

# Requirements Document

## Team 1

February 3, 2012

Table 1: Team

<b>Name</b>	<b>ID Number</b>
Jonathan Bergeron	9764453
Marc-André Faucher	9614729
Jeffrey How	9430954
Dmitry Kuznetsov	5679311
William Ling	9193480
Thomas Paulin	9333630
Alain Sakha	9770836
Kai Wang	5652723

# 1 System

## 1.1 Purpose

The purpose of this document is to define requirements for the Task Manager system.

The intended audience of this document is described in table 2.

Table 2: Targeted audience of this document.

Group of the readers	Reasons for reading
Users and customers	To give feedback about the requirements
System developers	To understand what functions and properties the system must contain
Testers	To test the system against the requirements

## 1.2 Context

Our goal is to develop a task and time management software system for game development projects.

Users of this system include programmers, graphic artists, model artists, webmasters, project managers and designers.

Our software creates and organizes a list of tasks on per user basis along with the time requirements so they can properly allocate resources needed to achieve each task.

## 1.3 Business Goals

Our goal is to offer game development companies a system to properly manage tasks and time related to a project.

Many methods of a various degree of efficiency are currently used to manage tasks and time. Some game developers use spreadsheets or paper solutions to name a few in order to manage resources.

Our solution to this problem is to offer them an easy to use system that standardizes the development process and helps coordination of tasks in order to achieve a common goal for the team.

## 2 Domain Concepts

**Task:** A piece of work assigned to an employee or a group of employees.

**Subtask:** An act that must be completed as an element of a task.

**Project:** A group of tasks that work towards the same vision.

**Gantt chart:** a type of bar chart which illustrates the start and finish dates of the tasks in a project.

**Dependency tree:** a directed graph representing dependencies of several tasks and sub-tasks.

**JVM:** Java Virtual Machine.

**OS:** Operating System, eg. Windows, Mac OSX, Linux.

## 3 Actors

Actors are the users that interact directly with the system. Different actors have different responsibilities and roles within the system.

**Manager:** Privileged employee responsible for coordinating a team of employees in order to accomplish a project. This includes executive producers, lead programmers, lead artists, and any other coordinator that the project might require. Some of his responsibilities are to manage tasks and employees who are assigned to the them.

**Administrator:** Another type of privileged employee who is responsible of managing the other employees. This is usually the responsibility of the human resources department. Some of his responsibilities include adding new employee and removing them from the system.

**Employee:** Any employee working for the company assumes this role. He is responsible for managing his own time.

## 4 Use Cases

### 4.1 System overview

The overall system is represented in figures 1-3.

