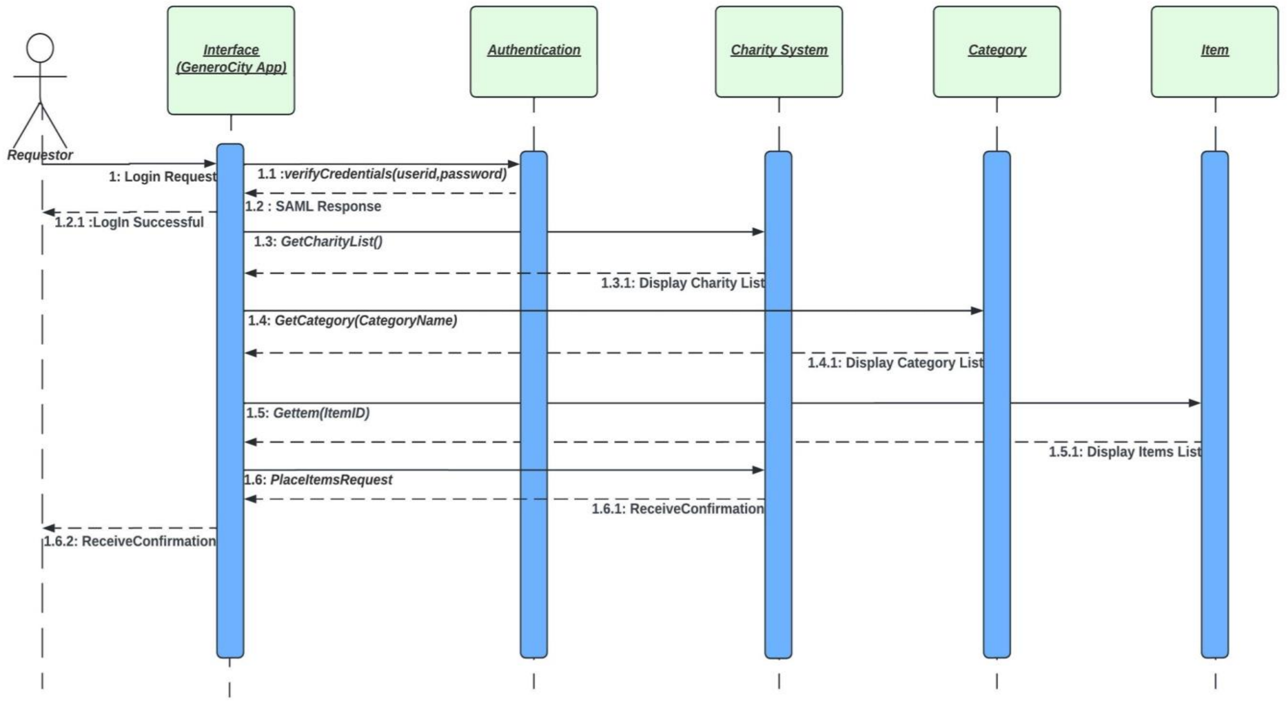
A. Volunteer_Charity



B. Donor_Charity


Donor
Interface (GeneroCity App)
Authentication
Charity System
Category
Item
Appointment
Route
Vehicle
1: Login Request
1.1: verifyCredentials(userid,password)
1.2: SAML Response
1.2.1: LogIn Successful
1.3: getCharityList()
1.3.1: Display Charity List
1.4: getCategory(CategoryName)
1.4.1: Display Category List
1.4.1.1: getRequestedItems()
1.4.1.2: chooseDonationItems()
1.4.1.3: Items Uploaded
1.5: getPickupSchedule(CharityID)
1.5.1: DisplayPickupSchedule
1.5.2: createAppointment(CharityID)
1.5.3: getRouteUpdate(RouteID)
1.5.3.1: DisplayRouteDetail
1.5.3.2: getScheduledVehicle()
1.5.3.3: updateRouteStatus(VehicleID)
1.5.3.4: ReceiveAppointmentConfirmation
1.5.3.5: AppointmentNotification
2: pickupItems()
2.1: PickupCompleteConfirmation
2.1.1: ReceiveDonationInvoice
2.1.2: ReceiveDonationInvoice
Alt
[Monetary Donation]
2.1.3 : SendMonetaryDonation(amount,paymentType)
2.1.4: ReceiveDonationInvoice
2.1.5: ReceiveDonationInvoice

