Group S Development Team

TeamMe

Software Requirements Specification

For CSc 32200 Project

Version 1.0

Revision History

Date	Version	Description	Author
12/Mar/20	0.3	First draft of specification report that contains basic information of specifications.	Sophie Huang, Ekramul Sawrid, Tohidul Islam, Dor Ulman
21/Mar/20	0.6	Second draft of specification report that includes details and a use-case diagram.	Sophie Huang, Ekramul Sawrid, Tohidul Islam, Dor Ulman
23/Mar/20	1.0	Final draft of report. Reviewed by every author and changes were finalized.	Sophie Huang, Ekramul Sawrid, Tohidul Islam, Dor Ulman

Table of Contents

Software Requirements Specifications

1. Introduction

This document will capture the software requirements for the TeamMe software system. The outline of this document based on a typical Software Requirements Specification for a project using use-case modeling. This artifact consists of a package containing use cases of the use-case model and applicable Supplementary Specifications and other supporting information.

The TeamMe software system will facilitate active teaming of people with similar interest and skill-set to forge groups for a certain do-good project (Wei, 2020a). The following subsections are an introduction to the software system and give an overview of the entire document.

1.1 Purpose

TeamMe is a platform that connects people and helps them to collaborate and achieve their goals. The system gives users the ability to form groups who share the same interests and ideas. Also, users can connect with other users, groups, and ideas in order to share knowledge. TeamMe's Software Requirements Specifications will clarify all the aspects of the system and will give the reader a full observation and understanding of its parts. The system includes the following types of users - OU, SU, VIP and visitors, groups formation, searching, registration, users verification, users polls, user's points counting, punishments of users and groups by the SU and even elimination of users and groups. Use-case diagrams are used to elucidate the working parts of the system and their integration, including each user requirements and scope of action, decision making (SU) and overall system and subsystems functionality.

Non-functional parts of the system are not discussed yet as it is still in its early stages of development and will be mentioned in the future.

GUI design prototype is available under “Supporting Information” in Appendix (pages 17-20). Programming language for implementation will be Python and libraries and operating systems used will be Windows 10 are detailed ahead in the document under the “Overall Description” section.(page 9)

Advanced UML diagrams for the GUI and more technical info remain unknown on this document as the system is still in its first stages of development. More will be added as development progresses.

1.2 Scope

The software application that is to be produced is called TeamMe, which is an active teaming system that will facilitate active teaming of people with similar interest and skill-set to forge groups for a certain do-good project. It is a Windows Desktop Application with Python being the primary language used to create it. We have based this software on the requirements given my Professor Wei (Wei, 2020a). This software is not intended for building of the actual projects on the platform but, rather, the general idea is for users to find and interact with other users, communicate with group members of a project group, and to have a professional and safe place where the soft-skills of users of a group project can be facilitated. The benefits that this software can provide is easing the finding of people with similar interests/skills, ease of collaboration of group members of a project, ease of management of human resources, and to provide a professional and safe environment for the aforementioned benefits.

The required features are explained as explained in this paragraph and the associated use-case diagram is Figure 1 of Section 2.1 (see page 11). Once a visitor registers and is approved to become an ordinary user (OU), he/she/they can form groups for a certain project by inviting other OUs. Once a group is formed, a web-page or window is made available, where some information is made public or private. On the web-page, OU members of that group can can certain activities like evaluating and warning other group members, post and moderate the group page, posting updates and scheduling meetings, tracking of assigned tasks, polling for meet-up times, voting out members, appealing to a SU for a reputation score change, voting to close a group, conduct a exit evaluation to other group members after the closing of a group, compliment other OUs to SU to increase reputation score of that complemented OU,. Visitors and OU can surf for projects, OU and SU profiles, and complain to SU about a group or OUs. OUs can become VIPs, users with more authorization than OUs, when their reputation source reaches a certain threshold. The SU can approve/reject visitors who want to become OUs, change reputation scores of OUs, shut down a group that was complained about and punish all involved in that group, kick out OUs from the system forever.

This software application will also have potential but not necessarily required creative features as of when the document was written. However, there will be at least 1 creative feature from the list of creative features discussed in section 3.2 (see page 16-17). Here is a brief scope of every creative feature:

1. Creative Feature 1: Project Tracker. This feature allows a OU to see graph of commits made to the project. This feature can be used to show trends of performance.

2. Creative Feature 2: Project Hierarchy Tree. This feature shows the user hierarchy tree for a project and the responsibilities of each person on the hierarchy tree.

We have also included images of some GUI (graphical user interface) prototypes (see Appendix).

1.3 Definitions, Acronyms, and Abbreviations

For all the definitions, go to the Glossary (page 21).

The following are important terms required to know to properly interpret this document.

Types of user:

Visitor: Not a registered user, can surf around the application, and isn't logged in.

OU (ordinary user): All self registered users who are approved by SU and need to login.

VIP (very important person): OUs whose reputation scores exceed a threshold set by SU and have more authorization than OUs.

SU (super user): One founding SU who initializes the system and one democratic SU who is voted by VIPs.

Framework (software framework): Platform used for developing a software or an abstraction in which software providing generic functionality can be selectively changed by additional user-written code, thus providing application-specific software.

GUI (graphical user interface): It is a graphical interface which uses visual elements such as windows, icons, buttons, cursor, etc.

Library (software library): Is a collection of resources used by computer programs, often for software development. These may include configuration data, documentation, pre-written code, subroutines, classes, and values.

Project Hierarchy Tree: A potential creative feature for the TeamMe software system that shows the user hierarchy tree for a project and the responsibilities of each person on the hierarchy tree.

Project Tracker: A potential creative feature for the TeamMe software system that allows an OU to see a graph of commits made to the project.

