Project Requirements

27/10/2014

Nirmaan Project Manager

Aditya Singh
Anshul R
Dharm Skandh Jain (Project Manager)
Janani Durairaj
Rohit H. Pandey
Shatrujit Aditya Kumar
Vishnu Nair

Submitted as part of the requirements
of group project (GROUP 21) for the course
Object Oriented Programming (CS F214)

Contents

List of Figures

1.0. Introduction

1.1. Purpose

Use case diagrams, CRC cards and various UML diagrams will be submitted along with this document over the course of submission in order to make the design concept clear to any reviewer who may want to study/emulate the project. The UML diagrams as a whole will explain the way the method calls and object interactions will work in this software. The use cases have been described clearly with the preconditions or post-conditions (if any) so that the user may be well versed with what they can extract from this application and to enrich their understanding of it.

The purpose of this document is to present a description of the Nirmaan Project Manager application. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli.

1.2. Scope of Project

This software system will be a Project Manager for the goa chapter of the NGO Nirmaan. This system is designed to effectively store the project details of the organization and to provide a notification system for the latest events by bringing all the schedules and major project details to the computers of all the organization members. This will ensure that the members stay up to date on the latest events, and on their roles as members of the organization in a very easy to use way.

The system will allow the Core members, along with the Project Managers and the Volunteers to interact more freely and diversely by bringing all the details together in one place. The software will send and receive messages and project details from the users and will make the necessary changes in the database and send notifications to the desktops of the users after taking into consideration the hierarchy and the role that specific member groups are supposed to play to keep the organization functioning without any hassles. The data will be broadcasted to all the members but will be stored selectively on the systems of only the members who are authorized to view the information. This authorization will be based on the hierarchy of the member in the existing structure of the Nirmaan Organization.

The systems will contain lists of the messages broadcasted to the users of the specific systems, as well as all the details of the projects that the members are enrolled in along with notifications of the recent events (meetings or some activities) for the projects that they have been enrolled in.

All communication between the systems will be mirrored across client machines and we will adopt a privacy system similar to Facebook. Every message will be broadcast to everyone and the client application will store/grant access to the message only when some guard conditions are met.

1.3. Glossary

Term	Definition
Person	This is the basic structure of any member, as viewed by the system. It has varying roles in the system, based on the

	assigned role which will be part of a hierarchy.
Core	A Core (member) is a Person with respect to the system, and a member with respect to the organization who is at the top of the member hierarchy and holds master privileges over all the Projects under the organization.
Database	Collection of all the information monitored by this system.
Project Head	Person who is directly below the Core member in the system and organization hierarchy. Has fewer privileges than the Core.
Volunteers	The Person who is placed lowest in the organization hierarchy and will have the least privileges of the three roles.
Message	A set of information that can be related to an Event or a Project.
Project	The basic group that any of the roles can be assigned to.
Event	Any event of a general importance to the organization. The notifications for this can be sent selectively.

1.4. Overview of Document

The next chapter, the Overall Description section, of this document gives an overview of the functionality of the product. It describes requirements and is used to establish the technical requirements specification in a simple language.

2.0.Overall Description

2.1 System Environment

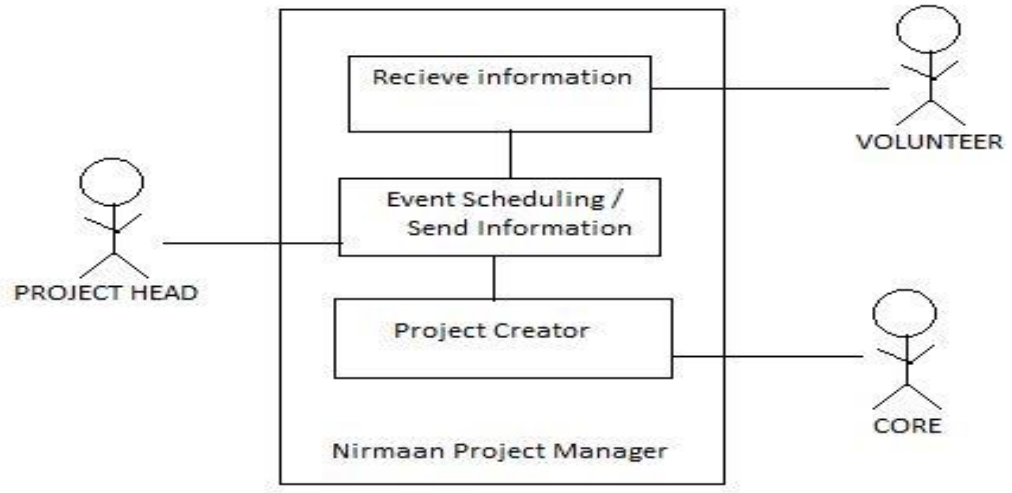


Figure 1 - System Environment

The Nirmaan Project Manager has three active actors – volunteer, project head and core member.

