



Computer Science and Engineering

Good Deed Project

System Requirements Specification (SRS)

Version 1.4

Document Number: SRS-003

Project Team A14

Project Team Members:

Jonah Sherman, jds854

Kerem Ulcay, ku350

Sofia Machado Lopes, sml856

Timur Gordon, tbg252

REVIEW AND APPROVALS

Team Members	Function	Date	Signature
Jonah Sherman	Author, Reviewer	11-19-20	JDS
Sofia Machado Lopes	Author, Reviewer	11-19-20	SML
Timur Blair Gordon	Author, Reviewer	11-19-20	TBG
Kerem Ulcay	Author, Reviewer	11-19-20	KU

REVISION LEVEL

Date	Revision Number	Purpose
10/8/2020	Version 1.0	Create SRS Analysis
10/15/2020	Version 1.1	Revise SRS Analysis
10/20/2020	Version 1.2	Update SRS Analysis
11/01/2020	Version 1.3	Revise SRS Analysis
11/17/2020	Version 1.4	Complete SRS Analysis

TABLE OF CONTENTS

DOCUMENT PURPOSE	4
Purpose	4
1.1.0 Purpose of SRS	4
1.1.1: Intended audience	4
INTRODUCTION	4
2.0.1 The system to be produced	4
2.0.2 The results of the needs	4
2.0.3 Description of system application	5
2.0.4 Product perspective	5
Scope	6
Identification	6
Bounds	6
Objectives	7
2.4.1 Project priority	7
2.4.2 Project lifecycle	7
2.4.3 Initial deliverable	7
Context Diagram	8
2.5.1 Website Diagram	8
Additional Descriptive Items	9
2.6.1 Product functions	9
2.6.2 User characteristics	9
2.6.3 Constraints	9
2.6.4 Assumptions and dependencies	10
2.6.5 Requirements subsets	10
GLOSSARY	10
REFERENCE DOCUMENTS	11
BUSINESS REQUIREMENTS	11
Technology	11
Economics	11
Regulatory and Legal	11
Market Considerations	12
Risks and Alternatives	12
Human Resources and Training	12
USER REQUIREMENTS (DESCRIPTIVE FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS)	12
6.1 Functional Descriptive Detailed Requirements	12

SYSTEM ARCHITECTURE	14
DETAILED SYSTEM REQUIREMENTS – USE CASES	15
Requirement Use Cases	15
Use Case Diagrams	16
SYSTEM MODEL (UML)	19
Static - Class Diagrams	19
Dynamic - Behavioral Models	20
EVOLUTION OF THE SRS	20
RATIONALE	22
NOTE	22
APPENDICES	22
System Test Plan Requirements	22
Qualification Provisions	22
Requirements Traceability	22
Schedule Tracking	24
Defect Tracking	26
13.6 Dictionaries	28
INDEX	33

1. DOCUMENT PURPOSE

1.1 Purpose

1.1.0 Purpose of SRS

The purpose of this document is to report on the progress and provide a detailed overview of our product, its parameters, and what we intend to achieve with it. This SRS will define all the project's requirements with respect to the system, its target audience, its user interface, and both the hardware and software needed.

1.1.1: Intended audience

The intended audience of the SRS is Professor Strauss, a client, or any designer or developer who is to assist in the software delivery lifecycle process.

2. INTRODUCTION

2.0.1 The system to be produced

The system will establish connections between users who are willing to commit to good deed projects. This will be accomplished through providing a website accommodating the needs of the users.

2.0.2 The results of the needs

Need: The current platforms are offering solutions. However, they are not very effective as it is not a prioritized aspect of the product that they offer. Consequently, a website devoted for this specific purpose is required.

Explanation: Social media giants are losing the trust that they have accumulated. Tik Tok is associated with stealing personal information, Facebook is responsible for Brexit in the eyes of many British, Twitter was prominent in the Arab Spring, Instagram is owned by Facebook. This is not the atmosphere you should be in when you are discussing a more effective way to clean a local dog shelter, looking for a cheaper way to fund a mountainous village's solar panels or donating your clothes to a recent earthquake centre.

Need: Other NGOs are accustomed to get the potential contributors to donate through their

website by providing links to their websites over Instagram, Facebook or Twitter. Although Facebook accepts donations through itself as well, numerous organizations including the United Nations still follow the normal path. However, there is a website maintenance fee associated with every website developed to service this sector. Every NGO, regardless of their size, has to have a website which can securely have money transfers, provide a friendly and trustworthy user interface and support a community connection for recruiting which can be accomplished through making information more available. The era of sending emails about donation invitations should be over.

Explanation: In its shell, the idea is: there are thousands of NGOs doing this, why not put a roof over them and connect them with the keyword “help”? This would relieve them from the stress of website maintenance, money transfer, transparency, informing the public and scaling their operations through recruits. If at the end of the day, they can donate to the organisation due to the savings it provided, we can thank them twice. Once for the “help” we achieved together and once for the “help” they give us.

2.0.3 Description of system application

Website. Combines all of the NGO websites to one. For “helpers”, people who are willing to help somehow, they can match their skills with the tasks that are needing help to be done. If I know about IoT devices for example, then I should be using my time with something more technical rather than carrying boxes. This is how the website actually creates value. By using the resources more efficiently and actually establishing a connection with the helper, this service has the potential to deliver a good deed more efficiently and in increased scale. In times of COVID, people are thinking less about natural disasters as we are already living one. However, climate change is continuing and the hurricanes will only get stronger. The question is: “How can we respond to such disasters more efficiently?” We believe that this service can be a hotspot for correct information, efficient distribution of locals for tasks and creating requests for help in a time of crisis.

2.0.4 Product perspective

Change is part of development. If there is a solution which would make things better for this service, it will be used. Consequently, at this stage it is early to give any definite answer.

2.1 Scope

The users should be able to join a specific group under an organization to offer their specific talents to the organization. Social entrepreneurs and volunteers involved in projects should be able to communicate through the project page that updates constantly with respect to input by the users of the project. A social entrepreneur would be the person creating a project. A volunteer would be a person who offers funds or labour. The users should voice their opinions freely without any fear of authority or punishment but with recognition that everyone is working towards a common goal. The users should not be discouraged if they fail in a project. Instead the users should be relocated. Good Deed will offer good options based on highlighted skills of a specific volunteer or social entrepreneur. This will include serving volunteer's or social entrepreneur's interests as well as serving more efficiently with their limited time. Good Deed recognizes the value of time and decreasing online presence costs for NGOs. For this offered decrease in costs and increase in impact, Good Deed expects donations to cover server costs. Good Deed will strive to offer an option of donations for the volunteers through the site, to a project or directly itself.

