Software Requirements Specification

for

OnTraq

Version 1.0 approved

Prepared by Valentin Bogachev

Basement Brogrammers

10/22/2018

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 1

1.5 References 1

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 2

2.6 User Documentation 2

2.7 Assumptions and Dependencies 3

3. External Interface Requirements 3

3.1 User Interfaces 3

3.2 Hardware Interfaces 3

3.3 Software Interfaces 3

3.4 Communications Interfaces 3

4. System Features 4

4.1 System Feature 1 4

4.2 System Feature 2 (and so on) 4

5. Other Nonfunctional Requirements 4

5.1 Performance Requirements 4

5.2 Safety Requirements 5

5.3 Security Requirements 5

5.4 Software Quality Attributes 5

5.5 Business Rules 5

6. Other Requirements 5

Appendix A: Glossary 5

Appendix B: Analysis Models 5

Appendix C: To Be Determined List 6

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

The purpose of this document is to detail the purpose and features of the OnTraq project management tool. It will present the reader with the operating procedures and user interactions pertinent to the application’s system. This document is intended for stakeholders, users, and the other developers.

## Document Conventions

There are no specific typographical conventions that are necessary to fully comprehend this document.

## Intended Audience and Reading Suggestions

This document is intended for any person, user or developer, seeking to gain a comprehensive understanding of the OnTraq system. It is recommended that the reader read this document in the order it is presented.

## Product Scope

Simply, as stated above, OnTraq is intended to be a lightweight and accessible project management tool. It will provide tools to help organize and manage development teams and the projects and individual tasks they will be working on.

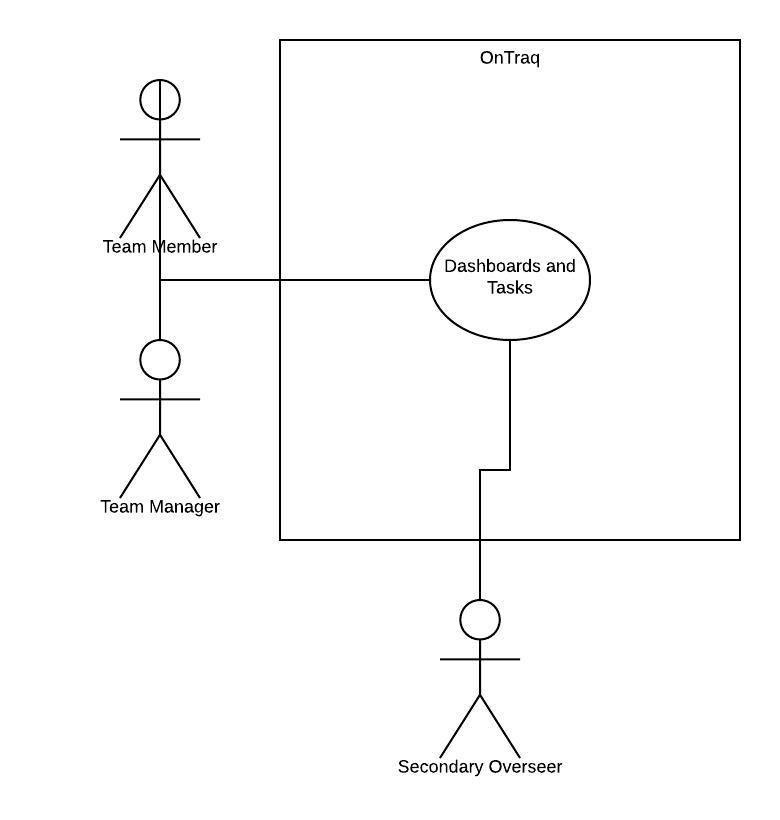
OnTraq allows team members to view the tasks assigned to them in order of priority while also giving managers and overseers the ability to change the structure of the teams and project while also being able to quickly assess the state of the project.

