

Software Specific Requirements

For

Quest Till Done

Prepared By

Mun Khong Chin

Kevin Huang

Mark Scheid

At

Drexel University

Advisor

Yuanfang Cai

Stakeholder

Rob Di Marco, CTO at eLocal USA LLC

Revision History

Version	Prepared By	Date
0.1	K. Huang	1/1/2014
0.2	The QTD Team	1/19/2014
0.3	The QTD Team	1/21/2014
1.0	M. Scheid	2/4/2014
1.1	M. Scheid	3/5/2014
1.2	The QTD Team	4/29/2014

Contents

1 Introduction	6
1.1 Purpose	6
1.2 Intended Audience	6
1.3 Scope	6
1.4 Definitions, Acronyms and Abbreviations	6
2 Overall Descriptions	8
2.1 Product Perspective	8
2.2 Product Features	8
2.3 User Characteristics	9
2.4 Requirement Apportioning	9
3 Specific Requirements	9
3.1 External Interface Requirements	9
3.1.1 GitHub API Integrations	9
3.1.2 Additional Developer API Integrations	9
3.2 Functional Requirements	10
3.2.1 Incorporating with VCS	10
3.2.2 Additional Developer API Integrations	12
3.2.2 User	12
3.2.3 Project Management	22
3.2.4 Group Mechanics	30
3.2.5 Timer Functionality	33
3.2.6 Record Management	36
3.2.7 Tagging, Filtration and Searching	39
3.2.8 Timeline Review	40
3.2.9 Global Notifications	44
3.2.10 Prioritization Functionality	44
3.3 Non-functional Requirements	46
3.4 Design Constraints	48
4 Use Cases	48
4.1 Authenticate	48
4.1.1 Logging in	48
4.1.2 Logging Out	48

4.2 Use cases for all logged in users	48
4.2.1 User Profile	49
4.2.2 VCS import.....	50
4.2.3 Manage Projects	50
4.2.4 Timer Setup	50
4.2.5 Manage Record	50
4.2.6 Search	50
4.3 Use case for administrators	50
4.3.1 Add/delete user in his/her group.....	51
4.3.2 Administrators can assign tasks to groups and users.....	51
4.3.3 Administrators can promote and demote users/administrators as administrators and QTD Group Member.	51
4.4 Use case for QTD Site Administrator	51
4.4.1 Site Online/Offline	52
4.4.2 Send global announcements.....	52

List of Figures:

[Figure 1 - Registering a Github Account](#)
[Figure 2 - User Settings Page](#)
[Figure 3 - Home Page](#)
[Figure 4 - User Control Modal Dialog](#)
[Figure 5 - Registration Page](#)
[Figure 6 - User Account Deletion Page](#)
[Figure 7 - User Metrics Page](#)
[Figure 8 - User Profile Page](#)
[Figure 9 - User Profile Page for Current User](#)
[Figure 10 - User Social Page](#)
[Figure 11 - Group Creation Modal Dialogue](#)
[Figure 12 - Project Manipulation Modal Dialog](#)
[Figure 13 - Create Project Modal Dialog](#)
[Figure 14 - Edit Project Modal Dialog](#)
[Figure 15 - Import/Export Project Modal Dialog](#)
[Figure 16 - Project Page Add Task Modal Dialog](#)
[Figure 17 - Project Page Edit Task Modal Dialog](#)
[Figure 18 - Project Page Delete Task Interface](#)
[Figure 19 - Group Overview Page](#)
[Figure 20 - Group Overview Page Edit Group Interface](#)
[Figure 21 - Timer Modal Dialog, Inactive Timer Interface](#)
[Figure 22 - Timer Modal Dialog, Active Timer Interface](#)
[Figure 23 - Add Record Modal Dialog](#)
[Figure 24 - Edit Record Modal Dialog](#)
[Figure 25 - Search Results Page](#)
[Figure 26 - User Timeline Page](#)
[Figure 27 - Group Timeline Page](#)
[Figure 28 - Project Timeline Page](#)
[Figure 29 - Priority Tasks Page](#)
[Figure 30 - Castle Siege Prioritization Overview Page](#)
[Figure 31 - Use Cases For QTD Member](#)
[Figure 32 - Use Cases For QTD Administrators](#)

1 Introduction

1.1 Purpose

This document specifies all the requirements for Quest Till Done. These requirements directly relate to features, functionalities, performance, attributes, constraints and interfaces of the system.

This program will address the lack of effective tools to keep a complete log of a programmer's development process, expanding upon existing source control and bug tracking systems. In addition, this program will provide gamification elements like achievements into the development log.

1.2 Intended Audience

The intended audience for this document is software developers, testers, and end users of the Quest Till Done program.

1.3 Scope

The scope of this project is to append another layer of logs to existing development and bug tracking tools like Git and Jira. This tool will fill the need of maintaining a more detailed state of unfinished projects and tasks, as well as providing a more detailed reference as to the process that lead to existing code. This program will include a full snapshot ability to capture a developer's whole workflow — commits, issue fix, fix time, development time, comments, and web references. In addition, metrics to evaluate the efficiency of the developer and mastery of a certain technology based on a tagging.

1.4 Definitions, Acronyms and Abbreviations

1.4.1 Definitions

Adventurer: Synonymous with user.

Adventurer Level: A count of all levels the user achieved since the creation of their account.

Campaign: Synonymous with Project.

Experience: A measure of user progress, relevant to Adventurer/Recent Level as well as Skill Points. Higher levels require more experience to level up.

Encounter: Synonymous with Time-block.

GitHub: Web based hosting service that uses Git revision control.

Group: A collection of users and their shared projects.

Level: A measure of the user's effort invested in projects in QTD.

Link: A record that represents a specific website, like a bookmark for browser.

Record: A fundamental data for annotation of tasks. This could be a note, link, commit, or image to provide additional log information.

Project: A project is a group of tasks that are strongly connected to each other by dependencies. The top-most task in the dependency hierarchy defines the project name and details.

Recent Level: A count of all levels achieved by the user in a rolling 30 day period.

SCM: Source control management system, a system that is responsible for management of changes or revisions for computer programs. This term is interchangeable with VCS.

Skill-points: A representation of a user's time investment in a given task type, such as a programming language, on a scale of 0-20, measured by reading tags on given tasks.

Tag: A label for classifying tasks and record.

Task: A task is some discrete, actionable item, the building block of the projects.

Timer: A timer functionality that allows the user to work on tasks for the duration of a time-block.

Time-block: A user defined duration of work for grouping user progress on tasks during the given time period.

QTD Site Administrator: One user account with maximum permissions, able to perform maintenance on member accounts as well as the QTD server and database.

QTD Group Administrator: A specific user who manage groups of QTD Members.

QTD Member: Registered users of QTD, with no special permissions.

QTD Group Member: A user registered as a part of a given group.

Quest: Synonymous with Task.

User: The generic term for a registered user of QTD. This includes QTD Members, QTD Group Administrators, QTD Group Members, and QTD Group Administrators. Will be used wherever the difference between these groups and their permissions are not relevant.

VCS: Synonymous with SCM.

Workflow: Representing all process of a work, including all setups, intermediate development, formal and informal work related to given task.

1.4.2 Abbreviations

EULA End User License Agreement

QTD Quest Till Done

SCM Source Control Management

VCS Version Control System

2 Overall Descriptions

2.1 Product Perspective

Students, especially those in programming related field such as Computer Science and Software Engineering, are often working on multiple projects or programs for multiple classes all the time. As a student developer this requires a source control system as well as an issue tracking system for each of the projects. QTD will take the approach of unifying these source control and issue systems with additional tools for viewing relationships between tasks, time since last activity, and annotation of the process of development across all current and past projects for a user. Small groups of developers will also benefit from the QTD by managing multiple projects and providing more detailed logging for users maintaining code written by others.

The program proposed in this document will address this issue by utilizing existing SCM tools like Git log, Jira log and more to build a comprehensive development log. With such log, the user will be able to acquire a deeper understanding of their achievements by viewing the provided performance metrics, better organized for completed tasks by keeping all projects in one place, simplifying the juggling of multiple projects by listing pending/important projects side by side.

2.2 Product Features

When completed, QTD's features will include:

- Integration with GitHub and other SCM hosting options through use of their respective APIs
- Deep development documentation, recording though process and references alongside commits
- Full Text search for projects and tasks
- User metrics to measure productivity
- Social tools to manage groups and encourage cooperation and competition
- Shared record of development history to simplify extended code maintenance
- In-depth filtering to provide a quick overview of a user's priorities
- Timeline review, allowing a user to reconstruct the thought process that lead to current codebase
- Device agnostic

2.3 User Characteristics

The users of QTD will primarily be Computer Science students and small software development teams (comprised of 2-15 developers). The targeted audience will be able to incorporate their existing VCS system and add additional information after they link their existing workflow with QTD.

2.4 Requirement Apportioning

Priority Description

- 1 This is the highest priority level; all requirements of this level must be fully satisfied and verified in order for the software system to be released.
- 2 Requirements of this priority are not expected to be fully implemented in the 1.0 release of QTD. The nonfulfillment of these requirements must not create hazards for the system or compromise core functionality. However, these requirements are significant to the overall design and goals of the system, and indicate areas that will be expanded as soon as resources permit.
- 3 These requirements are of the lowest priority and are not within the current scope of the system design. These requirements are included to show a more complete overview of the system's future capabilities and the direction of design and development.

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 GitHub API Integrations

3.1.1.1 Purpose: The external GitHub API will provide Git integration and issue assignments with QTD. Priority 1

3.1.1.2 Input: The input will be created by QTD based on user's project name and GitHub credentials. Priority 1

3.1.1.3 Output: The output will be the set of commits and issues pulled from GitHub, converted to records and tasks in QTD program. Priority 1

3.1.1.4 Data Format: The format will be JSON. This format is provided by GitHub and parsed by QTD. Priority 1

3.1.2 Additional Developer API Integrations

3.1.2.1 QTD will be platform independent, and additional APIs will be supported as user base

expresses needs for additional development platforms (Jira, Bitbucket, etc) as time progresses
Priority 3

3.2 Functional Requirements

3.2.1 Incorporating with VCS

3.2.1.1 GitHub Integration

3.2.1.1.1 When the user first register there will be a GitHub Integration page, as per [Figure 1](#)
Priority 1

3.2.1.1.2 The GitHub integration will be able to be added/modified/deleted, as per [Figure 2](#)
Priority 1

3.2.1.1.2.1 The user will be able to select which projects he wishes to sync with QTD Priority 1

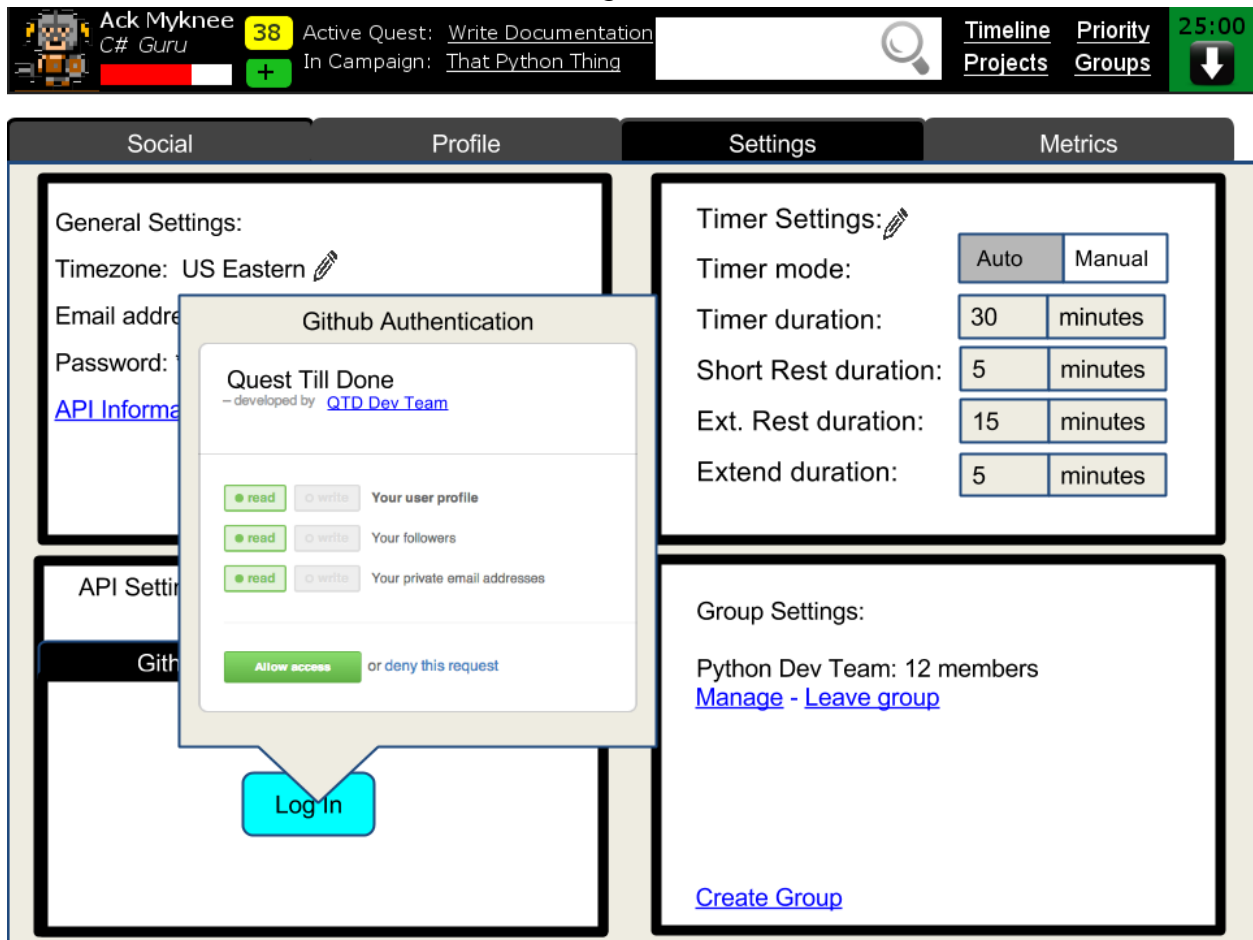
3.2.1.1.3 Once registered, the API will synchronize the following items for registered projects

3.2.1.1.3.1 Commits Priority 1

3.2.1.1.3.2 Issues Priority 1


3.2.1.1.3.3 Comments Priority 2

Figure 1



As referenced under heading(s) [3.2.1.1](#), [4.2.2](#), [4.2.6](#)

Figure 2



Ack Myknee


C# Guru

38

+

Active Quest: [Write Documentation](#)

In Campaign: [That Python Thing](#)



[Timeline](#)
[Projects](#)

[Priority](#)
[Groups](#)

25:00

↓


Social


Profile


Settings

Metrics

General Settings:

Timezone: US Eastern 

Email address: [something@website.com](#) 


Password: ***** 

[API Information](#)

API Settings:

Github

Jira

Logged in as: [that programming dude](#) 

Projects


That Python Thing

Some Java Stuff

Synced?

