

Software Requirements Specification for Let's Go!

Prepared by

Team Name: Group 7

Caroline Queiroz Santos

<carol.qs@gmail.br>

Graziela Basilio Pereira

<graziela.basilio@acad.pucrs.br>

University: PUCRS - PONTIFICAL CATHOLIC UNIVERSITY OF RIO GRANDE DO SUL -
FACULTY OF INFORMATICS.

Country: Brazil

October 20, 2014

Revision History

Date	Version	Description	Author(s)
October 08, 2014	V1.0	Document created	Graziela Basilio Pereira
October 20, 2014	V1.0	Document update	Caroline Queiroz Santos
October 20, 2014	V1.0	Document update	Graziela Basilio Pereira

1 Introduction

This document describes and specifies the needs of the customer, being the main source of information for design development of a product to support distributed software development teams.

1.1 Purpose

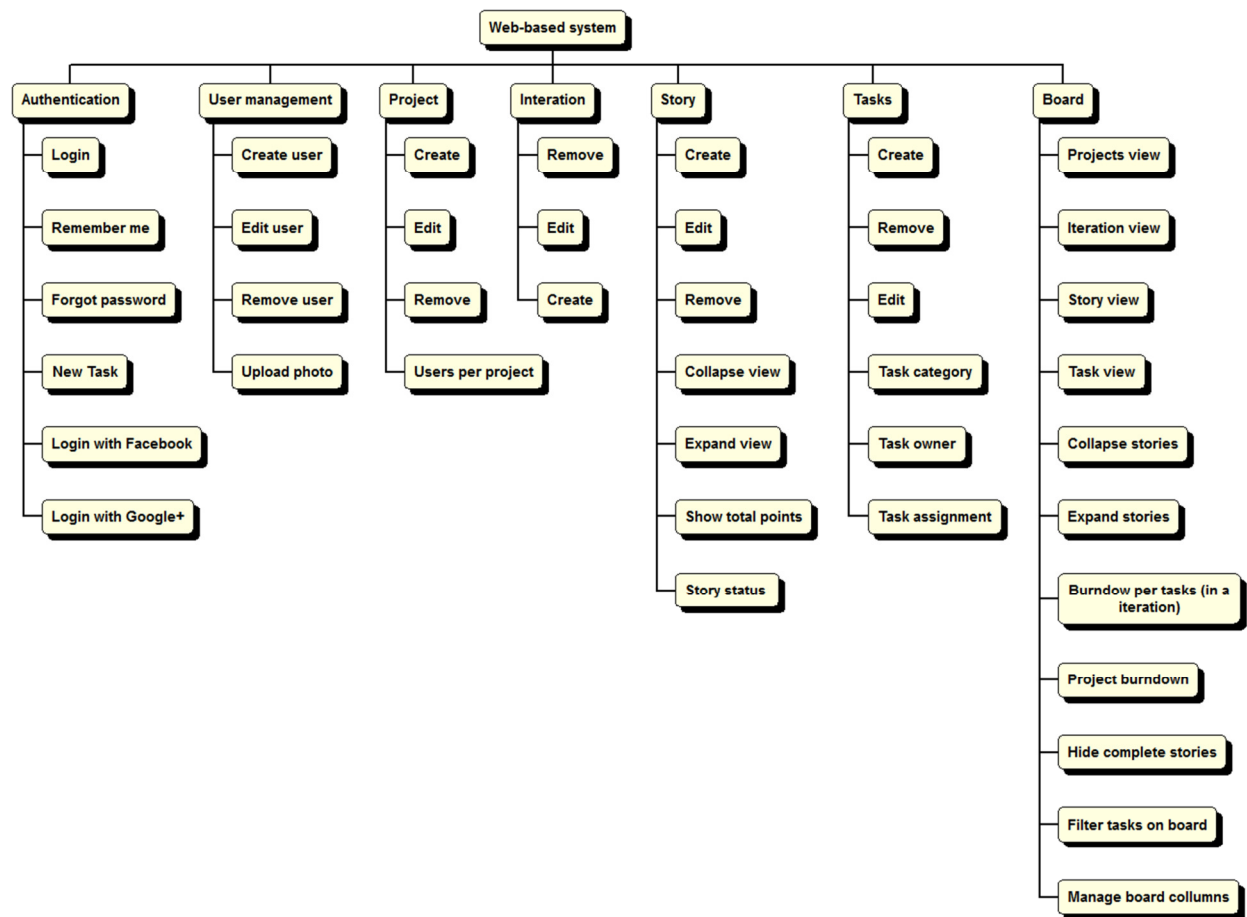
This requirements specification fully describes the external behavior of the application. It also describes nonfunctional requirements, design constraints, and other factors necessary to provide a complete and comprehensive description of the requirements for the software.

