

Contents

TeamMachine

A Tool for Connected Collaboration

If you are reading a printed or [PDF version](#) of this document, the current online version may be found at <http://bit.ly/teammachine-reqs> (UC LDAP login may be required).

Introduction

In today's fast paced world, it can be very challenging to network and meet new people with similar interests and experiences with technology and development. Developers may be working on very similar projects or may need assistance with a project, but have no readily accessible means of meeting other developers with knowledge and experience or find mentors to help further refine their own skills.

TeamMachine is a web service and Android mobile application which aims to help connect developers with similar interests in a collaborative environment would promote faster development and shared research, and build upon existing research.

These interests could be highlighted through a short survey, some options to like different projects, and even education and employment backgrounds. Users would also be able to form collaborative efforts on different projects with other users in the nearby area, stimulating development of new technology and other design applications. In essence, the application would be used to connect "Think Tanks" within local areas to collaboratively share knowledge and development with each other.

Scope

The TeamMachine project will initially target software developers in a single municipality (the city of Cincinnati, OH) who wish to find other programmers to participate in small-group collaborations.

The fundamental idea behind TeamMachine could, however, be easily expanded to different applications, such as art, music, web design, system design, and many other research and development activities. Users would be automatically matched up based on the projects/ideas that interest them, and can communicate with groups from within the application. Users would also be able to join a public project, such as "3D Printing," but could also join a private collaboration within 3d printing that aids in the development of individual projects and communication.

Objectives and Success Criteria

The objectives of the TeamMachine project are to:

- Facilitate communication between users based on projects/ideas that rank according to their provided interests profile
- Create a system of peer rating and credentialing which can be used to certify individuals' skills and evaluate their suitability for collaboration
- Provide basic groupware functions such as email discussion lists or live chat*

**Stretch goals: only if primary functionality is implemented ahead of schedule*

References

1. *Object Oriented Software Engineering Using UML, Patterns, and Java*, 3rd ed., Bernd Bruegge and Allen H. Dutoit

Current System

There is currently no single industry-leading service which provides a collaborative platform for creators seeking to recruit others to help with their projects. At the moment, creators are using a variety of different services to meet the collective needs which we intend to provide with TeamMachine.

Examples:

- Google Chat, GroupMe, HipChat, Campfire, and other IM services for real-time group collaboration
- Dropbox, Box.net, Google Drive for file sharing
- Facebook, Twitter, LinkedIn for social networking and outreach
- [Kickstarter](#), [Indiegogo](#) for crowdfunding and payment processing
- [Freelancer.com](#), [oDesk](#) (and many others) for hiring freelancers for small projects
- [Duet](#), [Asana](#), [Basecamp](#) (and many others) for project management

Separately, these services will exceed TeamMachine's capabilities for file sharing, business-related networking, job postings, hiring freelancers, and social media outreach. However, these services have not yet been implemented in a unified way as a single online service, with a central sign-on and permissions management.

Proposed System

Overview

We believe TeamMachine will differentiate itself on the market by providing a unified platform for makers, creators, and entrepreneurs to come together on and eventually bring to market hardware and software products, as well as artistic and civic collaborations.

Definitions, Acronyms, and Abbreviations

Acronyms Used in This Document

1. **JSON:** *JavaScript Object Notation*
2. **REST:** *Representational State Transfer*
3. **API:** *Application Programmer Interface*

TeamMachine Vocabulary

1. **Project:** the central focus of the TeamMachine site; a collaborative effort to produce some desired end result, which may be a hardware or software product, a work of physical or performance art, or a civic activity
2. **User/Member:** a person who uses the TeamMachine mobile app or web site while signed in to a registered account; a collaborator on an existing project
3. **Profile:** a collection of information about a User which lists some biographical information as well as a list of past projects and competencies
4. **Skill:** an area of emphasis or competency for which a project leader is seeking collaborators; an area of focus or competency which a User lists in his or her Profile
5. **Project Admin/Owner/Leader:** the User in charge of accepting applications to his or her project and assigning responsibilities to project Members
6. **Role:** a delegation of responsibilities by the Project Owner to Members of the project

Functional Requirements

The mobile device application will be the primary “point of entry” for new users of the TeamMachine service. Initially, the Android platform (over 70% of the smartphone market) will be the primary target for development of the mobile application. The Android app should provide the following functionality:

- **Search**

- User must be able to search for Projects by any of the following criteria:
 - * Project name
 - * Project location
 - * Stage of project: Working, Not Started, Completed, (Archived)
 - * Skills required by project
 - * If the project has open roles and is seeking collaborators
 - * Type of project: software application, web site, robotics experiment, etc.
 - * Project members and project leader
 - * Age: how long the project has been established and progressing toward completion
 - * Project creation date
 - * Last Used
 - * Group Size
- User must be able to search for other Users by any of the following criteria:
- Geographic location/proximity: in a ZIP code or municipality, or within a certain radius of the User's location

- **View and apply to Projects**

- User must be able to apply for projects
- Users must be able to view detailed information about the project, including but not limited to:
 - * Project Name
 - * Project Location
 - * Project Events
 - * Project Members
 - * Project Roles
 - * Both available and filled
 - * Project Description
 - * Project Type
 - * Project Skills
 - * Goal of the Project
- Application must support “Apply” functionality
 - * The Project Admin must be notified when an application is submitted
- Users must be able to watch/follow projects (stretch goal)
 - * Notifications: when a position opens or project changes state

