

DeepSink Partners

Give A Lot

Kids Next Door

Software Requirements and Design Specifications

Team Name: KidsNextDoor

Team Members: * - *indicates team leader*

Name	Surname	Student Number
Kgomotso	Moroke	u19104546*
Nqobani	Nhlengethwa	u17297542
Tshilidzi	Nekhavambe	u17090939
Joshua	Warneke	u18227369
Rashiidah Weira	Wanda	u17281017

Contents

1	Project Owner	4
2	Project Vision and Objectives	4
3	Functional Requirements	4
4	User classes and characteristics	5
4.0.1	NGOs/Organisations	5
4.0.2	Give Alot platform Admins	5
4.0.3	General public	5
5	User Stories	6
6	System Class Diagram	6
6.1	Description	7
7	Subsystems	7
7.1	Verification subsystem	7
7.1.1	Description	7
7.1.2	Use case Diagram	7
7.1.3	Domain model	8
7.1.4	Service contract	8
7.2	Search subsystem	9
7.2.1	Description	9
7.2.2	Use Cases	9
7.2.3	Domain Model	10
7.2.4	Service Contracts	10
7.3	Donation subsystem	11
7.3.1	Description	11
7.3.2	Use Cases	11
7.3.3	Domain Model	12
7.3.4	Service Contracts	12
7.4	User Subsystem	13
7.4.1	Description	13
7.4.2	Use Cases	13
7.4.3	Domain Model	14
7.4.4	Service Contracts	14
7.5	Admin Subsystem	16
7.5.1	Description	16
7.5.2	Use Cases	16
7.5.3	Domain Model	17
7.5.4	Service Contracts	17

8	Quality Requirements	19
8.1	Maintainability	19
8.2	Scalabilty	19
8.3	Security	19
8.4	Testibility	19
8.5	Usability	20
9	Traceability Matrix	20

1 Project Owner

Deepsink partners.

2 Project Vision and Objectives

Give a Lot is a relatively simple idea that is meant to solve the problem that individuals and organisations encounter when it comes to verifying the authenticity of charities and many other different types of organisations. Give a Lot aims to solve this problem by becoming a central point of reference for the verification for charities and other types of organisations.

The purpose of this project is to design and create a secure system for verifying the authenticity of charities and many other different types of organisations. The system must be secure to protect privacy, user data and other compliances.

3 Functional Requirements

Priority Listing: Must, Should, Could, Wish-List

Organization Portal

- FR01 The system should allow an Organisation to create a profile. (Must)
- FR02 The system should allow an Organisation to login and logout. (Must)
- FR03 The system should send a verification email to the user after they have Registered. (could)
- FR04 The system should allow an Organisation to manage their details.(Should)
- FR05 The system should allow an organisation to update their Password. (Must)
- FR06 The system should allow an organisation to delete their profile. (Should)

public user Portal

- FR07 The system should allow public users to create an account.(Should)
- FR08 The system should allow public users to login and logout. (Should)
- FR09 The system should send a verification email to the user after they have Registered. (could)
- FR10 The system should allow public users to update their profile details. (Should)
- FR11 The system should allow public users to update their Password. (Should)

- FR12 The system should allow public users to list all available organisations in the system. (Must)
- FR13 The system should allow public users to search for an organisation. (Must)
- FR14 The system should return search results(Must)
- FR15 The system should return results from other public platforms in case the organisation is not registered with GiveAlot. (Should)
- FR16 The system should allow the public users to donate to an organisation. (Should)
- FR17 The system should allow public users to view their transaction history. (Should)
- FR18 The system should allow public users to Report organisations

Admin Portal

- FR19 The system should allow an Admin to search for an organisation. (Must)
- FR20 The system should allow an Admin to manage organisations. (Must)
- FR21 The system should allow an Admin to verify an Organisation. (Must)
- FR22 The system should generate a certificate to legitimate organisations. (Must)
- FR23 The system should allow an Admin to verify certificates. (Should)
- FR24 The system should allow an Admin to revoke an organisations certificate. (Must)
- FR25 The system should send an email to the organisation after it has been verified to be legitimate.(could)

4 User classes and characteristics

4.0.1 NGOs/Organisations

- Register on the platform to obtain a digital certificate that shows that it is indeed Give Alot verified. The digital certificate is used to show people/general public that the organisation is authentic and therefore they should not worry about if the donations made are for the right cause.

4.0.2 Give Alot platform Admins

- The administrators are responsible for approving the organisations and NGOs that register on the Give Alot platform.

4.0.3 General public

- The General public uses the platform to verify the authenticity of an organisation that claims to be verified by Give Alot.

5 User Stories

- **As a basic user I want** to log in **so that** my privacy is ensured, and my application is personalized.
- **As a basic user I want** to be able to check the authenticity of an organisation **to ensure** that the donations made are used for the right cause
- **As an Admin I want** to be able to approve organisations using the certificates generated **so that** the organisations can use these certificates to prove their authenticity.
- **As an Admin I want** to log in **so that** my privacy is ensured, and my application is personalized.
- **As an Admin I want** to log in **so that** my privacy is ensured, and my application is personalized.
- **As an organisation I want** to log in **so that** my privacy is ensured, and my application is personalized.
- **As an organisation I expect** to receive a certificate **so as** to prove the organisation's authenticity