2.2 Identification

This document is identified as follows:

Good Deed Platform SRS, SRS-003, Version 1.4.

2.3 Bounds

- Growth should be sustainable, trust is a major issue that we are tackling to get an advantage against facebook. Uncontrolled development is not welcomed.
- Diverging all of the traffic of NGOs into one website will have server requirements increasing in size and speed. Performance under those conditions are important. It should be carefully programmed for lower cellular usage as well. This would facilitate its usage in cases of emergencies.
- Service should maintain to be better than the competition. After establishing trust, it would be a shame to lose the competitive advantage.
- Failure of the project could be associated with the failure of the idea rather than the implementation.

2.4 Objectives

2.4.1 Project priority

The project priority is to have a platform where users can post about and join others partaking in community service. The project aims to produce a platform that supports the creation and execution of social impact projects.

2.4.2 Project lifecycle

We will use the iterative method while developing the website. Therefore, the project will be developed incrementally, and an initial product will be launched prior to the development of the final product. The most fundamental functionalities will be included in the initial product, such that the initial product successfully fulfills the project priority. Once the initial product is launched, the system requirements will be reevaluated in accordance with the feedback received from the client, the system will be redesigned if necessary, and will be developed further to meet the client's requirements.

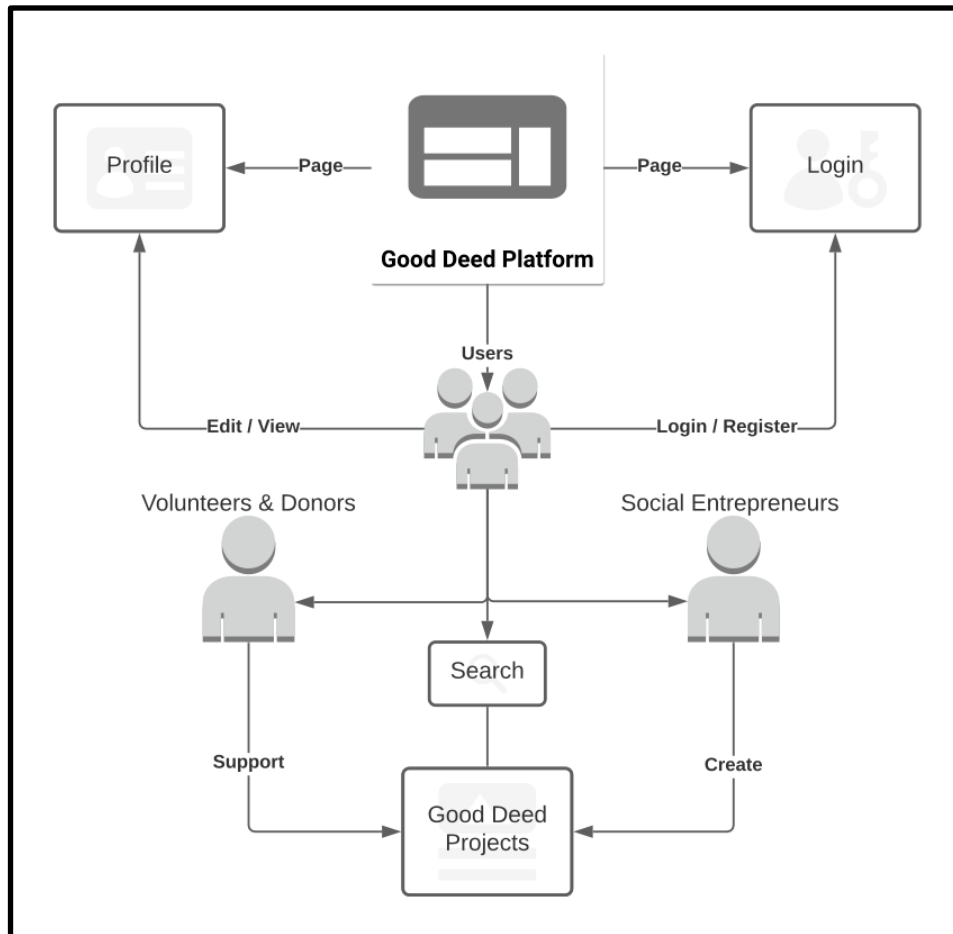
2.4.3 Initial deliverable

The product will roll out in two stages. The initial product / prototype will be designed, implemented and launched prior to the design of the final product. The feedback received from the initial product will be incorporated into the design of the final product. The anticipated milestones and dates for product development and launch are as follows:

Completion of SRS	<i>19 November 2020</i>
Initial product design	<i>1 January 2021</i>
Initial product implementation	<i>1 February 2021</i>
Final product design	<i>1 March 2021</i>
Final product implementation	<i>1 April 2021</i>
Final product evolution & maintenance	<i>1 May 2021</i>

2.5 Context Diagram

2.5.1 Website Diagram



The website diagram above provides a high level overview of the good deed platform. As the diagram illustrates there are 5 web pages that are a part of the website. These are the "Home", "Login/Register", "Profile", "Search", and the "Good Deed Project" pages. Users are thus presented with five functionalities. They can login and register to the platform. Users can also edit their profile and view other users profiles. The main emphasis of the platform is on developing good deed projects, and as such users are also able to create and support said projects. Finally, the search functionality allows for users to find other users and good deed projects to collaborate with.

2.6 Additional Descriptive Items

2.6.1 Product functions

- User account functionalities: The product will feature functionality for users to create an account on the platform and to partake in the platform with their accounts. Therefore the product will provide functions to log in to the platform and to register as a user.
- Profile functionalities: Users will be able to edit information on their profile and view other users' profiles. The profile function will also automatically generate content for the user's profile, as users partake in good deed projects. The profile will be composed of the user's areas of interest, and previous projects the user has contributed to.
- Project creation functionalities: Users will be able to create good deed projects, add a description to the project being created, select a category for the project, and add information about what sort of support (monetary, volunteers) is requested for the successful execution of the project.
- Project support functionalities: The platform will provide functions that allow users to support the project by volunteering or donating money, or spreading the word.
- Search functionality: The platform will have a search page in which a user can view and search through a ledger of projects and users, with the help of filters.