- New events
 - Registration deadline extended
 - Open positions
- **Create Projects**
 - Users must be able to create projects
 - Everything in Manage Project
- **Manage Projects**
 - Project owners must receive notifications of user applications
 - Project owners must be able to accept/deny applications
 - User Profiles (or Activity Stream) should show feedback when a user’s own applications are accepted/denied
 - * If accepted, user is automatically added to Project
 - Project owners must be able to edit their projects
 - Project owners (or delegates; *see Roles*) must be able to create events for their projects
 - Projects must support both private and public fields
 - * Every Item under every field must be able to be marked public or private, except for:
 - Name
 - Description
 - Skills
 - Type
 - Rating
 - Location
 - * Private fields will not match any search
 - Users should be to invite participants directly
 - Project owners should be able to delete their own Project
 - Project owners should be able to close/finish project
 - * Optionally (or automatically) “archive” the project, hiding it from basic search results
- **Search for Profiles**
 - Users must be able to search for other users by criteria, including but not limited to:
 - * Skills
 - * Location
 - * Name
 - * Status (“other metadata”)
 - Looking to join a project

- Completely free
 - Not looking (inactive)
- Star/Watch User (*stretch goal*)
 - * Users with an existing relationship may be notified when:
 - they create new projects
 - they join projects
 - they attend or host an event
- **View a Profile**
 - Users must be able to view profile details for other users
 - Profiles should show the number of projects done with a given skill
 - Profiles should showcase previous work as much as possible (non-functional)
 - Users must be able to contact other users
 - Email
- **Edit Profiles**
 - Users must be able to edit their profiles
 - Users should give a “level” to their skills
 - Users should be able to make their profiles private (invite only) or searchable
- **Create a Profile**
 - Users must be able to create their profile from the app
 - * all of “Edit Profiles”
- **Manage Project Roles**
 - Users will be able to add roles to their projects
 - Roles will contain
 - * A name
 - * Skills required to fill the role
 - * A description of the role
 - Be able to be filled by a user
 - Users must be able to add other users to fill a role

Non-functional Requirements

*Note that a web-front end (a publicly accessible web site where project owners can manage their projects) is **not** considered an initial goal of the first iteration of the TeamMachine service.*

Usability

- Profiles on the service should emphasize credibility by showing skills and project history
- Project owners should feel like they are hiring competent people
- Every page should be accessible to from anywhere within the interface in as few possible clicks/taps as possible
- Profile images should be encouraged to professional

Reliability

- Project should be hosted on a scalable infrastructure which anticipates very large scale growth in a short period of time
- Project should have monitoring and notification systems in place to inform developers of problems and downtime

Performance

- Search functionality should be prominent and easily accessible in both the Android and web interfaces
- The mobile application should be easily navigated with the fewest clicks possible
- The web front end should utilize Ajax to push page “fragments” rather than whole page updates, and maximize the use of browser cache and HTML5 offline storage APIs to improve responsiveness and resiliency to periods of disconnected use

Supportability

- Application should be developed on Android and target the widest range of form factors and screen sizes possible
- Allow users to use their own email program for project messaging

Implementation

- The mobile application should work on the widest range of current-generation Android devices possible (Android 3.x and up)
- The back end should utilize RESTful APIs to allow for future integrations and third-party hooks

Interface

- The application UI and web site should look and feel professional
- The mobile application should place notifications in the notification bar and/or lock screen for the targeted mobile platform(s)

Legal

- Service should have clear legal guidelines on who retains copyright to creative works submitted to the service
- Service should encourage works of software to be submitted under an [OSI-approved open source license](#); documentation and other creative works should be encouraged to use [Creative Commons](#) or [GNU FDL](#); allow “all rights reserved” copyright assignment as an optional fallback
- Service should have clear stipulations about privacy and disclosure of user data or personally-identifying information to law enforcement or other authorities

Secondary Requirements

The following requirements are considered “stretch goals” which are only to be undertaken if primary goals are on schedule for completion in the required time frame.

Data Export Users should be able to export their project data (membership, contact information, role assignments, shared files, communications logs) in an accessible format should they desire to leave the service at any time

Messaging Users must be able to contact other users 1. IM 2. Private messaging 3. Group Mailing lists (address to mail to) 4. auto-completing @-replies in comment/discussion areas on the site

Create a project After creating a project, ‘Tinder’ style matching for requesting participants to join the project

1. Swipe Left, Right style
2. Not deep review of users

Activity Feed Follow other users, view their project participation

Achievements An achievement system may be put in place to encourage user engagement and increase retention; see [Mozilla OpenBadges](#)

Project Management

1. Delegate Roles
2. Create Roles

Integrations

1. GitHub
2. Cloud storage (Drive, Dropbox, etc.)

Profiles/Professionalism/Community Regulations

1. Flag bad profiles (“Flag as inappropriate”)
2. Use Gravatar for user avatars (API allows you to restrict profile pictures to only “G” rated)

System Models

Scenarios

Scenario #1: New User A user is a new user to the application, so downloads the application to their Android device and runs the application. The first page asks the user to log in or create a new account, so the user creates a new account within the application with their email address and a password. The android device then locates the user’s GPS coordinates and adds them to the local users (the coordinates are updated each time the application is loaded).

After setting up a new account, the user is prompted to add specific information about research and development they have been involved in, along with personal interests and availability types (personal projects, business projects, research projects, remote projects, etc.). The user finishes the sign-up process and is able to access the next screen which gives the following options:

1. Create a Project
 - A user may post new ‘projects’ that are looking for certain experienced developers within the area that are interested in cross collaboration. The posting user automatically becomes the ‘project manager’ and can delegate that role to other users in the group. Posts are visible to users who have indicated experience in certain areas of development,

and may be listed as paid or unpaid based on the posting user's information.