6 System Class Diagram

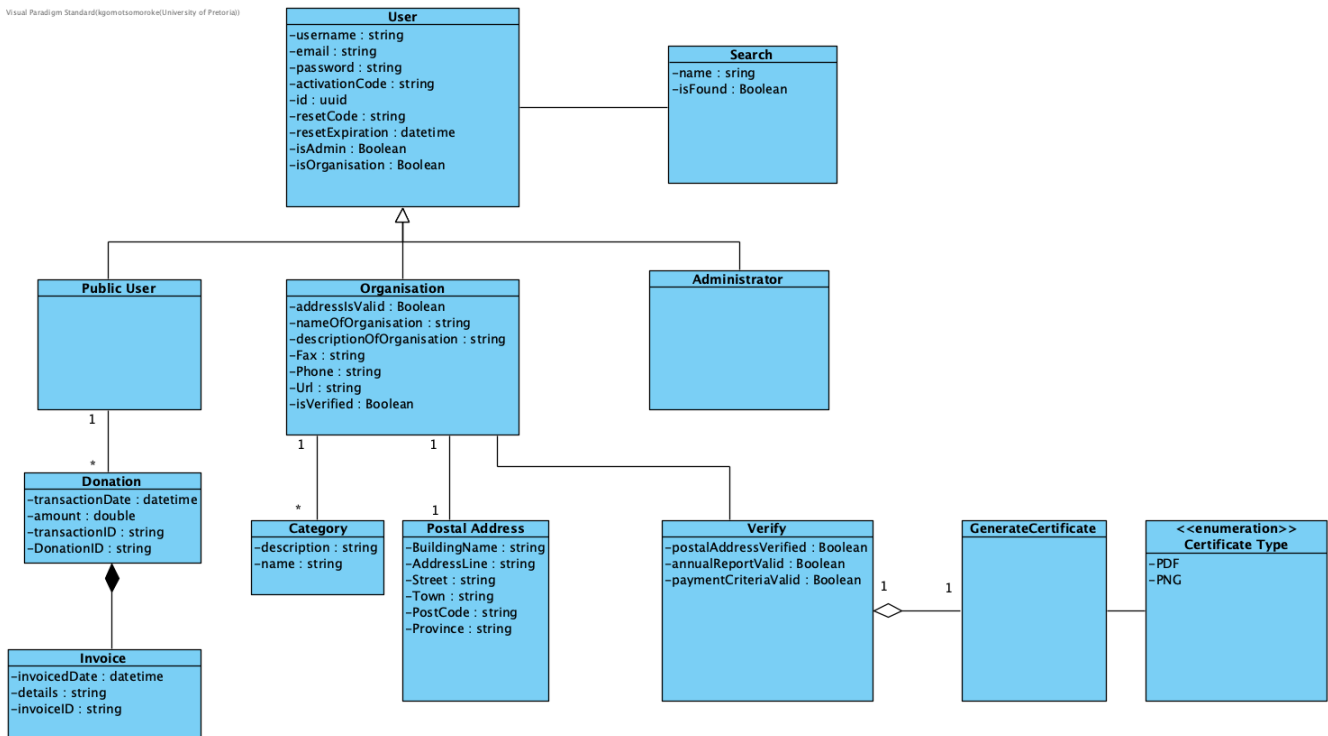


Figure 1: System Domain Model

6.1 Description

In order to make the Give a Lot more efficient to implement, a service-based organization approach is taken where the system is divided into several subsystems to make the system less tightly coupled and allow for concurrent implementation of features. To this end, the Give a Lot system comprises of 5 subsystems, namely User, Search, Donation, Verify, Generate Certificate System.

7 Subsystems

7.1 Verification subsystem

7.1.1 Description

The verification sub-system is responsible for checking the authenticity and integrity of the Organisations and NGOs that deal in charities. The verification applications get the verification information through the server, then the system authenticates the validation of the verification information by issuing a digital certificate.

7.1.2 Use case Diagram

The use cases of the search subsystem are shown in Figure 2.