The Volunteer machine has read-only privileges of the Message, Event, and Project information broadcasted by others as per the enrolled Project. The Project Head has access to all the information regarding the Project allotted to it. It can also receive the information broadcasted. Finally, the Core member has master access to all the activities that can be performed by the other two roles in addition to the privilege to create new projects and assign Project Heads to it.

The Messages and all communication between the systems will be mirrored across client machines and we will adopt a privacy system similar to Facebook. Every message will be broadcast to everyone and the client application will store/grant access to

the message only when some guard conditions are met. This will then be stored on the local machine for archival purposes.

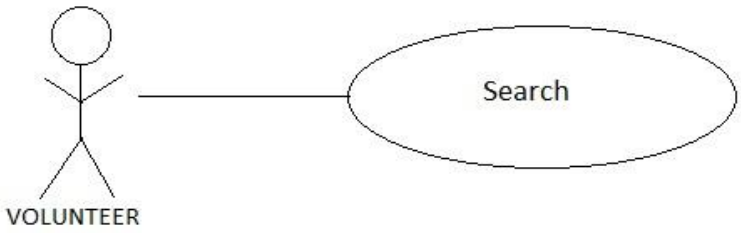
The Project, Event, and Message are different classes with their own attributes.

2.2 Functional Requirements Specification with Use Cases

This section will provide basic information about the main use cases of for each actor. The Person with the Volunteer role will have the least use cases as it has the least freedom in the system. The Core member will have the most as it is the main actor in the system and hence has the most privileges.

2.2.1 Volunteer Use Case

Use case: **Search**
Diagram:

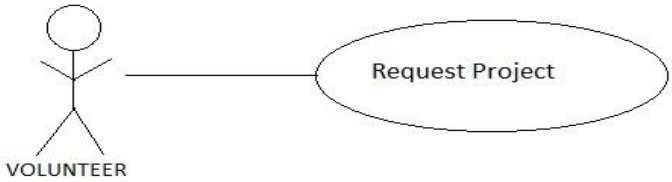


Brief Description
The Volunteer accesses the Database of the Project(s) stored on his system and searches and views the details of the Project(s) that he is enrolled in.

Initial Step-By-Step Description

1. The Volunteer chooses to search by Projects by Project name or Project Head.
2. The system displays the results to the Volunteer.
3. The Volunteer selects any one of the Projects that come up as a result and can view all the information about it like other Project co-volunteers, Schedules, etc provided that he is already enrolled in the selected Project.

Use case: **Request Project**
Diagram:



Brief Description
The Volunteer can request to be enrolled in any Project that has slots empty.

Initial Step-By-Step Description
For this step to be initiated, the Volunteer must have arrived at the Project list through search and the Project in question must have empty slots.

1. The Volunteer puts in request to enroll in the given course. The request will be accepted at the discretion of the Project Head.

General Mechanism for Creation/Modification of Projects and Events

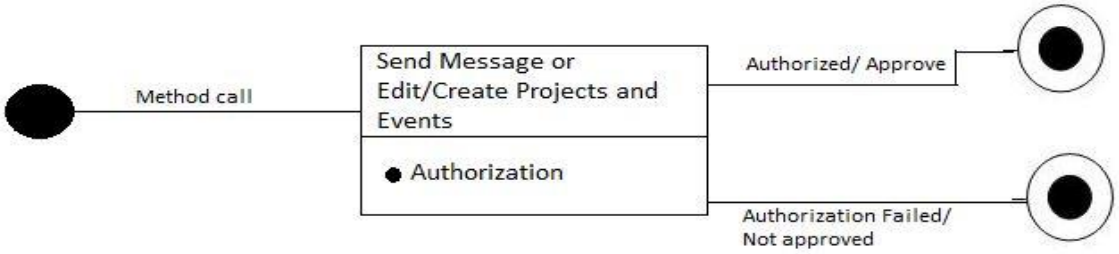


Fig 2 : Sending messages or Creating/Modifying Projects and Events

The above state transition diagram illustrates the processes behind the method calls for sending Messages, and modifying Messages and Projects. The Core member role is allowed to carry out all of the above transactions whereas the Project Head has access to only send Messages and edit Projects and Events. The Messages sent and Projects scheduled by Project Head will only be available for the enrolled volunteers of their own Project in addition to the Core members (who have sort of an all-access pass).