1.2 Scope

The scope of this project is to web-based project management system to support distributed software development teams.

- Project develops through successive iterations
- Team members, managers and other stakeholders need to record and track progress

The figure below shows the product structure and its scope.



1.3 Definitions, Acronyms and Abbreviations

UC: use case

Kanban: on going board.

1.4 References

Date	Document	Resources
August	Project overview	Prof ^a Sabrina
September, 12	Assignment #2, Part 1: To write clarification requestt	Clients Time of analysts

1.5 Overview

The SRS contains the description of the functional and non functional requirements about the systems inthe section 3.1 and 3.2. In the section 3.3 are presented the use case diagram and the user cases's description. Finally, to improve the comprehension and supporting information the documentation brings the main screen system's in the section 4.

2 Overall Description

2.1 Product perspective

2.2 Product functions

Basic Functions

- 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 where the source code is hosted. Github, Bitbucket, etc) or other platforms

2.3 User characteristics

The system users will be software developers teams.

2.4 Constraints

The system does not provide:

- cost management
- people management
- mobile interface

2.5 Assumptions and dependencies

- A project can be created by any user.
- Work management will be done throughout the project team.

3 Specific Requirements

3.1 Functional Requirements

Requirement ID	[FR1]
Title	Authentication
Description	Authentication system for user login Authorization services let users provide your application with access to read, write, update and delete the data they have stored in project management system.
Priority	Priority 1 - mandatory.
Risk	Critical (C) - It will break the main functionality of the system. The system cannot be used if this requirement is not implemented.

Requirement ID	[FR2]
Title	Task information
Description	The system shall provide tasks information like description, duration, resources and comments.
Priority	Priority 1 - mandatory.
Risk	Critical (C) - It will break the main functionality of the system. The system cannot be used if this requirement is not implemented.

Requirement ID	[FR3]
Title	Collaboration
Description	The system shall provide collaboration mechanisms like chat, comments
Priority	Priority 2 - feature nice.
Risk	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.

Requirement ID	[FR4]
Title	Schedule
Description	The system shall provide schedule mechanisms like Gantt chart, calendar and time tracking.

Priority	Priority 2 - feature nice.
Risk	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.

Requirement ID	[FR5]
Title	Resources agiles
Description	The system shall provide support agile methods like Burndown (project, sprint, release), Sprint backlog, Project backlog.
Priority	Priority 1 - mandatory.
Risk	Critical (C) - It will break the main functionality of the system. The system cannot be used if this requirement is not implemented.