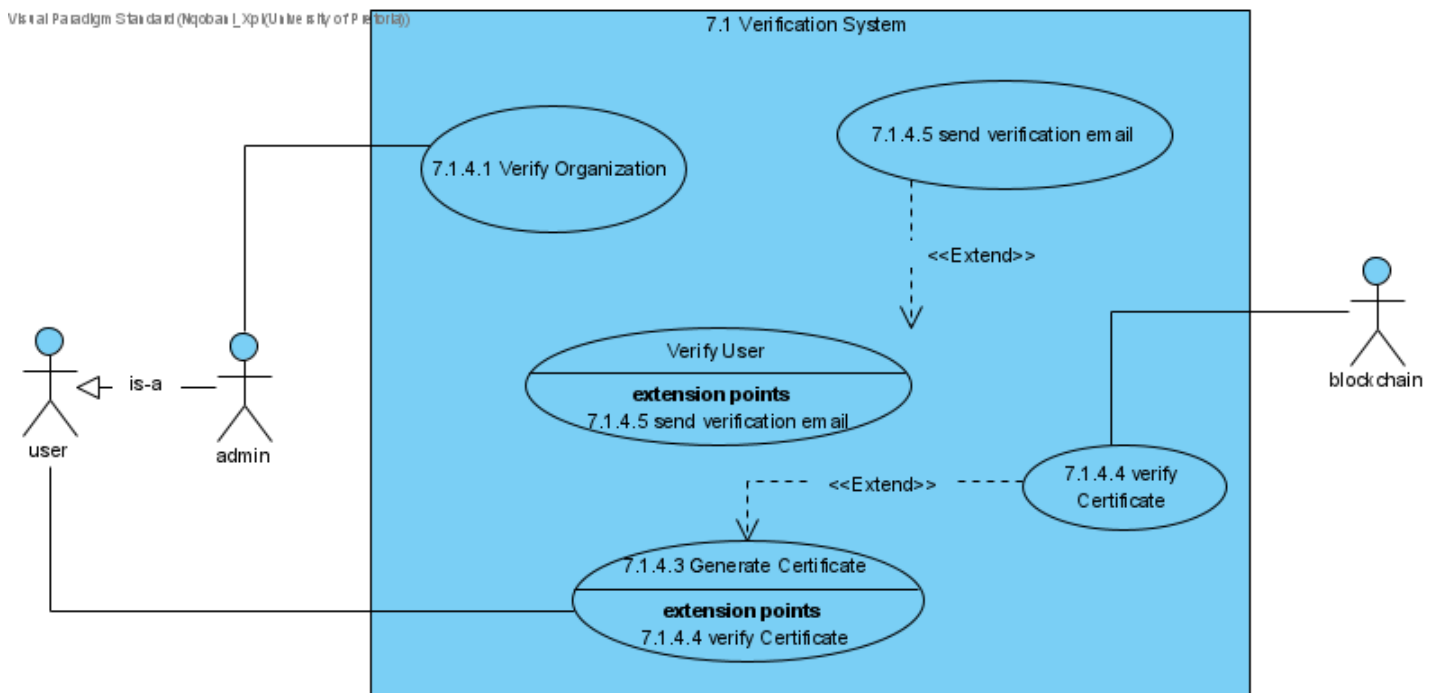


Figure 2: Verification Use Cases

7.1.3 Domain model

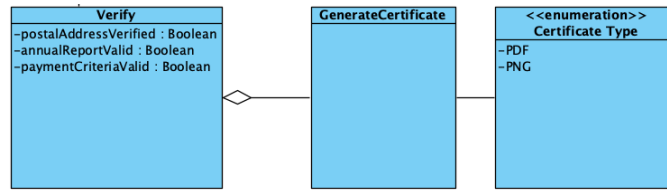


Figure 3: Verification Domain Model

7.1.4 Service contract

7.1.4.1 Verify Organization

This use case is used to verify the authenticity of an organisation before a certificate can be generated for such an organisation.

7.1.4.2 Verify User

The system automatically sends a verification email to any registering user to confirm their registration.

7.1.4.3 Generate Certificate

This use case is used to generate a give a lot verification certificate for an organisation to prove their authenticity.

7.1.4.4 Verify Certificate

This use case will use blockchain to verify the authenticity of a certificate.

7.1.4.5 Send Verification email

This email receives instructions from use case 7.1.4.2 and sends an email to a respective user.

7.1.4.6 Revoke Verification

text here

7.2 Search subsystem

7.2.1 Description

The search system is responsible for searching Give A Lot for the existence of an organisation. If the organisation does not exist within the Give A Lot data archive, the search is expanded to public platforms however all the results returned from public platforms should not be flagged as credible organisations.

7.2.2 Use Cases

The use cases of the search subsystem are shown in Figure 4.

