

# Software Requirements Specification

for

# Coffee

**Prepared by**

**Team Name:** OverDOSE

Ruslan Mustafin <r.mustafin93@gmail.com>

Emil Melnikov <emilmelnikov@gmail.com>

Timur Shakirov <shakirovtr23@gmail.com>

**University:** Innopolis

**Country:** Russia

**October 19, 2014**

---

# Contents

- 1 INTRODUCTION
    - 1.1 Purpose
    - 1.2 Scope
    - 1.3 Definitions, Acronyms and Abbreviations
    - 1.4 References
    - 1.5 Overview
  - 2 OVERALL DESCRIPTION
    - 2.1 Product perspective
    - 2.2 Product functions
    - 2.3 User characteristics
    - 2.4 Constraints
    - 2.5 Assumptions and dependencies
  - 3 SPECIFIC REQUIREMENTS
    - 3.1 Functional Requirements
    - 3.2 Non-Functional Requirements
    - 3.3 Use Cases
-

# Revision History

Date	Version	Description	Author(s)
2014-09-27	0.1	Initial draft	Emil, Ruslan, Timur
2014-10-05	0.2	Added requirements specs	Emil, Ruslan, Timur
2014-10-12	0.3	Use cases first draft	Emil, Ruslan, Timur
2014-10-13	0.4	Elaborated functional requirements	Emil, Ruslan, Timur
2014-10-19	1.0	First public release	Emil, Ruslan, Timur

# 1 Introduction

## 1.1 Purpose

This document will define the product to be delivered in terms of its functionality, quality attributes, interfaces and user interface elements. This document will provide a description for the product to be created as well as specific requirements that the final product will meet.

## 1.2 Scope

This document will define the Coffee. The project will allow a group of developers to work efficiently and effectively in a distributed environment. The system assumes that the project is being developed through successive iterations. The system will include means for tracking and recording progress on project development. Any functionality, quality attributes, interface elements or any other elements of the framework not described in the detailed requirements of this document are considered to be outside the scope of this project.

## 1.3 Definitions, Acronyms and Abbreviations

- The system — The Coffee software system
- Guest — an unregistered user of the system
- User — a registered and authenticated user of the system
- Developer — a user who belongs to a project
- Project manager — a developer who has additional responsibilities for project administration and requirements editing
- Requirement - a specific feature that must be implemented
- Task -
- Project backlog - a log that consists of requirements and tasks that are associated with requirements
- Sprint - a single development iteration
- Sprint backlog - a log that contains tasks that must be implemented during the sprint. If the sprint is in progress, the tasks cannot be modified

## 1.4 References

- IEEE Standard 830-1998, 'IEEE Recommended Practice for Software Requirements Specifications'
-

## 1.5 Overview

This document contains an overall description of the product to be developed including its functionality and quality attributes. This document begins with an introduction to the product. Next, it will contain an overview of the products functions and characteristics. The third section will contain specific requirements stated in individual statements. Finally, the document will end with an appendix containing additional information about Coffee needed to understand this document.

## 2 Overall Description

### 2.1 Product perspective

The proposed system is a stand alone and self-contained web based application. The system will run on commodity hardware.

### 2.2 Product functions

Basic functionalities:

- Backlog: a collection of requirements that need to / could be implemented
  - Sprint log: a log that is used as part of a development sprint
  - User management: users need to be able to register with the system; users need to be able to assign users to projects; basic user management (delete account, change personal details)
  - Support for different roles (for example developers and 'master' similar to SCRUM master). Functionalities are available depending on user roles.
  - User authentication
  - Users can be assigned to requirements; maybe requirements need to be broken down into smaller tasks
  - Requirements can be assigned with 'points' according to their complexity. One can also assign point to tasks. Users who implement the requirements/tasks get the 'points' of the implemented requirements
-

- Developers chart: all developers are ranked according to the 'points' received from the implemented tasks/requirements

Advanced functionalities:

- A dashboard that summarizes ongoing events of (a) project(s)
- Statistics that show how the project performances, compared to the estimates
- Team communication tools (e.g. a discussion board or group chat)
- Integration with issue trackers (e.g. the tracker from the where the source code is hosted.. Github, Bitbucket, etc) or other platforms

## 2.3 User characteristics

1. Developers — represent persons who are responsible for implementing the project that is tracked by the system
2. Managers — represent persons who are responsible for creating and distributing tasks among developers. They also track the progress and make sure that the project is being delivered on time
3. Quality Assurance — represent persons who are responsible for checking software quality
4. Product owners — represent persons who are able to see the progress and who can be interviewed about problem domain

## 2.4 Constraints

1. The system should be implemented in Eiffel programming language
2. The system should be a web based application

## 2.5 Assumptions and dependencies

Users are expected to use the latest versions of browsers.