C. Requestor_Charity


Requestor
Interface (GeneroCity App)
Authentication
Charity System
Category
Item
1: Login Request
1.1 :verifyCredentials(userid,password)
1.2 : SAML Response
1.2.1 :LogIn Successful
1.3: GetCharityList()
1.3.1: Display Charity List
1.4: GetCategory(CategoryName)
1.4.1: Display Category List
1.5: Gettem(ItemID)
1.5.1: Display Items List
1.6: PlaceItemsRequest
1.6.1: ReceiveConfirmation
1.6.2: ReceiveConfirmation

FUNCTIONAL SPECIFICATIONS

1. A donor can view the items requested by the Requestor (seeker) and make donations by selecting the category of the item which includes kids' clothing, toys, stationeries, accessories, and furniture.

2. A donor will be able to select the charity of interest (education, child welfare, homeless, social service) when making a donation and schedule a donation pickup appointment with the charity. GeneroCity will notify the donor through a reminder 24 hrs. prior to the pickup. Additionally, a donor will get notified via email and *push notification when the Requestor places an item request (*provided the push notifications option in the application is enabled) and be able to make monetary donations and get an invoice in return for the donation.

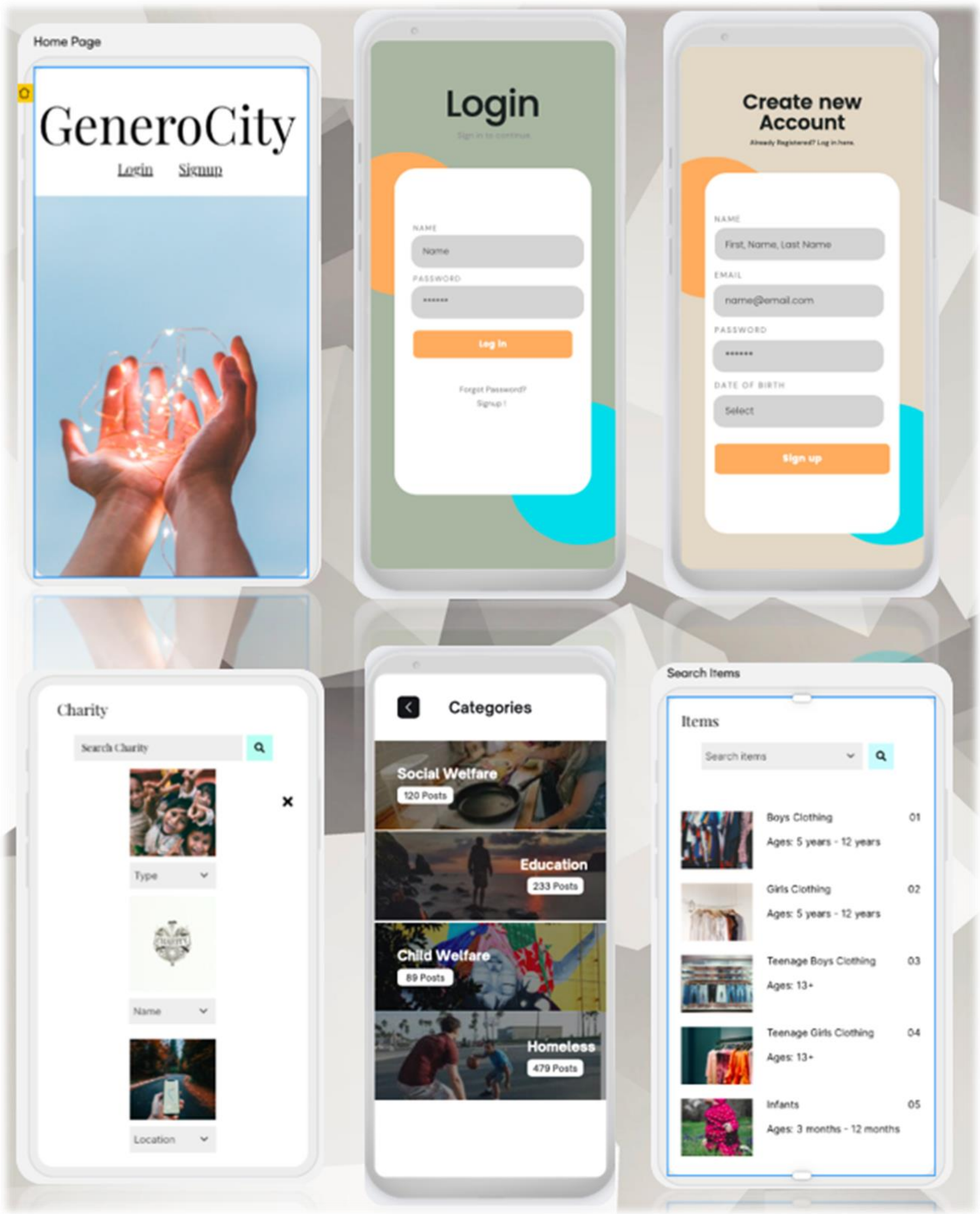
3. A Requestor will be able to view the donated items in the application, if required items are available, then they will be able to select the items for delivery. Otherwise, the Requestor will be able to make a new item(s) request by visiting the item's category in the application. GeneroCity will notify the Requestor through a reminder 24 hours prior to the drop-off. Additionally, the Requestor will get notified via email and *push notification when the donor donates an item (*provided the push notifications option in the application is enabled).