✓

✗

Timer Settings: 

Timer mode:

Auto

Manual

Timer duration:

30

minutes

Short Rest duration:

5

minutes

Ext. Rest duration:

15

minutes

Extend duration:

5

minutes

Group Settings:

Python Dev Team: 12 members

[Manage](#) - [Leave group](#)

[Create Group](#)

As referenced under heading(s) [3.2.1.1.2](#), [3.2.2.1](#), [3.2.2.5.1](#), [3.2.2.11](#), [3.2.5.4](#), [4.2.4](#),[4.2.6](#)

3.2.2 Additional Developer API Integrations

3.2.2.1 The software will provide settings alongside Github settings defined in 3.2.2.9 to register additional API credentials with their account, as per [Figure 2](#) Priority 3

3.2.2.2 Each such API will enable any supported subset of the synchronizations defined in 3.2.1.1.3

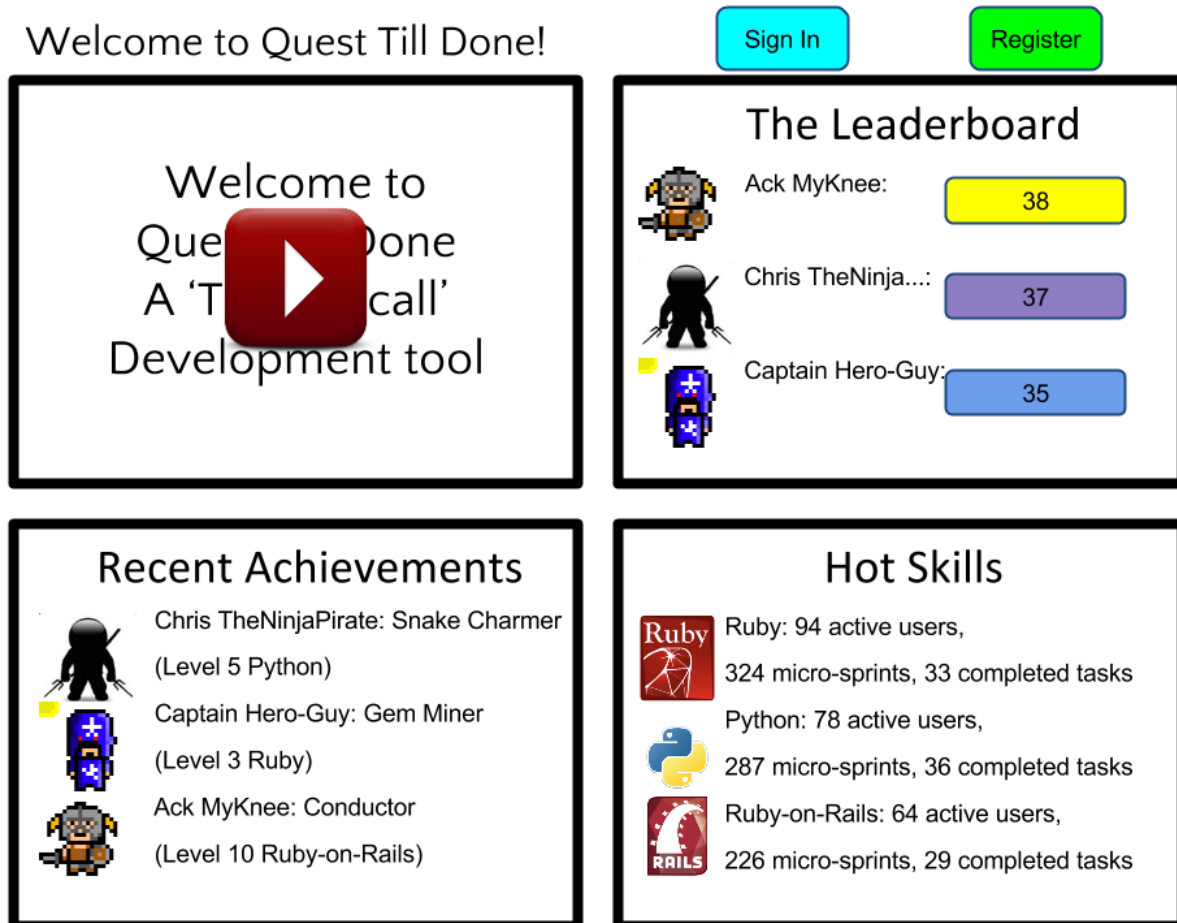
3.2.2 User

3.2.2.1 User Account Authorization

3.2.2.1.1 Users with existing user account will be able to login to the system using the login interface, as per [Figure 3](#) Priority 1

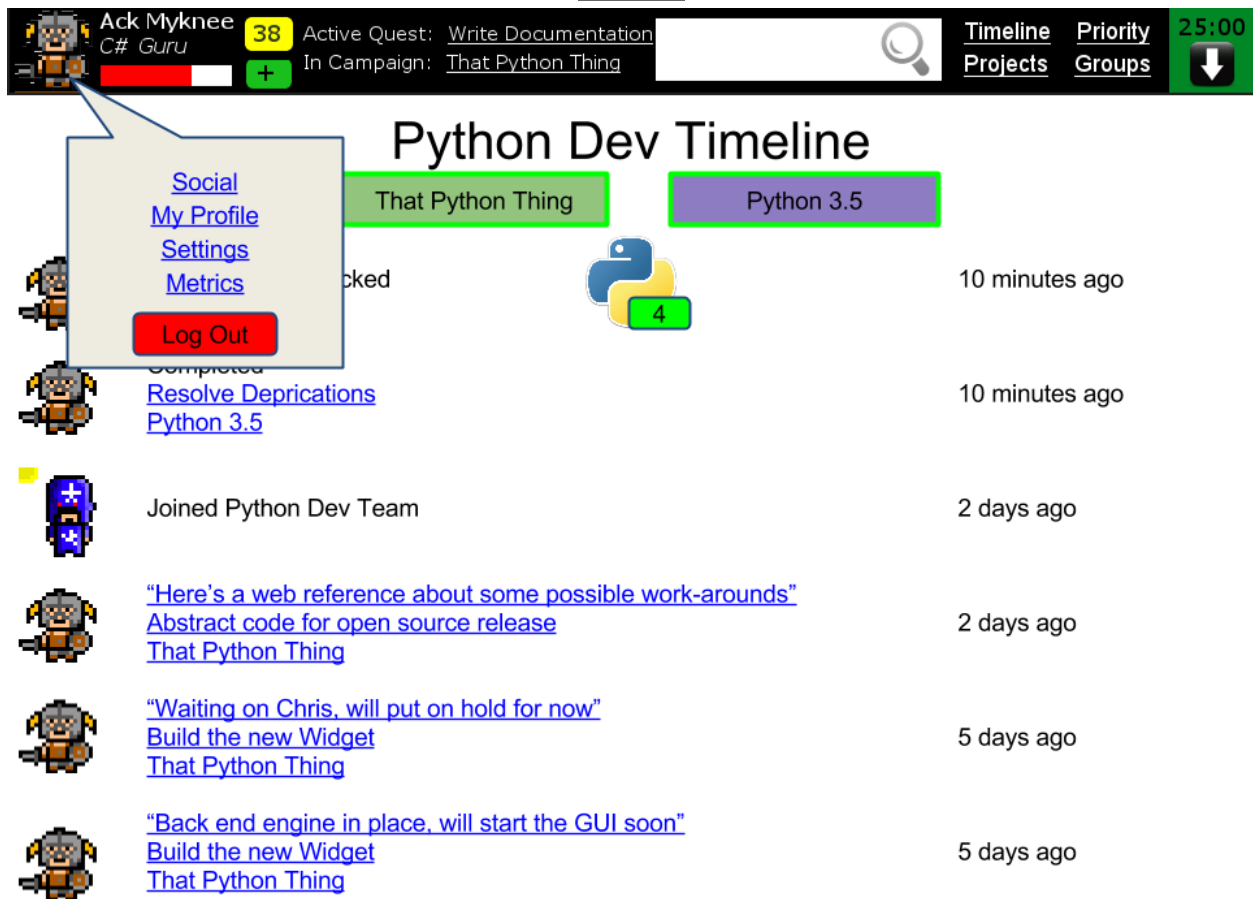
3.2.2.1.2 Logged in users will be able to logout from the system using the logout interface, as per [Figure 4](#) Priority 1

Figure 3



As referenced under heading(s) [3.2.2.1.1](#), [3.2.2.7.1](#), [4.1.1](#)

Figure 4



As referenced under heading(s) [3.2.2.1.2](#)

3.2.2.2 User Account Types

3.2.2.2.1 QTD Member Priority 1

All users without elevated permissions fall in this category. This will includes QTD Group Members.

3.2.2.2.2 QTD Site Administrator Priority 1

3.2.2.2.3 QTD Group Administrator Priority 1

3.2.2.3 User Privileges Level

The user privileges scheme will have the following requirements:

3.2.2.3.1 All users will be QTD Member and also a QTD Group Priority 2

3.2.2.3.2 All group members will be able to access group items based on their permission Priority 2

3.2.2.3.3 QTD Site Administrator will be able to manage user creation, deletion, and generate password resets. Priority 2

3.2.2.3.4 Only one QTD Site Administrator will exist. Priority 1

3.2.2.3.5 QTD Group Administrator will have access to modify group membership, member

access rights, and group-level projects. Priority 1

3.2.2.4 User Account Creation

3.2.2.4.1 QTD could have any number of users/groups created Priority 1

3.2.2.4.2 A user must have the following fields, validated prior to creation, as per [Figure 5](#)

3.2.2.4.2.1 Username Priority 1

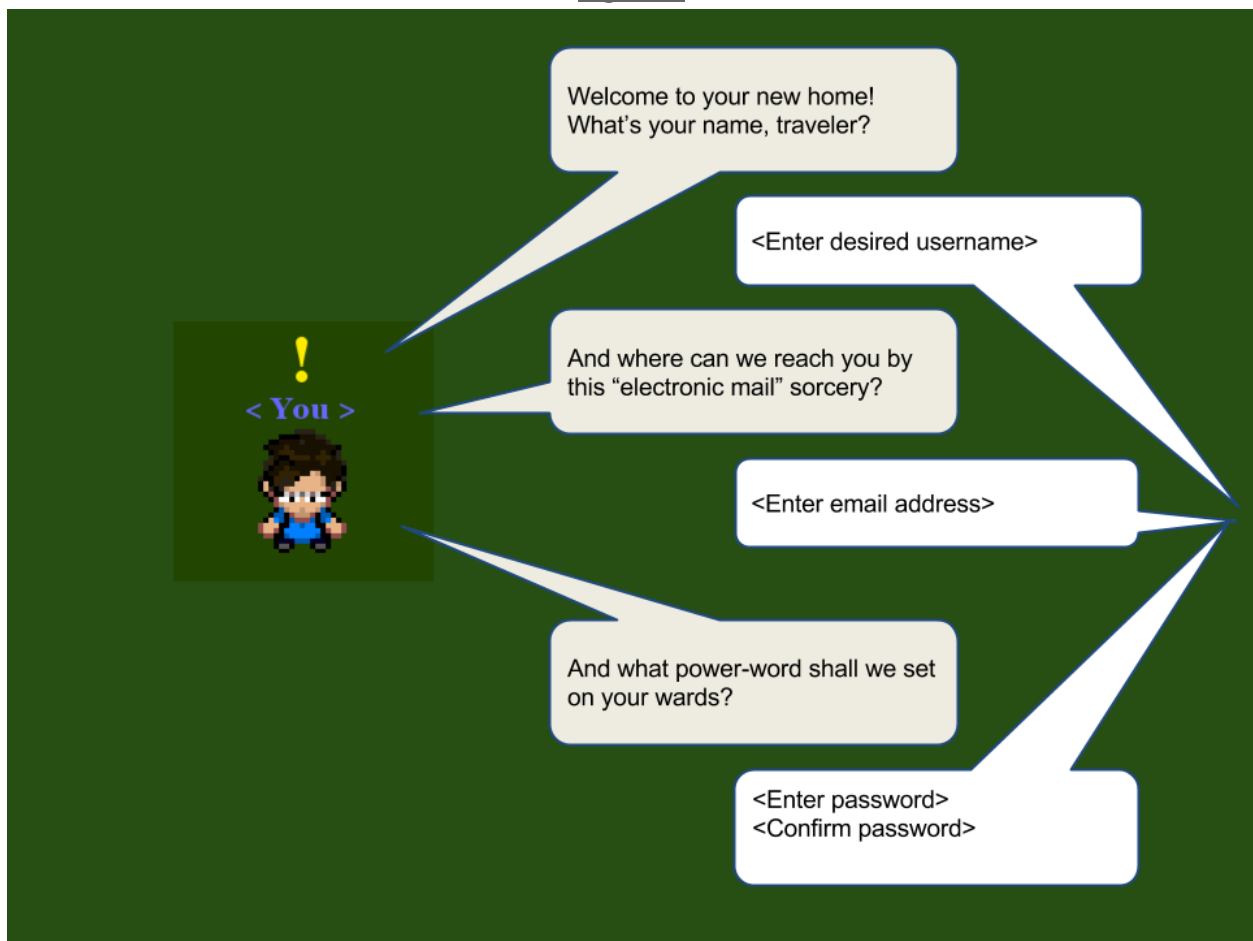
Names will be between 3- 64 characters in length, comprised of ASCII Decimal code from 33-126.