# 3 Specific Requirements

This section contains all software requirements both functional and non-functional. A requirement has the following properties:

**Priority** defines the order in which requirements should be implemented. Priorities are designated (highest to lowest) from 1 to 3. Requirements of priority 1 are mandatory; 2 represents features nice to have, and 3 represents optional features.

---

**Risk** specifies the risk of not implementing the requirement. It shows how critical the requirement is to the system as a whole. The following risk levels are defined over the impact of not being implemented correctly.

- Critical (C) It will break the main functionality of the system. The system cannot be used if this requirement is not implemented.
- High (H) It will impact the main functionality of the system. Some function of the system could be inaccessible, but the system can be generally used.
- Medium (M) It will impact some system features, but not the main functionality. The system can still be used with some limitation.
- Low (L) The system can be used without limitation, but with some workarounds

### 3.1 Functional Requirements

<b>Requirement ID</b>	FR-01
<b>Title</b>	The system shall allow guest to create and manage an account
<b>Description</b>	Guests can create accounts in the system. Users can edit account information and delete the account.
<b>Priority</b>	Mandatory (1)
<b>Risk</b>	Critical (C)

<b>Requirement ID</b>	FR-02
<b>Title</b>	The system shall allow user to create and manage a new project
<b>Description</b>	<p>Registered users can create a new project, edit project information. The user that created the project becomes a project manager. Project manager can add other registered users to the project as developers. Project manager can remove developers from the project.</p> <p>Manager can also transfer management responsibilities to a developer in the project. In this case the manager becomes a developer and a developer becomes the manager.</p>
<b>Priority</b>	Mandatory (1)
<b>Risk</b>	Critical (C)

<b>Requirement ID</b>	FR-03
<b>Title</b>	The system shall allow project manager to manage requirements in the project backlog
<b>Description</b>	<p>Project managers can edit project backlog by adding, removing or modifying requirements in the project backlog.</p> <p>When a requirement is removed from project backlog, all tasks that are associated with this requirement are deleted.</p>

---

	If the task is in the sprint backlog, it cannot be removed. Therefore, the requirement that this task is associated with cannot be deleted neither
<b>Priority</b>	Mandatory (1)
<b>Risk</b>	Critical (C)

<b>Requirement ID</b>	FR-04
<b>Title</b>	The system shall allow project manager to specify priority and risk for requirements.
<b>Description</b>	When adding the requirement to the project backlog, project manager should specify priority and risk of the requirement.
<b>Priority</b>	Nice to have (2)
<b>Risk</b>	Low (L)

<b>Requirement ID</b>	FR-05
<b>Title</b>	The system shall allow project manager to split requirements into tasks.
<b>Description</b>	Project managers can split requirements into tasks. These tasks are added to the project backlog for the later use in the sprint.
<b>Priority</b>	Mandatory (1)
<b>Risk</b>	Critical (C)

<b>Requirement ID</b>	FR-06
<b>Title</b>	The system shall allow project manager to run sprints.
<b>Description</b>	Project managers can create sprints. Tasks that should be implemented are added to the sprint backlog.
<b>Priority</b>	Mandatory (1)
<b>Risk</b>	Critical (C)

<b>Requirement ID</b>	FR-07
<b>Title</b>	The system shall allow project manager to estimate working time for the tasks.
<b>Description</b>	During the sprint creation, project manager should estimate working time for the tasks that they are added to the sprint backlog.
<b>Priority</b>	Mandatory (1)
<b>Risk</b>	Critical (C)

---



--	--

<b>Requirement ID</b>	FR-08
<b>Title</b>	The system shall allow developers to implement tasks from the sprint backlog
<b>Description</b>	Developers can take tasks from the sprint backlog and implement them.
<b>Priority</b>	Mandatory (1)
<b>Risk</b>	Critical (C)

<b>Requirement ID</b>	FR-09
<b>Title</b>	The system shall allow developers log to their working time
<b>Description</b>	When closing the taken task, developers should specify the time that they spent working on this task.
<b>Priority</b>	Mandatory (1)
<b>Risk</b>	Critical (C)