4. GeneroCity maintains the network of charities that are eligible to receive donations and displays the automated suggested charity list to the donor when the donor schedules the donation pickup appointment.

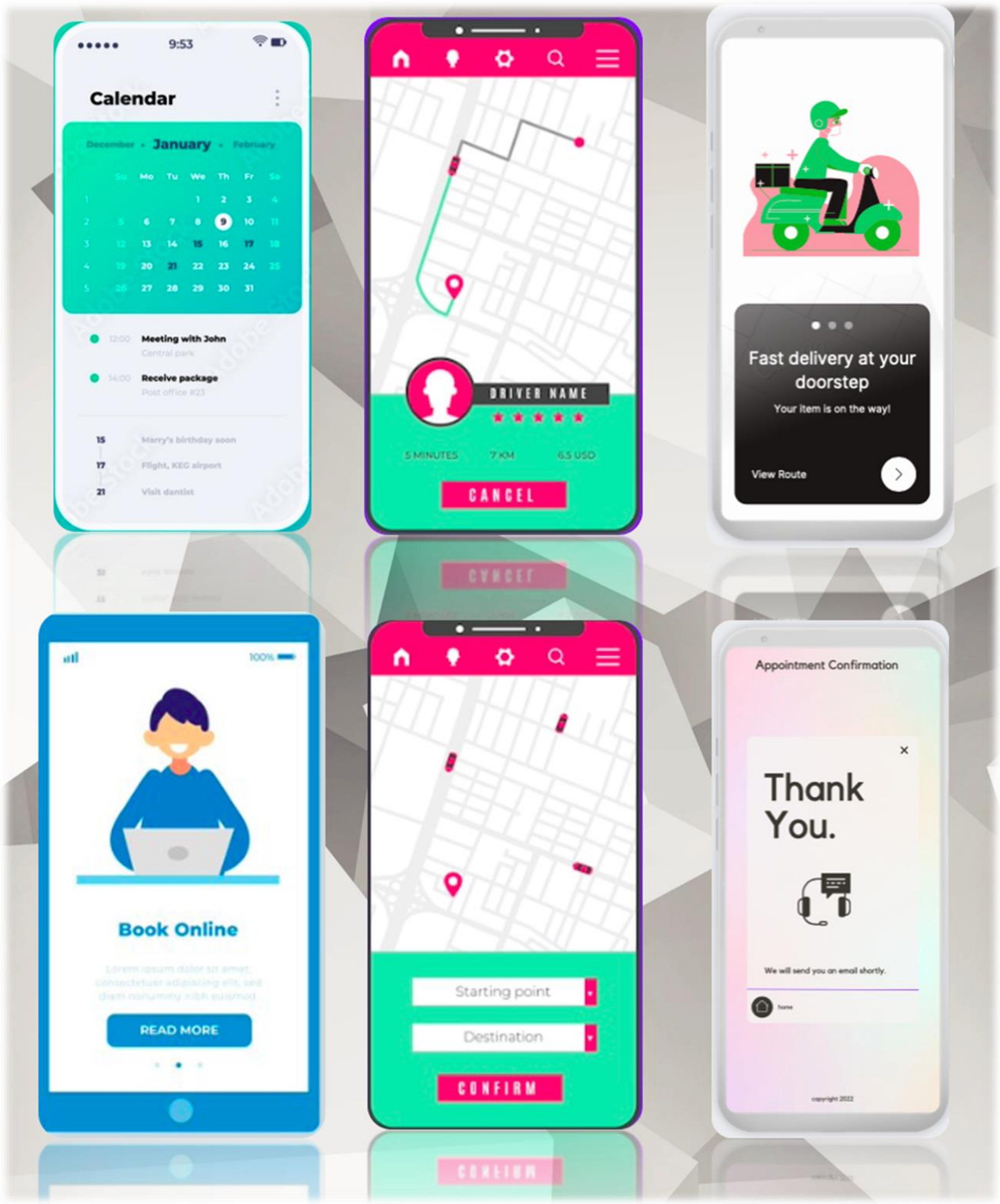
5. GeneroCity serves as an interface between the donor and the volunteer. The charity database receives donation pickup appointments from donors and allows volunteers to confirm the convenient slot for the donation pickup.