2. Search For Project

- A user may search for 'projects' in the local area that they have listed experience in one or more of the areas that the project covers. Users can send a private message to the 'project manager' and can be added to the 'project team' by the 'project manager'

3. Collaborate with project

- Users can collaborate privately within a project
- Post messages
- Send group updates
- Post files/pictures/etc
- Manage Project (for Project Owners or delegates)

4. Update profile

- Avatar
- Interests
- Skills

5. Collaborate with Community

- Create and communicate with subgroups based on common interests
- Users can post similar to forum style, but designed for mobile
- Allow users to ask and answer questions
- Forum points (Achievements)

Use Case Model

Primary Use Cases Create a Profile

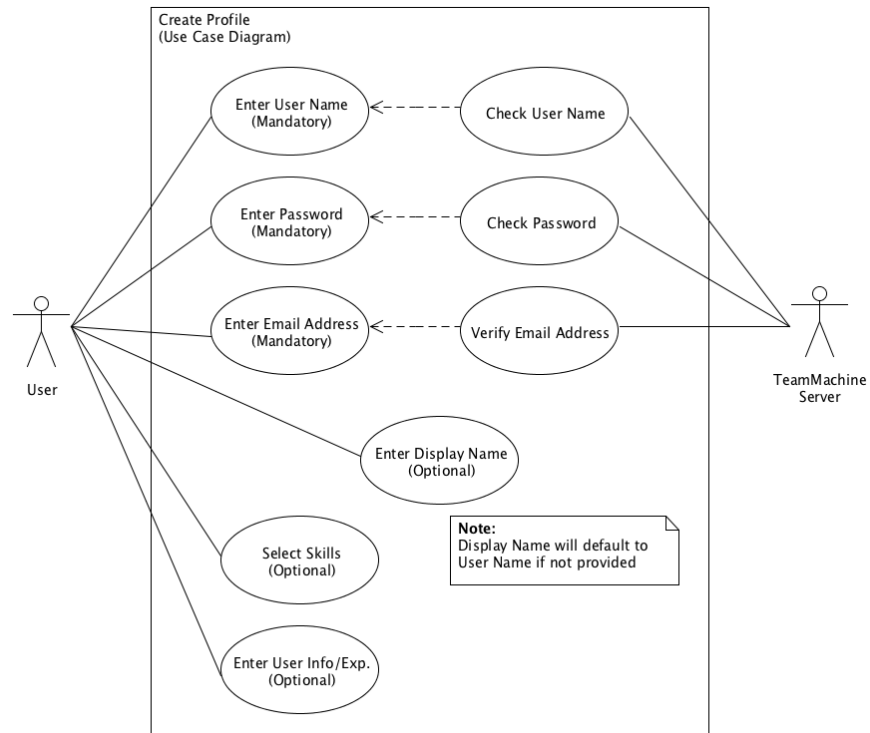


Figure 1: Create a profile use case

Create a Project

Search for and Apply to a Project

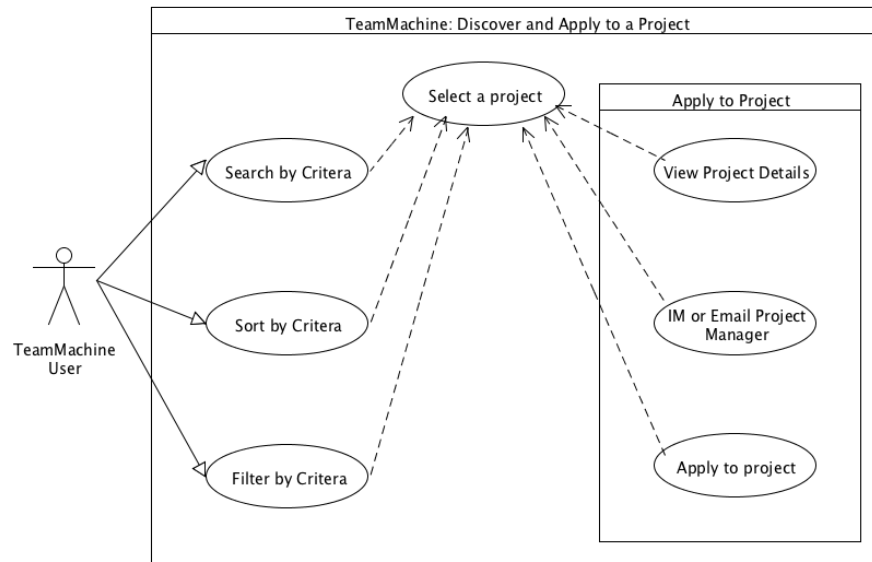


Figure 2: Search and apply to a project use case

Hire User(s) for Project

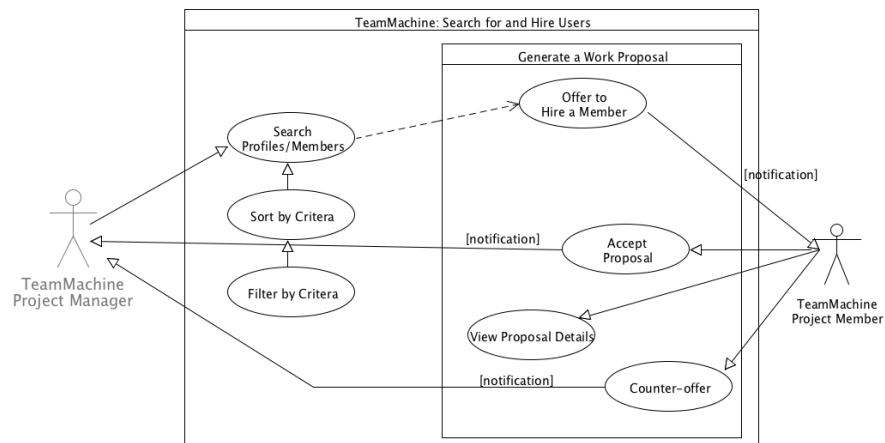


Figure 3: Search for users and hire for project use case

Search for User

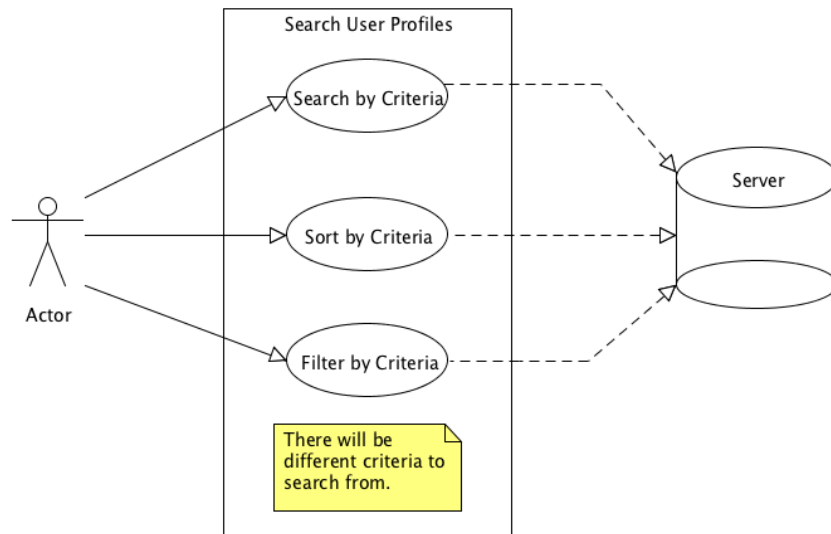


Figure 4: Search user profiles use case

View Project

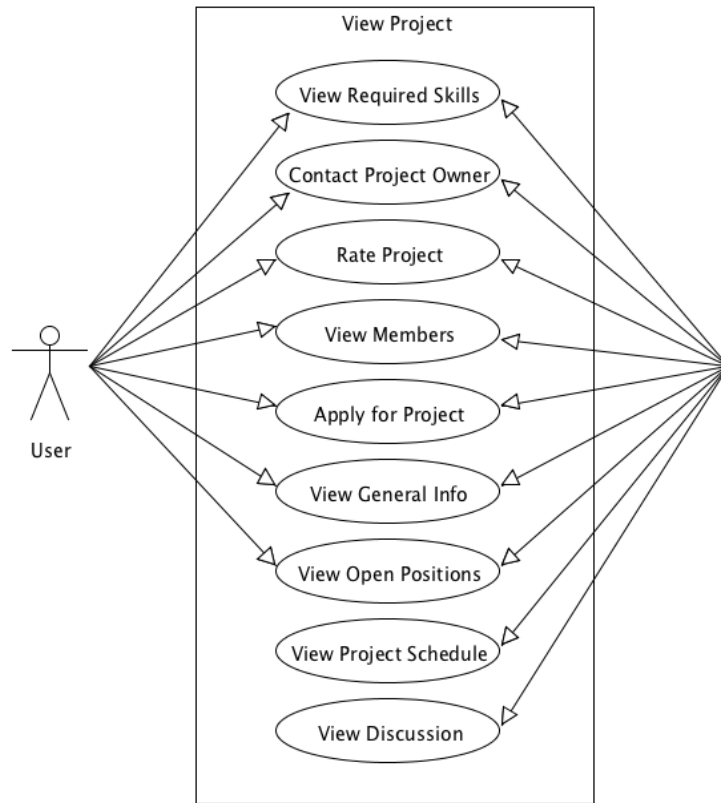


Figure 5: View project use case

View Profile

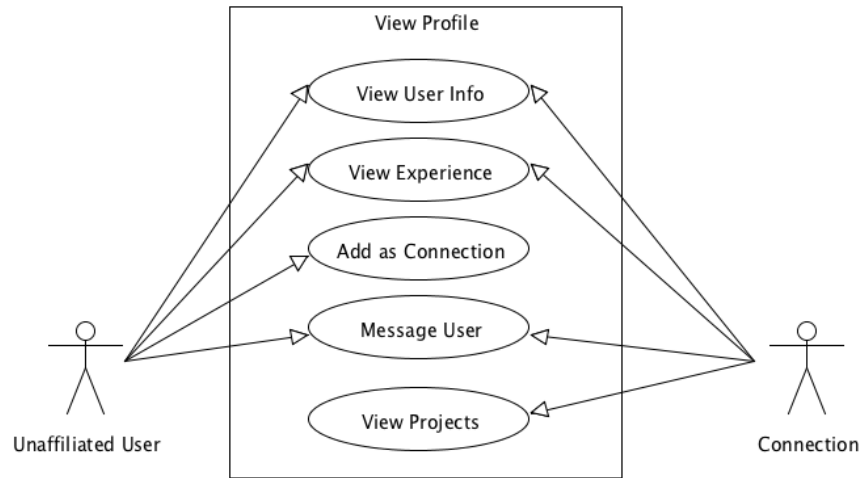


Figure 6: View profile use case

Edit Profile

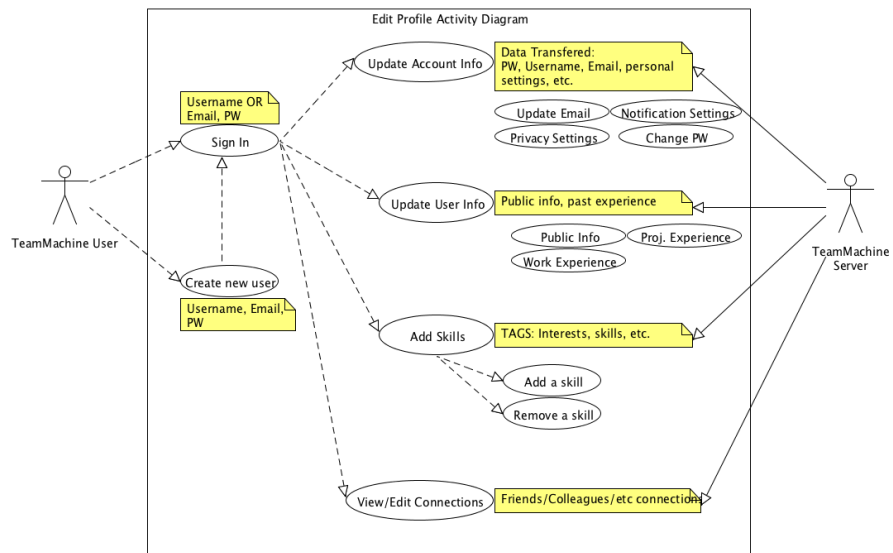


Figure 7: Edit user profile use case

Edit Project

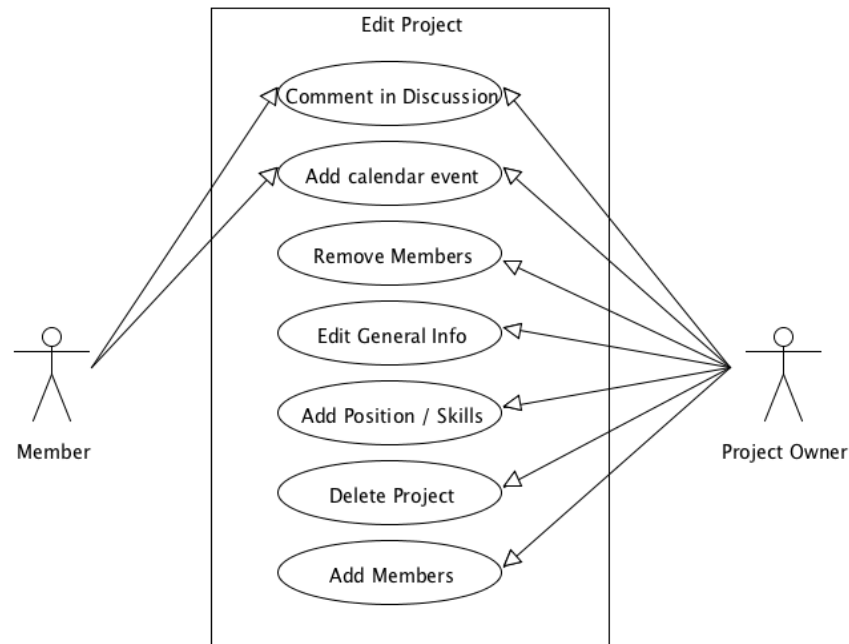


Figure 8: Edit project use case