<b>Requirement ID</b>	FR-10
<b>Title</b>	The system shall allow project members to view recent project activities
<b>Description</b>	Developers and project memebtrs can view the sprint history
<b>Priority</b>	Nice to have (2)
<b>Risk</b>	Low (L)

<b>Requirement ID</b>	FR-11
<b>Title</b>	The system shall allows project members to view overall project statistics
<b>Description</b>	The system generates statistics from completed sprints based on time estimation
<b>Priority</b>	Optional (3)
<b>Risk</b>	Low (L)

<b>Requirement ID</b>	FR-12
<b>Title</b>	The system shall move unfinished sprint tasks back to the project backlog
<b>Description</b>	If, at the end of the sprint, there are tasks that are not closed, they should be automatically moved back to the project backlog
<b>Priority</b>	Mandatory (1)

---

<b>Risk</b>	Critical (C)
-------------	--------------

<b>Requirement ID</b>	FR-13
<b>Title</b>	The system shall mark requirements with all finished tasks as completed
<b>Description</b>	When sprint has finished, the requirements for which all the tasks are implemented marked as completed. Implemented requirements should be moved to Finished backlog
<b>Priority</b>	Nice to have (2)
<b>Risk</b>	Low (L)

<b>Requirement ID</b>	FR-14
<b>Title</b>	The system shall rank all developers of a project according to their points
<b>Description</b>	All developers of a project must be ranked according to earned points
<b>Priority</b>	Nice to have (2)
<b>Risk</b>	Low (L)

## 3.2 Non-Functional Requirements

### 3.2.1. Security

The system shall not allow unauthenticated actors to access the system. The system should not expose any information about projects and users to unauthenticated actors.

Passwords must be stored securely.

The system should be protected against common attacks on web-application, such as CSRF, XSS and code injections.

## 3.3 Use Cases

<b>Use Case ID</b>	UC-01
<b>Title</b>	Registartion
<b>Primary Actor</b>	Guest
<b>Precondition</b>	-
<b>Postcondition</b>	Guest becomes a user
<b>Trigger</b>	Guest initiates a registration process
<b>Main flow</b>	1. Guest specifies the following information:

---

	<ol style="list-style-type: none"> <li>1.1. Email (Required)</li> <li>1.2. Password (Required, no shorter than 6 symbols)</li> <li>1.3. First name (Optional)</li> <li>1.4. Last name (Optional)</li> <li>2. Guest submits entered information</li> <li>3. The system informs the guest that the registration is successful</li> </ol>
<b>Alternate flow</b>	<ol style="list-style-type: none"> <li>1. The user with the same email address is already registered:               <ol style="list-style-type: none"> <li>1.1. The system informs guest that submitted email is already in use</li> <li>1.2. The system prompts the guest to specify information again, preserving previously submitted information, excluding the password</li> </ol> </li> <li>2. One of the required fields is missing or has invalid format:               <ol style="list-style-type: none"> <li>2.1. The system informs guest that some specified information is invalid</li> <li>2.2. The system prompts the guest to specify information again, preserving previously submitted information, excluding the password</li> </ol> </li> </ol>
<b>Notes</b>	

<b>Use Case ID</b>	UC-02
<b>Title</b>	Delete account
<b>Primary Actor</b>	User
<b>Precondition</b>	-
<b>Postcondition</b>	User becomes a guest
<b>Trigger</b>	User initiates a account deletion process
<b>Main flow</b>	<ol style="list-style-type: none"> <li>1. User requests to delete his/her account</li> <li>2. The system prompts user to confirm the action</li> <li>3. User confirms the action</li> </ol>
<b>Alternate flow</b>	<ol style="list-style-type: none"> <li>1. The user doesn't confirm the action               <ol style="list-style-type: none"> <li>1.1. The system informs the user that his/her account is still active</li> </ol> </li> <li>2. The user is a project manager of one or more projects:               <ol style="list-style-type: none"> <li>2.1. The system informs the user that it is impossible to delete his/her account while the user is a project manager of one or more</li> </ol> </li> </ol>

	projects. The system suggests to delegate project management responsibilities to another developer
<b>Notes</b>	The system must not delete user statistics in all projects that they participated in