3.2 Non-Functional Requirements

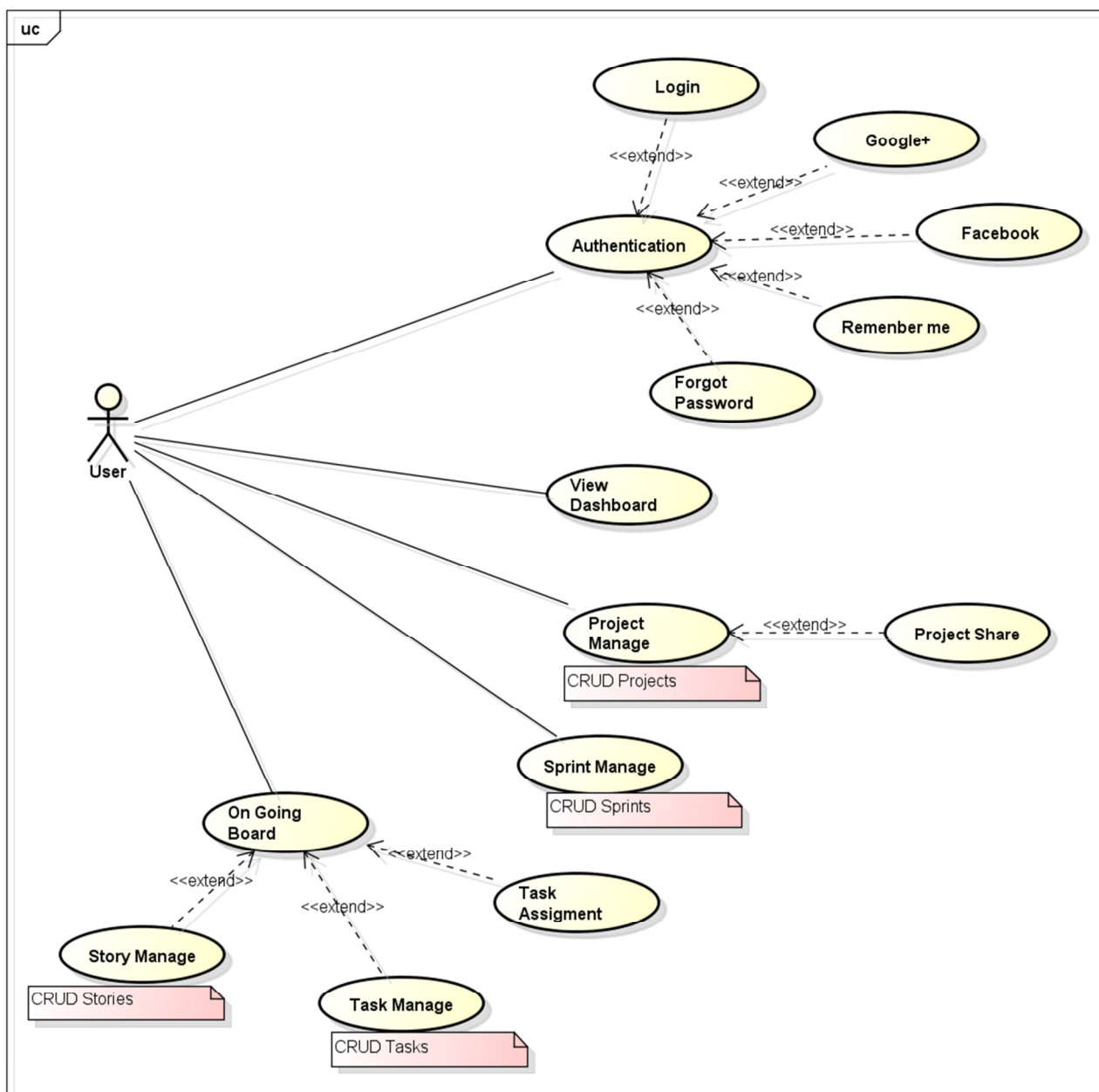
Requirement ID	NFR1
Title	Performance
Description	<ol style="list-style-type: none"> 1. Perform of the system to read and write data. 2. Response times 3. Processing times 4. Query and Reporting times
Priority	Priority 2 - feature nice.
Risk	Low (L) The system can be used without limitation, but with some workarounds.

Requirement ID	NFR2
Title	Availability
Description	<ol style="list-style-type: none"> 1. Hours of operation 2. Locations of operation
Priority	Priority 1 - mandatory.
Risk	Critical (C) - It will break the main functionality of the system. The system cannot be used if this requirement is not implemented.

Requirement ID	NFR3
-----------------------	------

Title	Security
Description	<ol style="list-style-type: none"> 1. Login requirements (authentication) - access levels 2. Password requirements - length, special characters, expiry date. 3. Inactivity timeouts - durations and actions
Priority	Priority 1 - mandatory.
Risk	Critical (C) - It will break the main functionality of the system. The system cannot be used if this requirement is not implemented.