2.6.2 User characteristics

The user does not need any previous experience, knowledge, or technical expertise. The website will be user friendly and will be categorised in terms of area of impact (education, environment, sustainability, health, etc.), from a view of either a potential participant or a potential benefactor. Throughout the platform, sections will be labeled clearly and simple road maps will be developed ensuring that users can reach their website destination easily.

2.6.3 Constraints

The constraints to ensure the successful and secure completion of the projects are as follows:

- The projects must be developed using well known coding languages and libraries to minimize potential vulnerabilities and bugs related to the languages being used.
- The donations being made to projects must be limited and audited in order to avoid potential money laundering issues.
- It is important that boundaries to enter this website should be low. That would mean a 7

year old phone should be able to process the website. We need backward compatibility in order to avoid a case where a person with an old phone, potentially from a less financially fortunate background, would not be kept outside of the system.

- It is critical that emergency response mode functions without errors and manages to go through stress tests powerfully. This component is vital for saving lives.
- Some deadlines are important for grades, but not for the course of the project. It is more important to have a good product before initial beta testing.
- International fund transfers are harder in developing countries as the cost of transfer increases. There has to be an international partnership to facilitate the transfer of funds by a Turkish family in United States of America, to a city in Turkish Republic.
- This is not kickstarter.com , there should not be projects which target personal capitalist interests. This is different from developing a product to serve for NGOs.

2.6.4 Assumptions and dependencies

We assume that the platform will not be accessed through web browsers on mobile devices. Furthermore we assume that users will access the platform with up-to-date web browsers. If such assumptions fail, then the requirements must be modified to accommodate said methods of accessing the platform.

2.6.5 Requirements subsets

The subset of requirements that apply to the initial product are as follows. Of the five functionalities mentioned above in section 2.6.1, a subset of three will be implemented in the initial prototype, and the rest will be added in subsequent iterations. The three are user account, project creation and project support functionalities. As such, of the five pages the website context diagram illustrates in section 2.5, only the home page, the login/registration page, and good deed project pages will exist. This subset of requirements suffice for the project priority to be met and will allow the team to gather adequate feedback from the client.

3. GLOSSARY

CSS: CSS is a coding language, and will be used for programming the front end of the website.

Git repository: A git repository is a directory on Github, which team members can access and collaboratively develop software by coding.

HTML: HTML is a coding language, and will be used for programming the front end of the website.

Javascript: Javascript is a coding language, and will be used for programming the front end of the website.

Python: Python is a coding language, and will be used for programming the backend of the website.

React: React is a Javascript library that allows for additional features to aid front end development.

4. REFERENCE DOCUMENTS

A14, Project_Proposal.pdf, SRS-000, Version 1.0, 15 September 2020

5. BUSINESS REQUIREMENTS

5.1 Technology

For development purposes, required tools include basic backend and frontend web development tools. We plan to use community versions Javascript, React, CSS, HTML and Python to develop the website. Additionally, a Git repository will be used for collaboration purposes. Data servers will be necessary to deploy, run, and maintain the website efficiently.

5.2 Economics

For development purposes, not much cost is involved. The development tools cited in 5.1 are free to use and open source. We plan to use community versions Javascript and Python to develop the website. A Git repository will be used for collaboration purposes. Costs arise in the deployment and maintenance of the website. Data servers will need to be purchased in the capacity necessary to maintain a sizable social platform. All costs associated with purchasing, running, and maintaining said data servers can be expected. Additional hires may need to be made for website maintenance purposes, as well as website monitoring (see 5.3).

5.3 Regulatory and Legal

We, as a company, will strive to strengthen social well-being through our website. In order to ensure our platform positively impacts the communities which it reaches, we will establish a

strict set of community guidelines which postings on the website must adhere to. Furthermore, we expect a great diversity of users on the website. Specifically, we expect users from a variety of nations. We will monitor the postings on the website to make sure they follow both our own community guidelines as well as the legal guidelines for the respective country or municipality.

5.4 Market Considerations

Our company is a not for profit organization; our market involves individuals who can freely use our platform to incite positive change. Therefore, there are not significant limitations to our potential market size. The types of postings individuals can organize or join is only limited by the guidelines stated in 5.3.

5.5 Risks and Alternatives

Risks during development include individual software issues leading to loss of code. Additionally, there is always a risk that a bug could work its way into the repository early in development. These development risks will be remedied by running tests often after changes, as well as making and testing changes locally before adding to the shared repository. After the website is deployed, risks include data server issues resulting in the site crashing and data being lost. These can be remedied by the hiring of additional employees to ensure the proper functioning of servers (see 5.2).

5.6 Human Resources and Training

Minimal human resources and training would be required given the small group this project requires. Training and human resources will only become a necessity upon the hiring of additional staff in the future (see 5.2)

6. USER REQUIREMENTS (DESCRIPTIVE FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS)

6.1 Functional Descriptive Detailed Requirements

1. The product shall feature functionality for users to create an account on the platform and

- to partake in the platform with their accounts.
- The product will provide functions to log in to the platform and to register as a user.
 - By creating an account, Good Deed will create an entry for the database and start collecting data about the user.
 - No matter who they are, if they can provide a legitimate identification, then they will be able to have a profile. There will not be a case where a Syrian refugee trying to aid his people will be rejected for not having a Turkish passport.
2. Users shall be able to edit information on their profile and view other users' profiles.
- The profile function will also automatically generate content for the user's profile, as users partake in good deed projects.
 - The profile will be composed of the user's areas of interest, and previous projects the user has contributed to.
3. Users shall be able to create good deed projects, add a description to the project being created, select a category for the project, and add information about what sort of support (monetary, volunteers) is requested for the successful execution of the project
- This will be achieved by using an updated version of the HTML which should be able to run with lower data consumption as well. It is critical and essential that the website can manage to serve when a natural disaster occurs.
 - Strict categorization of the projects would allow that level of specific homepage. As long as server costs are maintainable, higher levels of data related to the individual will be stored to increase the chances of increasing in magnitude or efficiency of the performance of the users.
 - Projects will be created by the users or an NGO. If the project is not well categorized, it will be less likely to match with a user. Therefore, projects will be expected to have at least a proper abstract. In order to achieve this, a proper guide on how to create a project will be shared with the users in a user friendly manner.
4. Every created project of the Good Deed shall spread through other social media platforms flawlessly, supporting various fund transferring options and offering your labour.
- Good Deed will use all conventional money transferring options of the online world. Unless the transfer option's charges are unacceptable, if someone wants to donate, they will not be discouraged from carrying their good deed by any means.
 - Good Deed will efficiently allocate your labour to the project that fits the most to the skills unless you want to do something completely different. This will be achieved through the delivery of potential interesting projects selected through the skills of the

subject, to be posted on the home of the subject.