<b>Use Case ID</b>	UC-03
<b>Title</b>	Create project
<b>Primary Actor</b>	User
<b>Precondition</b>	-
<b>Postcondition</b>	<ol style="list-style-type: none"> <li>1. A new project is created in the system</li> <li>2. User becomes project manager of a newly created project</li> </ol>
<b>Trigger</b>	User requests to create a new project
<b>Main flow</b>	<ol style="list-style-type: none"> <li>1. The user specifies the following information: <ol style="list-style-type: none"> <li>1.1. Project name (Required)</li> <li>1.2. Project ID (Required)</li> <li>1.3. Project description (Optional)</li> </ol> </li> <li>2. The user submits information</li> <li>3. The system informs user that the project is created</li> </ol>
<b>Alternate flow</b>	<ol style="list-style-type: none"> <li>1. The project with the same ProjectID is already created: <ol style="list-style-type: none"> <li>1.1. The system informs user that submitted ProjectID is already in use</li> <li>1.2. The system prompts the user to specify information again, preserving previously submitted information</li> </ol> </li> <li>2. One of the required fields is missing or has invalid format: <ol style="list-style-type: none"> <li>2.1. The system informs the user that some specified information is invalid</li> <li>2.2. The system prompts the user to specify information again, preserving previously submitted information</li> </ol> </li> </ol>
<b>Notes</b>	It is recommended to use only English uppercase and lowercase letters digits, hyphen and underscore for Project ID.

<b>Use Case ID</b>	UC-04
<b>Title</b>	Add users to project
<b>Primary Actor</b>	Project Manager
<b>Precondition</b>	-
<b>Postcondition</b>	User becomes developer
<b>Trigger</b>	Project Manager requests to add users to project
<b>Main flow</b>	<ol style="list-style-type: none"> <li>1. Project manager specifies the users that they want to add to the project</li> </ol>

	<ol style="list-style-type: none"> <li>2. Project manager submits the information</li> <li>3. The system informs that the request is successful</li> </ol>
<b>Alternate flow</b>	<ol style="list-style-type: none"> <li>1. Some of the listed users were not added to the project:               <ol style="list-style-type: none"> <li>1.1. The system informs the project manager which users were not added and the reason</li> </ol> </li> </ol>
<b>Notes</b>	Implementation suggestion: use list of email addresses as user IDs

<b>Use Case ID</b>	UC-05
<b>Title</b>	Developer leaves a project
<b>Primary Actor</b>	Developer
<b>Precondition</b>	-
<b>Postcondition</b>	Developer is not associated with the project anymore
<b>Trigger</b>	Developer initiates the process of removing themselves from the project
<b>Main flow</b>	<ol style="list-style-type: none"> <li>1. Developer submits removal request</li> <li>2. System asks for confirmation</li> <li>3. Developer confirms the action</li> <li>4. The system informs that the developer has left the project</li> </ol>
<b>Alternate flow</b>	<ol style="list-style-type: none"> <li>1. The developer is a project manager of this project               <ol style="list-style-type: none"> <li>1.1. The system informs the developer that it is impossible to remove them from the project. The system suggests to delegate project management responsibilities to another developer</li> </ol> </li> </ol>
<b>Notes</b>	

<b>Use Case ID</b>	UC-06
<b>Title</b>	Add requirement
<b>Primary Actor</b>	Project Manager
<b>Precondition</b>	-
<b>Postcondition</b>	<ol style="list-style-type: none"> <li>1. Unique requirement ID is generated by the system and assigned to the requirement</li> <li>2. Requirement is added to project backlog</li> </ol>
<b>Trigger</b>	Project manager requests to add requirement
<b>Main flow</b>	<ol style="list-style-type: none"> <li>1. Project manager specifies the following information about a requirement:               <ol style="list-style-type: none"> <li>1.1. Title (Required)</li> <li>1.2. Description (Optional)</li> <li>1.3. Priority (Required)</li> </ol> </li> </ol>

---

	<ol style="list-style-type: none"> <li>1.4. Risk (Required)</li> <li>2. Project manager submits the information</li> <li>3. The system confirms that the requirement was successfully created</li> <li>4. System prompts to add another requirement</li> </ol>
<b>Alternate flow</b>	<ol style="list-style-type: none"> <li>1. Some required information was not specified:               <ol style="list-style-type: none"> <li>1.1. The system informs the user that some specified information is invalid</li> <li>1.2. The system prompts the user to specify information again, preserving previously submitted information</li> </ol> </li> </ol>
<b>Notes</b>	