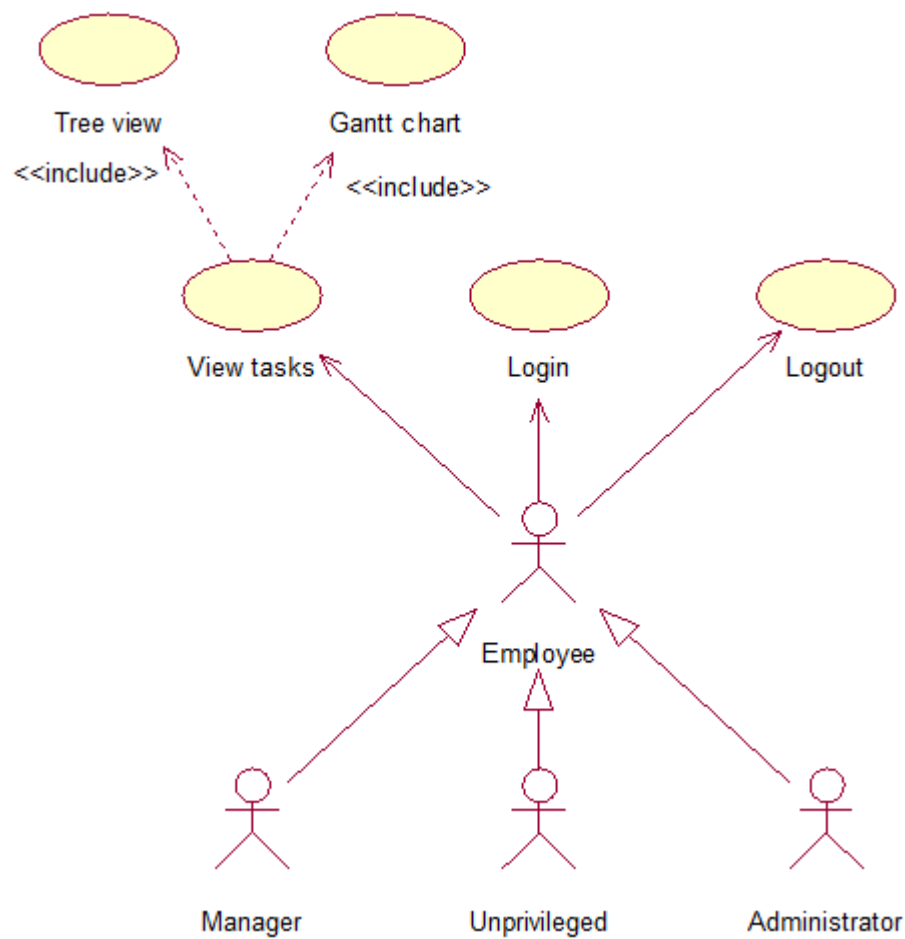


Figure 1: Use Case Diagram

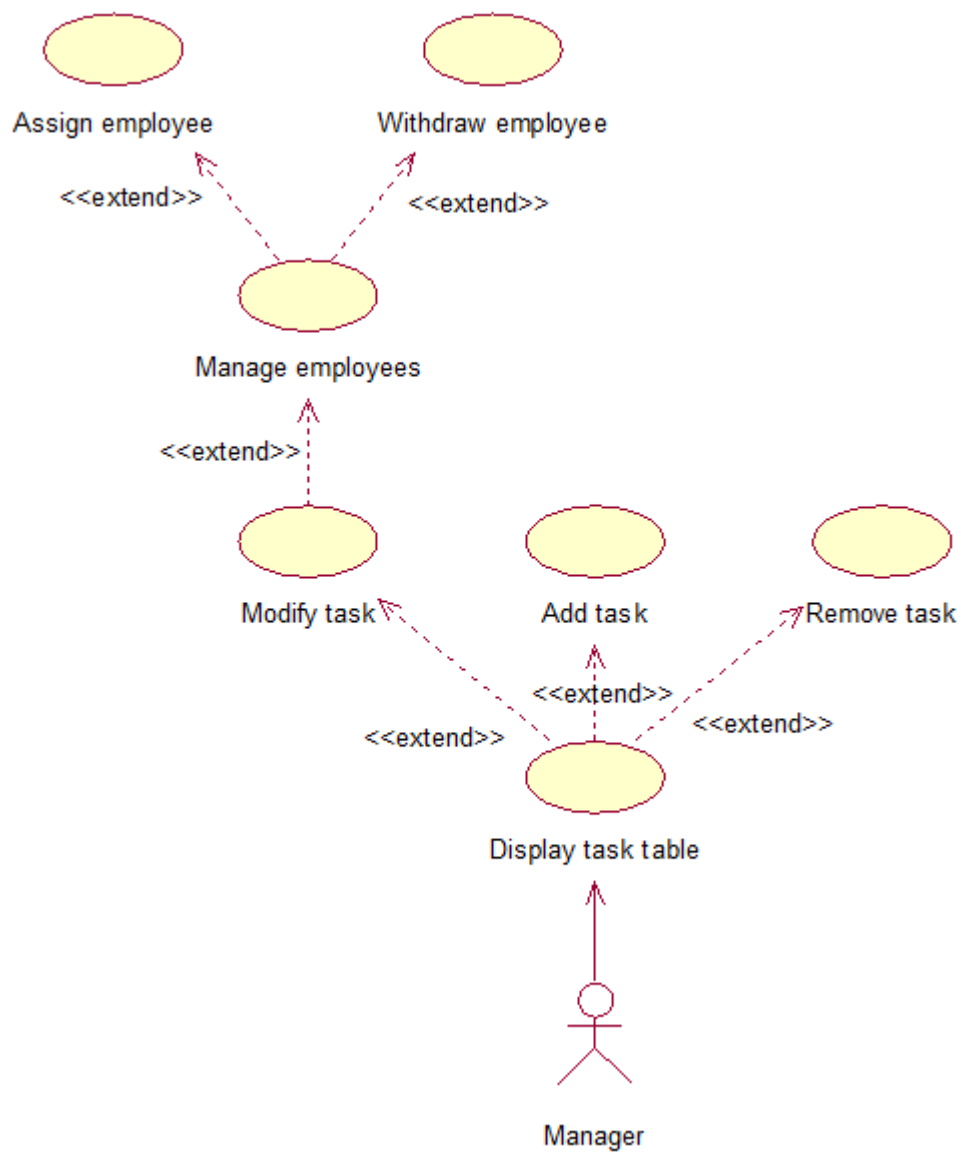


Figure 2: Use Case Diagram

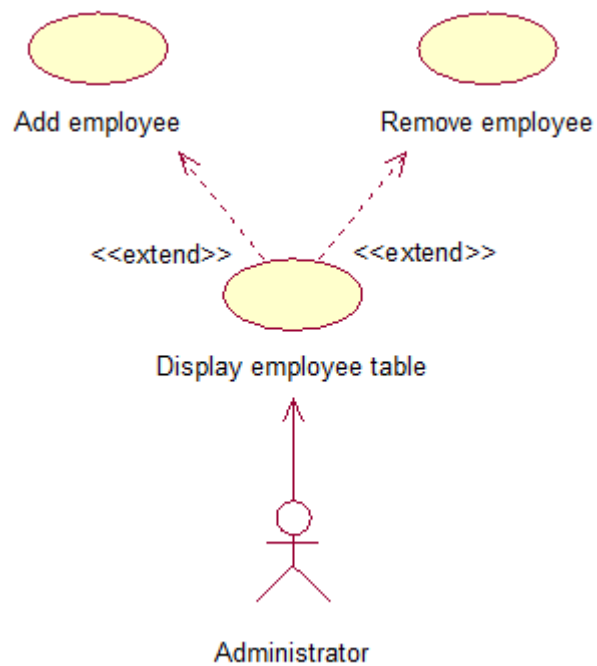


Figure 3: Use Case Diagram

## Use Case 1

### Name

Login

### Summary

The user requests access to the system. The user must be identified.

### Actors

Any user.

### Precondition

The system is active.

### Main Scenario

1. The system requests an identification (user name).
2. The user selects their name from a list.
3. The system grants the user access to the system.

### Exceptions

1. The person XML file is missing.

### Postcondition

The user is granted access to the system.

**Priority**

Must.

**Traces to requirements**

F1, N1.

**Use Case 2****Name**

Logout

**Summary**

The user terminates their session.

**Actors**

Any user.

**Precondition**

The user is logged into the system.

**Main Scenario**

1. The user clicks the logout button available after login.
2. The system terminates the session and requests a new identification.

**Postcondition**

The system remains active and awaits a new identification.

**Priority**

Must.

**Traces to requirements**

F1.

**Use Case 3****Name**

View tasks

**Summary**

The user requests a list of their tasks. They are presented with a menu from which they can choose different types of views for the data to be shown.

**Actors**

Any user.

**Precondition**

The user is in the “view tasks” menu.