3.2.2.4.2.2 Email Address Priority 1

3.2.2.4.2.3 Password Priority 1

Passwords will be between 8-128 characters in length, comprised of ASCII Decimal code from 33-126.

Figure 5



As referenced under heading(s) [3.2.2.4.2](#)

3.2.2.5 User Account Modification

3.2.2.5.1 The following fields will be able to be modified by the user, as per [Figure 2](#) Priority 1

3.2.2.5.1.1 Password Priority 1

3.2.2.5.1.2 Email Address Priority 1

3.2.2.6 User Account Deletion

3.2.2.6.1 The user will be able to delete their account on the user setting page as per [Figure 6](#) Priority 1

3.2.2.6.2 QTD Site Administrator will be able to permanently remove a user from the system Priority 3

Figure 6

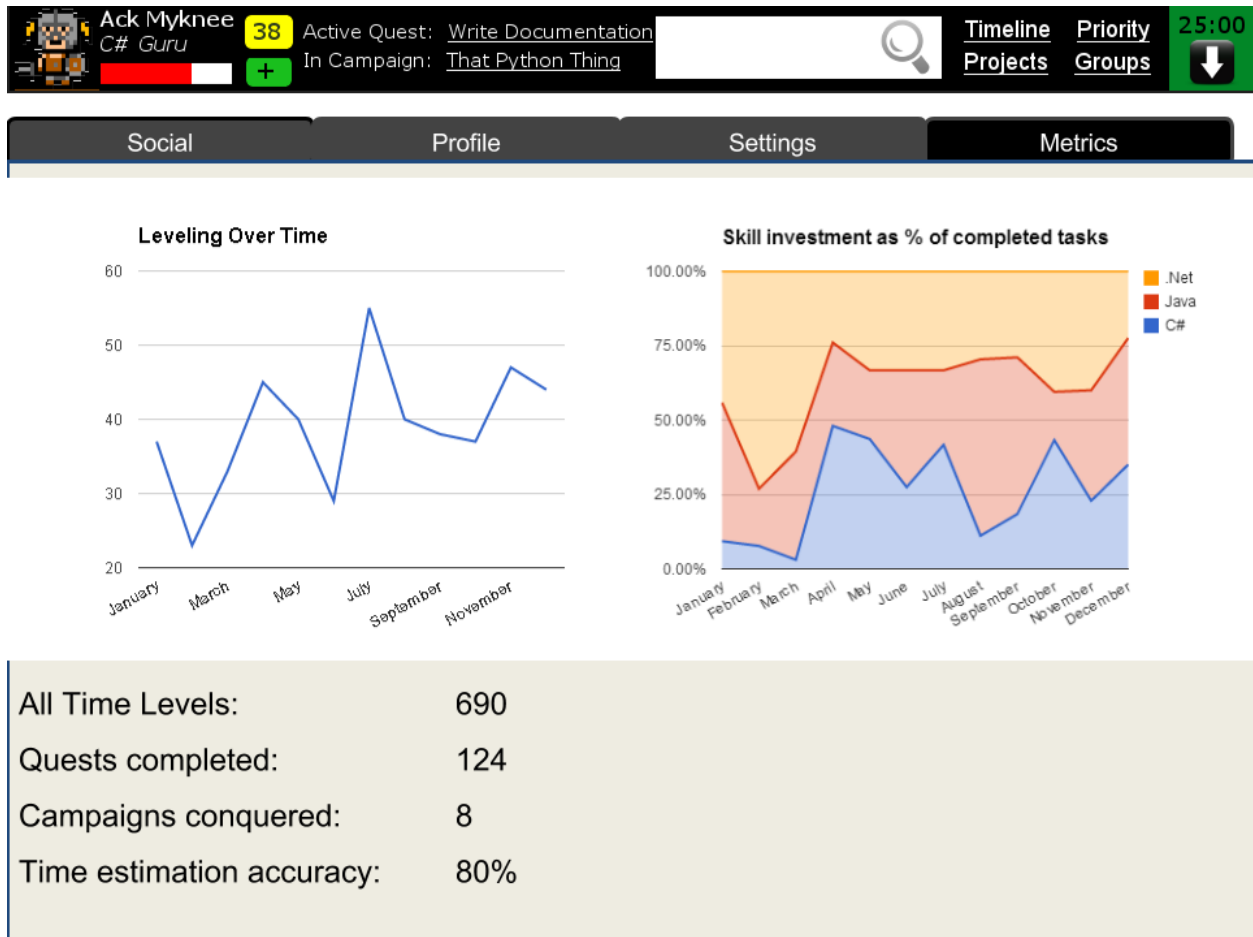
The screenshot shows a user interface for a user named 'Ack Myknee' (C# Guru, level 38). The top navigation bar includes 'Active Quest: Write Documentation', 'In Campaign: That Python Thing', a search icon, and links for 'Timeline', 'Priority Projects', and 'Groups'. A clock shows 25:00. The main content area is divided into four tabs: 'Social', 'Profile', 'Settings', and 'Metrics'. The 'Settings' tab is active, showing 'General Settings' (Timezone: US Eastern, Email address: something@website.com, Password: masked, API Information link, and a red 'Delete account' button) and 'Timer Settings' (Timer mode: Auto/Manual, Timer duration: 30 minutes, Short Rest duration: 5 minutes, Ext. Rest duration: 15 minutes, Extend duration: 5 minutes). Below 'General Settings' is the 'API Settings' section with 'Github' and 'Jira' tabs and a 'Log In' button. To the right of 'API Settings' is the 'Group Settings' section showing 'Python Dev Team: 12 members' with 'Manage - Leave group' links and a 'Create Group' link at the bottom.

As referenced under heading [3.2.2.6.1](#)

3.2.2.7 User Performance Metrics

A user account will have the following performance metrics, as per [Figure 7](#)

Figure 7



As referenced under heading(s) [3.2.2.7](#)


- 3.2.2.7.1 Adventurer level: A measure of the amount of productivity generated by the user through use of the system in documenting and completing their tasks. Priority 3
- 3.2.2.7.2 Recent level: The Recent Level measures the levels gained in the past rolling 30 day period. Priority 3
- 3.2.2.7.3 Skill Points: The Skill Points will be earned through resolving tasks associated with QTD supported programming languages Priority 3
 - 3.2.2.7.3.1 QTD supported languages: C, C#, Java, JavaScript, Python, and Ruby Priority 3
 - 3.2.2.7.3.2 Achievements: Achievements will be granted upon reaching certain Skill Points or levels as defined from [3.2.2.7.1](#) to [3.2.2.7.3](#) Priority 3
 - 3.2.2.7.3.3 Statistics on Skill Points progress over time. Priority 3
- 3.2.2.7.4 Count of completed tasks
- 3.2.2.7.5 Count of completed projects
- 3.2.2.7.6 Task Estimation Efficiency: A measure of how close the user's estimates for time taken are to actual time spent to complete tasks.
- 3.2.2.7.7 User metrics will be used to calculate aggregate metrics for display in public areas of the website Priority 3
 - 3.2.2.7.7.1 Such metrics will include activity levels of different Skill Point categories, as per [Figure 3](#)

3.2.2.8 User Profile

A user's status will be publicly visible in a profile page, as per Figure 8. The page will include

- 3.2.2.8.1 A user overview displaying:
 - 3.2.2.8.1.1 Their avatar Priority 1
 - 3.2.2.8.1.2 Their Recent Level Priority 1
 - 3.2.2.8.1.3 Their Adventurer Level Priority 1
 - 3.2.2.8.1.4 Their title, selected from the complete list of their achievements Priority 3
- 3.2.2.8.2 A Friend's list, as detailed in 3.2.2.9.1 Priority 2
- 3.2.2.8.3 A list of recent achievements Priority 2
- 3.2.2.8.4 A list of the user's current Skill Point levels Priority 3
- 3.2.2.8.4 A user's progress towards future Adventurer Level, Recent Level, and skill point thresholds, as an experience bar, as per [Figure 8](#) Priority 3

Figure 8



Ack Myknee

C# Guru

38

Active Quest: [Write Documentation](#)

In Campaign: [That Python Thing](#)

Timeline

Priority

Projects

Groups


25:00

Social

Profile

Settings

Metrics



Ack Myknee

Title: C# Guru


Recent Level

38

Adventurer Level


690

Friends(20)




Shady Business Conductor

38



Captain Hero-guy Gem-Miner


35



Chris TheNin... Snake Charmer


37

Recent Achievements




Snake Charmer,

4



C# Guru,


20



Snake Handler


3

Skills




C#

20




Java

12




.net

9



Python

4



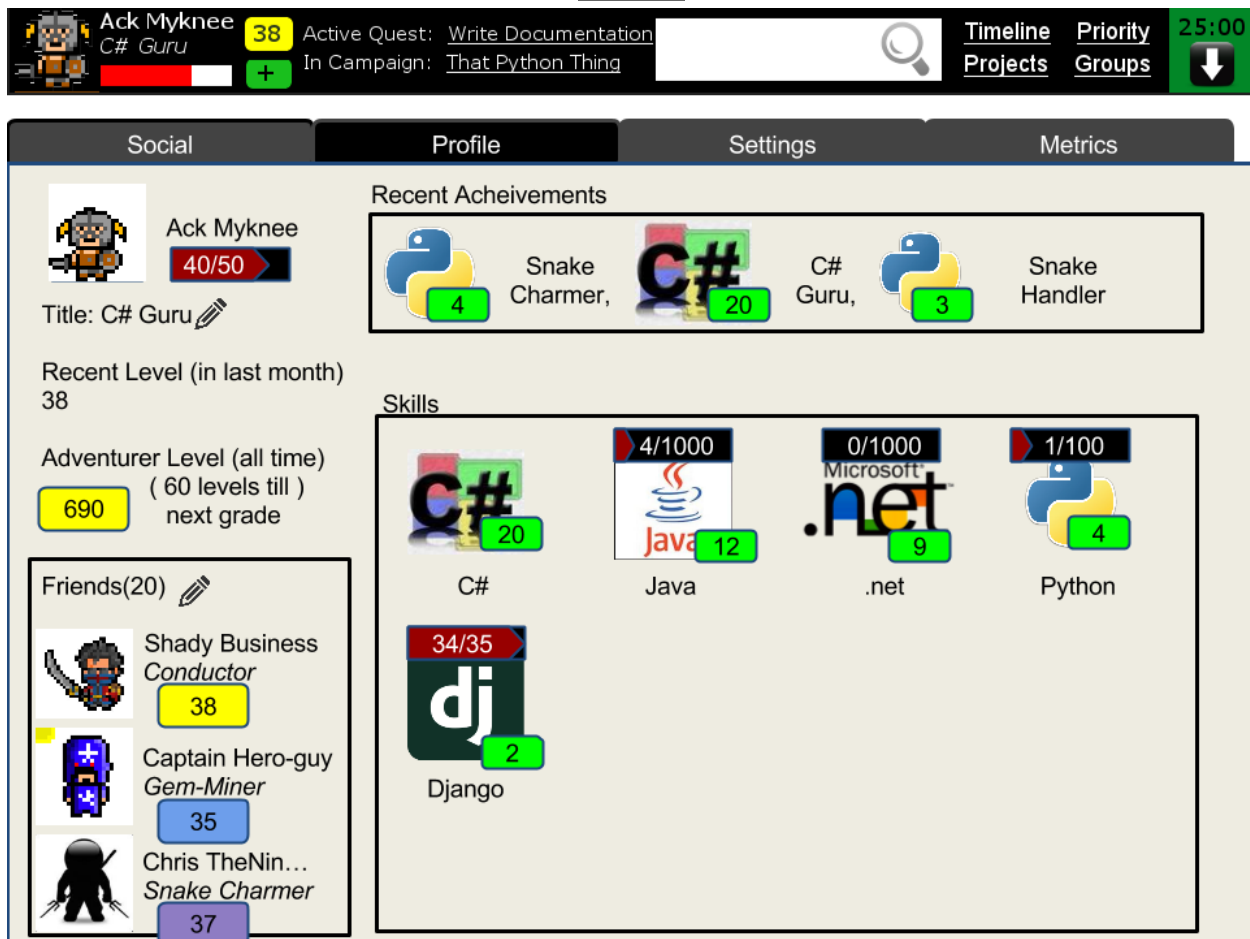
Django

2

As referenced under heading(s) [3.2.2.8](#), [4.2.1](#)