## References

# Overall Description

## Product Perspective

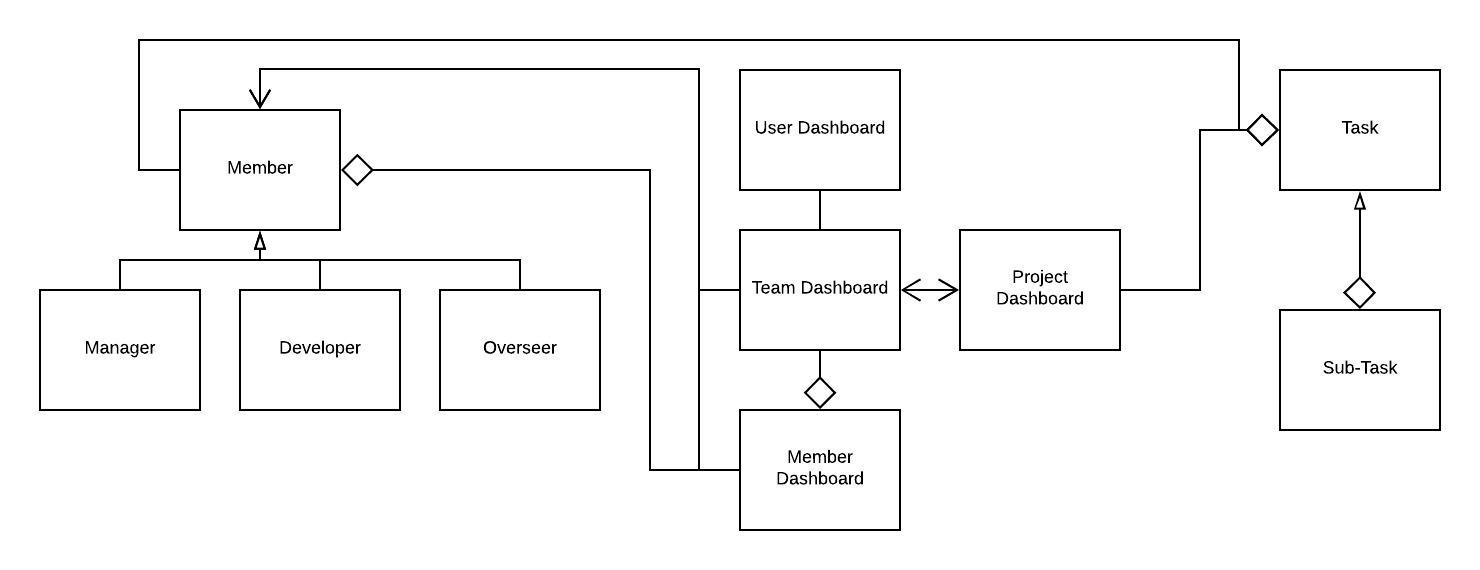
OnTraq is a new, self-contained, product that does not require any other systems or software to effectively function.



## Product Functions

The primary functions of the system include:

* Task assignment
* Task subdivision
* Task prioritization
* Team/group assignment
* Task/project progress update



## User Classes and Characteristics

1. Developer
   1. Most common user, and close second in importance
   2. Can edit task status (Completed, Started, Not Started)
   3. Can upload files of completed tasks
2. Manager
   1. Most important
   2. Can assign members to teams
   3. Can assign teams to projects
   4. Can divide projects into tasks and assign team members to them
   5. Can divide tasks into sub-tasks and assign team members to them
   6. Can prioritize tasks
3. Overseer
   1. Meant for other team members that only need to view the status of the project
   2. Can view the different dashboards and team members, but not edit them.

## Operating Environment

The environment is rather basic. The application will likely only be able to run on windows. OnTraq will not need any specific software to run.

## Design and Implementation Constraints (Optional)

The only constraint would be internet access. Most, if not all, the information will be stored on a server.