<b>Use Case ID</b>	UC-07
<b>Title</b>	Break down requirement into tasks
<b>Primary Actor</b>	Project Manager
<b>Precondition</b>	-
<b>Postcondition</b>	<ol style="list-style-type: none"> <li>1. The system generates a unique to the whole system TaskID and assigns it to the task</li> <li>2. The system marks the task as "Open"</li> <li>3. Associated tasks are added to project backlog</li> </ol>
<b>Trigger</b>	
<b>Main flow</b>	<ol style="list-style-type: none"> <li>1. Project manager selects requirement that they want to break down</li> <li>2. Project manager creates a new task for selected requirement</li> <li>3. Project manager inputs following information about task:               <ol style="list-style-type: none"> <li>3.1. Title (Required)</li> <li>3.2. Description (Optional)</li> <li>3.3. Points (Required, positive number)</li> </ol> </li> <li>4. project manager submits the information</li> <li>5. The system informs project manager that the task is successfully added</li> <li>6. The system prompts project manager to add another task to the same requirement</li> </ol>
<b>Alternate flow</b>	
<b>Notes</b>	

<b>Use Case ID</b>	UC-08
<b>Title</b>	Create sprint
<b>Primary Actor</b>	Project Manager
<b>Precondition</b>	-
<b>Postcondition</b>	The sprint is created
<b>Trigger</b>	
<b>Main flow</b>	<ol style="list-style-type: none"> <li>1. Project manager creates new sprint</li> <li>2. Project manager selects tasks from project backlog</li> <li>3. For each selected task, project manager specifies estimated implementation time</li> <li>4. Project manager specifies start date</li> <li>5. Project manager specifies sprint duration</li> <li>6. Project manager submits the information</li> <li>7. System informs project manager that the sprint will launch at the specified start date</li> </ol>
<b>Alternate flow</b>	<ol style="list-style-type: none"> <li>1. Selected start date is in the past:               <ol style="list-style-type: none"> <li>1.1. The system informs project manager that sprint could not be launched from the selected date</li> <li>1.2. The system asks project manager to select another date while preserving all specified information</li> </ol> </li> <li>2. No tasks selected:               <ol style="list-style-type: none"> <li>2.1. The system informs project manager that sprint could not be launched with empty tasks</li> <li>2.2. The system asks project manager to add at least one task while preserving all specified information</li> </ol> </li> <li>3. Some tasks are not estimated with respect to time:               <ol style="list-style-type: none"> <li>3.1. The system informs project manager that sprint could not be launched while there are tasks with no time estimation</li> <li>3.2. The system asks project manager to estimate all tasks while preserving all specified information</li> </ol> </li> </ol>
<b>Notes</b>	

<b>Use Case ID</b>	UC-09
<b>Title</b>	Select task from sprint backlog
<b>Primary Actor</b>	Developer
<b>Precondition</b>	-
<b>Postcondition</b>	1. The task is marked "In Progress" in the project backlog

	2. The task is added to the developer's task list
<b>Trigger</b>	
<b>Main flow</b>	<ol style="list-style-type: none"> <li>Developer selects task from a sprint backlog</li> <li>Developer changes the state of the task to "In progress"</li> <li>System informs the developer that the task is added to his/her backlog</li> </ol>
<b>Alternate flow</b>	
<b>Notes</b>	

<b>Use Case ID</b>	UC-10
<b>Title</b>	Close task
<b>Primary Actor</b>	Developer
<b>Precondition</b>	-
<b>Postcondition</b>	<ol style="list-style-type: none"> <li>The task is closed</li> <li>Working hours are added to the task</li> <li>The task is removed from the developer's task list</li> </ol>
<b>Trigger</b>	
<b>Main flow</b>	<ol style="list-style-type: none"> <li>Developer selects the task that they want to close</li> <li>Developer specifies the amount of hours that they spent implementing the task</li> <li>Developer changes the state of the task to "Closed"</li> <li>System informs the developer that the task is closed</li> </ol>
<b>Alternate flow</b>	<ol style="list-style-type: none"> <li>Developer did not specify the working hours <ol style="list-style-type: none"> <li>System informs the developer that they need to specify working hours</li> <li>The system prompts the user to specify information again, preserving previously submitted information</li> </ol> </li> </ol>
<b>Notes</b>	