18

Figure 9



As referenced under heading(s) [3.2.2.8.4](#)

3.2.2.9 User Socialization

The user will be able to review an overview of social interaction, as per [Figure 10](#)

3.2.2.9.1 Friends List Priority 2

3.2.2.9.1.1 The user will be able to add/delete users from their friends list in their profile Priority 2

3.2.2.9.1.2 The user will be able to view an aggregation of the recent achievements of the friends on their friend's list Priority 3

3.2.2.9.1.3 The user will be able to view a 'leaderboard' of their most active friends, as ranked by their Recent Level, and then by all-time level Priority 3

3.2.2.9.2 User Groups, further detailed in 3.2.4 Priority 1


3.2.2.9.2.1 Users will be able to create groups to collaborate on group projects Priority 1

3.2.2.9.2.2 Users will be able to invite others to join their groups, and accept invitations to join groups created by other users, as per [Figure 11](#) Priority 1

3.2.2.9.2.3 Users will be able to leave groups in which they have membership, as per [Figure 11](#) Priority 1

3.2.2.9.2.4 Users will be able to view an overview of their group membership, and those groups that have activity Priority 2

Figure 10



Ack Myknee

C# Guru

38

Active Quest:

Write Documentation

In Campaign:

That Python Thing

Timeline

Priority

Projects

Groups

25:00

↓


Social

Profile

Settings


Metrics

Friend Leaderboard




Shady Business:

38



Chris TheNinja...:


37



Captain Hero-Guy:


35

Recent Achievements




Chris TheNinjaPirate:

Snake Charmer
(Level 4 Python)



Captain Hero-Guy:

Gem Miner
(Level 3 Ruby)



Shady Business:

Conductor
(Level 10 Ruby-on-Rails)

My Groups

Python Dev Team - 1 Project - Admin

Invitation: Java-champions [Join](#) [Abort](#)

Group Activity

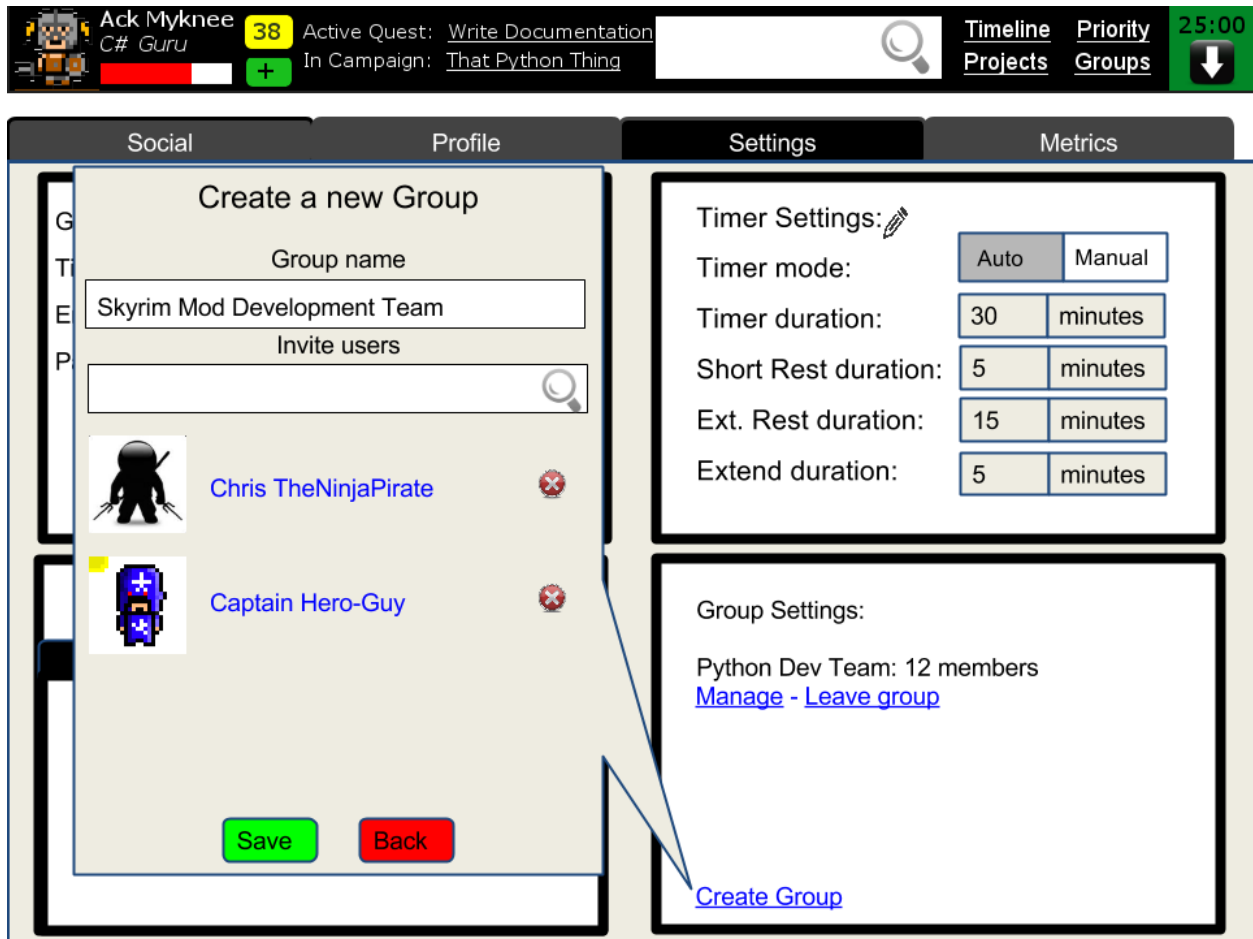
That Python Thing - Python Dev Team:

3 updates since last visit - 12 tasks completed this week

As referenced under heading(s) [3.2.2.9](#), [3.2.4.3.1](#)

20

Figure 11



As referenced under heading(s) [3.2.2.9.2.2](#), [3.2.2.9.2.3](#), [3.2.4.1.1](#)

3.2.2.10 User Security

3.2.2.10.1 QTD Member will be able to change the password of their respective user profile or group Priority 2

3.2.2.10.2 Password strength meter will show during QTD Member creation process, measured based on brute force and dictionary attack bit complexity. Priority 3

3.2.2.10.3 QTD Site Administrator will be able to generate a password reset to all QTD Members Priority 2

3.2.2.10.4 Passwords are known to no one except the user, Administrators will only be able to send reset request to users, cannot reveal passwords. Priority 1

3.2.2.10.5 Passwords input

3.2.2.10.5.1 Password will not display during user input or sent in plain text to user through any communication from QTD to user. Priority 2

3.2.2.10.5.2 Users will be locked out for 15 minutes if user had 5 fail login attempts. Priority 2

3.2.2.10.5.3 A user will be alerted of a failure with an alert, as well as how many attempts remaining until lockout. Priority 3

3.2.2.10.5.4 An email will be sent to user if they have more than 2 failed attempts, providing them choice of resetting of their password. Priority 2

3.2.2.10.6 User Session

An active user session is whenever a user logged into the system and it will stay active for 60 minutes and if user did not interact with QTD in any way it will be logged out. Priority 2

3.2.2.11 User Settings

In addition to previously mentioned modifications such as email and password changes, the user will be able to customize the followings, as per [Figure 2](#)

3.2.2.11.1 API and Secret keys This item is defined to offer open API, for future integration and authentication purposes. Priority 3

3.2.2.11.2 Time zone settings Priority 2

3.2.2.11.3 Timer settings, as detailed in [3.2.5](#) Priority 1

3.2.3 Project Management

Project Management entails the following features, as per [Figure 12](#)

3.2.3.1 Add Project

3.2.3.1.1 The user will be able to add project, as per [Figure 13](#). Each project will have the following attributes


3.2.3.1.1.1 **Project Name** Priority 1

3.2.3.1.1.2 **Project SCM** containing the URL for the online repository housing the project Priority 2

3.2.3.1.1.3 **Project Description** Priority 1

3.2.3.1.1.4 **Project Deadline** Priority 1

Figure 12



Ack Myknee

C# Guru

38

+

Active Quest:

Write Documentation

In Campaign:

That Python Thing

Timeline

Priority

25:00

Projects

Groups

↓

Campaigns

That Python Thing: Currently Active	<div>total remaining</div> <div>2 important</div> <div>4 pressing deadlines</div>
Service Integration w/ New Partners Last Activity: 2 Days ago	<div>total remaining</div> <div>2 important</div> <div>2 pressing deadlines</div>
Something Else: Last Activity: 2 weeks ago	<div>11 total remaining</div> <div>2 important</div> <div>1 pressing deadlines</div>
Misc ToDo's: Last Activity: Yesterday	<div>4 total remaining</div> <div>0 important</div> <div>0 pressing deadlines</div>

[Add Campaign](#)
[Delete Campaign\(s\)](#)
[Import/Export](#)
[Old Campaigns](#)

As referenced under heading(s) [3.2.3](#)

Figure 13

Campaigns

[That Python Thing:](#)
Currently Active

[Service Integration w/ New Par](#)
Last Activity: 2 Days ago

[Something Else:](#)
Last Activity: 2 weeks ago

[Misc ToDo's:](#)
Last Activity: Yesterday

Add Campaign

Name:

Description:

Deadline:

7 total remaining

4 pressing deadlines

2 pressing deadlines

1 pressing deadlines

0 pressing deadlines

As referenced under heading(s) [3.2.3.1.1](#), [4.2.3.1](#)

3.2.3.2 Modify Project

A User will be able to modify any of the attributes specified under heading [3.2.3.1](#), as per [Figure 14](#) Priority 1

Figure 14

The screenshot shows a user interface for managing campaigns. At the top, a header bar displays the user's profile (Ack Myknee, C# Guru, 38 points), active quest (Write Documentation), and in-campaign (That Python Thing). Navigation links for Timeline, Priority, Projects, and Groups are present, along with a 25:00 timer and a download icon. The main section is titled 'Campaigns' with a green checkmark and a red X icon. Below this, a list of campaigns is shown:

- [That Python Thing:](#) Currently Active
- [Service Integration w/ New Par...](#) Last Activity: 2 Days ago
- [Something Else:](#) Last Activity: 2 weeks ago
- [Misc ToDo's:](#) Last Activity: Yesterday

An 'Edit Campaign' modal is open for 'That Python Thing'. The modal contains the following fields:

- Name: That Python Thing
- Description: You know....when you want do do things...with the language named after a snake
- Deadline: (empty field)
- SCM URL: https://github.com/someuser/that-python-thing

At the bottom of the modal are 'Save' and 'Back' buttons. To the right of the campaign list, there are summary statistics for each campaign:

- That Python Thing:** 7 remaining, 2 important, 4 pressing deadlines
- Service Integration w/ New Par...** 2 pressing deadlines
- Something Else:** 1 pressing deadlines
- Misc ToDo's:** 0 pressing deadlines

As referenced under heading(s) [3.2.3.2](#), [3.2.3.3](#), [4.2.3.2](#), [4.2.3.3](#)

3.2.3.3 Remove Project

A QTD Member will be able to delete a project they own, as per [Figure 14](#) Priority 1

3.2.3.4 Import Project

A QTD Member will be able to import project(s) based on exported project file, as per [Figure 15](#). Priority 3

Figure 15

Campaigns

[That Python Thing:](#)
Currently Active

[Service Integration w/ New Partners:](#)
Last Activity: 2 Days ago

[Something Else:](#)
Last Activity: 2 weeks ago

[Misc ToDo's:](#)
Last Activity: Yesterday

Import/Export

Import from QTD Backup File

Export all to QTD Backup File

Select Projects to Export

- ☐ [That Python Thing](#)
- ☐ [Service Integration w/ New Partners](#)
- ☐ [Something Else](#)
- ☐ [Misc ToDo's:](#)

Save Back

As referenced under heading(s) [3.2.3.4](#), [3.2.3.5](#), [4.2.3.4](#), [4.2.3.5](#)

3.2.3.5 Export Project

A QTD Member will be able to export project(s), as per [Figure 16](#). The following formats are supported:

- 3.2.3.5.1 **JPG** A QTD Member will be able to export to jpeg image Priority 3
- 3.2.3.5.2 **PDF** A QTD Member will be able to export to PDF file Priority 3
- 3.2.3.5.3 **QTD** A QTD Member will be able to export to QTD format Priority 3

3.2.3.6 Task Management

Each project will have a number of tasks that are required for their completion, each task having the following features

3.2.3.6.1 Add Task

The user will be able to add Tasks as per [Figure 16](#). Tasks will have the following attributes:

3.2.3.6.1.1 Task Name Priority 1

3.2.3.6.1.2 Task Description Priority 1

3.2.3.6.1.3 Task Tags A user will be able to label the project with any number of tags Priority 1

3.2.3.6.1.3.1 Tags related to various programming skills will be available selections Priority 1

3.2.3.6.1.3.2 The user can add arbitrary number of user-defined tags Priority 2

3.2.3.6.1.4 Task Importance A user will be able to label a task as important. Priority 1

3.2.3.6.1.5 Task Deadline A user will be able to set a deadline for the task Priority 1

3.2.3.6.1.6 Task Dependents A user will be able to list any number of other tasks that cannot be completed without the completion of this task Priority 1

3.2.3.6.1.7 Task Prerequisites A user will be able to list any number of other tasks that must be completed prior to the completion of this task Priority 1

3.2.3.6.1.8 Task Estimation An estimation of the amount of time necessary to complete the task. Priority 2

3.2.3.6.1.9 Task Ownership A user who is assigned to complete the task Priority 2

3.2.3.6.2 Modify Task

A User will be able to modify any of the attributes specified under heading [3.2.3.6.1](#), as per [Figure 17](#) Priority 1

3.2.3.6.3 Delete Task

A QTD Member will be able to delete a project they own, as per [Figure 18](#) Priority 1


3.2.3.6.4 Task Overview

A task will maintain the following information for display

3.2.3.6.4.1 Task Timeline, will display all Time-blocks registered to the task, and their associated Records related to the task Priority 1

3.2.3.6.4.2 Task Time Investment, will show how much time was invested in the task as compared to its estimated time cost for completion Priority 2


Figure 16



Ack Myknee
 C# Guru


38

Active Quest: [Write Documentation](#)
 In Campaign: [That Python Thing](#)



[Timeline](#)
[Projects](#)

[Priority](#)
[Groups](#)

25:00


New Item!