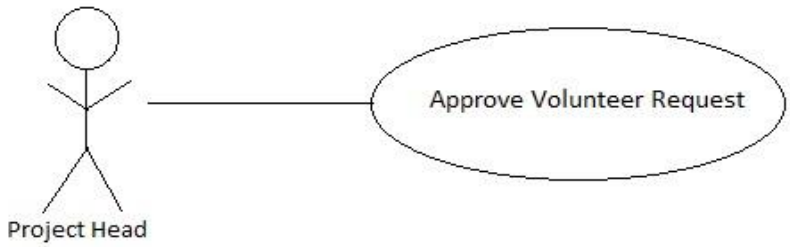
When the method is called to carry out any of the above-mentioned activities, the system takes in the input for the action that is to be carried out by the method. Then, before the action is carried out, the method checks whether the actor performing the action is authorized to do it. If he is, then the action is carried out and if he isn't, then the function just exits with a warning.

These actions are further described in the use cases for the Project Head and the Core members.

2.2.2 Project Head Use Case

There can be many one Project Head per Project. Every Project Head can send messages and create events and schedules of the Project for only his own project.

Use case: Approve Volunteer Request
Diagram:

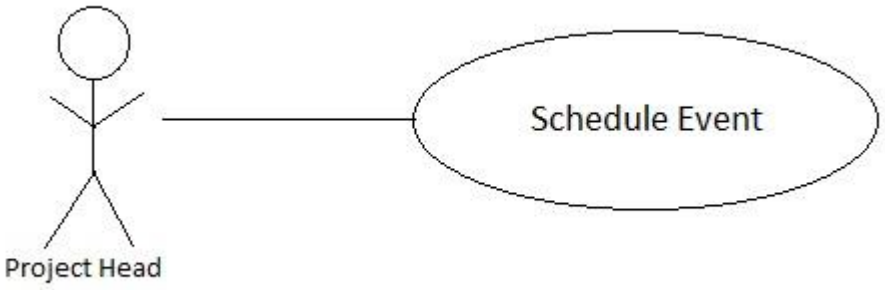


Brief Description
The user can approve/reject the Volunteer requests to join his group.

Initial Step-By-Step Description
Before this function can be used, an authorization procedure will take place.

1. The Head calls the method and gets a list of all the Volunteers that have requested an enrollment to his Project.
2. The Project Head can accept/reject the request and then.
3. If the volunteer is accepted, an acceptance message is sent to the Volunteer.
4. All the clients with access to the Project in question will get their database updated with the new Volunteer added.
5. If the volunteer is rejected, a message informing of the rejection will be sent to the volunteer.

Use case: Schedule Event
Diagram:

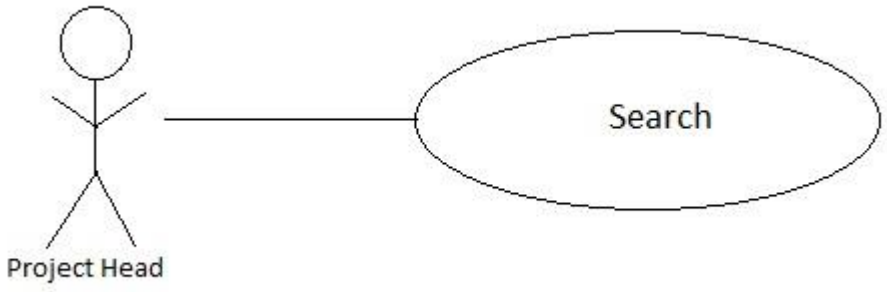


Brief Description
The Project Head can schedule an event and invite members of his Project to it by sending notifications to all.

Initial Step-By-Step Description
Before this function can be used, an authorization procedure will take place.

1. The schedule event option is selected from the list of functions that the user gets at the main screen.
2. Event date, timings, location and Name is entered by the user on prompt.
3. System sends the message to the members of his Project.

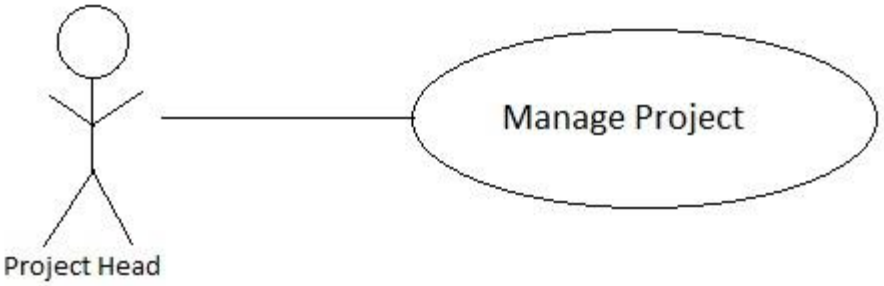
Use case: Search
Diagram:



Brief Description
The user can search for all projects and can view the data associated with the Project that he is in-charge of.
Initial Step-By-Step Description

1. The user selects the Search option from the functions list on the main screen.
2. Projects and Volunteers can be searched by providing the Project Head or the Project name.
3. Any Project or Volunteer from the search results can be selected to view from the given options.
4. A user is authorized to view the information of only his own Projects and the volunteers enrolled in his own project. System does authorization verification and if the user is authorized, all the requested data is made accessible.