3.3 Use Cases



Use Case ID	UC 1.0
Use Case Name	Authentication
Description	This use case Authentication (login access) details the login steps of a user; it is necessary to gain access to other functionality of the system.
Actors	User
Preconditions	None
Normal flow	<ol style="list-style-type: none"> 1. The system presents fields for login, password and a list of language. 2. The user inserts her/ his e-mail (username). 3. The user inserts her/ his password. 4. The user selects a language. 5. The user clicks on Login button. 6. The use case ends successfully.
Alternative flows	None
Exceptions	<ol style="list-style-type: none"> 1. Invalid User or Password <ol style="list-style-type: none"> a. If username or password cannot be located, the system presents a message "Incorrect username or password combination. Try again!". b. The system presents two options: "Forgot your password?" and "Not Registered?"
Posconditions	The system locates the user and verifies the access rights.
Notes and issues	<ol style="list-style-type: none"> 1. Logging or Username is usually used to enter a specific software (web page) or computer system. A password is a secret word or string of characters that is used for authentication, to prove identity or gain access to a systems. 2. The system default language is English.

Use Case ID	UC 2.0
Use Case Name	Project Manage
Description	This UC allows to perform the actions to: create, read (retrieve), delete and update the projetcts
Actors	User
Preconditions	User can be logged
Normal flow	<ol style="list-style-type: none"> 1. Create: <ol style="list-style-type: none"> 1.1. User chooses create new project 1.2. User fills the project's description 1.3. User defines the date to start and completion date 1.4. User indicates the current projetc's status: open, close 1.5. User submit the project 1.6. The system shows a confirmation message "Projetc created successfully!" 2. Retrieve:

	<ol style="list-style-type: none"> 2.1. User chooses the option to view a list of projects 2.2. The system presents a list of projects 2.3. User select a project 2.4. The system open the project selected 2.5. 3. Update: <ol style="list-style-type: none"> 3.1. User chooses to edit the project 3.2. User submit the project edited 3.3. The system shows a confirmation message "Projetc edited successfully!" 4. Delete (Remove Project): <ol style="list-style-type: none"> 4.1. User chooses to delete the project 4.2. The system presents a message to confirm the action 4.3. User confirms the action (remove project) 4.4. The system presents a confirmation message "Projetc removed successfully!"
Alternative flows	The project already exists
Exceptions	<ol style="list-style-type: none"> 1. The user aborts the use case. 2. The system doesn't conclude the action. 3. The use case ends.
Posconditions	The system locates the project.
Notes and issues	None

Use Case ID	UC 2.1
Use Case Name	Project Share
Description	This UC allows users to share the projetcs with other stakeholders or key people of the company.
Actors	User
Preconditions	<ol style="list-style-type: none"> 1. User can be logged 2. Just will be able to invite registered emails
Normal flow	<p>Invite People</p> <ol style="list-style-type: none"> 1. User indicates the emails of the people that will recive the project 2. User writes some message about the project or other information 3. User submit the invitation 4. User submit the message 5. System shows the dialogue box "Message sent with success!" <p>Clear Invitation</p> <ol style="list-style-type: none"> 1. User can decide to clear the fields: invite people or message 2. User press the "clear option" 3. System delete the description's field <p>Edit Invitation</p> <ol style="list-style-type: none"> 1. User edits the fields: invite people and message 2. System shows the dialogue box

Alternative flows	<ol style="list-style-type: none"> 1. User invites members that not have access on the platform 2. The platform notifies that the user don't have a login (or valid login)
Exceptions	<ol style="list-style-type: none"> 1. The user aborts the use case. 2. The system doesn't conclude the action. 3. The use case ends.
Posconditions	The system to share the projetcs with other stakeholders.
Notes and issues	Any project can be sharing with others (or just current projects?)