Python: Is an interpreted, high-level, general purpose programming language. It can be used to make desktop software. Python 3.7 is a version of Python.

Matplotlib: Is a plotting library for the Python programming language and its numerical mathematics extension Numpy.

Seaborn: Is a Python data visualization library based on Matplotlib. It provides a high-level interface for drawing informative statistical graphics.

Tkinter: Is a Python binding to the Tk GUI toolkit. It is the standard Python interface to the Tk GUI toolkit, and is Python's de facto standard GUI.

Serialization: The conversion of an object into a byte stream which allows it to be saved on a local disk or over a network.

Software System: Is a system of intercommunicating components based on software forming part of a computer system.

TeamMe: Software system that will facilitate active teaming of people with similar interests and skill-set to forge groups for a certain do-good project.

Use-Case diagrams: A form or drawing to display a system or software behavior that is under development. It contains both visual and textual representation, including entities that are involved in the system.

Window: Is a framed area on a display screen for viewing information or content.

Windows 10 (Microsoft Windows 10): Series of operating systems produced by Microsoft.

1.4 References

"Functional and Nonfunctional Requirements: Specifications and Types", May, 2018. https://www.altexsoft.com/blog/business/functional-and-non-functional-requirements-specification-and-types/

"Project Requirements" Wei Jie, March 21, 2020a. http://www-cs.engr.ccny.cuny.edu/~csjie/322.html

"Spec Report Template" Wei Jie, March 21, 2020b.

http://www-cs.engr.ccny.cuny.edu/~csjie/322.html

"What are System Requirements Specifications/Software (SRS)?" Inflectra, Jan 28, 2020.
https://www.altexsoft.com/blog/business/functional-and-non-functional-requirements-specification-and-types/

1.5 Overview

The remaining portion of this documentation goes over the specific system requirements needed for the application, TeamMe, to function and provides a visualization of the program's user hierarchy through detailed use-case diagrams and descriptions; in addition, specific features and potential features of the program are introduced. A general background on the product is first provided in section 2 to allow a better understanding of the program and its purpose. A use-case diagram is then provided in section 2.1 along with descriptions of all actors and entities. Following this section a brief description of the system, subsystem, and components that are assumed to exist when using this software is given. Section 3 completely details the software and other specifications needed for this application to be compatible with a given system. The following two subsections thoroughly explain each portion of the use case model and describe application specifications that are not introduced in the use-case diagram such as potential features that may be added to the application in future updates. Section 4 of this documentation contains tools to help readers navigate through the document and provides prototypes of the application user interface and application features. In addition all application related terms used within this document are defined in section 4 to reduce mutual misunderstanding.

2. Overall Description

TeamMe aims to create a collaborative environment where registered users with similar interests can work together on theorizing and planning projects. The program does not serve as a replacement to integrated development environments, instead it is meant to facilitate the development of a project by allowing users to plan meetings and offer advice or ideas on ongoing projects. TeamMe also seeks to eliminate the enigma behind users by establishing a reputation system which will allow anyone to see how trustworthy a specific user is; helping to facilitate teamwork between users who otherwise are complete strangers.

The program allows users to build their profiles and improve their reputations through partaking in multiple group projects and providing valuable contributions and criticism. All users can view the current top three project ideas, ordinary user profiles, and super-user profiles upon program startup; users are also able to browse and search for more user profiles and projects. The program creates a group web-page for each project group, which allows members to post updates and schedule meetups. Meetup scheduling is based on polling in order to determine a time suitable to a majority of the group. Group members are allowed to vote on issuing a warning or praise to other members of the group, if any member receives three warnings they will automatically be removed from the group and be deducted five points from their reputation. The program will also allow group members to vote to directly remove an individual which will also deduct ten points from the removed person's reputation. If any individual goes below zero in their reputation, the program will remove them from the system and place them onto a blacklist. There is also an option that allows group members to vote on closing the group; following group closure, members will be prompted to evaluate the other members in the group and then be assigned the median of all scores between the members. The ranking of closed groups are stored and sorted in a list in order to display the top three on the main page. The program also filters messages and deducts reputation for inappropriate words in order to keep the environment work-friendly.

There are four main categories of users within the application: visitors who can browse around on the application without needing to log in, ordinary users who have registered and logged in, VIPs who are ordinary users with a reputation score above a certain threshold, and super users who have heightened access to the program's features. The program will only have two super users at any given moment, the first being the one who initialized the system and the second being voted to be a superuser by the VIPs. Each category of users are limited to the actions they can take within the program. Visitors can only browse through the application and register to become an ordinary user. Ordinary users can create groups, evaluate other ordinary users and VIPs, complain about or compliment other users in their groups, vote to kick an individual, vote to make a meeting, and vote to close a project group. VIPs have all the quirks of ordinary users in addition to being able to determine the reputation score to be added for an entire group, assign a higher initial reputation score, and vote for one super user. The super user is the privileged actor in this program, they have permission to shut down groups, remove users from the system, deduct reputation points, approve registrations, review appeals, and assign VIPs to evaluate groups.