Use case: Manage Project
Diagram:



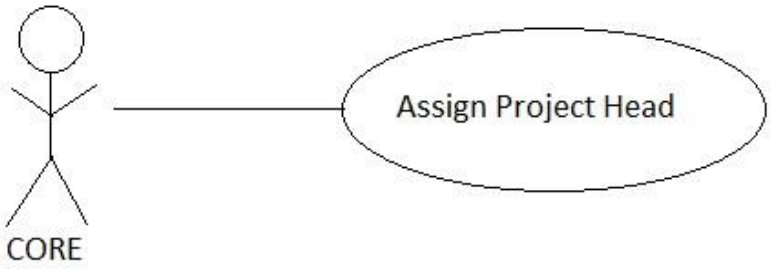
Brief Description
The user can change the project dates and schedule meetings by sending messages directly to the enrolled Volunteers.
Initial Step-By-Step Description
Before this function can be used, an authorization procedure will take place.

1. The Project Head will choose the option to call on this method from the list.
2. He selects one of the options from amongst changing dates and scheduling meetings.
3. Then, the user is required to enter the message body.
4. After authentication by the system, the required changes are made to the database on the systems of the affected parties by the system.

2.2.3 Core Member Use Case

A core member has an all access pass to database – they manage almost everything at grass-roots level

Use case: Assign Project Head
Diagram:

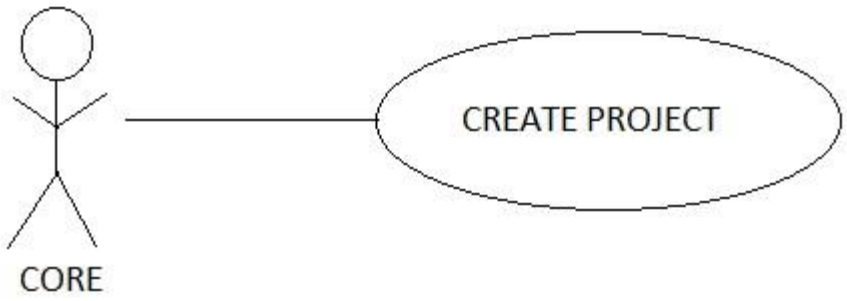


Brief Description
The core member assigns a project head to an existing project.

Initial Step-By-Step Description
Before this function can be used, an authorization procedure will take place.

1. The project name and the project head are specified by the core member.
2. A method retrieves the list of projects.
3. If the project is found, it sets the project head of a project regardless of the existing value for project head.
4. If the project is not found, a message is displayed conveying the same.
5. If things go successfully, a broadcast is sent to reflect the changes in other relevant machines' copies of the database.

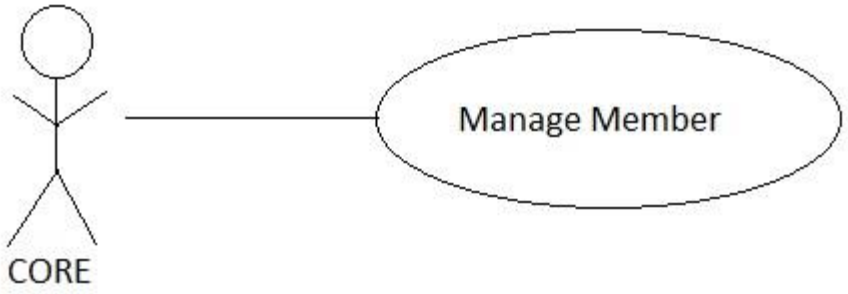
Use case: Create Project
Diagram:



Brief Description
The core member can create projects through this method.
Initial Step-By-Step Description

1. The core member specifies the details of the project, optionally, a project head.
2. The method searches in the database to check if the project exists.
3. If the project does not exist, a new entry is created with the specified details.
4. If the project already exists, a message is displayed conveying the same.
5. If a project was created, a message is broadcasted to reflect the changes in the local copies of databases on relevant machines.