Use Case ID	UC 3.0
Use Case Name	Sprint Manage
Description	This UC allows to perform the actions to: create, read (retrieve), delete and update sprints
Actors	User
Preconditions	User can be logged
Normal flow	<ol style="list-style-type: none"> 1. Create: <ol style="list-style-type: none"> 1.1. User chooses create new sprint 1.2. User fills the sprint's name 1.3. User defines the date to start and completion date 1.4. User indicates the current sprint's status: future, closed, planned, in progress 1.5. User submit the sprint 1.6. The system presents a confirmation message "Sprint created successfully!" 2. Retrieve: <ol style="list-style-type: none"> 2.1. User chooses the option to view a list of sprints 2.2. The system presents a list of sprints 2.3. User select a sprint 2.4. The system open the sprint selected 3. Update: <ol style="list-style-type: none"> 3.1. User chooses to edit the sprint 3.2. User changes data(s) 3.3. User submit the sprint edited 3.4. The system presents a confirmation message "Sprint edited successfully!" 4. Delete (Remove Sprint): <ol style="list-style-type: none"> 4.1. User chooses to delete the sprint 4.2. The system presents a message to confirm the action 4.3. User confirms the action (remove sprint) 4.4. The system presents a confirmation message "Projetc removed successfully!"
Alternative flows	None
Exceptions	<ol style="list-style-type: none"> 1. The user aborts the use case. 2. The system doesn't conclude the action.

	3. The use case ends.
Posconditions	None
Notes and issues	A sprint belongs to a project.

Use Case ID	UC 4.0
Use Case Name	On Going Board (story board)
Description	This UC represents the system to control the software development process. The users will be able to add new stories, management stories, and include new tasks
Actors	Users
Preconditions	User can be logged
Normal flow	<ol style="list-style-type: none"> 1. User chooses create new story board 2. User fills the story board description 3. User manages stories (UC 4.1), and manages tasks (UC4.2) 4. The system presents a confirmation message "Story Board created successfully" 5. User moves the tasks to other process steps. (not started, in progress, completed and blocked)
Alternative flows	None
Exceptions	<ol style="list-style-type: none"> 1. The user aborts the use case. 2. The system doesn't conclude the action. 3. The use case ends.
Posconditions	None
Notes and issues	None

Use Case ID	UC 4.1
Use Case Name	Story Manage
Description	This UC allows to perform the actions to: create, read (retrieve), delete and update stories.
Actors	User
Preconditions	User can be logged
Normal flow	<ol style="list-style-type: none"> 1. Create: <ol style="list-style-type: none"> 1.1. User chooses create new story 1.2. User fills story's title (mandatory) 1.3. User fills story's description (optional) 1.4. User fills story's points (mandatory) 1.5. User fills notes (optional) 1.6. User submit the story

	<ol style="list-style-type: none"> 1.7. The system presents a confirmation message “Story created successfully!” 2. Retrieve: <ol style="list-style-type: none"> 2.1. User chooses the option to view a list of stories 2.2. The system presents a list of stories 2.3. User select a story 2.4. The system open the story selected 3. Update: <ol style="list-style-type: none"> 3.1. User chooses to edit the story 3.2. User changes data(s) 3.3. User submit the story edited 3.4. The system presents a confirmation message “Story edited successfully!” 4. Delete (Remove Story): <ol style="list-style-type: none"> 4.1. User chooses to delete the story 4.2. The system presents a message to confirm the action 4.3. User confirms the action (remove story) 4.4. The system presents a confirmation message “Story removed successfully!”
Alternative flows	None
Exceptions	<ol style="list-style-type: none"> 1. The user aborts the use case. 2. The system doesn't conclude the action. 3. The use case ends.
Posconditions	None
Notes and issues	A story belongs to a sprint.