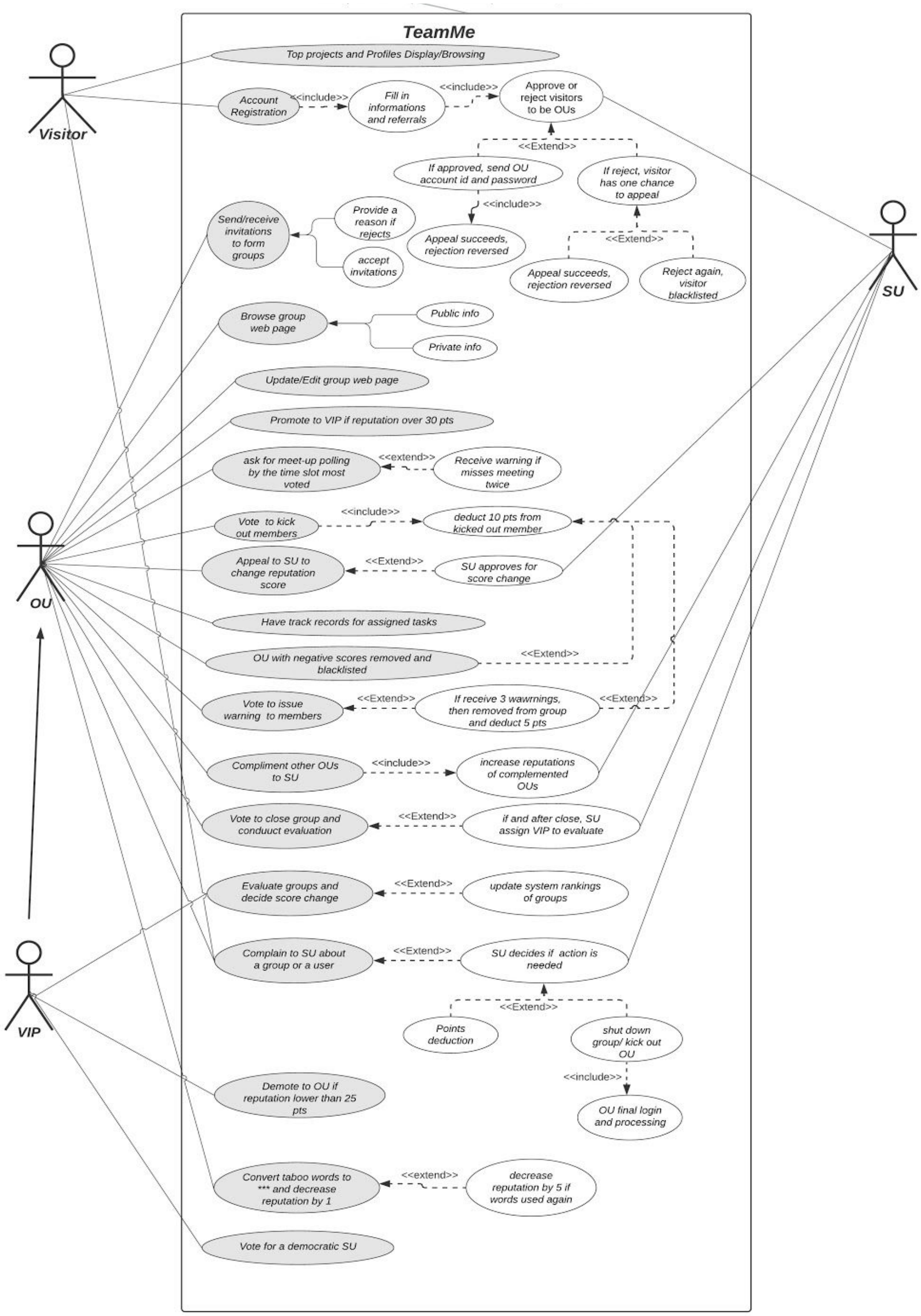
TeamMe will be based on a Windows 10 operating system and implemented in python 3.7. The program makes use of the Tinker library to initialize the interface for the app; in addition to utilizing the Matplotlib and Seaborn to display user hierarchy and allow polling votes. There must be some available storage space in order to use the program due to data storage through file serialization. It is assumed that the user will have python version 3.7 and the required libraries working on their system.

2.1 Use-Case Model Survey

For the TeamMe application, we design the use-case diagram based on the specification sheet provided by Professor Wei. The use-case diagram has four categories of actors: Visitor, Ordinary User (OU), VIP user, and Super User (SU). For each use case, you can refer to section 3.1 Use-Case Reports for more detailed descriptions.

Directly below is the entire use-case diagram. And after that, the entire use-case diagram is split into 2 larger parts for viewing convenience.

Figure 1 of Section 2.1: Use-case diagram



TeamMe
Visitor
OU
VIP
SU
Top projects and Profiles Display/Browsing
Account Registration
<<include>>
Fill in informations and referrals
<<include>>
Approve or reject visitors to be OUs
<<Extend>>
If approved, send OU account id and password
If reject, visitor has one chance to appeal
<<include>>
Appeal succeeds, rejection reversed
<<Extend>>
Appeal succeeds, rejection reversed
Reject again, visitor blacklisted
Send/receive invitations to form groups
Provide a reason if rejects
accept invitations
Browse group web page
Public info
Private info
Update/Edit group web page
Promote to VIP if reputation over 30 pts
ask for meet-up polling by the time slot most voted
<<extend>>
Receive warning if misses meeting twice
Vote to kick out members
<<include>>
deduct 10 pts from kicked out member
Appeal to SU to change reputation score
<<Extend>>
SU approves for score change
Have track records for assigned tasks
OU with negative scores removed and blacklisted
<<Extend>>
Vote to issue warning to members
<<Extend>>
If receive 3 wawrnings, then removed from group and deduct 5 pts
<<Extend>>
Compliment other OUs to SU
<<include>>
increase reputations of complemented OUs
Vote to close group and conduuct evaluation
<<Extend>>
if and after close, SU assign VIP to evaluate
Evaluate groups and decide score change
<<Extend>>
update system rankings of groups
Complain to SU about a group or a user
<<Extend>>
SU decides if action is needed
<<Extend>>
Points deduction
shut down group/ kick out OU
<<include>>
OU final login and processing
Demote to OU if reputation lower than 25 pts
Convert taboo words to *** and decrease reputation by 1
<<extend>>
decrease reputation by 5 if words used again
Vote for a democratic SU