Create Project Role

Secondary Use Cases *These are “stretch goals” which are not primary aims of the first iteration of the TeamMachine product.*

- Contact User from Profile Page
- Contact User from Project Page
- Add/Modify Events
- Export Calendars
- Subscribe to a Project / Follow a User
- Rate a Project
- Thumbs-up/down post or comment
- Give Feedback to Project Owner
- Vote (on Project Decisions)
- Forum Use Cases
 - View Post

- Search Posts
- Create Post

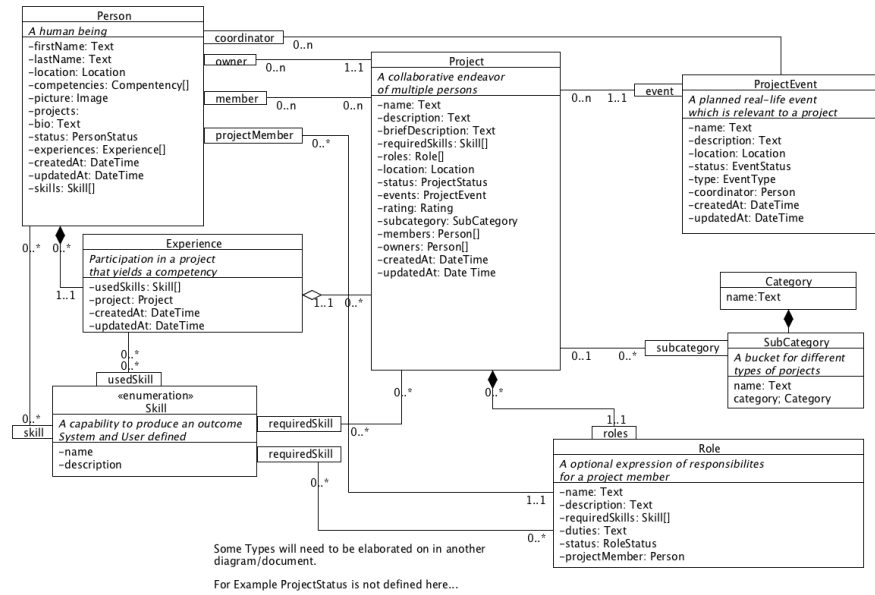


Figure 9: TeamMachine class overview diagram

Object Model

Field Name	Public/Private	Required
1. Name	Public	Yes
2. Description	Public	Yes
3. Brief Description	Public	No
4. Skills involved	Both	Yes
5. Roles	Both	No?
6. Location	Both	Yes
7. Status	Public	Yes
8. Events	Both	No
9. Rating	Public	N/A