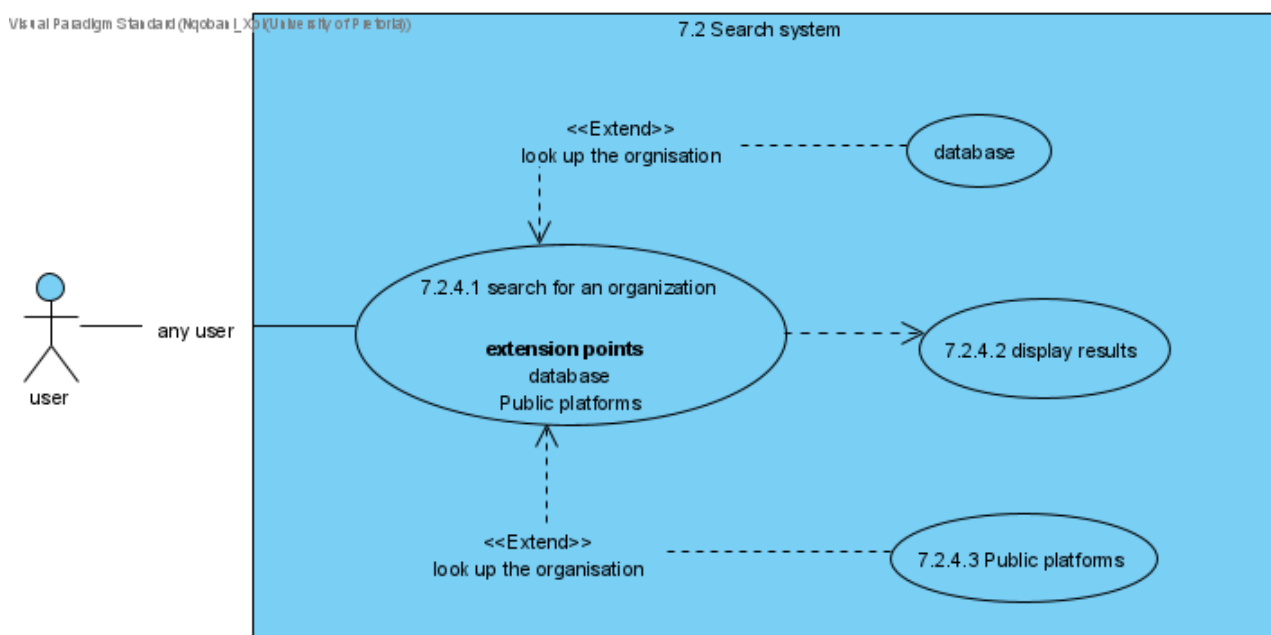


Figure 4: Search Use Cases

7.2.3 Domain Model

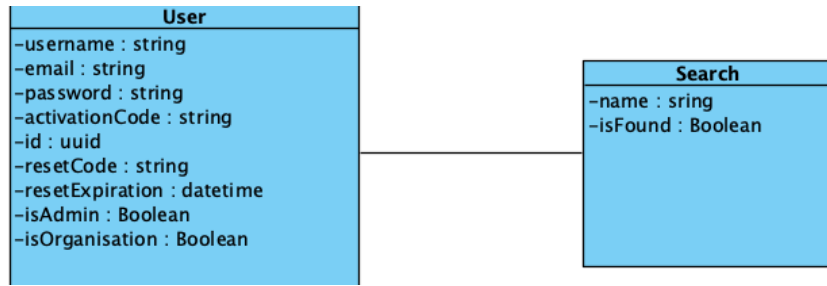


Figure 5: Donation Domain Model

7.2.4 Service Contracts

The service contracts outlined above deals with all aspects related to search. Our system may search public domains for an organisation and it aims to do so without compromising the security of the whole system and it's users.

7.2.4.1 search for an organisation

The search for an organisation use case can be used by all users of the system. The search looks up an organisation by it's name from the Give A lot data archive and returns a result. The search can be expanded to public domains should the system return no result however all the results returned from other public public archives are flagged as unverified with no exception. This use case is triggered by selecting the search box in the system.

7.2.4.2 Display search results

After the user enters the name of an organisation by triggering use case 7.2.4.1 - the results for that search are shown on the screen for the user to interact with. On trigger by clicking on the search box, the display search results use case should show top 5 most searched organisations that a user can click on as they type their organisation of choice.

7.2.4.3 Search public platforms

After the user enters the name of an organisation by triggering use case 7.2.4.1 - if use case 7.2.4.2 does not return any results, this use case is triggered to widen the search into public platforms. All the results returned by this use case are not verified.

7.3 Donation subsystem

7.3.1 Description

The donation subsystem defines how the management of donations, invoicing and the transaction lifecycle are managed, and include actions such as how the payment is verified against the chosen payment gateway. The payment gateway select for the Give a lot system is the Payfast system. Utilizing a online banking has various benefits one of these benefits is the additional layer of security provided to the user's transaction provided by the payment gateway.

7.3.2 Use Cases

The use cases of the donation subsystem are shown in Figure 6.