+

Name

Owner

Description

Tags

Estimated Time

Deadline


Save

Back

Service integration with new partners

Deploy to production

directional data accuracy


Write mappings to translate between their data format and our own 

Owner: John | Points: 5 | Due: Tuesday

Api documentation available here: [www.partner.com/api](#)

Timeline:
John pushed a commit "category mappings complete" 70 minutes ago

calls



finished "Update and Distribute API documentation"(Steve) -10 hrs

My History

Most recent issue: API documentation missing legend -

Most recent note: Waiting on John to forward the mapping table

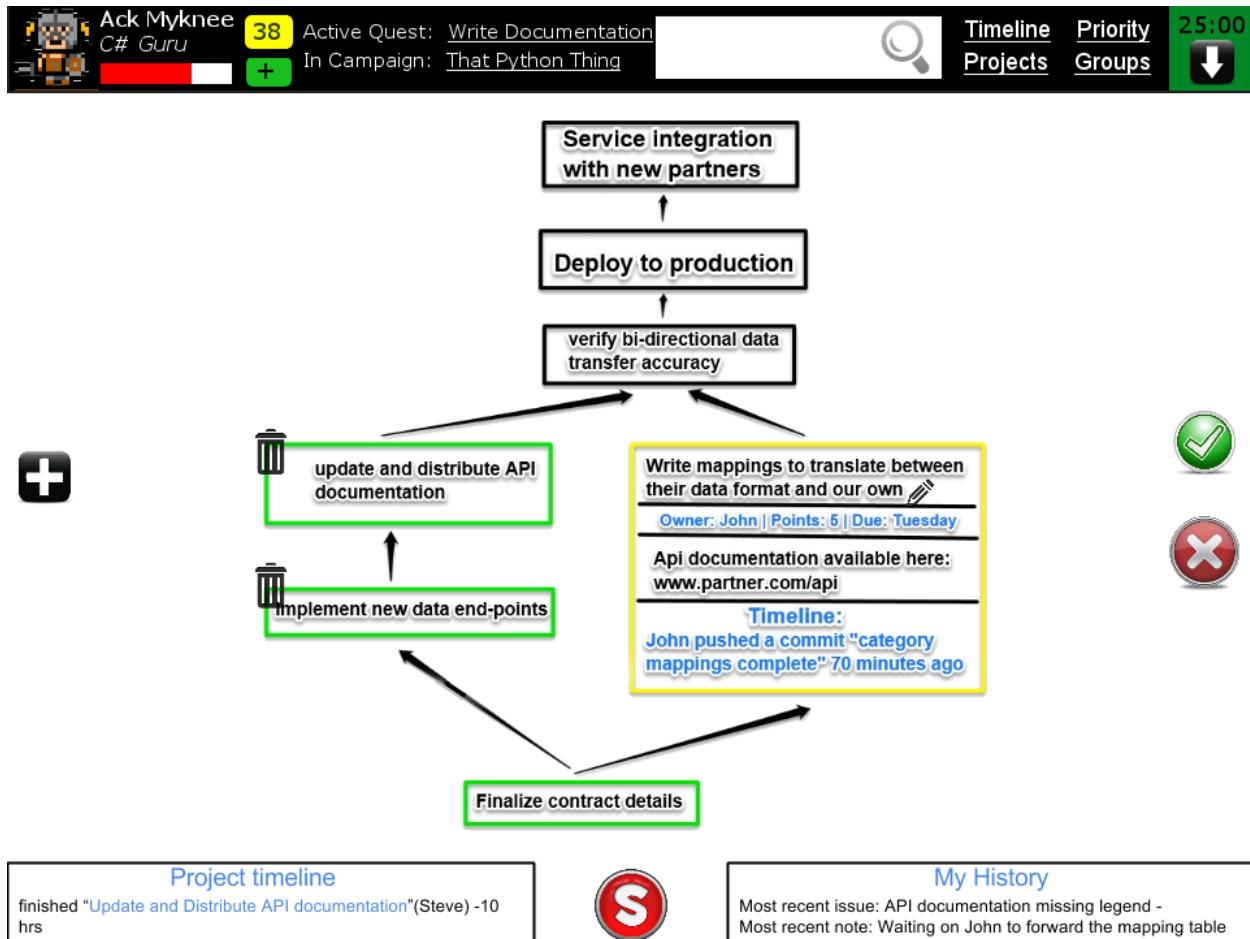
As referenced under heading(s) [3.2.3.6.1](#)

28

[illegible]

29

Figure 18



As referenced under heading(s) [3.2.3.6.3](#)

3.2.4 Group Mechanics

3.2.4.1 Group Creation and Deletion

- 3.2.4.1.1 A user will be able to create one or more groups, as per [Figure 11](#) Priority 1
- 3.2.4.1.2 The creator of the group will become QTD Group Administrator automatically Priority 1
- 3.2.4.1.3 A QTD Group could have more than one QTD Group Administrator Priority 3
- 3.2.4.1.4 A QTD Group Administrator could delete their group Priority 2
- 3.2.4.1.5 A QTD Group is identified by group name.

3.2.4.2 Group Information

Group members will have access to the following information, as per [Figure 19](#)

- 3.2.4.2.1 Users in the current group Priority 2
 - 3.2.4.2.1.1 For each user, the time since they last submitted activity Priority 3
 - 3.2.4.2.1.1.1 If currently working on group tasks, the current active task of the user Priority 3
 - 3.2.4.2.1.1.2 For each user, their role in the group Priority 3
 - 3.2.4.2.1.1.3 Metrics showing how much the user contributed towards group task completion over

the last 30 days Priority 3

3.2.4.2.2 A list of all projects for the current group Priority 1

3.2.4.2.2.1 For each project, the current count of tasks remaining unfinished Priority 3

3.2.4.2.2.2 For each project, the current count of tasks with deadlines in the next week Priority 3

3.2.4.2.2.3 For each Project, the current count of tasks tagged as important Priority 3

3.2.4.2.3 Group activity stream Priority 3

A recent activity stream will be available to group members detailing the following events:

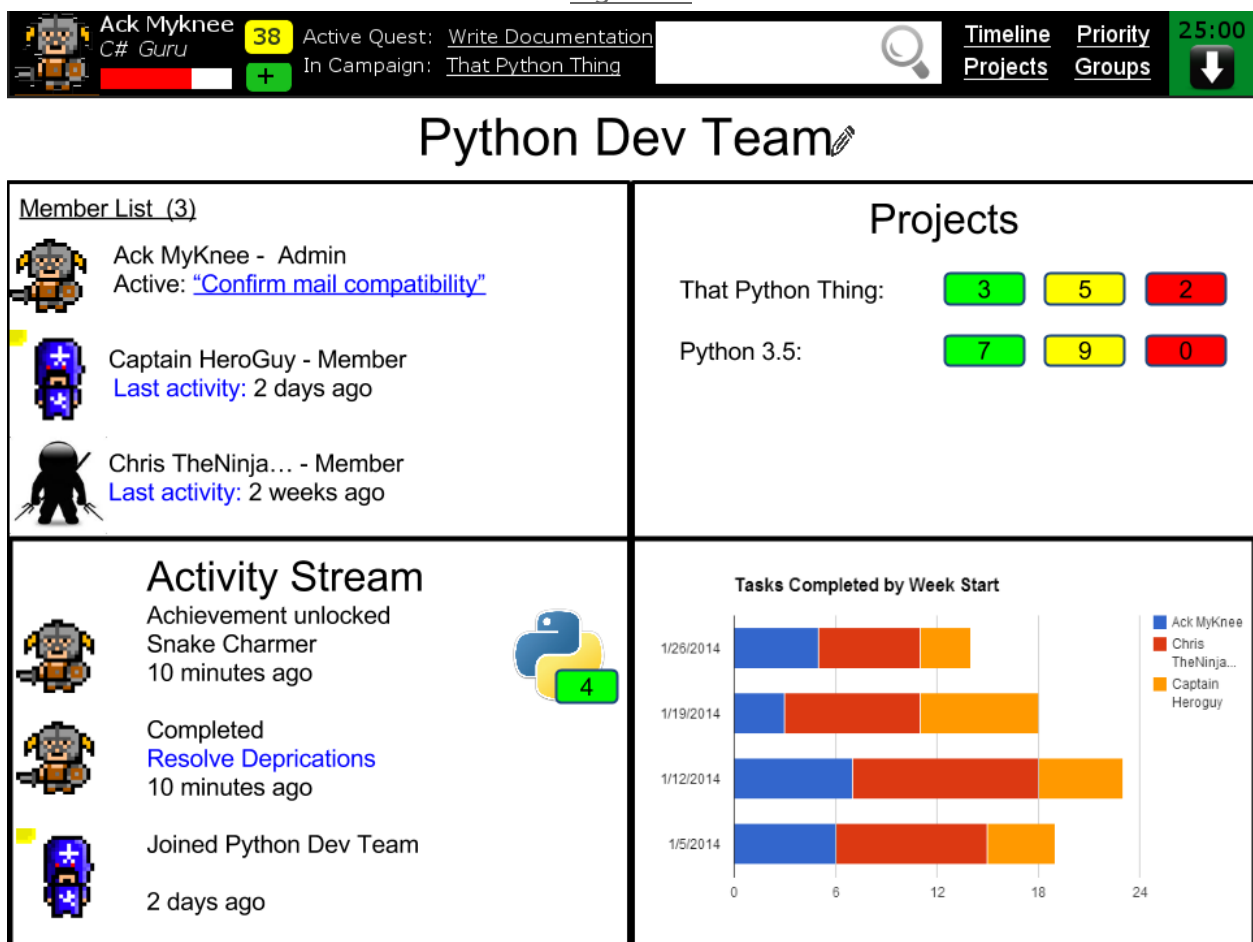
3.2.4.2.3.1 Newly created, closed, completed, overdue tasks

3.2.4.2.3.2 Newly created, closed, completed, overdue projects

3.2.4.2.3.3 Member achievements

3.2.4.2.3.4 Members joining or leaving the group.

Figure 19



As referenced under heading(s) [3.2.4.2](#)

3.2.4.3 QTD Group Member

3.2.4.3.1 A QTD Member could accept invite from QTD Group Administrator to become a QTD Group Member as per [Figure 10](#) Priority 1

3.2.4.3.2 A QTD member will be able to leave the group Priority 1

3.2.4.3.3 A QTD member will be able to modify or view a project associated with the group based on permission Priority 2

3.2.4.3.4 A QTD member will be able to assign tasks to themselves or another QTD Group Member Priority 1

3.2.4.4 QTD Group Administrator

QTD Group administrator will be able to manage their group from the Group page, as per [Figure 20](#)

3.2.4.4.1 A QTD Group Administrator could invite other QTD Member to the group to become one of their QTD Group Members Priority 1