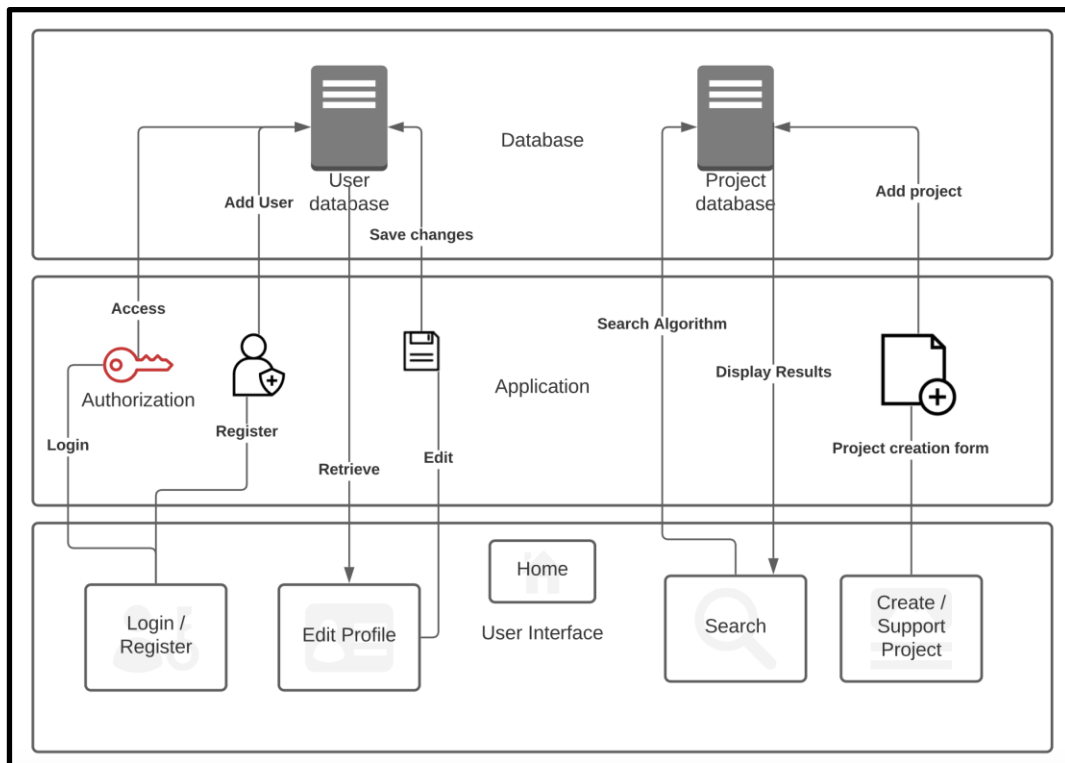
- Currency fluctuations will be prevented by immediate conversion to a hedge fund of globally established and stable currencies. Overnight fluctuations in valuation of a developing country's currency, especially during a time of crisis, is critically high. Good Deed will not let funds gathered to be melting in its hands due to mismanagement and slow pace in response.

5. The platform shall have a search page in which a user can view and search through a ledger of projects and users, with the help of filters.

- Through the definition of filters, a person's profile will be updated through the database to reevaluate the interests of the person.
- Keeping a log of interest and updating them accordingly would ameliorate the chances for creating a self-tailored homepage experience.

7. SYSTEM ARCHITECTURE

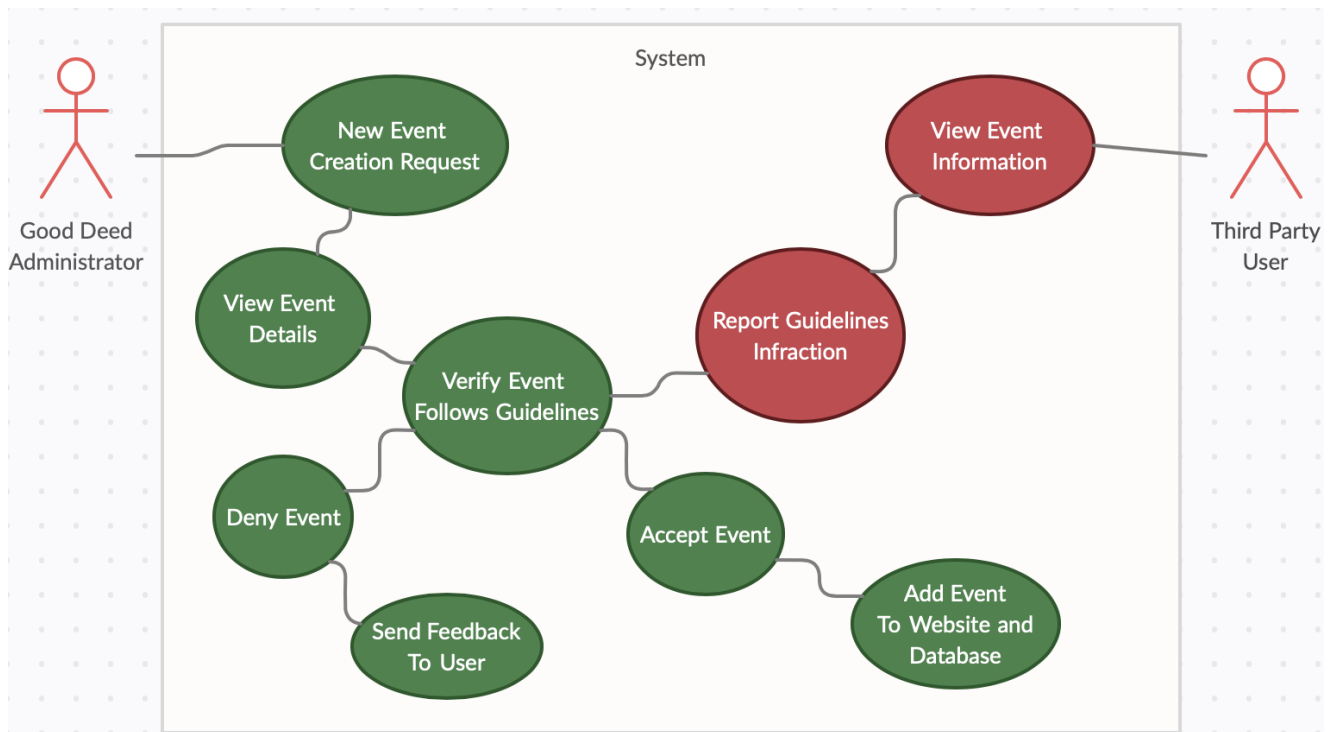
The system has a three-tiered architecture, these are respectively the database, application, and user interface tiers. The overall system architecture diagram below shows how components in each tier interact with each other. The data stored in the database will be in two domains: the user data and the project data. It is worth noting that each user will also store a pointer to the address of project data that the user has participated in. Similarly, projects will hold pointers to addresses of users which have contributed to the said project. Otherwise, the two classes of data will be used by different functionalities of the platform. The application tier is where the processes to support the functions provided in the platform occur. This includes algorithms for authorization for user login, and a search and indexing algorithm for the search functionality. Finally, the user interface tier includes components that the users of the platform will be able to interact with. These tiers and the processes between them are illustrated below.