Field Name	Public/Private	Required
10. General Type	Public	Yes
11. Sub Type	Public	No
12. Members	Both	Yes
13. Project Lead/Admin	Public	Yes
14. Goal	Public	No
15. Creation Date	Public	N/A
16. Last Updated Date	Public	N/A

Project Object Fields

1. Name
2. Description
 - In depth description of the project
3. Brief Description
 - Description given when viewed from search
 - If not given the first x characters will be taken from Description for this field
4. Skills needed in the projects
5. Project Roles
6. Project Location
 - The general location the project will be worked on.
 - Given at a city level
7. Project Status
 - Not Started, Working, Complete
8. Project Events
 - Events related to the project
 - See also: Events fields
9. Rating
 - The user given rating of the project
 - The rating is not set but gathered over time
10. General Type

- The general category that the project belongs to
 - Hardware, Software, etc..
- 11. Sub Type
 - More detailed type.
 - C++ Applications, Webserver, etc...
- 12. Members
- 13. Project Lead/ Admin
- 14. Goal
 - General goal of the project.
 - “We want to learn what goes into building a Web server”
 - “We want to produce an application that produced revenue”
 - “It’s for a school project and we need X expertise” (e.g., DAAP/CEAS collaborations)
- 15. Creation Date
 - The date the project was created
- 16. Last updated date
 - The last date when the project was updated

	Field Name	Public/Private	Required
1.	Name	Public	Yes
2.	Email	Either	Yes
3.	Location	Either	No
4.	Skills	Public	No
5.	Profile Picture	Public	No
6.	Projects Joined	Either	N/A
7.	Bio	Public	No
8.	Status	Public	Yes
9.	Previous work	Either	No

Profile Object Fields

1. Name
2. Email
 - Signup email

3. Location
 - Location of user
4. Skills
 - The skills of the user
 - Will have a level associated with them
5. Profile Picture
6. Projects Joined
 - A list of all the projects the user has joined
 - Must be part of the project for at least x% of the project life for it to show up
7. Bio
 - Description of the user, similar to resume skills list
8. Status
 - The status of the users project hunt
 - Looking for project, Not looking for project, etc...
9. Previous work done by the user
 - Any work the user wants to share on their profile

	Field Name	Public/Private	Required
1.	Name/Job Title	Public	Yes
2.	Description	Public	Yes
3.	Skills Required	Public	No
4.	Duties/Responsibilities	Public	Yes
5.	Filled	Public	Yes
6.	Filled by	Private to non-members	No

Role Object Fields

1. Name/Job Title
 - The name or title of the role
2. Description

- A brief description of the role and what it entails
3. Skills Required
 - The skills that will be required of the user to fill this role
 4. Duties/Responsibilities
 - What the user will be doing if they take this role
 - Should be detailed
 5. Filled
 - If the role is filled or not
 6. Filled by
 - Whom the role is filled by

	Field Name	Public/Private	Required
1. Name		Public	Yes
2. Description		Public	Yes
3. Level of Expertise		Public	No

Skill Object Fields

1. Name
 - The name of the skill
2. Description
 - Very brief description of the skill
3. Level of expertise
 - Level of expertise with the skill

	Field Name	Public/Private	Required
1. Name		Public	Yes
2. Description		Both (default to “private” for members-only events)	Yes

	Field Name	Public/Private	Required
3. Location	Both (default to “private” for members-only events)		Yes
4. Status	Public		Yes
5. Event Type	Public		Yes
6. Attendance	Public		Yes (but default to “open”)
7. Event Coordinator	Both (default to “private” for members-only events)		Yes
8. Creation Date	Public		N/A
9. Last Updated Date	Public		N/A

Event Object Fields

1. Name
2. Description
 - In depth description of the event
3. Location
 - Where will project members meet for the event (online or IRL)
4. Status
 - Completed (archived; do not show in “basic” search results)
5. Event type
 - Social or work oriented
6. Attendance
 - Members-only (private, registration required)
 - Open attendance
 - Entrance fee/cover?
7. Event Coordinator
 - Point of contact for directions, registering for event, what to bring
8. Creation Date
9. Updated Date