3.2.4.4.2 A QTD Group Administrator could remove any QTD Member from the group Priority 1

3.2.4.4.3 A QTD Group Administrator could assign any QTD Member as Group Administrator Priority 2

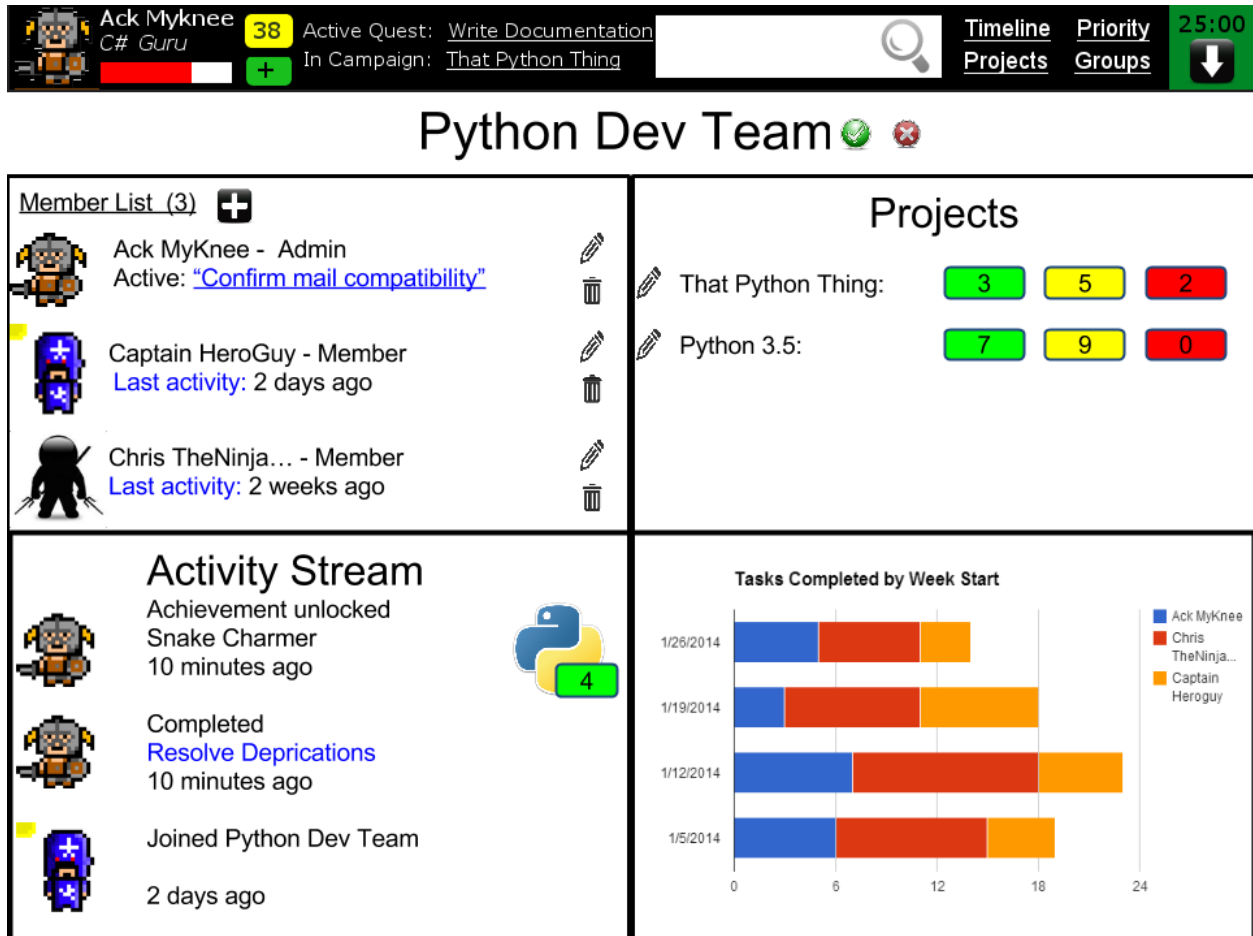
3.2.4.4.4 A QTD Group Administrator could remove administrator status of other QTD Member Priority 2

3.2.4.4.5 For a group without an active QTD Group Administrator, if there are other users in the group, the oldest QTD Group Member will be promoted to QTD Group Administrator Priority 3

3.2.4.4.6 If the QTD Group Administrator leave the group the QTD Group and there is no more QTD Member or QTD Group Administrator in the group, the group will be deleted Priority 2

3.2.4.4.7 A QTD Group Administrator could create group-based project Priority1

Figure 20

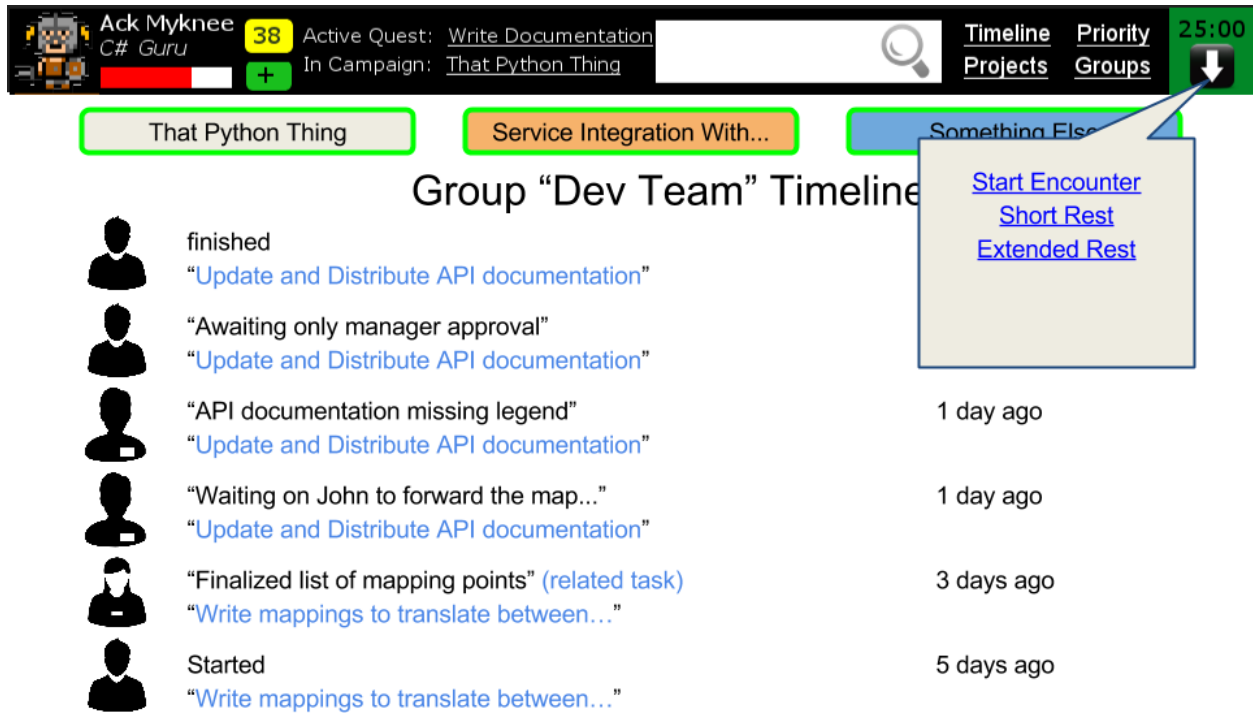


As referenced under heading(s) [3.2.4.4](#)

3.2.5 Timer Functionality

In order to chunk the work-day into discrete chunks a user can choose to use the timer, as per [Figure 21](#) and [Figure 22](#). Recorded time-block data is used in the generation of timelines as specified in [3.2.8](#). The timer entails the following functionalities:

Figure 21



As referenced under heading(s) [3.2.5](#), [4.2.4](#)

Figure 22

Ack Myknee
C# Guru 38
Active Quest: [Write Documentation](#)
In Campaign: [That Python Thing](#)

Timeline Projects Priority Groups -1:32

That Python Thing Service Integration With... Something Else

Group "Dev Team" Timeline

	finished	" Update and Distribute API documentation "	
	"Awaiting only manager approval"	" Update and Distribute API documentation "	
	"API documentation missing legend"	" Update and Distribute API documentation "	1 day ago
	"Waiting on John to forward the map..."	" Update and Distribute API documentation "	1 day ago
	"Finalized list of mapping points" (related task)	" Write mappings to translate between... "	3 days ago
	Started	" Write mappings to translate between... "	5 days ago

[Reset Encounter Time](#)
[Pause Encounter](#)
[Extend Encounter](#)
[Short Rest](#)
[Extended Rest](#)

As referenced under heading(s) [3.2.5](#), [4.2.4](#)

3.2.5.1 Start Timer

The user can start the timer and after user start it, it will count down from set time period to zero. Once it reaches zero, system will alert user such that they may take a break, record their progress, or switch tasks as necessary Priority 1

3.2.5.2 Pause Timer

The user can choose to pause the timer due to extended leave or any other reason he/she wants to stop working on projects Priority 1

3.2.5.3 Stop Timer

The user can choose not use the timer function by stopping the timer and the timer will simply stay at set time period Priority 1

3.2.5.4 Configure Timer

The user will be able to configure the timer to match the needs of their personal workflow, as per [Figure 2](#)

3.2.5.4.1 Manual or automatic, automatic by default Priority 1

3.2.5.4.2 Set up a time block, 30 minutes by default Priority 2

3.2.5.4.2.1 Range from 10 minutes to 120 minutes Priority 2

3.2.5.4.3 Break system. In manual mode, once a user's time block runs up, the user will be able to set a default break period Priority 2

3.2.5.4.3.1 A 'Short break' duration can be set

3.2.5.4.3.2 A 'Extended break' duration can be set

3.2.5.4.4 Extend duration, a duration in minutes to add to the timer before the user will be alerted again when in manual mode

3.2.5.5 Extend Encounter

A button for manual mode that allows you to extend the countdown by the current 'extend duration' setting, and then again alert the user of the timer expiration Priority 2

3.2.5.6 Short Rest

A button for manual mode to give the user a predefined break period Priority 2

3.2.5.7 Extended Rest

A button for manual mode to give the user a predefined, longer break period Priority 2

3.2.6 Record Management

3.2.6.1 Add Record(s)

A QTD Member will be able to add following types of records to a given Task, as per [Figure 23](#).

Figure 23

Ack Myknee
38 Active Quest: [Write Documentation](#)
Timeline [Projects](#) [Priority Groups](#) 25:00

- Service integration with new partner
v Deploy to Production

Deploy to Production

Record Data

☐ Note
 ☒ Link
 ☐ Image
 ☐ File

Name

URL

Description

Quote

Save
Back

acy
partner

se and website to support new integration

mentation

en...

r data and theirs
racy

se and website to support new integration

mappings complete" - 70 minutes ago

[Project timeline](#)

finished "Update and Distribute API documentation"(Steve) -10 hrs

[My History](#)

Most recent issue: API documentation missing legend -
Most recent note: Waiting on John to forward the mapping table

As referenced under heading(s) [3.2.6.1](#), [4.2.5](#)

3.2.6.1.1 Add Note(s)

A QTD Member will be able to add a note with a character limit of 3000 characters for each entry Priority 1

3.2.6.1.2 Add Link(s)

A QTD Member will be able to add link(s) with a short description of 300 characters or less Priority 1

3.2.6.1.3 Add Image(s)

A QTD Member will be able to add image smaller than 1MB with short description of 300 characters or less, the image's format will be either in jpeg, bitmap or png format Priority 1

3.2.6.2 Modify Record

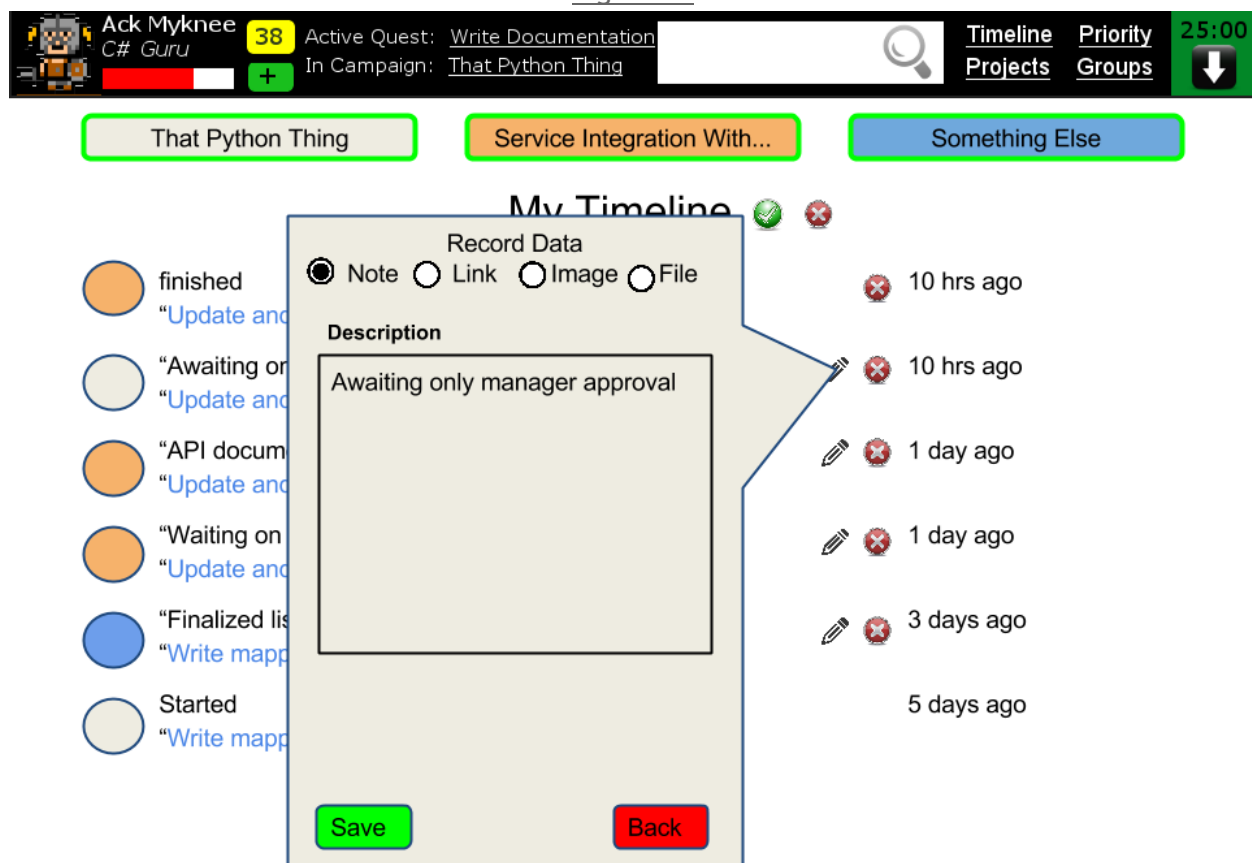
A QTD Member will be able to modify contents of the following types of item defined in 3.2.5.1, as per [Figure 24](#)