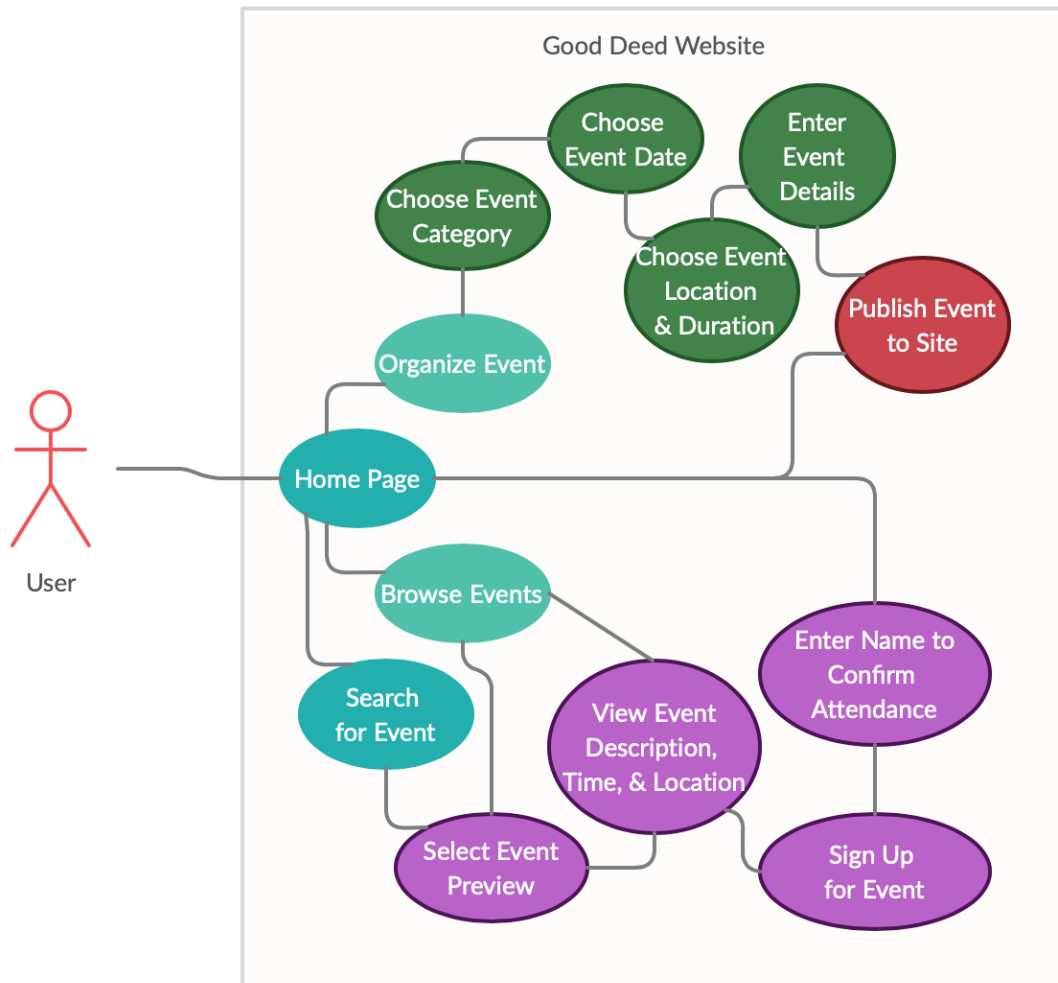


8. DETAILED SYSTEM REQUIREMENTS – USE CASES

8.1 Requirement Use Cases

8.1.1 Use Case Diagrams



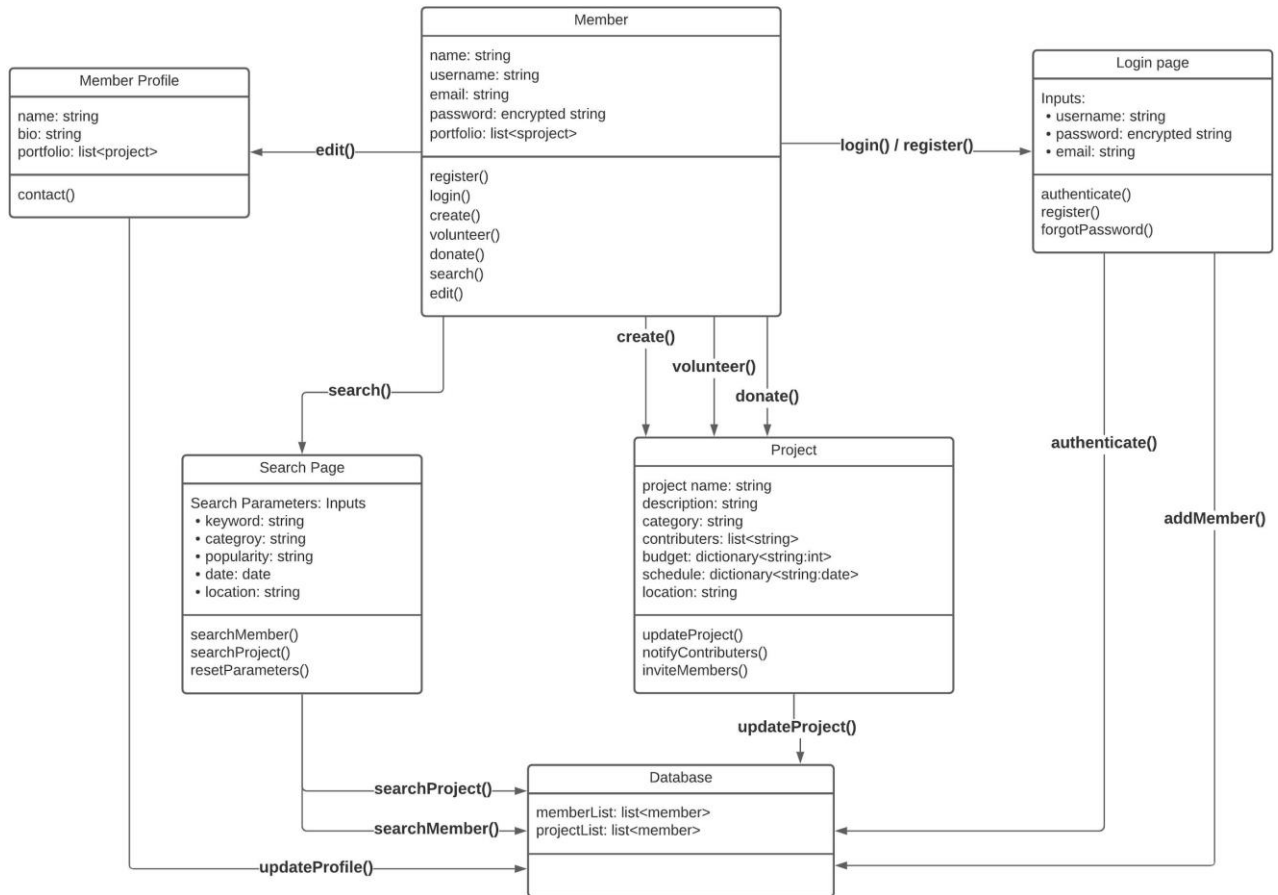


8.1.2 Use Case Descriptions

Use Case Name		
Description	This is the envisioned usage flow for the website. Users will log onto the home page. They will then have the option to organize an event or browse events. There will also be an option to search event keywords or locations. From there, they will follow the necessary criteria to create their event or join events of their choosing. Upon finishing creation or joining, the user is given a confirmation notice/email and returned to the home page.	
Pre-Conditions	Anyone can browse the site. Must have email in order to participate.	
Flows	Basic or Normal Flows	1. Choose creation or join option 2. Enter new event info or choose event to join 3. Finish task, navigate back to home page
	Alternative Flows	1. Navigate back to event browser without joining 2. Choose to browse events after specific search 3. Cancel and navigate back to home page anytime during event creation