Figure 2 of Section 2.1: Enlarged use-case diagram, part I

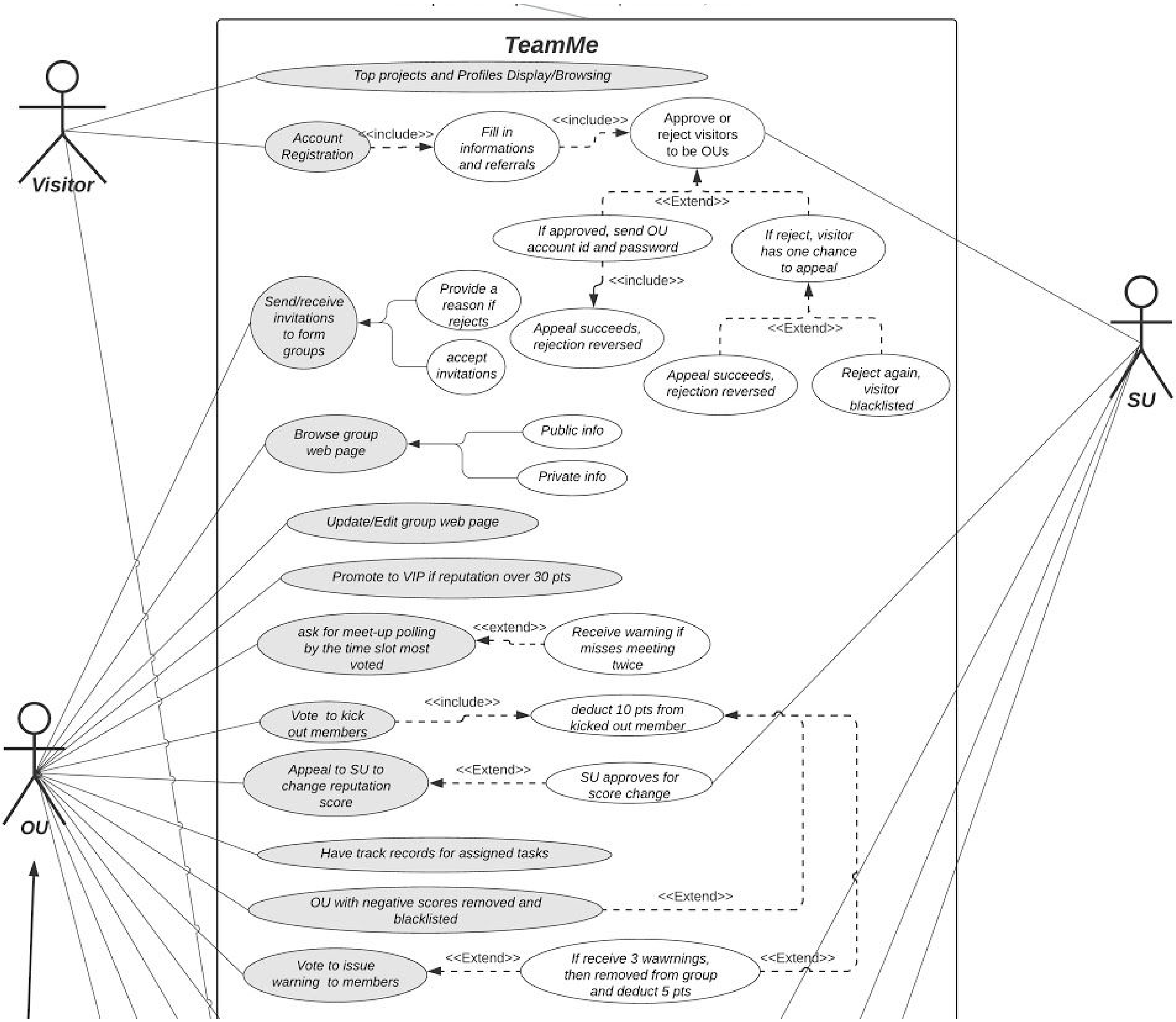
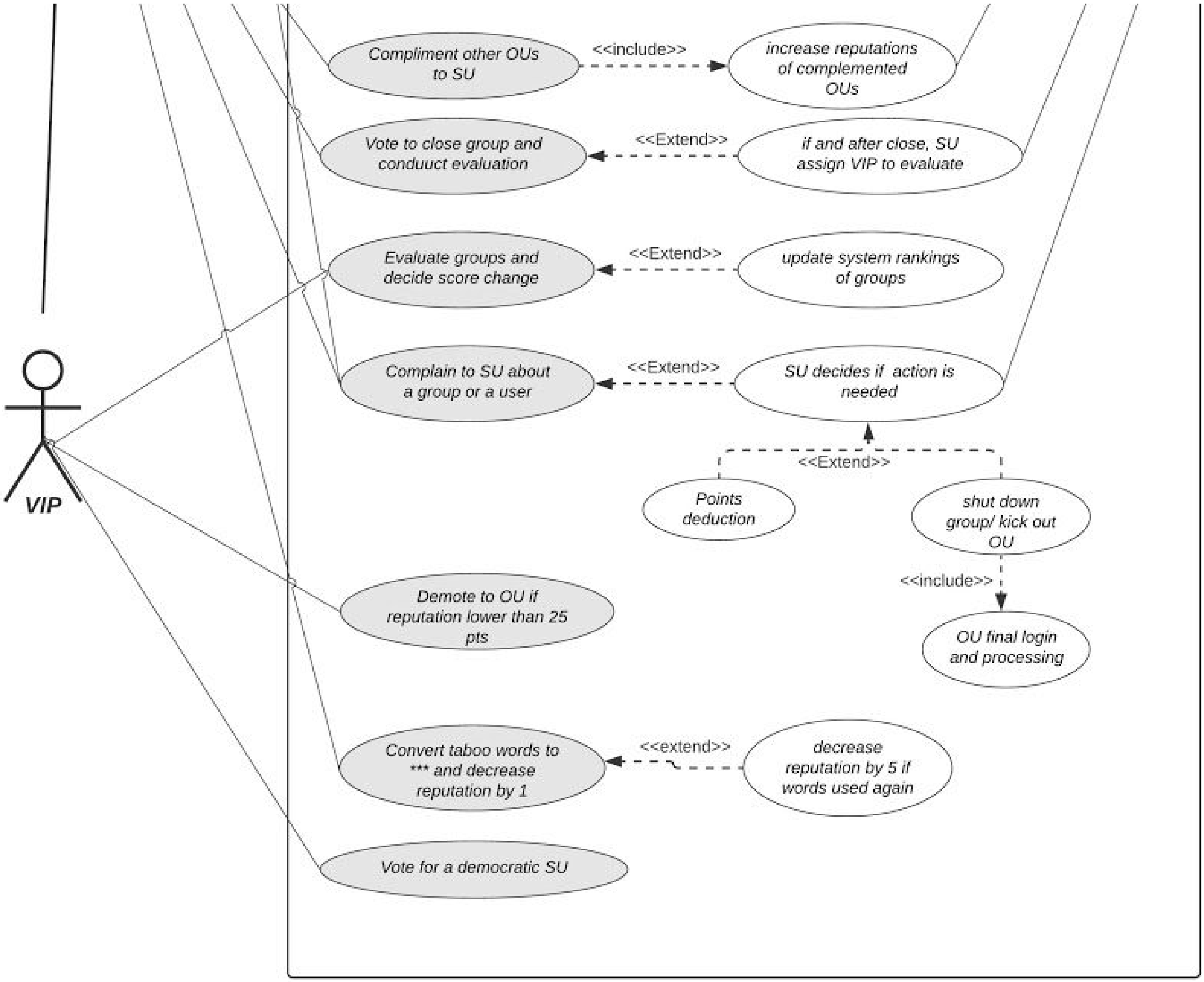