3.2.6.2.1 Modify Note(s) Priority 1

3.2.6.2.2 Modify Link(s) Priority 1

3.2.6.2.3 Modify Image(s) Priority 1

Figure 24



As referenced under heading(s) [3.2.6.2](#), [3.2.6.3](#), [4.2.5](#)

3.2.6.3 Delete Record

A QTD Member will be able to delete following types of item defined in [3.2.5.1](#), as per [Figure 24](#)

3.2.6.3.1 **Delete Note(s)** Priority 1

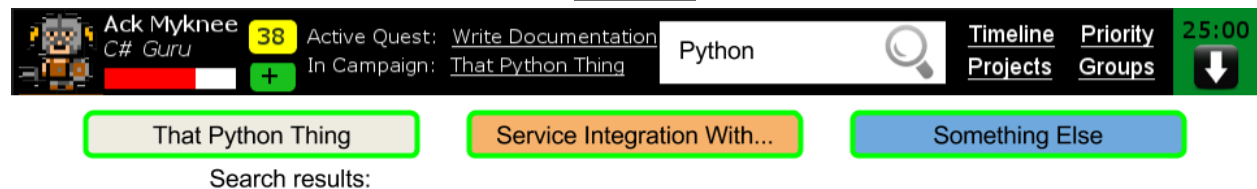
3.2.6.3.2 **Delete Link(s)** Priority 1

3.2.6.3.3 **Delete Image(s)** Priority 1

3.2.7 Tagging, Filtration and Searching

Searching is primary based on the tagging system, as per [Figure 25](#). Tagging system provides the following features:

Figure 25



As referenced under heading(s) [3.2.7](#)

3.2.7.1 Search

A QTD Member will be able to search the following types of information,

3.2.7.1.1 Project name and description Priority 1

3.2.7.1.2 Tasks content and record content Priority 1

3.2.7.1.3 Records Priority 1

3.2.7.1.4 Search scope

A user can search fields described from [3.2.7.1.1](#) to [3.2.7.1.3](#)

3.2.7.1.4.1 The user will be able to search across their entire QTD account Priority 1

3.2.7.1.4.2 The user will be able to do a search within an individual project Priority 1

3.2.7.1.4.3 The user will be able to do a search within arbitrary subsets of their projects
Priority 2

3.2.7.1.4.4 The user will be able to search within their group project if they have active
membership within a QTD Group. Priority 2

3.2.7.2 Tag Types

3.2.7.2.1 **System Wide Tagging** Priority 1

The following are built-in types of tags provided by QTD:

3.2.7.2.1.1 Programming Languages tags Priority 1

3.2.7.2.1.2 Project/Record type tags Priority 1

3.2.7.2.1.3 Skill tagging related to system recognized skills Priority 1

3.2.7.2.2 **User-defined Tagging** This will be any tags user would like to specify Priority 2

3.2.7.3 Filtration

3.2.7.3.1 A user will be able to filter results by specific projects Priority 1

3.2.7.3.1.1 A user will be able to filter results by an arbitrary subset of projects Priority 2

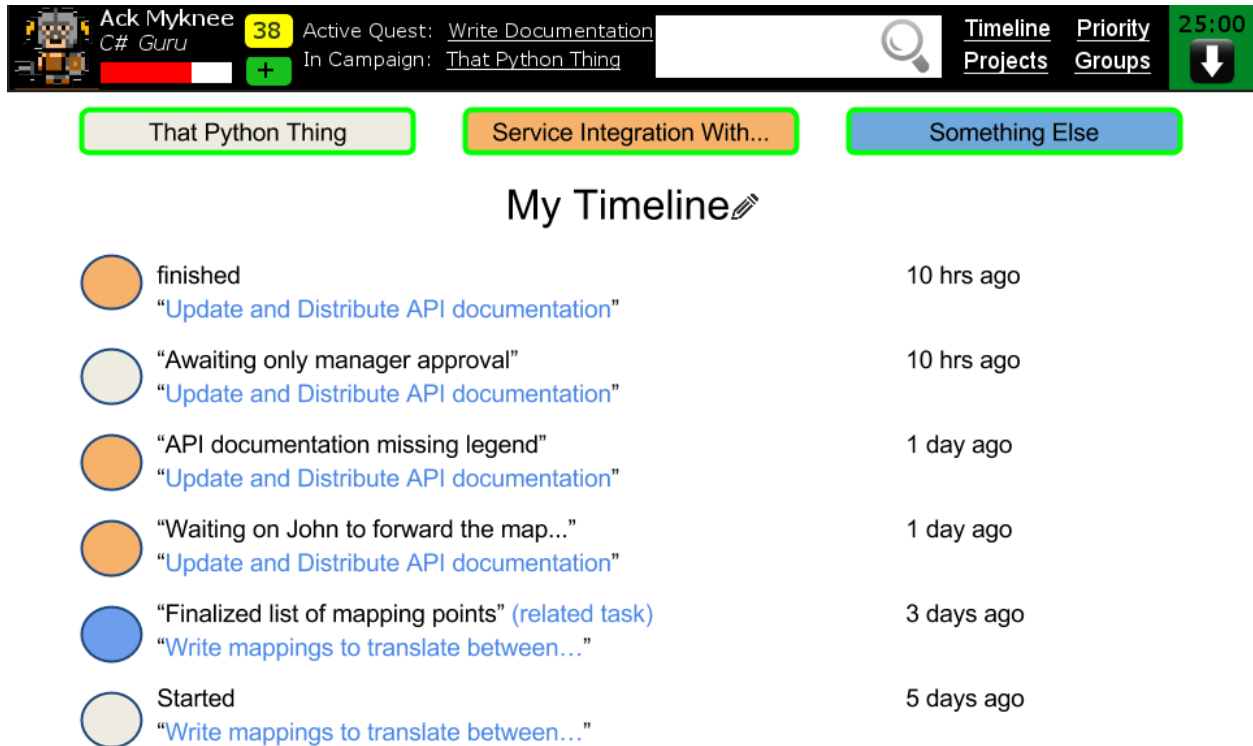
3.2.8 Timeline Review

A timeline of all user activity will be aggregated into various feeds, for purposes of reviewing
development process, group management, and social interaction. Priority 1

3.2.8.1 User Timeline Priority 1

This will list project management, record management, achievements and level-ups in a general
feed to give users a way of recreating their work-day, as per [Figure 26](#) Priority 1

Figure 26



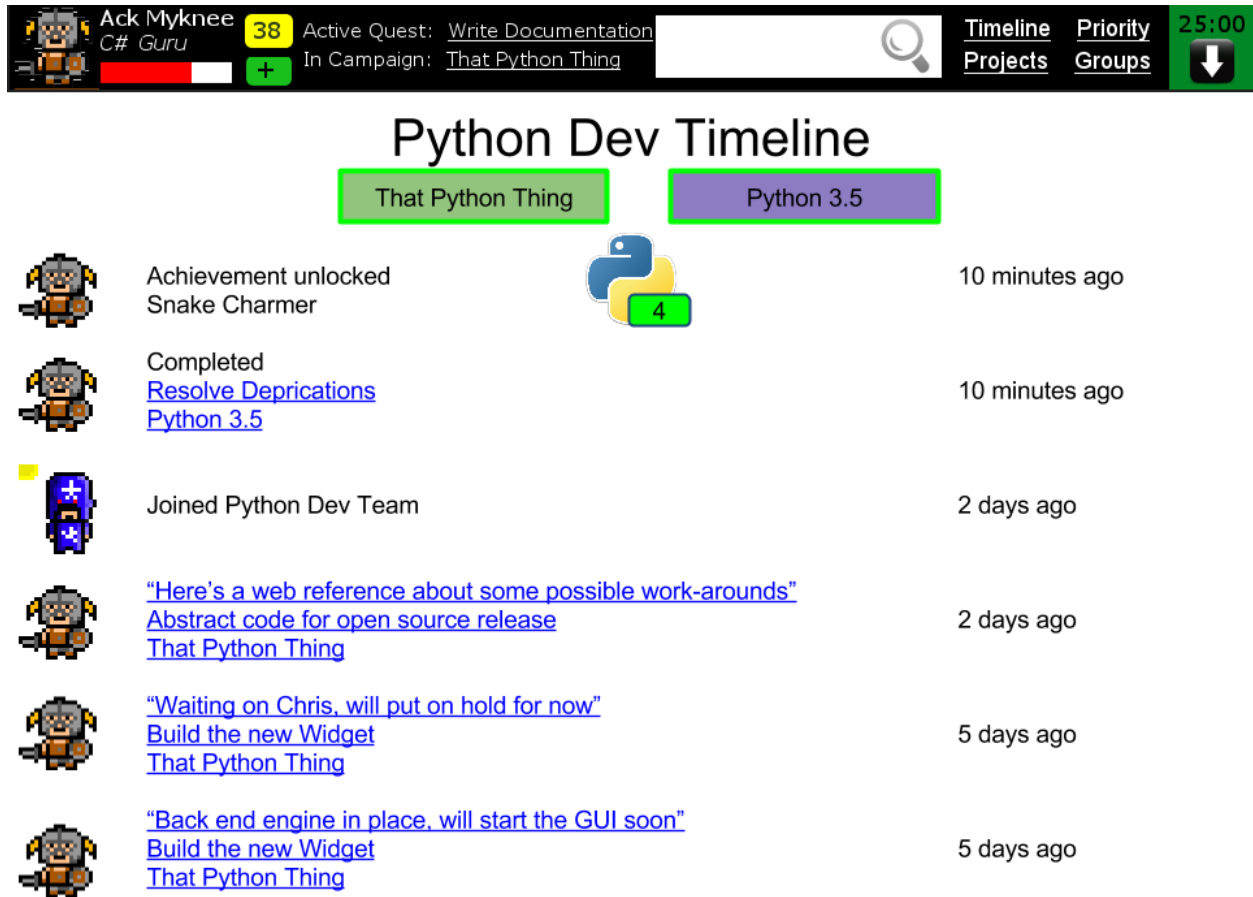
As referenced under heading(s) [3.2.8.1](#)

3.2.8.2 Project Timeline

A timeline of any given project, showing only project and record management for the given project, as per [Figure 27](#) Priority 1

3.2.8.2.1 If applicable each activity will be related to the appropriate user by their avatar, as in any group project *Priority 2*

Figure 27



As referenced under heading(s) [3.2.8.2](#)

3.2.8.3 Group Timeline

Contains all the content of a User timeline, only for all members of the group. In addition, it will display group management (members leaving or joining), as per [Figure 28](#) Priority 1

Figure 28



Ack Myknee

C# Guru

38

Active Quest: [Write Documentation](#)

In Campaign: [That Python Thing](#)



[Timeline](#)

[Projects](#)







[Priority](#)

[Groups](#)

25:00



Service Integration With New Partners Timeline

	finished "Update and Distribute API documentation"	10 hrs ago
	"Awaiting only manager approval" "Update and Distribute API documentation"	10 hrs ago
	"API documentation missing legend" "Update and Distribute API documentation"	1 day ago
	"Waiting on John to forward the map..." "Update and Distribute API documentation"	1 day ago
	"Finalized list of mapping points" (related task) "Write mappings to translate between..."	3 days ago
	Started "Write mappings to translate between..."	5 days ago

As referenced under heading(s) [3.2.8.3](#)

3.2.9 Global Notifications

QTD Site Administrator will be able to direct global notifications to all users. Such notifications will appear as simple pop-up alerts requesting user confirmation *Priority 3*

3.2.10 Prioritization Functionality

Users will be able to quickly select the most pressing tasks with an overview designed to highlight pressing and important issues, as per [Figure 29](#)

3.2.10.1 Factors to filter by include:

3.2.10.1.1 Due Date *Priority 1*

3.2.10.1.2 Ownership *Priority 2*

3.2.10.1.2.1 Owned by User

3.2.10.1.2.2 Unassigned, from group projects

3.2.10.1.2.3 Owned by others, from group projects

3.2.10.1.3 Importance *Priority 1*

3.2.10.1.4 Number of unfulfilled prerequisites *Priority 3*

3.2.10.2 Priority View *Priority 1*

3.2.10.2.1 Shows prioritized items as scrolling list

3.2.10.2.2 For each item, displays additional information

3.2.10.2.2.1 Items which list this as a prerequisite *Priority 2*

3.2.10.2.2.2 Unfulfilled prerequisites of this item *Priority 2*

3.2.10.3 Map Overview, as per [Figure 30](#) *Priority 2*

3.2.10.3.1 Shows all items flagged by the priority system on a single screen in a generalized overview *Priority 3*

Figure 29

The interface displays a user profile for 'Ack Myknee' (C# Guru) with 38 points. The active quest is 'Write Documentation' and the campaign is 'That Python Thing'. Below the profile are three filter buttons: 'That Python Thing' (highlighted), 'Service Integration With...', and 'Something Else'. A checkbox labeled 'Show my assignments only' is present. The main area shows a list of tasks:

Task	Status	Assignee	Urgency
Write Documentation	Assigned to: you	1 day left	2
Update use case diagram	Unassigned	2 days left	2
Don't forget the milk	Unassigned	2 days left	0
Finish test framework	Assigned to: John	6 days left	0
Update use case diagram	Assigned to: Steve	7 days left	0
Make the cheese platter	Assigned to: you	Important	1
Fix email bug #4	Unassigned	Important	4
Address security exploit in...	Assigned to: Steve	Important	2
Call the caterer	Assigned to: Frankie	Important	1

As referenced under heading(s) [3.2.10](#)

Figure 30



As referenced under heading(s) [3.2.10.3](#)

3.3 Non-functional Requirements

3.3.1 Server side

3.3.1.1 Response time

The server will fetch the response to a client's request in maximum 2 seconds under the maximum number of concurrent connections (100 users by default).