**Main Scenario**

1. The user clicks the button that allows them to select a view.
2. The system verifies the user’s role.

3. The system displays a list of different views, some views are specific to the user's role.

**Exceptions**

1. The person XML file is missing.

**Postcondition**

The system displays a list of available views.

**Priority**

Must.

**Traces to requirements**

F6, F7.

**Use Case 4****Name**

Tree view

**Summary**

The user requests a dependency graph of their tasks.

**Actors**

Any user.

**Precondition**

The user has selected the tree view.

**Main Scenario**

1. The user selects the tree view option.
2. The system verifies the user's ID and role.
3. The system constructs a dependency tree based of the user's tasks.
4. The system renders and displays the graph.

**Exceptions**

1. The person XML file is missing.
2. The task XML file is missing.

**Postcondition**

The system displays a tree view.

**Priority**

Optional.

**Traces to requirements**

F6.



## **Use Case 5**

### **Name**

Gantt chart

### **Summary**

The user requests a schedule of their tasks.

### **Actors**

Any user.

### **Precondition**

The user is in the “view tasks” menu.

### **Main Scenario**

1. The user selects the gantt chart option.
2. The system verifies the user’s ID and role.
3. The system constructs a schedule of the user’s tasks.

### **Exceptions**

1. The person XML file is missing.
2. The task XML file is missing.

### **Postcondition**

The system displays a gantt chart.

### **Priority**

Optional.

### **Traces to requirements**

F7.

## **Use Case 6**

### **Name**

Display employee table

### **Summary**

The user requests a table of all employees in order to manage them.

### **Actors**

Administrators.

### **Precondition**

The user must be logged in.

### **Main Scenario**

1. The user selects the employee table button.
2. The system constructs a table of all employees.
3. The system renders and displays the employee table.

**Exceptions**

1. The person XML file is missing.

**Postcondition**

The system displays a view of the current employees and options related to employee management.

**Priority**

Must.

**Traces to requirements**

F10, F11, F12.

**Use Case 7****Name**

Add employee

**Summary**

The user wants to add an employee into the system.

**Actors**

Administrators.

**Precondition**

The user is in the employee table view.

**Main Scenario**

1. The user selects the option to add a new employee.
2. The system requests information about the new employee.
3. The user submits the information.
4. The system adds the employee into the system.

**Exceptions**

1. The person XML file is missing.

**Postcondition**

A new employee is added into the system.

**Priority**

Must.

**Traces to requirements**

F11.

**Use Case 8****Name**

Remove employee

**Summary**

The user wants to remove an employee.

**Actors**

Administrator.

**Precondition**

The user is in the employee table view.

**Main Scenario**

1. The user selects the option to remove a specific employee.
2. The system confirms his request.
3. The system removes the employee.

**Exceptions**

1. The person XML file is missing.

**Postcondition**

An employee is removed from the system.

**Priority**

Must.

**Traces to requirements**

F12.

**Use Case 9****Name**

Display task table

**Summary**

The user wants to see a view of tasks in order to manage them.

**Actors**

Managers.

**Precondition**

The user must be logged in.

**Main Scenario**

1. The user requests to see the task table.
2. The system shows the user a table with tasks and offers task management options.

**Exceptions**

1. The person XML file is missing.
2. The task XML file is missing.

**Postcondition**

A view is shown to the user that shows current tasks and options related to task management.

**Priority**

Must.

**Traces to requirements**

F2, F3, F4, F5, F8, F9, F13.

**Use Case 10****Name**

Add task

**Summary**

The user wants to add a task to the pool of tasks.

**Actors**

Managers.

**Precondition**

The user is in the task table view.

**Main Scenario**

1. The user requests to add a task.
2. The system requests information on the task to be added.
3. The user submits the information.
4. The system creates the task.

**Exceptions**

1. The person XML file is missing.
2. The task XML file is missing.

**Postcondition**

A new task is added into the system.

**Priority**

Must.

**Traces to requirements**

F3.

**Use Case 11****Name**

Remove task

**Summary**

The user wants to remove a task from the pool of tasks.