TeamMe
Visitor
SU
OU
Top projects and Profiles Display/Browsing
Account Registration
<<include>>
Fill in informations and referrals
<<include>>
Approve or reject visitors to be OUs
<<Extend>>
If approved, send OU account id and password
If reject, visitor has one chance to appeal
<<include>>
Send/receive invitations to form groups
Provide a reason if rejects
accept invitations
Appeal succeeds, rejection reversed
<<Extend>>
Appeal succeeds, rejection reversed
Reject again, visitor blacklisted
Browse group web page
Public info
Private info
Update/Edit group web page
Promote to VIP if reputation over 30 pts
ask for meet-up polling by the time slot most voted
<<extend>>
Receive warning if misses meeting twice
Vote to kick out members
<<include>>
deduct 10 pts from kicked out member
Appeal to SU to change reputation score
<<Extend>>
SU approves for score change
Have track records for assigned tasks
OU with negative scores removed and blacklisted
<<Extend>>
Vote to issue warning to members
<<Extend>>
If receive 3 wawrnings, then removed from group and deduct 5 pts
<<Extend>>

Figure 3 of Section 2.1: Enlarged use-case diagram, Part II


Compliment other OUs to SU
<<include>>
increase reputations of complemented OUs
Vote to close group and conduuct evaluation
<<Extend>>
if and after close, SU assign VIP to evaluate
Evaluate groups and decide score change
<<Extend>>
update system rankings of groups
Complain to SU about a group or a user
<<Extend>>
SU decides if action is needed
<<Extend>>
Points deduction
shut down group/ kick out OU
<<include>>
OU final login and processing
VIP
Demote to OU if reputation lower than 25 pts
Convert taboo words to *** and decrease reputation by 1
<<extend>>
decrease reputation by 5 if words used again
Vote for a democratic SU

2.2 Assumption And Dependencies

Based on the specification sheet and the use-case diagram, we conclude that TeamMe would be a Windows desktop. We will use Python as the programming language to build TeamMe. The potential libraries and frameworks that will be used are: Tkinter, Matplotlib, Seaborn. Tkinter is a Python library for building desktop GUI; Matplotlib and Seaborn are libraries that create and decorate charts. We may also use other libraries if we discover that they are more functional or suitable. In addition, TeamMe will use simple text files to serialize and store the data.

In addition, since this is a toy system, we will not use the internet to achieve user communications. We will use the application from a single user perspective and use serialization to update and save user information locally.

3. Specific Requirements

The application will be implemented using version 3.7 of the python language so systems that intend to edit or test this program must have a compatible version of python installed. Current usage of the program is geared towards Windows 10 users with build version 18362 and above; both 32 and 64 bit systems are supported. There has not been any testing done on other operating systems so the intended results are not guaranteed. The system used must also be able to run methods from the Tkinter, Matplotlib, and Seaborn libraries as these are essential to the visual and interactive components of the application. The system must grant read/write permissions and file access to the app in order for data to be serialized when closing the program and deserialized when re-opening.

3.1 Use-Case Reports

This section provides detailed descriptions for each use case on the use-case diagram (section 2.1). The order of descriptions is from top to bottom of the use-case diagram.