Dynamic Model

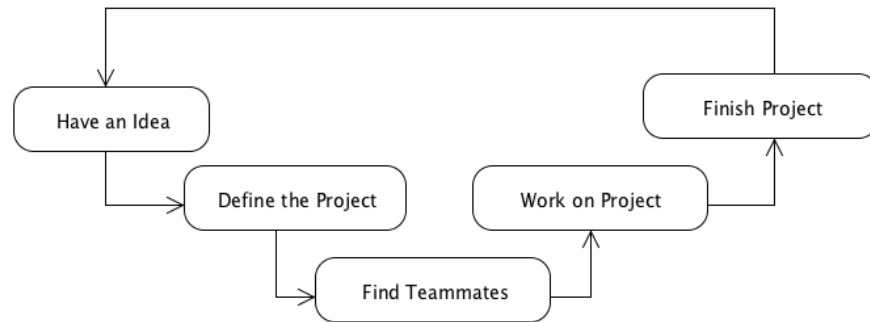


Figure 10: Overview activity diagram

Overview

Activity Diagrams Create a Project Workflow

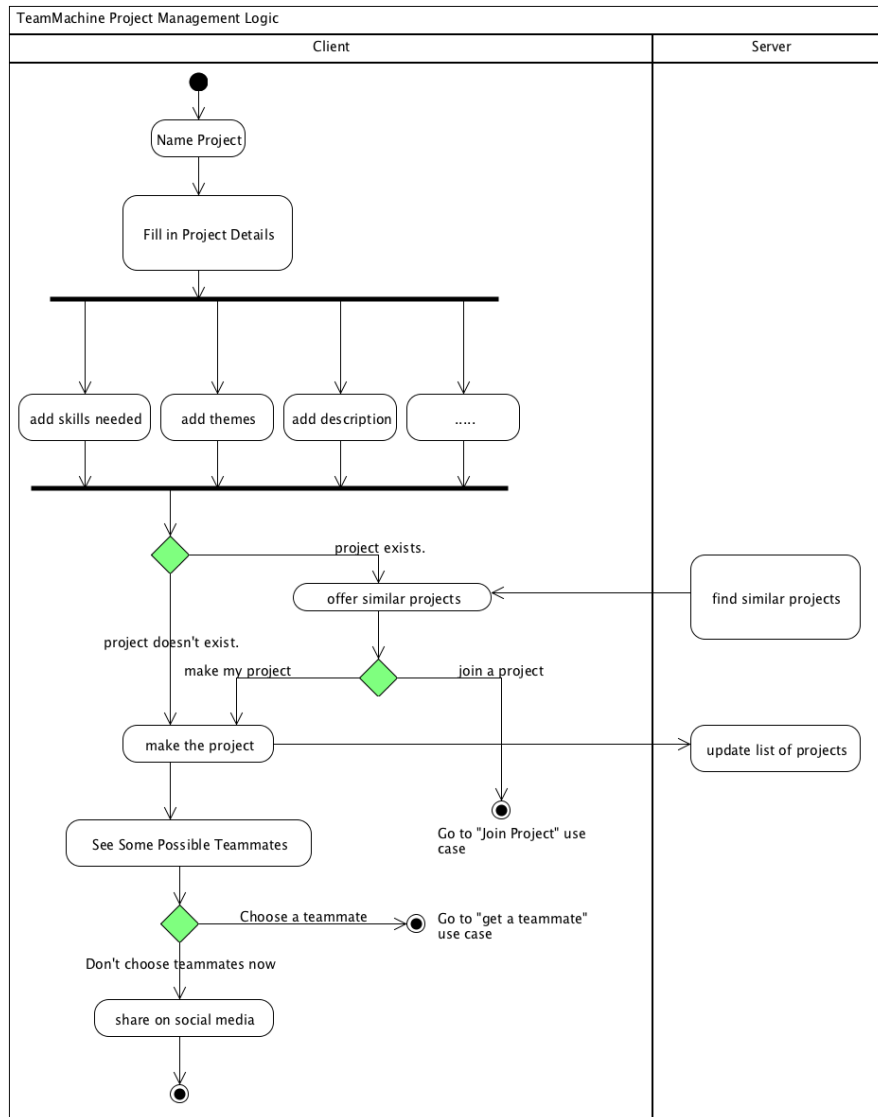


Figure 11: Create a project activity

Find and Add Team Members Workflow

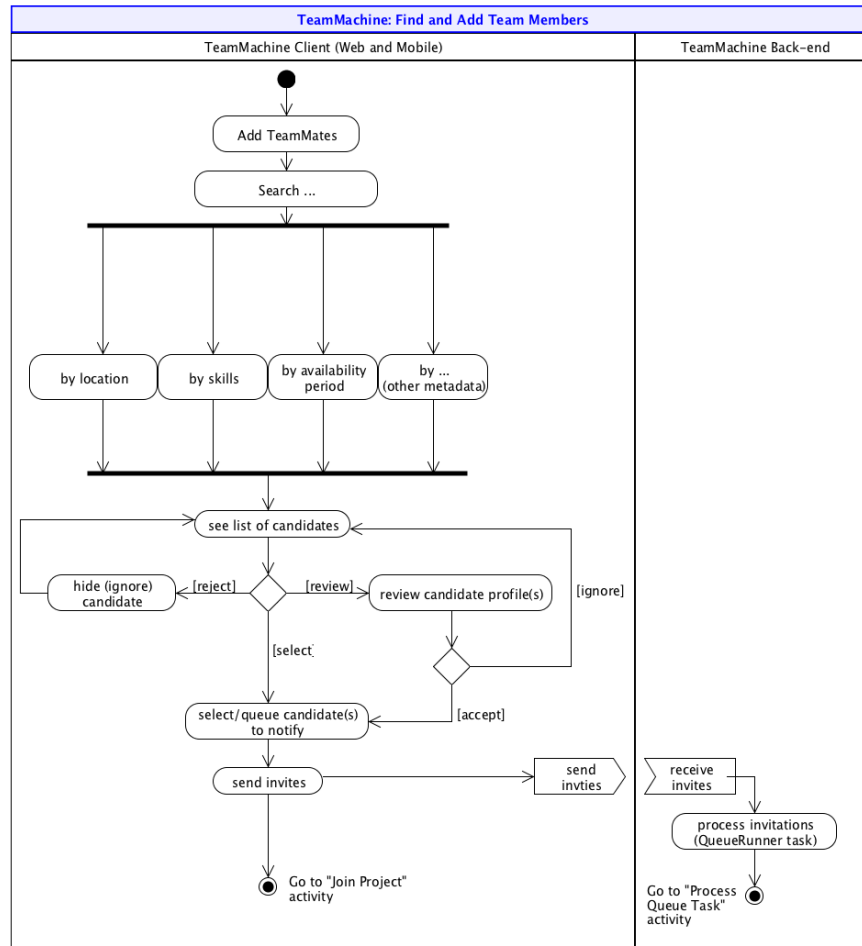


Figure 12: Find/add teammates activity

Hire a User Workflow

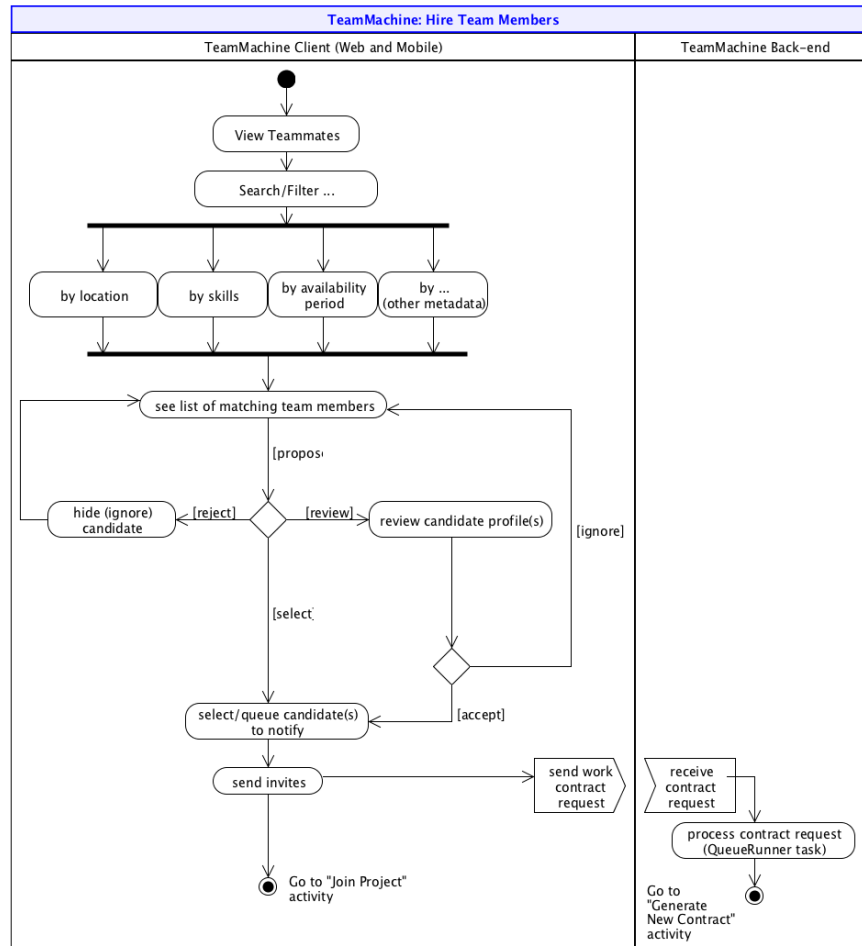


Figure 13: Hire a user activity