Use Case ID	UC 4.2
Use Case Name	Task Manage
Description	This UC allows to perform the actions to: create, read (retrieve), delete and update tasks.
Actors	User
Preconditions	User can be logged User is a member allocated in the project/sprint
Normal flow	<ol style="list-style-type: none"> 1. Create: <ol style="list-style-type: none"> 1.1. User chooses create new task 1.2. User fills task's description and task's number 1.3. User defines assigned tasks 1.4. User submit the task 1.5. The system presents a confirmation message “Task created successfully!” 2. Retrieve: <ol style="list-style-type: none"> 2.1. User chooses the option to view a list of tasks 2.2. The system presents a list of tasks 2.3. User select a task 2.4. The system open the task selected 3. Update:

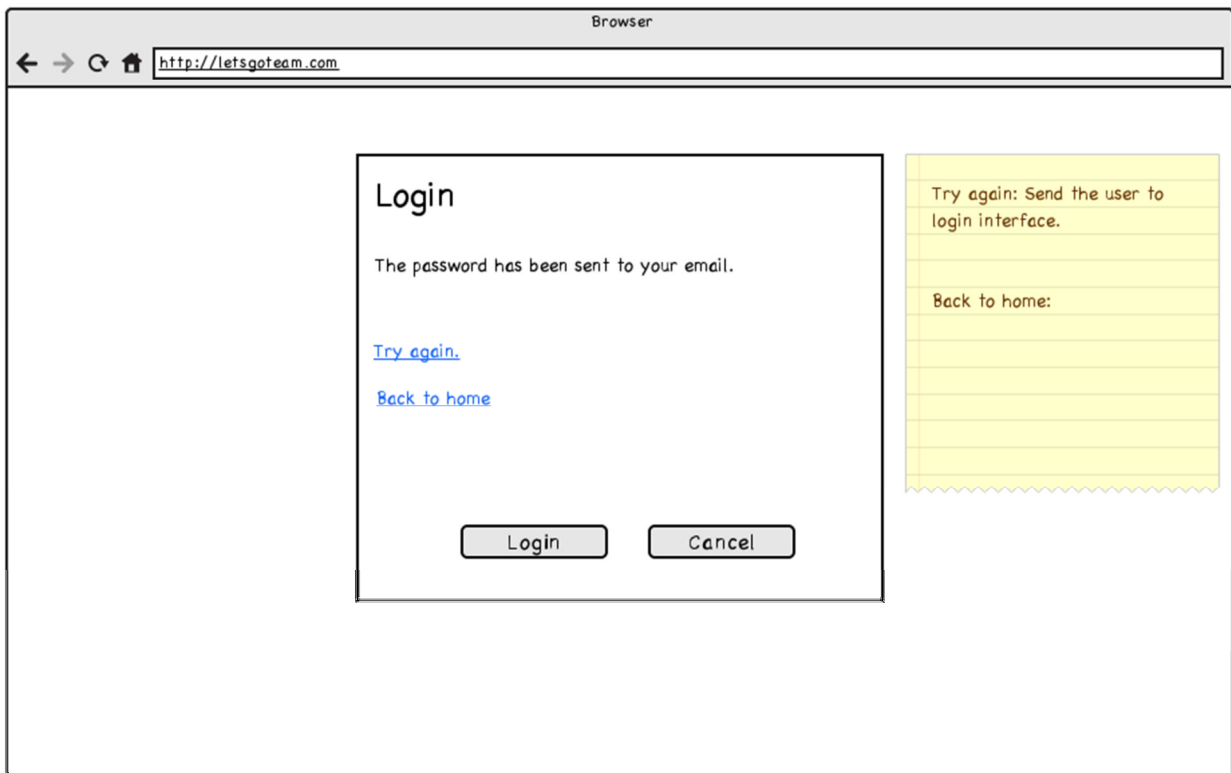
	<ul style="list-style-type: none"> 3.1. User chooses to edit the task 3.2. User changes data(s) 3.3. User submit the task edited 3.4. The system presents a confirmation message "Task edited successfully!" 4. Delete (Remove Task): <ul style="list-style-type: none"> 4.1. User chooses to delete the task 4.2. The system presents a message to confirm the action 4.3. User confirms the action (remove task) 4.4. The system shows a confirmation message "Task removed successfully!"
Alternative flows	None
Exceptions	<ul style="list-style-type: none"> 1. The user aborts the use case. 2. The system doesn't conclude the action. 3. The use case ends.
Posconditions	None
Notes and issues	<ul style="list-style-type: none"> 1. A task belongs to a story. 2. Anyone can take to assigned the task. 3. Tasks don't need to have a user assigned when the task is added to the user story