Use case explanations:

Top projects and profiles display/browsing : a visitor without account registered can browse the application to see top rated projects and user profiles

Account registration & Approving visitors to be OUs: visitors can register to be OUs by filling in basic information and an OU as reference. The SU will then check and approve the application. If rejected, the visitor can appeal to SU to reverse the rejection. If rejected again, the visitor will be blacklisted. When the user is registered, he or she will be given a starting reputation score based on the reference. OU reference may have 0-10 score and VIP reference may have 0-20 score.

Form groups: OUs can form groups by invitations. OUs can either reject or accept a group invitation. For rejection, a reason must be provided. OU can also reject or accept all invitations at one time.

Browse group web page & Update group web page: each group has its own web page. The web page has public information that can be browsed by visitors and other OUs; the web page also has private information only open for its own group members. Web page can be edited by its own group members.

Promote to VIP if reputation over 30 pts: An OU will be promoted to a VIP when its score is above 30 points.

Ask for meet-up polling by the time slot most voted: a group member can ask for a poll to find a meet up time for all members. The most voted time slot will be selected. An OU will receive a warning after missing the meeting twice.

Vote to kick out members: Group members can vote to kick out a group member. The kicked out member will be deducted 10 points.

Appeal to SU to change reputation score: All OU, including the kicked out member can appeal to the SU to change their reputation scores. Each group member should have a track record for the number of assigned tasks that have been done, which is the foundation for the group warnings and the appeals of the affected group member(s).

Have track records for assigned tasks: every OU has a track record of assigned tasks. It shows if the task is complete, and can be used as a reference by SU for deciding reputation score.

OU with negative scores removed and blacklisted: with a negative reputation score, an OU is removed from a group and blacklisted.

Vote to issue warning to members: within a group, every group member can vote together to issue a warning to other members. If a member receives 3 warnings, he or she will be removed from the group and its reputation score will decrease by 5.

Compliment other OUs to SU: every OU can issue a compliment on another OU. This compliment is received and evaluated by SU. If an OU receives 3 compliments, then SU will increase the OU's reputation score.

Vote to close group and conduct evaluation: group members can vote to close their groups. When a group is closed, each group member will conduct an evaluation of other members. Each group member will receive a median reputation score based on the group evaluation. After the group is closed, the SU will assign a VIP to evaluate the group and calculate reputation scores for the group members.

Evaluate groups and decide score change: VIP has the feature to calculate reputation scores for group members after the group is closed. After the new reputation scores are calculated, the system will update the group rankings.

Complain to SU about a group or a user: visitors and OU can issue complaints to SU about a group or another OU. Then the SU will evaluate and decide if the complaint is valid. If valid, then SU can close the group or kick out the OU. All involved members will suffer reputation score decrease or blacklisting.

Demote to OU if reputation is lower than 25 pts: when a VIP's reputation score is lower than 25 pts, then the VIP is demoted to an regular OU.

Convert taboo words: in the system, there is a list of taboo words, if these words are used, they will be automatically transformed into ***. Users who use these words will be deducted 1 reputation score for the first use. If the words are used again by the same users, the users' reputation scores will decrease by 5.

Vote for a democratic SU: all VIPs can vote for one democratic SU

3.2 Supplementary Requirements (Potential Creative Features)

In addition to the required features requested by the instructor, our team has come up with 2 potential creative features. Please note that these are potential features, whether or not they will be delivered depends on the progress made on required features.

Creative Feature 1: Project Tracker

This feature allows an OU to see a graph of commits made to the project. The graph shows the commits made every day or every month, and throughout the entire project. The goal of this feature is to show trends of performance to the OU. The project tracker should be within the private section of the group website.

Creative Feature 2: Project Hierarchy Tree

This feature shows a hierarchy tree for a project. The tree has the project owner or group leader as the root, and has different members branching out. The tree also shows who is responsible for what part of the project. The hierarchy tree should be on the public section of the group website.

4. Supporting Information

Table of Contents

See page 3 for the Table of Contents.

Appendix

Not considered a requirement of the software system.

User-Interface Prototypes

The images below are what some of the graphical user-interface of the TeamMe application might look like. Note that this is an early design and not a finalized idea.

Figure A1: Early design of a TeamMe Account Registration window.