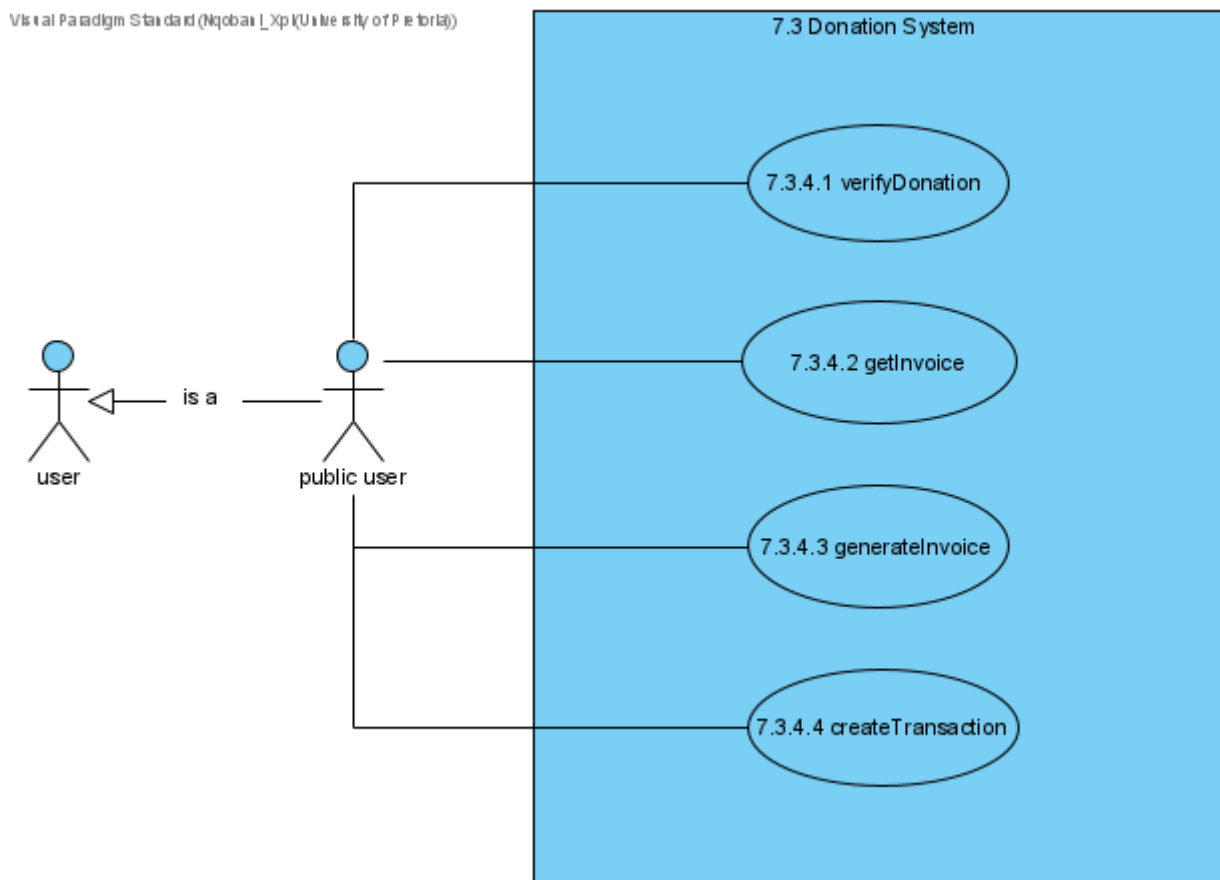


Figure 6: Donation Use Cases

7.3.3 Domain Model

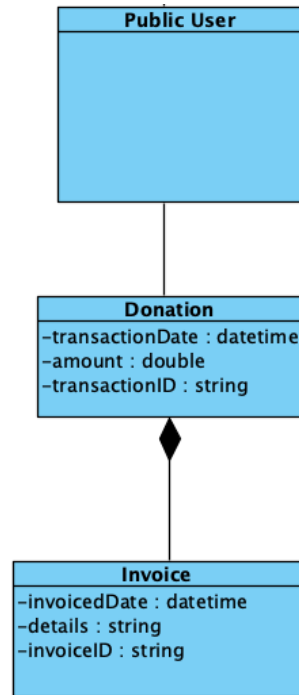


Figure 7: Donation Domain Model

7.3.4 Service Contracts

The service contracts outlined above deals with all aspects related to donations. As our system allows for the donations between parties for a monetary amount, it is critical to ensure that the donation caters for all possible use cases, as well as edge cases as such proper error handling and retry logic would be critical to ensure a good user experience.

7.3.4.1 Verify Donation

A verify donation use case receives an donation ID for a transaction that should be validated. The transaction ID field should be marked as unique with the online banking system. It is essential in validating a transaction, not only to ensure the donation amount is correct but whether the receiving account is correct as well. Upon donation being verified, the use cases in section — and — should be called to generate an invoice for the transaction in question.

7.3.4.2 Get Invoice

This use case is used to retrieve an invoice for a transaction Id. It is important to note that users should only be able to execute this use case for donations where they are the donor. If no security was implemented, it would allow users to retrieve other users invoices, which would provide personally identifiable information such as first names, last names and addresses, which would be a privacy issue.

7.3.4.3 Generate Invoice

This use case should be triggered when the use case 7.3.4.1 is successfully executed. A PDF invoice should be generated and persisted to a storage mechanism for retrieval later by the use case defined in section 7.3.4.2.

7.3.4.4 Create Transaction

Donation in the Give a lot system will make use of online banking .

7.4 User Subsystem

7.4.1 Description

The user subsystem defines how a general public user will make use of the Give A Lot platform by browsing organisations, searching for specific organisations, viewing the details of organisations and checking the verification certificates of organisations.

7.4.2 Use Cases

The use case of the User subsystem is shown below:

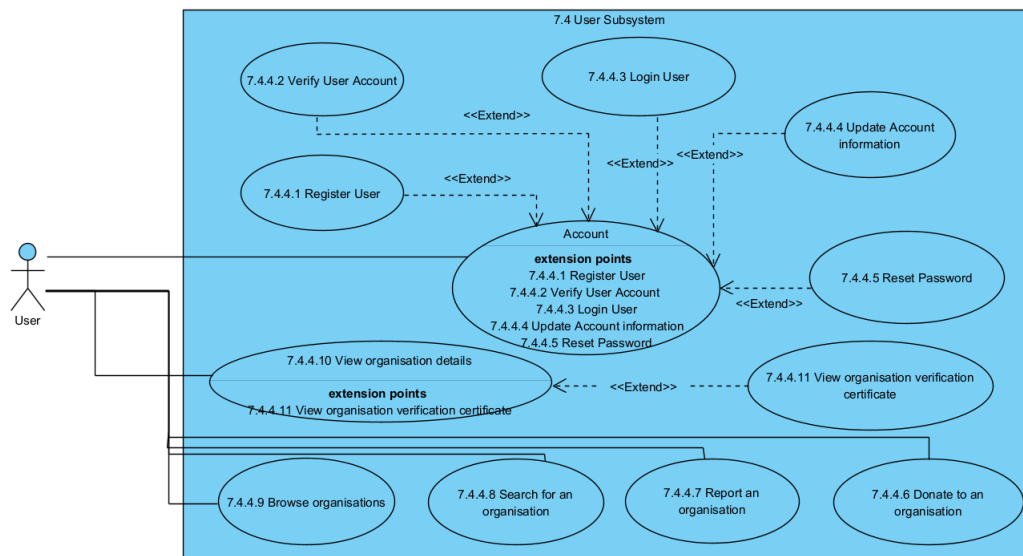


Figure 8: User Use Case

7.4.3 Domain Model

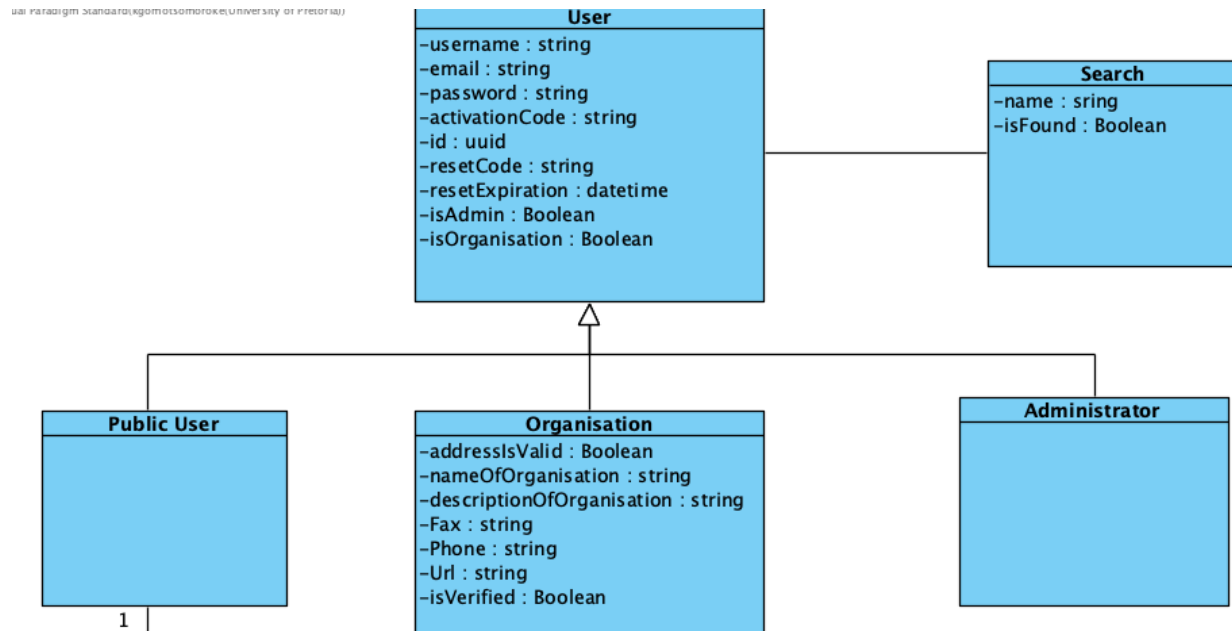


Figure 9: User Domain Model

7.4.4 Service Contracts

7.4.4.1 Register User

The register user use case allows the user to create an account on the Give A Lot platform, said account is required to use many functionalities of Give A Lot. The use case will make use of the following User attributes; Email, Username, First Name, Last Name and Password. When registering the user will enter their details into a form, upon submitting, system will validate that the email and/or username are not already in use in the system

7.4.4.2 Verify User Account

The verify user use case ensures that the person who registered the account has access to the email address provided. The use case will make use of the following User attributes; `isVerified`. Once a user has registered their account, an email will be sent to the email address provided. The email will contain the options to verify the creation of the account or reject the creation of the account. If the user chooses to verify the account, their account status will be updated to verified and they will have more accessibility on the website. If the user chooses to reject the account, the account is removed from the system.

7.4.4.3 Login User

The login user use case requires the user's username/email and password. The system will then validate that the provided username/email is in the system and then validate that the username/e-

mail and password correspond. If the provided credentials are correct, the user will be issued a JWT.

7.4.4.4 Update Account information

The update account information use case allows the user to update the first name, last name, and email address they provided when registering their account. The use case will make use of the following User attributes; First Name, Last Name and Email. No validation is required for the change, and if the user requires a password change, they can make use of the Reset Password use case.

7.4.4.5 Reset Password

The reset password use case allows the user to change their password if they forgot it or want to change it for security reasons. The use case will make use of the following User attributes; Password. An email will be sent to the user's provided email with a link to the change password page which will allow them to change passwords.

7.4.4.6 Donate to an organisation

Please refer to the Donation subsystem.

7.4.4.7 Report an organisation

The report an organisation use case allows the user to report an organisation that they think is being dishonest or the organisation is misrepresenting itself. The user will be redirected to a form where they must provide information such as the name of the organisation, what offense it committed and a description of the offense. This service can only be used by registered users to prevent malicious reports and spam.

7.4.4.8 Search for an organisation

The search for an organisation use case allows the user to search for a specific organisation using a search bar and filters. The user will be presented a search bar with filters with which they can enter the name of the organisation and search for it from the registered organisations on the platform and can use the filters to narrow down the results.

7.4.4.9 Browse organisations

The browse organisations use case allows the user to peruse through the list of organisations registered on the platform, they can use filters to specify the type of organisation they are looking for. The user will be presented with a list of the organisations on the platform, ordered in some way (such as by date added or alphabetically) and will be able to scroll through the list and filter the list to get a list more tailored for what they are looking for.

7.4.4.10 View organisation details

The view organisation use case allows the user to view the details on the selected organisation. The user will be shown the details of the organisation once they select it from the list they were presented, or the results returned from a search. The details show included the organisations name, their chairperson, their contact details, a brief description of the organisation and a link to their verification certificate.

7.4.4.11 View organisation verification certificate

The view organisation verification certificate use case allows the user to view and share the verification certificate of their selected organisation. The user can choose to view the organisation's verification certificate of the organisation profile they are viewing. If selected, the user will be redirected to the certificate and have the option to share the certificate or save a pdf copy of it.