4 Supporting Information

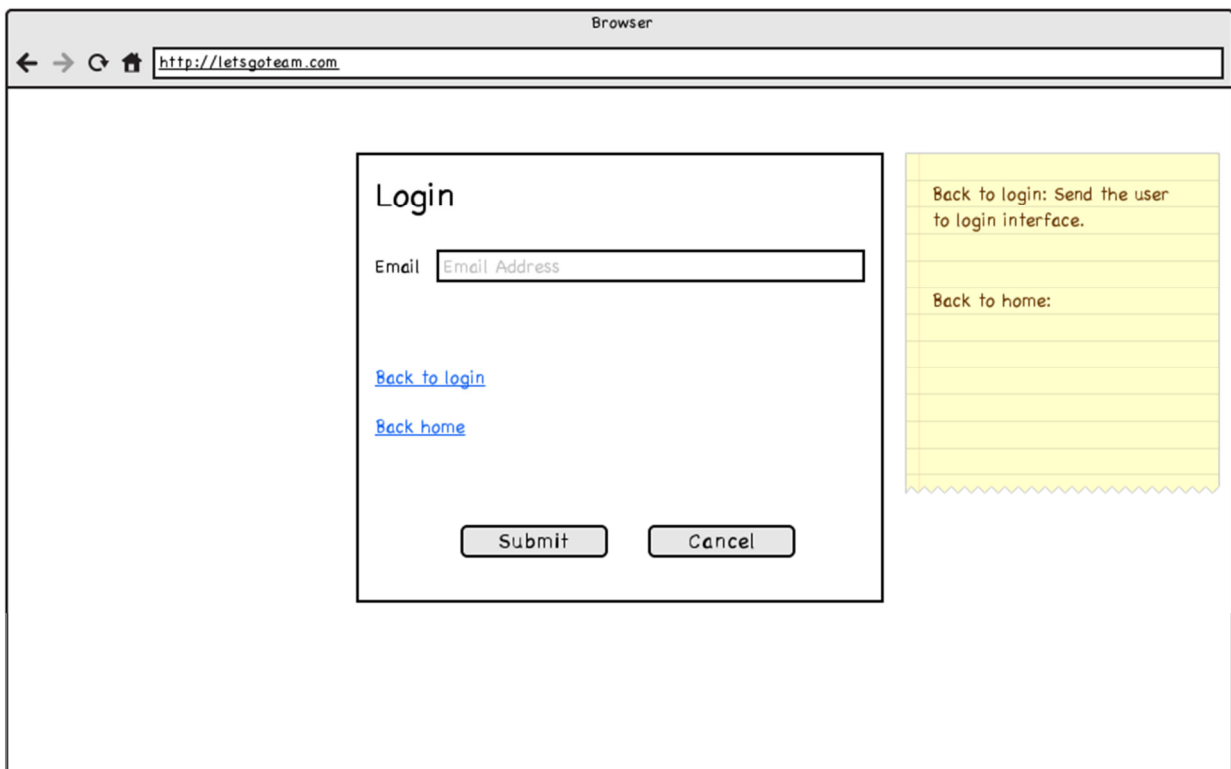
User-interface prototypes.

The image shows a user-interface prototype of a login page, displayed within a browser window. The browser's address bar shows the URL `http://letsgoteam.com`. The login form is titled "Login" and includes a placeholder text: "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla quam velit, vulputate eu pharetra nec, mattis ac neque." The form contains three input fields: "Email:" with a placeholder "Email Address", "Password:" with a placeholder "*****", and "Language:" with a dropdown menu showing "English". Below these fields are a checkbox labeled "Remember me" and a link "Forgot password?". A "Login" button is positioned at the bottom of the form. To the right of the form, there is a yellow sticky note with the following text: "Note 1: Language: the default value is english." and "Forgot password: will send an email with the password."