Post Conditions	Verification of event creation or participation sent to valid email. Additional email will be sent in build-up to the event to confirm attendance.	
Special Requirements	Organizations, rather than individuals, can contact Good Deed in order to set up a priority event that will reach more people.	
Extension Points	Possible need for verification system to allow reputable organizations to reach a broader audience.	

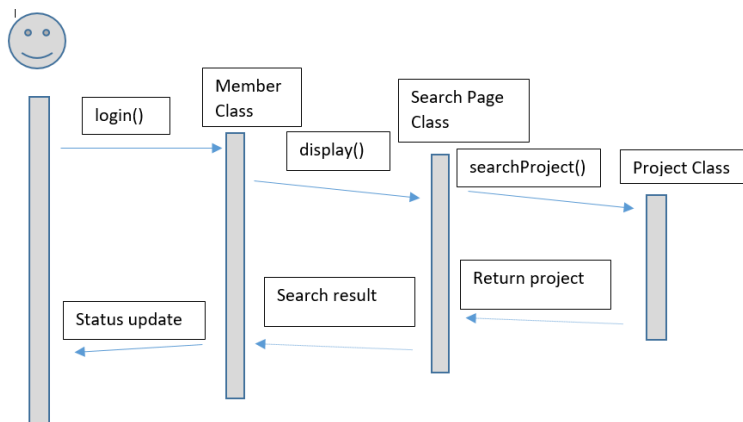
9. SYSTEM MODEL (UML)

9.1 Static - Class Diagrams

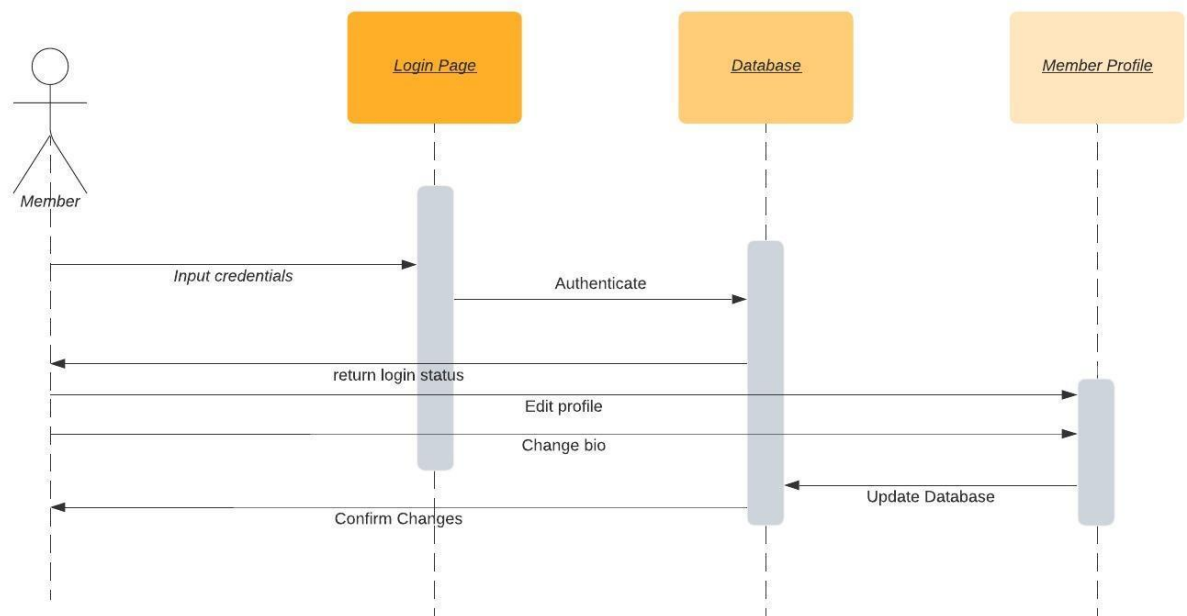


The class diagram above displays the objects that are involved in the good deed platform. The main classes at play are the member and project classes. Once logged in, the members can display pages on the website, and through these pages can create, search for, and contribute to projects. The members also have public profiles in which the good deed projects they are involved in are displayed, and can search for other members on the platform, to contact them and collaborate with them on good deed projects.