**Actors**

Managers.

**Precondition**

The user is in the task table view.

**Main Scenario**

1. The user requests to remove a specific task.
2. The system confirms his request.
3. The system removes the task.

**Exceptions**

1. The task XML file is missing.

**Postcondition**

A task is removed from the system.

**Priority**

Must.

**Traces to requirements**

F4.

**Use Case 12****Name**

Modify task

**Summary**

The user wants to modify a task in the pool of tasks and is given the option to manage employees related to that task.

**Actors**

Managers.

**Precondition**

The user is in the task table view.

**Main Scenario**

1. The user requests to modify a specific task.
2. The system displays a menu that shows properties of task and allows the user to modify them.
3. The system confirms his request.
4. The system modifies the task.

**Exceptions**

1. The person XML file is missing.
2. The task XML file is missing.

**Postcondition**

A previously created task is modified.

**Priority**

Must.

**Traces to requirements**

F5, F8, F9, F13.

**Use Case 13****Name**

Manage employees

**Summary**

The user wants to manage employees on a previously created task.

**Actors**

Managers.

**Precondition**

The user must be modifying a task.

**Main Scenario**

1. The user requests to manage employees.
2. An option menu appears to the user to give a choice of how to manage his employees.

**Exceptions**

1. The person XML file is missing.
2. The task XML file is missing.

**Postcondition**

A menu appears to show the user how he can manage employees.

**Priority**

Must.

**Traces to requirements**

F8, F9, F13.

**Use Case 14****Name**

Assign employee

**Summary**

The user wants to assign an employee to a task.

**Actors**

Managers.

**Precondition**

The user must be in the manage employees menu.

**Main Scenario**

1. The user chooses to assign an employee to the task.
2. The system shows a list of employees that can be assigned.
3. The user selects an employee.
4. The system updates the task.

**Exceptions**

1. The person XML file is missing.
2. The task XML file is missing.
3. No employees are registered into the system.

**Postcondition**

An employee is assigned to a task.

**Priority**

Must.

**Traces to requirements**

F9.

**Use Case 15****Name**

Withdraw employee

**Summary**

The user wants to withdraw an employee from a task.

**Actors**

Managers.

**Precondition**

The user must be in the manage employees menu.

**Main Scenario**

1. The user chooses to remove an employee from the task.
2. The system shows a list of people that are assigned to the task.
3. The user selects an employee.
4. The system confirms the choice of the user.
5. The system removes the employee from the task.

**Exceptions**

1. The person XML file is missing.
2. The task XML file is missing.

3. No employees are assigned on the task.

**Postcondition**

An employee is withdrawn from a task.

**Priority**

Must.

**Traces to requirements**

F13.

## 5 Functional requirements

### F1

**Requirement**

Only one person should be logged in at a time.

**Rationale**

Different actors have access to different functionalities in the system.

**Priority**

Must.

**Traces to use cases**

Login, logout.

### F2

**Requirement**

A view of tasks in the company is to be shown to the user in order for him to manage them.

**Rationale**

A manager can manage tasks and assign employees to work on them.

**Priority**

Must.

**Traces to use cases**

Display task table.

### F3

**Requirement**

The user is to be able to enter a new task.

**Rationale**

A manager must be able to create new tasks in order to help him manage his employees'



time.

**Priority**

Must.

**Traces to use cases**

Display task table, add task

**F4**

**Requirement**

The user is to be able to remove a previously created task.

**Rationale**

A manager must be able to remove tasks in order to help him manage his employees' time.

**Priority**

Must.

**Traces to use cases**

Display task table, remove task.

**F5**

**Requirement**

The user is to be able to modify a previously created task.

**Rationale**

A manager must be able to modify tasks in order to help him manage his employees' time.

**Priority**

Must.

**Traces to use cases**

Display task table, modify task.

**F6**

**Requirement**

The user is to be able to view a dependency graph of tasks.