3.3.1.2 Availability

QTD system will be expected to be up and running as long as hardware and network condition remain functional, with availability of 99.9% throughout a year which is roughly 9 hours downtime per year.

3.3.1.3 Capacity

The server will have at least 500GB of physical storage capacity, with at least of 6 gigabytes of RAM. The server will at least support 100 concurrent users at the same time. The storage capacity as well as the concurrent users supported will be expandable according to the affordable hardware and software on the server side.

3.3.1.4 Reliability

The server should be fault tolerant and the rate of failures of the server should be less than once a week. A critical failure is defined such that the server has encountered a critical error and is no longer accessible or experiencing data corruption.

3.3.1.5 Scalability

The storage of the database will be scalable by utilizing the major Database Management System, SQLite or PostgreSQL to minimize performance impact when the database's size grows.

3.3.1.6 Maintainability

QTD will be modularized so that it is extensible to allow continuous improvement and implementation of new requirements without impacting the existing system's integrity.

3.3.1.7 Testability

QTD will be broken down into individual modules (Client and Server side) so that each piece could be unit tested separately. The system could also provide the necessary means for integration test between modules.

3.3.1.7 Documentation

Any code that is not auto-generated by scaffolding or similar techniques by using the intended framework (Ruby on Rails in this case) will be documented to specify the purpose of the code block, and the expected result.

3.3.2 User side

3.3.2.1 Supported Platforms

The client software will be run in a browser, which is in compliance with current HTTP protocol and support JavaScript. The major supported browsers are Google Chrome version 30.x and above, Microsoft Internet Explorer 9.x and above, and Mozilla Firefox version 20.x and above. Any other major browsers are not officially supported but are expected to be able to run the system with the adequate requirement as specified previously.

3.3.2.2. Security/privacy

The user account's password will be encrypted for storage in the database. Sensitive user's information such as personal information, contact data and email address will be hidden and inaccessible to the other users to maintain privacy.

3.3.2.3 Internet Connectivity

The client is required to have an internet connection to be able to use the application, with at least 1 Mbps connection speed to ensure smooth usage of the system.

3.4 Design Constraints

3.4.1 Software Constraints

3.4.1.1 The communication between the server and client will mostly be dealt in text format such as JSON or XML to minimize bandwidth usage. Any response of client's request will not be more than 10MB except downloading file from the server.

3.4.2 Hardware Constraints

3.4.2.1 The application's binaries must not take more than 100MB of space on the server. The server must use less than 6GB of memory on peak concurrent usage due to hardware limitation to ensure the data is served to the current users fluently.

4 Use Cases

4.1 Authenticate

4.1.1 Logging in

The user accesses the correct website and is logged in by providing following correct information

*A field to enter username, labeled Username

*A field to enter password, labeled Password

After the user enters he/she username and password and hit the Submit option. If the username and password is valid, the user will be logged in and the system will proceed to the welcome Homepage. [Figure 3](#) shows the login page.

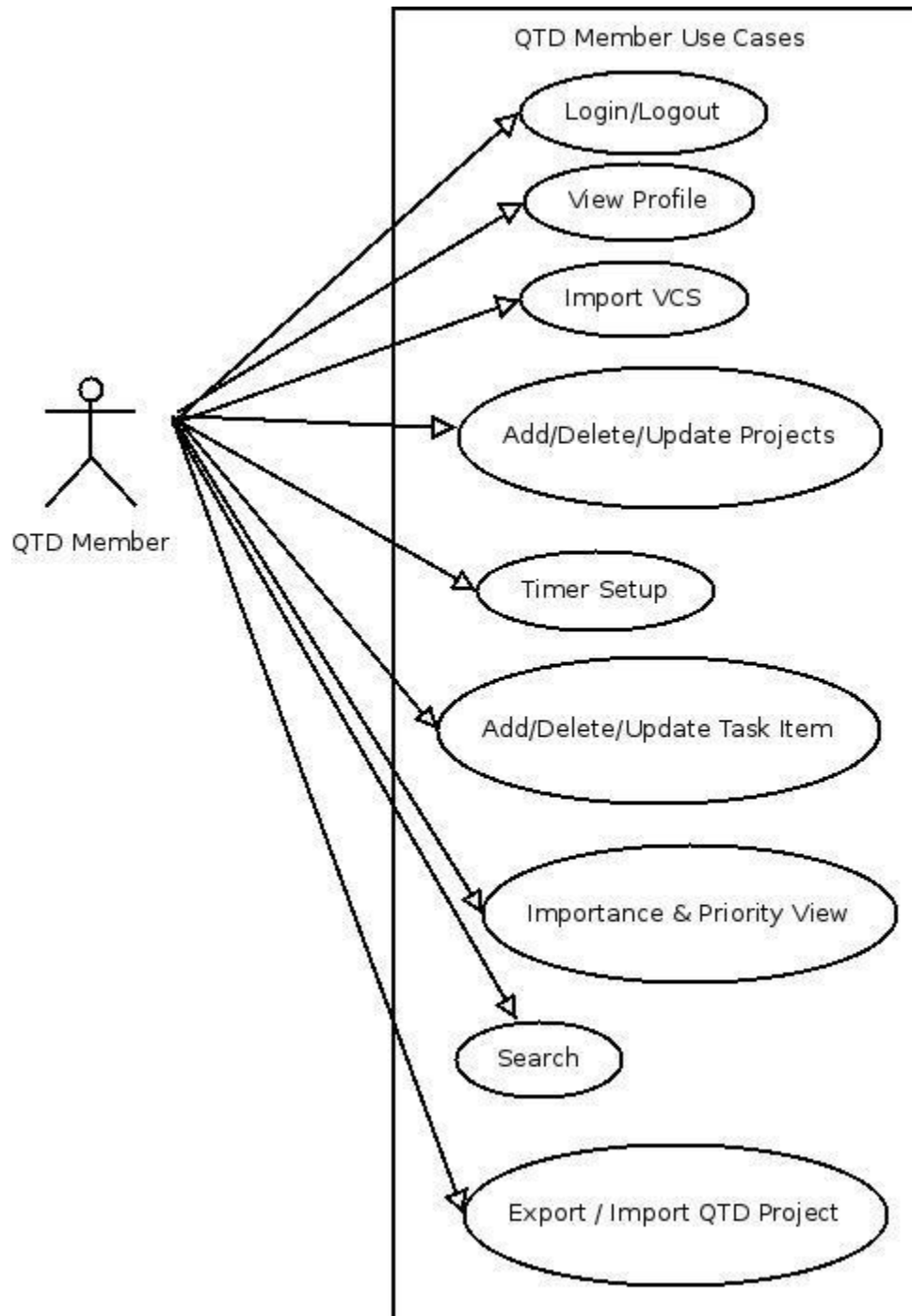
4.1.2 Logging Out

The user may log out if they are logged in. After they logged out, they will be redirect to home page. All their logged in information will not be available for the logged out user.

4.2 Use cases for all logged in users

[Figure 31](#) shows all user use cases.

Figure 31



As referenced under heading(s) [4.2](#)

4.2.1 User Profile

A user will be able to view his/her profile for performance metrics defined in 3.2.2.8. [Figure 8](#) shows the user profile page.

4.2.2 VCS import

Upon first time login, or in the user profile page, a user will be able to set a VCS credentials. Details defined in 3.2.1 and [Figure 1](#) shows the import interface.

4.2.3 Manage Projects

A user will be able to add, delete, update, import and export projects they own

4.2.3.1 Add Project(s)

A user will be able to add project to existing project lists. [Figure 13](#) shows adding project process.

4.2.3.2 Delete Project(s)

A user will be able to delete project from existing project lists. [Figure 14](#) shows the delete project interface.

4.2.3.3 Update Project(s)

A user will be able to update project with following properties. [Figure 14](#) shows the update project interface.

4.2.3.4 Import Project(s)

A user will be able to import QTD file and get exported projects to display on current system. [Figure 15](#) shows the import project interface.

4.2.3.5 Export Project(s)

A user will be able to export QTD workflow data into JPG, PDF, QTD file format for sharing and backup purpose. [Figure 15](#) shows the export project interface.

4.2.4 Timer Setup

A user will be able to setup a timer to alert for switching tasks or take a break. [Figure 21](#) and [Figure 22](#) show this interface. [Figure 2](#) shows the timer configuration interface.

4.2.5 Manage Record

Users will be able to add/delete/update records described in 3.2.6. [Figure 23](#) and [Figure 24](#) show the record management interface.

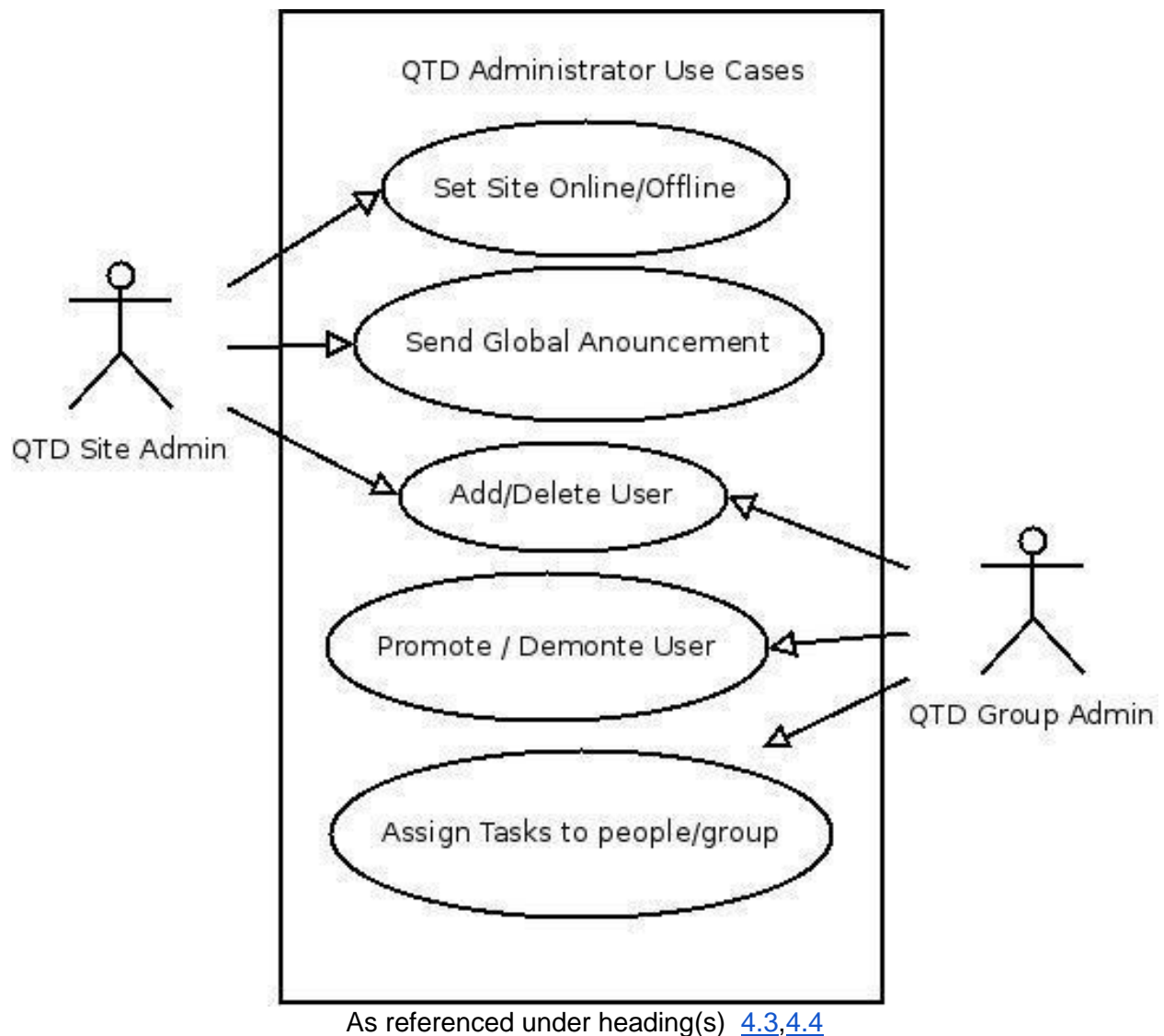
4.2.6 Search

A user will be able to fully search on created projects and tasks. The search interface is located in most user interfaces. Such as [Figure 1](#) and [Figure 2](#)

4.3 Use case for administrators

[Figure 32](#) is the use case diagram for administrator roles. Administrators consists the QTD site administrator and QTD Group Administrator. All administrators have the following use cases:

Figure 32



4.3.1 Add/delete user in his/her group

QTD Group Administrators will be able to add and delete group members for their group through the user profile page.

4.3.2 Administrators can assign tasks to groups and users

4.3.3 Administrators can promote and demote users/administrators as administrators and QTD Group Member.

4.4 Use case for QTD Site Administrator

[Figure 32](#) shows the use cases for QTD Site Administrator

4.4.1 Site Online/Offline

QTD Site Administrator will be able to set the online/offline for the purpose of upgrading system or enable/disable service to customers.

4.4.2 Send global announcements

QTD Site Administrator will be able to send global notifications to alert users for downtime, service upgrades, EULA changes. Defined in 3.2.9.