7.5 Admin Subsystem

7.5.1 Description

The admin subsystem defines how a admin user will make use of the Give A Lot platform by managing organisations, searching for specific organisations, updating the details of organisations, changing the verification certificates of organisations and adding or removing organisations.

7.5.2 Use Cases

The use case of the Admin subsystem is shown in figure 10 below:

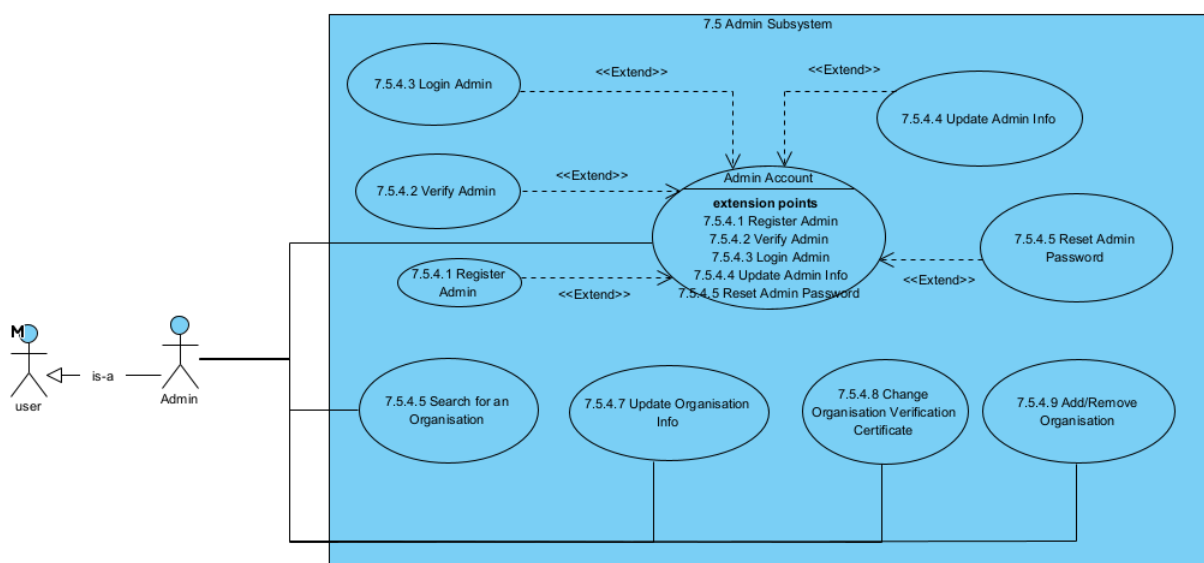


Figure 10: Admin Use Case

7.5.3 Domain Model

use: paraaigbm 31aanaar01egom05somoroke(University of Pretoria)

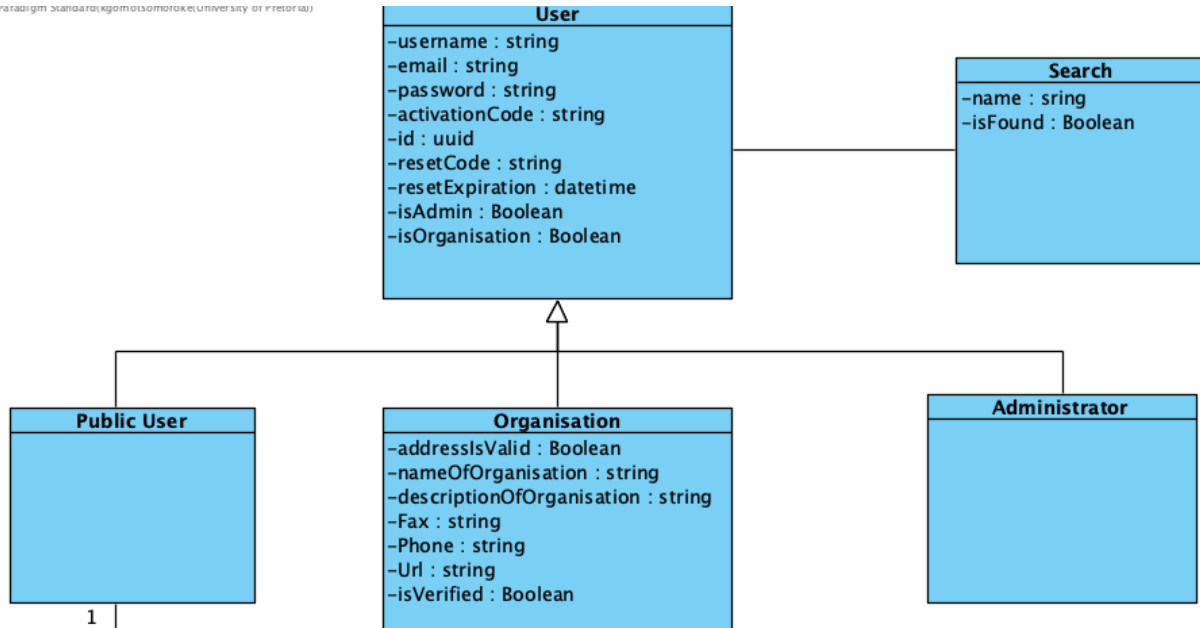


Figure 11: Admin Domain Model

7.5.4 Service Contracts

7.5.4.1 Register Admin

The register admin use case allows the admin user to create an administrative account on the Give A Lot platform, said account is required to manage many of the inner workings of Give A Lot. The use case will make use of the following User attributes; Email, Username, First Name, Last Name and Password. When registering the user will enter their details into a form, upon submitting, system will validate that the email and/or username are not already in use in the system.