**Rationale**

The user must be able to view the dependency relations between tasks in order for him to manage his own time.

**Priority**

Optional.

**Traces to use cases**

View tasks, Tree view.

**F7****Requirement**

The user is to be able to view a Gantt chart of tasks.

**Rationale**

The user must be able to view the a schedule of their tasks in order for him to manage his own time.

**Priority**

Optional.

**Traces to use cases**

View tasks, Gantt chart.

**F8****Requirement**

The user must be able to manage employees on a task.

**Rationale**

A manager must be able to modify the list of employees that are working on a task.

**Priority**

Must.

**Traces to use cases**

Display task table, modify task, Manage employees

**F9****Requirement**

The user must be able to assign employees to a task.

**Rationale**

A manager must be able to assign an employee to a task in order for the employee to receive information about the task.

**Priority**

Must.

**Traces to use cases**

Display task table, modify task, manage employees, assign employee.

## **F10**

### **Requirement**

The user must be able to view a list of employees in the company in order for him to manage them.

### **Rationale**

An administrator has to manage the workforce of the company since these might change during the course of a project.

### **Priority**

Must.

### **Traces to use cases**

Display employee table, add employee, remove employee.

## **F11**

### **Requirement**

The user is to be able to add a new employee.

### **Rationale**

An administrator must be able to add new employees into the system when hired to work on the project.

### **Priority**

Must.

### **Traces to use cases**

Add employee.

## **F12**

### **Requirement**

The user is to be able to remove an employee.

### **Rationale**

An administrator must be able to remove employees from the system when they cease to work on the project.

### **Priority**

Must.

### **Traces to use cases**

Remove employee.

## **F13**

### **Requirement**

The user must be able to withdraw employees from a task.

### **Rationale**

A manager must be able to withdraw an employee from a task to stop the system from showing information about the task to that employee.

### **Priority**

Must.

### **Traces to use cases**

Display task table, modify task, manage employees, Withdraw employee.

## **6 Non-Functional requirements**

### **N1**

#### **Requirement**

The system must know who the user is in order to offer the appropriate options.

#### **Rationale**

Different actors should have different options available to them.

#### **Priority**

Must.

#### **Traces to use cases**

Login.

### **O1**

#### **Requirement**

It shall be possible to run the system on any computer system that runs the JVM.

#### **Rationale**

The system should be accessible to the largest customer base, regardless of their OS.

#### **Priority**

Optional.

#### **Traces to use cases**

All use cases.

## **7 References**

## A Description of File Format: tasks.xml

`<tasks>` : Root node used to specify the beginning of the listing of tasks

`<task>` : Node used to specify a new task

`<identifier>` : A unique, alphanumeric identifier for each task

`<title>` : Node to store a short title for the task

`<description>` : Node to store a full description for the task

`<startdate>` : Node to store the appropriate start date for the task

`<duration>` : The duration, in hours of the task.

`<deliverable>` : The expected physical deliverable upon task completion

`<deadline>` : The date by which the task must be completed

`<peopleassigned>` : A listing of the individuals assigned to the task. The duration will be split evenly among assignees.

`<id>` : The corresponding identifier for each person assigned

`<completion>` : The percentage of completion for the task

## B Description of File Format: people.xml

<people> : Root node used to specify the beginning of the listing of people

<person> : Node used to specify a new person

<identifier> : A unique, alphanumeric identifier for each person

<fname> : Node used to store the first name of the person

<lname> : Node used to store the last name of the person

<jobtitle> : Node used to track the job title of the person. Used to differentiate between roles

<jobdescription> : Node used for the person's job description.

<clearance> : Node used to specify the clearance a person within the organization. Restricts permission based on value. 0 - Employee, 1 - Manager, 2 - Administrator

## C Description of File Format: people.txt

The list of people with a summary of their names, the total amount of time spent on all projects, and the specific projects to which they are assigned.

Name	Total Hours	Project List
LName, FName	X.X	A, B, C