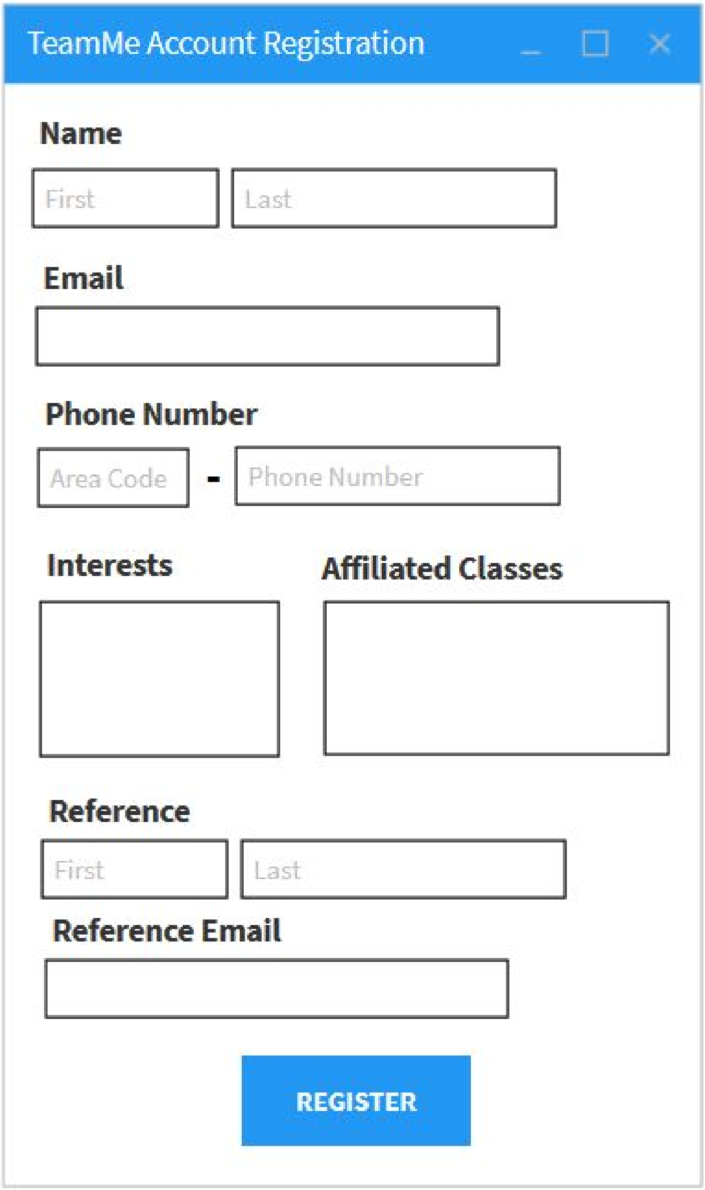
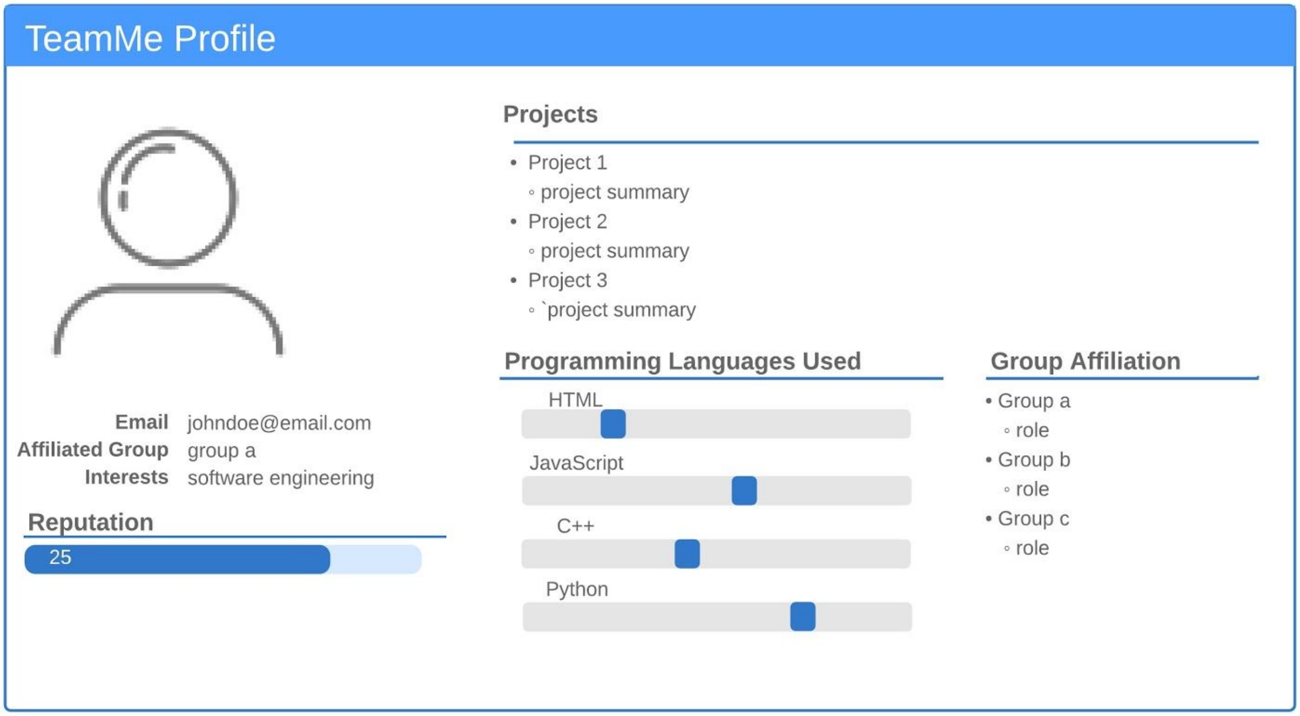


Figure A2: Early design of a TeamMe Profile window.