6. A volunteer will be able to view available donation pickups, and select time slot(s) and charity for which donation will be routed.

INTERFACE DESIGN



Donor and Volunteer Perspective:



SOFTWARE DESIGN

Signature:

Method Name: verifyCredentials()	**Class Name:** UserLogin
Client (Consumers): Donor, Requestor, Volunteer	
Associated Use cases: USER LOGIN	
Description of responsibilities: UserLogin must verify the credentials i.e. userid and password for successful authentication	
Arguments Received: userId, userPassword	
Type of Value Returned: Boolean	
Preconditions: user enters userId and userPassword	
Post Conditions: user navigates to the application	

Logic:

```
DISPLAY LOGIN FORM
FETCH userid FROM Genorocity app
FETCH pwd FROM Genorocity app
IF Member
THEN
DISPLAY “Login Successful”
```

Table of Contents

```
ELSE

DISPLAY "Please try again!!"

RETURN DISPLAY LOGIN FORM
```

Signature:

Method Name: getCategory()	**Class Name:** Category
Client (Consumers): Donor, Requestor	
Associated Use cases: GET CATEGORY	
Description of responsibilities: User is required to select the category of the item (Kids clothing, toys, accessories, furniture or monetary donation)in order to make donation or request items	
Arguments Received: categoryName	
Type of Value Returned: String	
Preconditions: user enters categoryName	
Post Conditions: user will be able to view itemsList from the selected category	

Logic:

```
IF "Search Category" = "successful"

THEN

DISPLAY categoryName

ELSE
```

Table of Contents

```
THEN

DISPLAY “select from the categories available”

RETURN DISPLAY Category option
```

Signature:

Method Name: chooseDonationType()	**Class Name:** Item
Client (Consumers): Donor	
Associated Use cases: CHOOSE DONATION TYPE	
Description of responsibilities: Donor will select items list to make a donation and update details associated with the items.	
Arguments Received: itemID, itemName, itemsize	
Type of Value Returned: string	
Preconditions: user selects itemName	
Post Conditions: user will be able to view items name, and size (if applicable) from the selected category	

Logic:

```
Select Items to make donation

IF “itemsName” = “Available”

THEN
```

Table of Contents

```
Update item(s) and size(if applicable)
ELSE
THEN
DISPLAY "select items from the available items list"
 RETURN DISPLAY Item option
```

Signature:

Method Name: getInvoice()	**Class Name:** Charity
Client (Consumers): Donor	
Associated Use cases: GET INVOICE	
Description of responsibilities: Donor will send a monetary donation invoice request to charity	
Arguments Received: charityType, charityName	
Type of Value Returned: string	
Preconditions: Donor selects charityType, charityName to request an invoice from the charity to which the monetary donation is made.	
Post Conditions: Donor will receive monetary donation invoice from the charity	

Logic:

```
FETCH "itemCategory" FROM "Category"
IF "itemCategory" = "MonetaryDonation"
THEN
ENTER charityType, charityName to get invoice
ELSE
THEN
RECEIVE "Donation invoice"
```

Signature:

Method Name: getRequestedItems()	**Class Name:** Item
Client (Consumers): Requestor	
Associated Use cases: CHECKOUT CART "Request Items", "Add Item To Cart"	
Description of responsibilities: Requestor will search for required items in the application, if available, then place a request to get those items delivered. Otherwise, place a new request for the required items.	
Arguments Received: itemID, itemName, itemsize	
Type of Value Returned: string	
Preconditions: Requestor selects itemName	

Post Conditions: Requestor will be able to view items name, and size (if applicable) from the selected category

Logic:

```
FETCH "itemName"
EXECUTE "itemName search"
IF "itemName" = "Found"
THEN
Place request for items from the donation list
ELSE
THEN
ADD new request by selecting items from the category.
```

Signature:

Methods Name: getCharityList()	**Class Name:** Charity
Client (Consumers): Donor, Requestor	
Associated Use cases: GET CHARITY LIST	
Description of responsibilities: User will search for charity in the networks based on the preferred location, type, and/or name	
Arguments Received: location, CharityName, CharityType	

Table of Contents

Type of Value Returned: string
Preconditions: user enters the location and/or CharityName and/or CharityType to find the available charity.
Post Conditions: user will be able to donate /request items from charity

Logic:

```
FETCH “Location”, “CharityType”, “CharityName”
IF “search criteria” = “Location” and “search status” = “Successful”
THEN
DISPLAY “Charities in the entered location “
ELSE IF “search criteria” = “CharityType” and “search status” = “Successful”
THEN
DISPLAY “Charities that matches the requested charity type”
ELSE IF “search criteria” = “CharityName” and “search status” = “Successful”
THEN
DISPLAY “Charity”
ELSE
THEN
DISPLAY “Search not found!”
 RETURN DISPLAY search option
```

Signature:

Methods Name: getVolunteerSchedule()	**Class Name:** Volunteer
Client (Consumers): Volunteer	
Associated Use cases: GET VOLUNTEER SCHEDULE	
Description of responsibilities: Volunteer will serve donation pickup	
Arguments Received: pickUpDate, pickUpTime, Route	
Type of Value Returned: Date Time	
Preconditions: Volunteer signup for donation pickup by selecting preferred pickUpTime, pickUpDate	
Post Conditions: Volunteer is assigned with donation pickup route	

Logic:

```
FETCH “pickUpTime”, “pickUpDate”

SELECT preferred “pickUpDate” and “pickUpTime”

IF “pickUpDate” = “Confirmed” and “pickUpTime” = “Confirmed”

THEN

ASSIGN route to volunteer

ELSE
```

```
THEN

DISPLAY “choose from available slots”.
```

Signature:

Methods Name: createAppointment()	**Class Name:** Appointment
Client (Consumers): Donor	
Associated Use cases: No use case present- write pickup schedule use case as well, so that I could use it here as a use case covered instead of creating another signature for it	
Description of responsibilities: MAKE APPOINTMENT	
Arguments Received: charityID, appointmentDateTime	
Type of Value Returned: string	
Preconditions: Donor select CharityID, appointmentDateTime for donation pickup	
Post Conditions: Donor receives appointment confirmation from charity	

Logic:

```
FETCH “charityID”, “appointmentDateTime”

SELECT charityID

IF “appointmentDateTime” = “Available”
```

Table of Contents

```
THEN
DISPLAY “Donation pickup appointment is confirmed “
ELSE
THEN
DISPLAY “Select appointment from available slots!”
```

Signature:

Methods Name: getRouteUpdate()	**Class Name:** Route
Client (Consumers): Charity Admin	
Associated Use cases: CRUD - SCHEDULE ROUTE	
Description of responsibilities: Charity assign route to Volunteer for DonationPickUp and delivery	
Arguments Received: RouteID, vehicleNumber	
Type of Value Returned: varchar	
Preconditions: Charity fetches route information via RouteID when appointment is created for donation pickup/delivery	
Post Conditions: Volunteer is assigned with a route for donation pickup /delivery	

Logic:

```
FETCH “RouteID”,”vehicleNumber”

CASE

WHEN Volunteer =”available” and serviceType = “DonationPickUp”

THEN

ASSIGN RouteID and vehicleNumber to volunteer for donation pickup

WHEN Volunteer = “available” and serviceType = “ItemDelivery”

THEN

ASSIGN RouteID and vehicleNumber to volunteer for donation delivery

END CASE
```

Table of Contents