7.5.4.2 Verify Admin

The verify admin use case ensures that the person who registered the account has access to the email address provided and requires another admin to authorize this admin account. The use case will make use of the following User attributes; `isVerified`. Once an admin has registered their account, an email will be sent to the email address provided. The email will contain the options to verify the creation of the account or reject the creation of the account. If the admin chooses to verify the account, their account status will be updated to verified and they will have more accessibility on the website. If the admin chooses to reject the account, the account is removed from the system. If the account is verified, the application will appear on the admin homepage, which then needs to be approved or rejected by an admin.

7.5.4.3 Login Admin

The login admin use case requires the admin user's username/email and password. The system will then validate that the provided username/email is in the system and then validate that the username/email and password correspond. If the provided credentials are correct, the user will be issued a JWT.

7.5.4.4 Update Admin Info

The update admin information use case allows the admin user to update the first name, last name, and email address they provided when registering their account. The use case will make use of the following User attributes; First Name, Last Name and Email. No validation is required for the change, and if the user requires a password change, they can make use of the Reset Password use case.

7.5.4.5 Reset Admin Password

The reset admin password use case allows the admin user to change their password if they forgot it or want to change it for security reasons. The use case will make use of the following User attributes; Password An email will be sent to the admin user's provided email with a link to the change password page which will allow them to change passwords.

7.5.4.6 Search for an Organisation

The search for an organisation use case allows the admin user to search for a specific organisation using a search bar and filters. The admin user will be presented a search bar with filters with which they can enter the name of the organisation and search for it from the registered organisations on the platform and can use the filters to narrow down the results.

7.5.4.7 Update Organisation Info

The update organisation info use case allows the admin user to change/update the details on the selected organisation. The admin user will be shown the details of the organisation once they select it from the list they were presented, or the results returned from a search. The details that can be changed include the organisations name, their chairperson, their contact details, a brief description of the organisation and a link to their verification certificate.

7.5.4.8 Change Organisation Verification Certificate

The change organisation verification certificate use case allows the admin user to change the verification certificate of their selected organisation. The admin user can choose to change the level the organisation's verification certificate, of the organisation profile they are viewing. They can either increase or decrease the level of the certificate, or revoke the certificate in its entirety. If the certificate has been revoked, there is an option to reinstate it.

7.5.4.9 Add/Remove Organisation

The add/remove organisation use case allows the admin user to add an organisation to or remove an organisation from Give A Lot. The admin user can choose to manually add an organisation to the Give A Lot platform or remove an organisation that is already listed on Give A Lot.

8 Quality Requirements

The quality requirements are the requirements around the quality attributes of the systems and the services it provides. This includes requirements like maintainability, performance, scalability, security, usability, and testability requirements.

8.1 Maintainability

Amongst the most important quality requirements for the system is maintainability. It should be easy to maintain the system in the future. To this end

- future developers should be able to easily understand the system,
- the technologies chosen for the system can be reasonably expected to be available for a long time,
- and developers should be able to easily and relatively quickly
 - change aspects of the functionality the system provides, and
 - add new functionality to the system.

8.2 Scalability

The future scale-out of the application looks as follows. It is hoping to cover the entire globe hence it is of such importance that the system can allow for both vertical and horizontal scaleout to ensure that demand can be met while keeping costs low.

8.3 Security

Security is the most important quality requirement due to the high security nature of the generated certificates. Special security measures must be deployed not only for traditional means such as authentication against the back-end via an OAuth2 implicit grant but other means of securing a High value certificate such as those provided by Givealot.

8.4 Testability

All services offered by the system must be testable through:

- automated unit tests testing components in isolation using mock objects, and
- automated integration tests where components are integrated within the actual environment.

In either case, these functional tests should verify:

- that the service is provided if all pre-conditions are met (i.e. that no exception is raised except

if one of the pre-conditions for the service is not met), and

- that all post-conditions hold true once the service has been provided.

In addition to functional testing, the quality requirements like scalability, usability, performance and so on should also be tested.

8.5 Usability

The system should be intuitive and efficient to use where computer literacy can be assumed, buy the time it takes users to search for an organisation should be kept to a minimum. Error messages should be self-explanatory and as much as possible of the input validation should be done on the client.

9 Traceability Matrix

		Requirements																								
		R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15	R16	R17	R18	R19	R20	R21	R22	23	24	25
Use cases	7.5.4.1	X						X																		
	7.5.4.2			X						X																
	7.5.4.3		X						X																	
	7.5.4.4				X						X															
	7.5.4.5					X						X														
	7.5.4.6																									
	7.5.4.7																									
	7.5.4.8																									
	7.5.4.9																									
	7.5.4.10																									
	7.5.4.11																									
	7.5.4.12																									
	U7.1.4.1																				X					
	U7.1.4.3																					X				
	U7.1.4.4																						X			
	U7.1.4.5																								X	
	U7.2.4.1													X												
	U7.2.4.2													X												
	U7.2.4.3														X											
	U.3.7.4.1															X										
	U.3.7.4.2																X									
	U.3.7.4.3																X									
	U.3.7.4.4															X										

Table 1: Requirement traceability matrix for Give A Lot