TeamMe Profile

Email	johndoe@email.com
Affiliated Group	group a
Interests	software engineering

Reputation

25

Projects

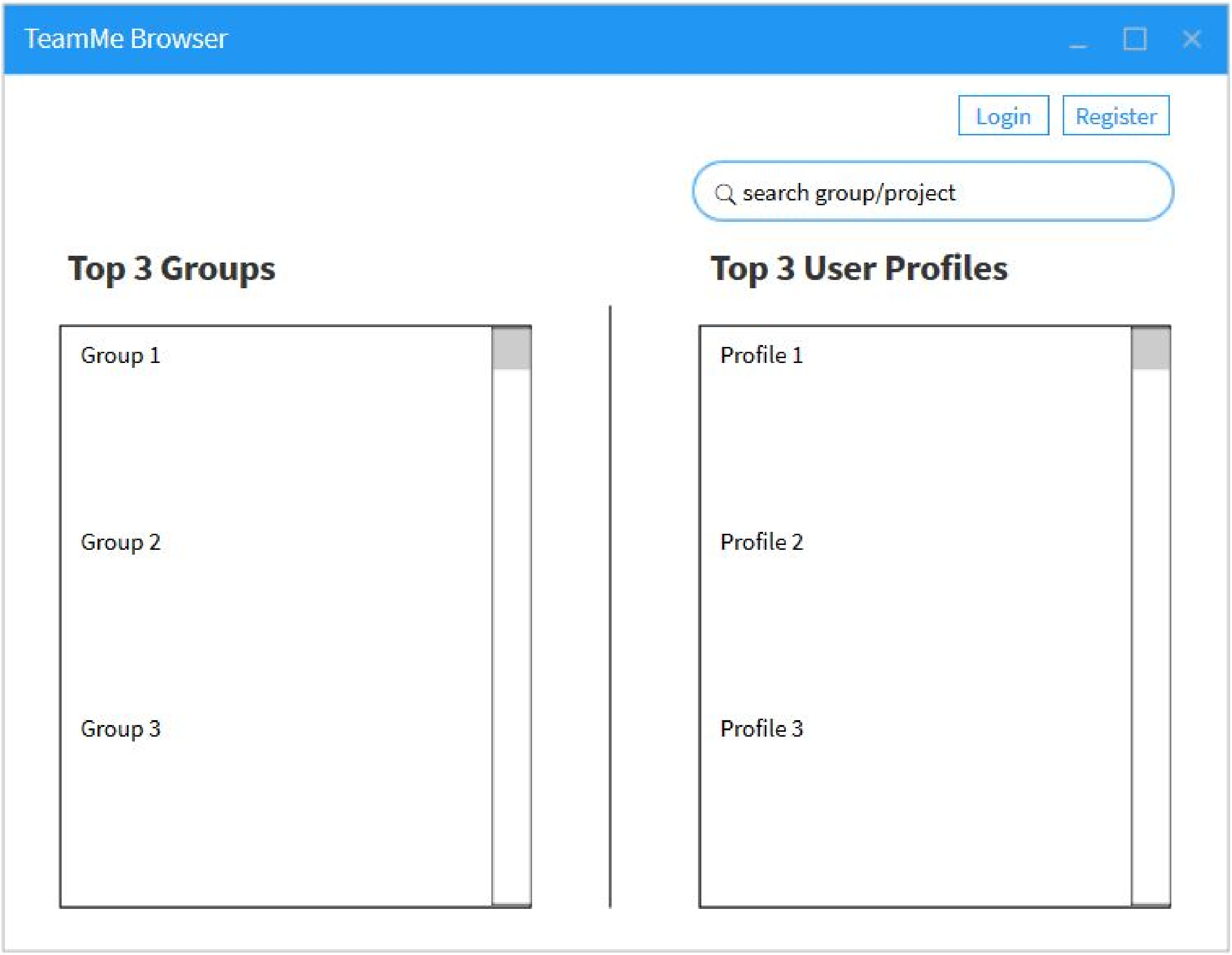
- Project 1
  - project summary
- Project 2
  - project summary
- Project 3
  - `project summary

Programming Languages Used

Group Affiliation

- Group a
  - role
- Group b
  - role
- Group c
  - role

Figure A3: Early design of a TeamMe Browser window that displays the top 3 groups and user profiles.



Glossary

A

B

C

D

E

F

Feature: a unit of functionality of a software system.

Framework (software framework) Platform used for developing a software or an abstraction in which software providing generic functionality can be selectively changed by additional user-written code, thus providing application-specific software.

G

GUI (graphical user interface) It is a graphical interface which uses visual elements such as windows, icons, buttons, cursor, etc.

H

I

Internet A global computer network providing a variety of information and communication facilities, consisting of interconnected networks using standardized communication protocols.

J

K

L

Library (software library) Is a collection of resources used by computer programs, often for software development. These may include configuration data, documentation, pre-written code, subroutines, classes, and values.

M

Matplotlib Is a plotting library for the Python programming language and its numerical mathematics extension Numpy.

N

O

OU (ordinary user) All self registered users who are approved by SU and need to login.

Operating System Is a system software that manages computer hardware, software resources, and provides common services for computer programs.

P

Project Hierarchy Tree A potential creative feature for the TeamMe software system that shows the user hierarchy tree for a project and the responsibilities of each person on the hierarchy tree.

Project Tracker A potential creative feature for the TeamMe software system that allows a OU to see a graph of commits made to the project.

Prototype A first, typical or preliminary model of something.

Python Is an interpreted, high-level, general purpose programming language. It can be used to make desktop software. Python 3.7 is a version of Python.

Q

R

S

Seaborn Is a Python data visualization library based on Matplotlib. It provides a high-level interface for drawing informative statistical graphics.

Serialization The conversion of an object into a byte stream which allows it to be saved on a local disk or over a network.

(SRS) Software Requirements Specification Is a document that describes the software system to be developed. A SRS can follow a specific outline.

Software System Is a system of intercommunicating components based on software forming part of a computer system.

SU (super user) One founding SU who initializes the system and one democratic SU who is voted by VIPs.

Subsystem A self contained system within a larger system.

System functional requirements Requirements that describe how a software system must behave, what its features and functions are.

System nonfunctional requirements Requirements that describe the general characteristics of a system. They are also known as quality attributes.

T

TeamMe Software system that will facilitate active teaming of people with similar interests and skill-set to forge groups for a certain do-good project.

Tkinter Is a Python binding to the Tk GUI toolkit. It is the standard Python interface to the Tk GUI toolkit, and is Python's de facto standard GUI.

U

Use-Case diagrams A form or drawing to display a system or software behavior that is under development. It contains both visual and textual representation, including entities that are involved in the system.

User hierarchy Organized structure of users by their rank (level of access).

V

Visitor Not a registered user, can surf around the application, and isn't logged in.

VIP (very important person) OUs whose reputation scores exceed a threshold set by SU and have more authorization than OUs.

W

Window Is a framed area on a display screen for viewing information or content.

Windows 10 (Microsoft Windows 10) Series of operating systems produced by Microsoft.

X

Y

Z

Index