## User Documentation

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>

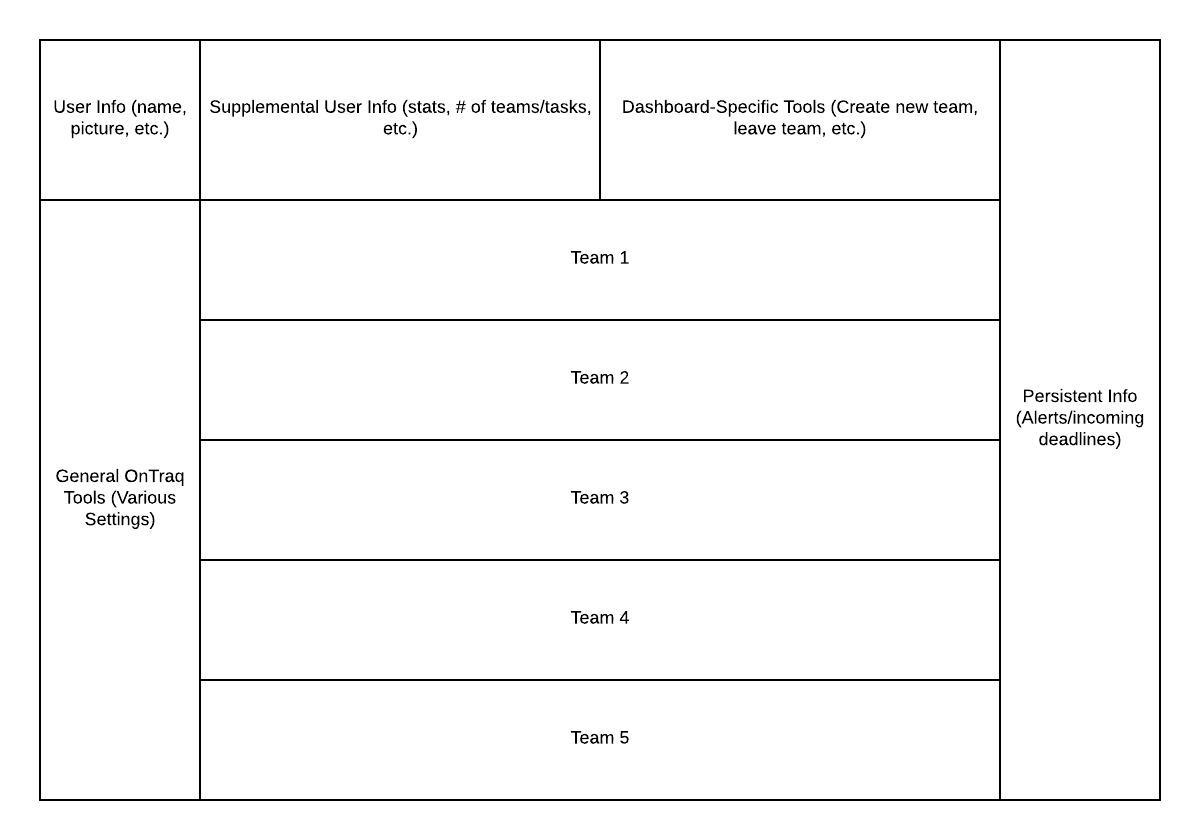
## Assumptions and Dependencies

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>

# External Interface Requirements

## User Interfaces

The primary interface will be a series of dashboards which are nested within one another. For example, the first dashboard a user will be presented with is one consisting of a list of teams the user belongs to, along with critical information displayed along with each one. The user can select one and will be brought to another dashboard which lists all the team members within that team.



## Hardware Interfaces (Optional)

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

## Software Interfaces

OnTraq will have an accompanying database which will store all objects (team members, tasks, etc.) necessary for functionality. The application will send changes to the server when a dashboard is closed and changes were made.

## Communications Interfaces

As stated above, OnTraq will send updated versions of edited objects and overwrite the ones in the database. The formatting of these updates will be rather simple, with the server doing most of the computation and the message simply containing what object needs to be changed, and which of its parameters need to be changed.

# System Features

This section outlines the various use cases for each possible actor (Developer, Manager, Overseer) in each page of the application.

## Primary User Dashboard

4.1.1 Description and Priority

This is the first dashboard a user will find themselves in once the program is run. It is therefore of greatest importance

4.1.2 Stimulus/Response Sequences

1. Create new team
   1. User selects the option to create a new team
   2. User names the team
   3. User can add team members to the team through previous teammates or through invite code.
2. Leave team
   1. User can leave a team by selecting the option
   2. Then checking a box next to the team(s) he or she wants to leave
3. View team dashboard
   1. User can enter a new dashboard view by clicking on one of the teams he or she is a part of.

4.1.3 Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use “TBD” as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1:

REQ-2:

## Team Dashboard

4.2.1 Description

This is the dashboard detailing all of the members of the team. The user can also change between member view and task view, displaying all of the tasks assigned to the team.

4.2.2 Stimulus/Response Sequences

* 1. Add/remove members
     1. Allows the user to add a new member to the group if they have been granted managerial permissions
  2. Switch to task view
     1. Can change to task view which displays all of the tasks which have been assigned to the group.
  3. View member
     1. Can change to the dashboard of an individual member.

4.3 Member Dashboard

4.3.1

This is the dashboard where the user can view the information specific to the selected member including member statistics, the tasks they are working on, and any groups they belong to.

4.3.2

* 1. View subtasks
     1. View the smaller components of the task that it has been divided into.
  2. Set task priority
     1. Set how important the task is in relevance to other tasks.
  3. Upload version (if applicable)
     1. If the task is something that can be completed and uploaded, the user can upload the latest version here.
  4. Set task status
     1. The user can change the task status between not started, started, and completed.

# Other Nonfunctional Requirements

## Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.

## Security Requirements

In the final version of the product, we will have encrypted communication between the server and the client. Additionally, the program will require a secure login.

## Software Quality Attributes

The program won’t demand any specific attributes, the interface also won’t allow for much confusion.

# Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>