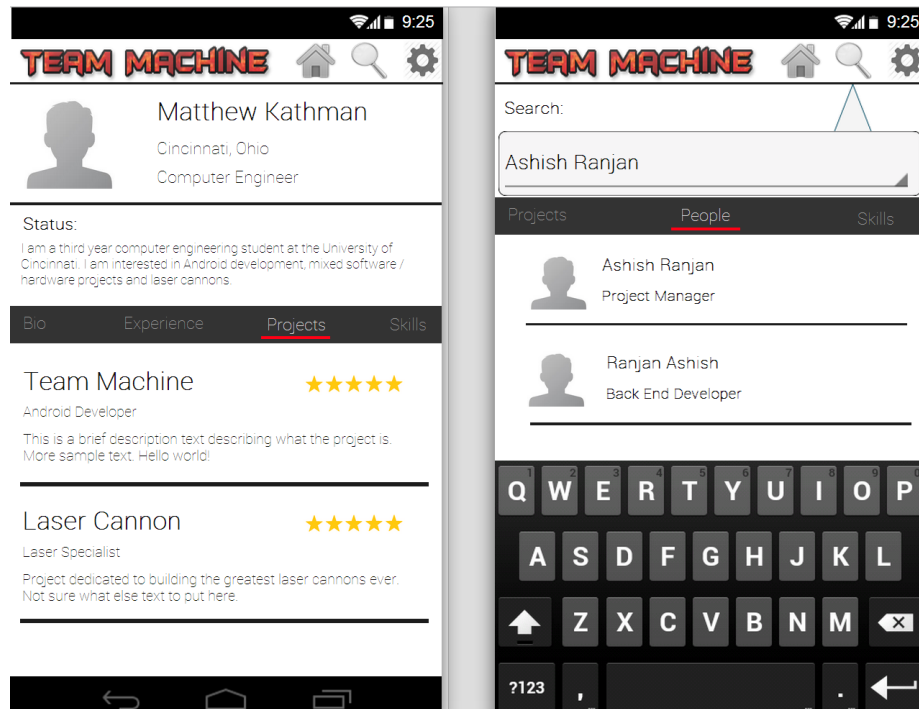


Figure 14: Version 1 of Android profile and search UI

User Interface

- [Version 2 of Android profile UI](#)
- [Android project view UI](#)
- [Android sidebar menu UI](#)

Credits

The HTML version of this document makes use of Junil Um's [markdown-toc](#), which in turn makes use of the [Tocify](#) jQuery plugin by Greg Franko,

The PDF version of this document was created with John MacFarlane's [Pandoc](#) document processing system.