9.2 Dynamic - Behavioral Models



This would be how a person pairs with a project of interest. Then there could be a difference with display function once the project is found. Then there could be involvement of createProject(params) which would mean a new project because there is not a project that fits her/his needs. Similarly, there could also be a donate or volunteer component regarding a project of choice.



10. EVOLUTION OF THE SRS

Until the design, development and implementation stages of the project are initiated, the SRS will evolve through additions to the document in accordance with the deadlines provided in the syllabus of the course. As the team gathers feedback relating to the recently added portions, the document will be revised. Once the system is being designed in accordance with the SRS, if there appear to be any changes to be made, the SRS will be revised. Furthermore, if the project team determines any shortcoming or deficiencies in the SRS in future stages of the project lifecycle, the SRS may be evolved further.

Formal Change Process (anticipated)*Version 1.0 - 8 October 2020*

The SRS document has been created, and sections 1-5, 10, 13.4 and 13.5 have been generated.

Version 1.1 - 15 October 2020

The completed components of the SRS will be reviewed. The feedback received from the course instructor or the course TA will be analysed by the project team and necessary changes will be incorporated into the previous version of the document.

Version 1.2 - 20 October 2020

Sections 6, 7 and 8 will be completed by the project team.

Version 1.3 - 1 November 2020

Sections 6-8 of the SRS will be reviewed. The feedback received from the course instructor or the course TA will be analysed by the project team and necessary changes will be incorporated into the previous version of the document.

Version 1.4 - 17 November 2020

The entire SRS document will be completed. The appendices will be updated and any relevant documents such as the project management plans will be referenced within the document. Shall changes to the initial project idea occur, the SRS will be updated accordingly to accommodate the necessary changes as discussed by the project team.

Future Changes

As the project team gathers feedback from the target audience of the project and the course instructor, the SRS might be subject to change. Since the project development process will follow

an iterative model, prototypes will be made and the SRS will be updated as the project progresses. Any changes made will be documented in the formal change process.

11. RATIONALE

To be included at a later date

12. NOTE

To be included at a later date

13. APPENDICES

13.1 System Test Plan Requirements

The testing process will involve automated creation of events, signing up for events, and admin actions. The creation of events test will be performed in both small and large quantities; i.e. we will run a verification and stress test on the system. The system should be able to handle lots of users worldwide creating events nearly simultaneously. The signing up for event function will be similarly tested. Events have the potential for lots to attend them, and therefore need to be rigorously stress tested to verify the site can handle large volume. Finally, the administrator actions such as verifying events or reporting guidelines infractions will be tested for functionality.

13.2 Qualification Provisions

Each member of the team was assigned certain sections of this document to complete. Team members are responsible for reviewing their completed sections. Afterwards, any section that a team member does not do, that team member must review. This will ensure that each section is correct, unambiguous, and complete. A team-wide walkthrough and inspection will also take place. The cross-verification will also allow for team members who did related sections to check for consistency and stability. If a team member can not verify all of the aforementioned attributes of a section during review, the team member who needs to modify the section will be contacted.

13.3 Requirements Traceability

Each section makes appropriate references to previous documents where needed. Also each team member has an in-depth understanding of each section and can add references wherever deemed necessary in the future. This document is reviewed in its entirety after each iteration, therefore each section is defined by its most recent version number of the SRS. This document can be referenced in any future document by citing the SRS version and section number.

13.4 Schedule Tracking

Artifact or Deliverable	Who (individual and team)	Estimated	Actual	Difference
SRS – Business Domain	Jonah Sherman	2 hours	1 hour	1 hour
SRS – Business Domain	Sofia Machado Lopes	2 hours	½ hour	1.5 hour
SRS – Business Domain	Timur Blair Gordon	1.5 hours	½ hour	1 hour
SRS – Business Domain	Kerem Ulcay	2 hours	1.5 hours	½ hour
	Summary for entire team	7.5 hours	3.5 hours	4 hours

Artifact or Deliverable	Who (individual and team)	Estimated	Actual	Difference
SRS – Requirements	Jonah Sherman	1.5 hours	1 hour	½ hour
SRS – Requirements	Sofia Machado Lopes	1.5 hours	1.5 hours	0 hour
SRS – Requirements	Timur Blair Gordon	1.5 hours	1.5 hours	0 hour
SRS – Requirements	Kerem Ulcay	1.5 hours	1 hour	½ hour
	Summary for entire team	6 hours	5 hours	1 hour

Artifact or Deliverable	Who (individual and team)	Estimated	Actual	Difference
SRS – Analysis - Complete	Jonah Sherman	2 hours	1 hour	1 hour
SRS – Analysis - Complete	Sofia Machado Lopes	2 hours	½ hour	1.5 hour
SRS – Analysis - Complete	Timur Blair Gordon	2 hours	2 hours	0 hour
SRS – Analysis - Complete	Kerem Ulcay	2 hours	1.5 hours	½ hour
	Summary for entire team	8 hours	5 hours	3 hours

Artifact or Deliverable	Who (individual and team)	Estimated	Actual	Difference
SPMP	Jonah Sherman	2 hours	2 hour	0 hour
SPMP	Sofia Machado Lopes	2 hours	2 hours	0 hour
SPMP	Timur Blair Gordon	2 hours	1.5 hours	½ hour
SPMP	Kerem Ulcay	2 hours	1.5 hour	½ hour
	Summary for entire team	8 hours	7 hours	1 hour

Cumulative

Who	Estimated	Actual	Difference
Jonah Sherman	7.5 hours	5 hours	2.5 hours
Sofia Machado Lopes	7.5 hours	4.5 hours	3 hours
Timur Blair Gordon	7 hours	5.5 hours	1.5 hours
Kerem Ulcay	7.5 hours	5.5 hours	2 hours
	29.5 hours	20.5 hours	9 hours

13.5 Defect Tracking

Artifact or Deliverable	Who (individual and team)	Estimated	Actual	Difference
SRS – Business Domain	Jonah Sherman	2	0	2
SRS – Business Domain	Sofia Machado Lopes	2	1	1
SRS – Business Domain	Timur Blair Gordon	2	0	2
SRS – Business Domain	Kerem Ulcay	2	0	2
	Summary for entire team	8	1	7