Use case: **Manage Member**
Diagram:

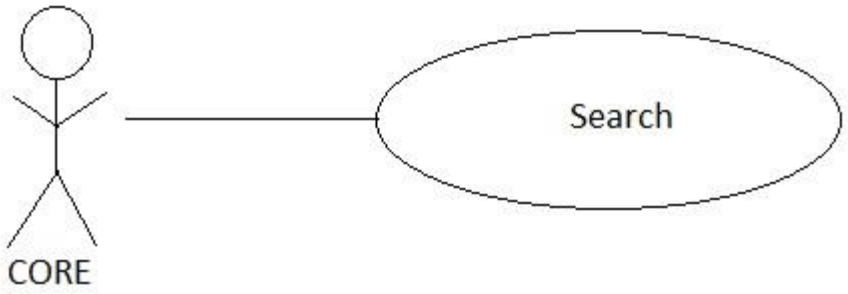


Brief Description
The core member can view and modify the details of every volunteer and project head. Used to set privileges and assign roles for different members in Nirmaan.

Initial Step-By-Step Description

1. The core members makes a request for a member's details and optionally, specifies a change in an attribute.
2. If the person is found, the changes are reflected in the database.
3. If the person is not found, a message is displayed conveying the same.
4. If the changes are successful, a message is broadcasted to reflect changes in the local copies of the database.

Use case: **Search**
Diagram:



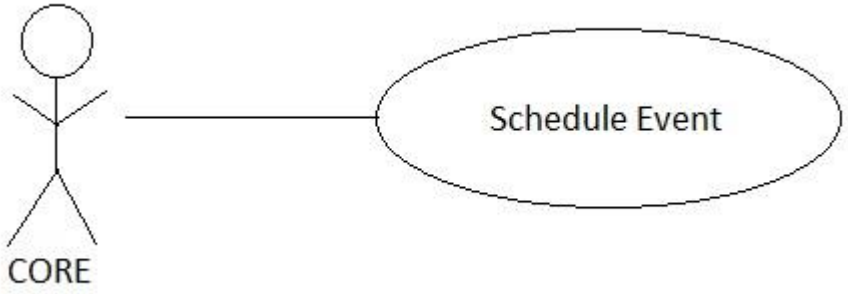
Brief Description
The core member can search for every detail of any Nirmaan member.

Initial Step-By-Step Description

1. The core members makes a request for a member or Project through an attribute like name.
2. If the person or Project is found, the details are displayed after authorization.

3. If the person is not found, a message is displayed conveying the same.

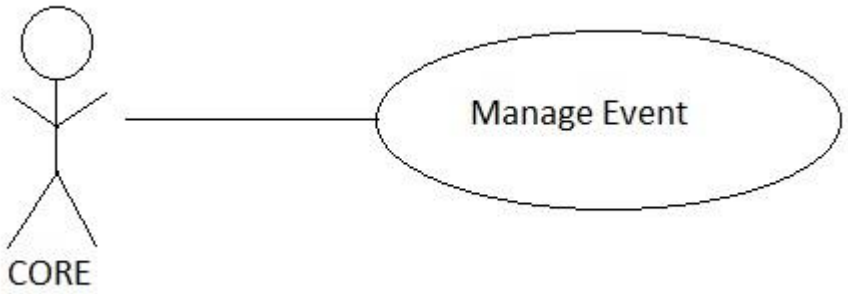
Use case: Schedule Event
Diagram:



Brief Description
The Core can schedule an event and invite members of one or more Projects to it by sending notifications to all.
Initial Step-By-Step Description
Before this function can be used, an authorization procedure will take place.

1. The schedule event option is selected from the list of functions that the user gets at the main screen.
2. Event date, timings, location and Name is entered by the user on prompt.
3. The recipients are selected.
4. System sends the message to the members of his Project.

Use case: Manage Event
Diagram:



Brief Description
The Core can see and re-schedule an event and invite members of one or more Projects to it by sending notifications to all.
Initial Step-By-Step Description
Before this function can be used, an authorization procedure will take place.

1. The manage event option is selected from the list of functions that the user gets at the main screen.
2. All the scheduled events from all groups are shown to the user. The user can select one of these to make changes to.
3. The changes are taken in from the user one by one.
4. Final recipients are selected.
5. System sends the message to the recipients and updates the data on every client's database.

2.3 Non-Functional Requirements

The speed of the message transmission/reception depends on the speed of the internet connection. The members are supposed to know their own authorization/ hierarchy in the organization. The members should be basic computer literate in opening applications and following clear instructions in English.

2.4 User Characteristics

The user is expected to have a working internet connection without any block on IM servers and without any proxies set up.