Created with Moqups (Free Version)
[Upgrade your account](#) to remove this banner



Created with Moqups (Free Version)
[Upgrade your account](#) to remove this banner



Created with Moqups (Free Version)
[Upgrade your account](#) to remove this banner

Browser

[←](#)
[→](#)
[↻](#)
[🏠](#)

Login

Incorrect username / password combination.

[Try again.](#)

[Forgot password?](#)

Not registered. [Click here.](#)

Try again: Send the user to login interface.

Not registered. Click here: Send the user to register interface.

Forgot password: will send an email with the password.

Created with Moqups (Free Version)
[Upgrade your account](#) to remove this banner

Browser

[←](#)
[→](#)
[↻](#)
[🏠](#)

[Home](#) | [Login](#) | [Register](#)

Register

First name:*

Last name:

Email *

Password:*

Confirm password:*

*Indicates required field

Note 1:

The email, registered, is unique.

Created with Moqups (Free Version)
[Upgrade your account](#) to remove this banner

16

Browser

← → ↻ 🏠 <http://letsgoteam.com>

[Home](#) | [Login](#) | [Register](#)

Project

Project name:*

Project description:

Start date: 📅

Completion date: 📅

Status: ▼

*Indicates required field

Completion date > Start date

Status: default "open", the other option is "close"

When submit save the informations: owner (user logged on) and date creation (system date)

Created with Moqups (Free Version)
[Upgrade your account](#) to remove this banner

Browser

← → ↻ 🏠 <http://letsgoteam.com>

[GraziBP \(online\)](#) | [Your account](#) | [Log out](#)

Projects

Project 1 | Owner: Graziela Pereira Start Date: 10/20/2014
 Status project: Open

[Properties](#) | [Members](#)

Project 2 | Owner: Letícia Machado Start Date: 10/01/2014
 Status project: Open

[Properties](#) | [Members](#)

Sprints

Sprint 1 | Start Date: 10/01/2014 End Date: 10/15/2014
 Summary: 2 User Stories | 5 Tasks

[Edit](#) | [View board](#)

Open the "Project" page to edit properties

Open the "Project Members" to edit members on project

Open the "Edit Sprint" to edit properties

Open the "Board" page to view the project board

Note 1:
 Add Project (button) - Open the "Project" page to edit properties

The project name is unique and

Note 2:
 Add Sprint (button) - Open the "Sprint" page to edit properties

The sprint name is unique (in the project context) and mandatory.

Browser

← → ↻ 🏠 <http://letsgoteam.com>

[Home](#) | [Login](#) | [Register](#)

Sprint

Sprint name:*

Start date:* 📅

Completion date:* 📅

Status:* ▼

*Indicates required field

Note 1:

Completion date > Start date

Status: default "Future", the other options are "closed", "Planned" and "In Progress".

Created with Moqups (Free Version)
[Upgrade your account](#) to remove this banner

Browser

← → ↻ 🏠 <http://letsgoteam.com>

[Home](#) | [Login](#) | [Register](#)

Project Share

Invite people:

Message:

*Indicates required field

Note 1:

Invite people: Use commas to separate multiple email addresses. Just will be able to invite registered emails.

Message: Write an specific message to invited people.

Created with Moqups (Free Version)
[Upgrade your account](#) to remove this banner

Browser

[←](#)
[→](#)
[↻](#)
[🏠](#)
<http://letsgoteam.com>

Project name | Sprint name

[Grazibp \(online\)](#) | [Your account](#) | [Log out](#)

Story	+	Not started	In progress	Completed	Blocked
US #1 - Login page - 8 points Edit Add task		Task #2 - develop page Edit	Task #1 - design page Edit		
		Task #3 - test page Edit			
US #2 - Forgot password - 13 points Edit Add task		Task #2 - send mail function Edit		Task #1 - design solution forgot password Edit	
		Task #3 - test function forgot password Edit			

Created with Moqups (Free Version)
[Upgrade your account](#) to remove this banner

Created with Moqups (Free Version)
[Upgrade your account](#) to remove this banner

Created with Moqups (Free Version)
[Upgrade your account](#) to remove this banner