Artifact or Deliverable	Who (individual and team)	Estimated	Actual	Difference
SRS – Requirements	Jonah Sherman	2	1	1
SRS – Requirements	Sofia Machado Lopes	2	0	2
SRS – Requirements	Timur Blair Gordon	2	1	1
SRS – Requirements	Kerem Ulcay	2	0	2
	Summary for entire team	8	2	6

Artifact or Deliverable	Who (individual and team)	Estimated	Actual	Difference
SRS – Analysis - Complete	Jonah Sherman	2	0	2
SRS – Analysis - Complete	Sofia Machado Lopes	2	0	2
SRS – Analysis - Complete	Timur Blair Gordon	2	0	2
SRS – Analysis - Complete	Kerem Ulcay	2	0	2
	Summary for entire team	8	0	8

Artifact or Deliverable	Who (individual and team)	Estimated	Actual	Difference
SPMP	Jonah Sherman	2	1	1
SPMP	Sofia Machado Lopes	2	0	2
SPMP	Timur Blair Gordon	2	0	2
SPMP	Kerem Ulcay	2	1	1
	Summary for entire team	8	2	6

Cumulative

Who	Estimated	Actual	Difference
Jonah Sherman	8	2	6
Sofia Machado Lopes	8	1	7
Timur Blair Gordon	8	0	8
Kerem Ulcay	8	1	7
	32	4	8

13.6 Dictionaries

Class			
Name	Description	Methods	Attributes
Member	The member class is a class for each user registered on the platform. It's attributes are member's information and the methods are functionalities available to users on the platform.	register() login() edit() search() create() volunteer() donate()	Name Username Email Password Portfolio
Member Profile	The member profile class is the class responsible for publicly presenting members' profiles on the platform. It allows people to learn about and contact members on the platform.	contact()	Name Bio Portfolio
Login Page	The login page class manages functions to login and register to the platform. It's attributes are string input fields that allow users to input credentials	authenticate() register() forgotPassword()	Inputs: - Username - Password - Email
Search Page	The search page class is responsible for searching the platform database for members and projects. It displays parameters and filters that users can use to narrow their search results.	searchMember() searchProject() resetParameters()	Inputs: - Keyword - Category - Popularity - Date - Location
Project	The project class holds all information regarding the good will project as its attributes, and can be created or supported through user functionalities.	updateProject() inviteMembers() notifyContributors()	Project Name Description Category Contributors Budget Schedule Location
Database	The database class is where all data on projects and members are held.		memberList projectList

Methods			
Name	Description	Class	Arguments
register()	Register method registers new users by calling register method on the login page.	Member Login page	Name Username Email Password
login()	Logs the user in to the system, calls the authenticate method in the login page class.	Member	Username Password
create()	Creates a project, gets input from the user about the project. Once the project is created, call the updateProject method in Project class. Arguments are inputted by the member once the method is called.	Member	Project Name Description Category Contributors Budget Schedule Location
volunteer()	Member becomes a volunteer of the project once the method is called.	Member	Project object
donate()	Member donates to project budget, referred to outsourced money transfer protocols such as PayPal.	Member	Project object Donation amount
search()	Directs member to search page to gather search inputs.	Member	
edit()	Allows member to edit information displayed on their profile, members can change their bio's and contact information.	Member	
authenticate()	Called by the login page class, the method checks encrypted user credentials with the ones stored in the database class.	Login page	Username Password(encrypted)
forgotPassword()	Called at the login page by the user if the user requires assistance logging in.	Login page	Email

contact()	The method allows one member to request the contact information for another member for collaboration on good will projects.	Member profile	Member object
searchMember()	Searches member in the list of members stored in the database given search parameters.	Search Page	Keyword Category
searchProject()	Searches project in the list of projects stored in the database given search parameters.	Search Page	Keyword Category Popularity Date Location
resetParameters()	Resets inputted search parameters.	Search Page	
updateProject()	Updates data on project stored in the database if changes to the project occur.	Project	
notifyContributors()	Broadcasts contributors of the project a message or important update on the project.	Project	
inviteMembers()	Allows project contributors to invite members to join the project.	Project	Username

Attributes			
Name	Description	Type	Authorization (r/w)
portfolio	A list of projects that a member has contributed to.	List of projects	Read only
bio	Biography, description of members displayed on their profile.	string	Write
keyword	Search parameter used to find project or members by keyword.	string	Write

category	Category of good will project such as sustainability, education, etc.	string	Write
popularity	Sorting preference, either popularity ascending or descending. Based on number of contributors to a project.	string	Write
description	Project description attribute that defines what the good deed project aims to achieve and how.	Timur Gordon	Write
contributors	List of member objects contributing to a project	List of members	Read only
Budget	Budget attribute of expenses of the project and the amount.	Dictionary with string keywords and integer values.	Read only (unless creator)
Schedule	Schedule of milestones and dates	Dictionary with string keywords and date values.	Read only (unless creator)
memberList	List of all member objects on platform	List of members	Read only
projectList	List of all project objects	List of projects	Read only

Attributes which are named in a self-explanatory manner (such as “username” and “password”) haven’t been included in the dictionary for concision. Refer to section 9.1 to see all attributes. Also please refer to section 9.1 to see attribute types, all attributes have been labeled with their corresponding types in the class diagram. Complex attribute types are mentioned above.

Relationship					
Name	Description	From Class	To Class	Optional/ Mandatory	Cardinality
Owns	The owns relationship means each member class owns a member profile class. Owns implies the owner can edit information.	Member	Member Profile	Mandatory	One-to-one

Key Events						
Name	Description	Motive	Action	Pre-conditions	Post conditions	State change
Create project	The platform member creates a good deed project.	To add a project to the platform.	Member object calls create() method	Member is directed to input project parameters	Project is added to database attribute projectList.	One project added to database. Member's portfolio attribute updated to contain newly contributed project.
Volunteer	The member volunteers to a project.	To support good deed project	Member object calls volunteer()	Member isn't already a volunteer of the project	Project contributors attribute is reflected to include new volunteer member.	Database is updated to reflect changes to project, the project is added to member's portfolio.
Donate	The member donates money to a project	To support good deed project	Member object calls donate() method	Member is directed to outside money transfer	Project budget and status is updated to reflect	Database is updated to reflect changes to project, the

				platform	money income.	project is added to member's portfolio.
--	--	--	--	----------	------------------	--

14. INDEX

To be included at a later date