

**SANTA CLARA UNIVERSITY**  
**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

Date: November 20, 2015

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY

**Alberto Diaz-Tostado**  
**Amy Nguyen Tran**

ENTITLED

**Project Management Application**

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREES OF

BACHELOR OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING  
BACHELOR OF SCIENCE IN WEB DESIGN AND ENGINEERING

---

Thesis Advisor

---

Department Chair

# **Project Management Application**

by

Alberto Diaz-Tostado  
Amy Nguyen Tran

Submitted in partial fulfillment of the requirements  
for the degrees of  
Bachelor of Science in Computer Science and Engineering  
Bachelor of Science in Web Design and Engineering  
School of Engineering  
Santa Clara University

Santa Clara, California  
November 20, 2015

# **Project Management Application**

Alberto Diaz-Tostado  
Amy Nguyen Tran

Department of Computer Science and Engineering  
Santa Clara University  
November 20, 2015

## **ABSTRACT**

Individuals currently live in a society that revolves around productivity. People from managers to students are bombarded with numerous tasks and daunting deadlines, making it extremely difficult to stay organized. In addition, people have multiple commitments in their lives, adding to the stress of keeping up with their already busy schedules. Currently, people are faced with a variety of organization methods. However, the two tools most people resort to, e-mail and notepad, become a burden to organize over time. In addition, current project management solutions are targeted towards large enterprises and not accessible for the general public. We propose a simple and user-friendly web-based solution that will be accessible to all. With the creation of a project management application, we will allow users to host a collaborative environment by providing them with a web based interface where they can easily communicate with their team members, find an organized list of their tasks and keep track of their schedules. In addition, we want individuals will use our project management application to plan personal projects, such as weddings or birthday parties, thus allowing them to enjoy their lives doing the things that make life memorable.

# Table of Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Current Solutions . . . . .	1
1.3	Proposed Solution . . . . .	2
1.4	Requirements . . . . .	2
1.4.1	Introduction . . . . .	2
1.4.2	Functional Requirements . . . . .	2
1.4.3	Non-functional Requirements . . . . .	3
1.4.4	Design Constraints . . . . .	4
<b>2</b>	<b>Design</b>	<b>5</b>
2.1	Conceptual Model . . . . .	5
2.1.1	State Diagram . . . . .	5
2.1.2	Activity Diagrams . . . . .	5
2.1.3	Mockups . . . . .	8
2.2	Use Cases . . . . .	8
2.3	Architectural Design . . . . .	12
2.4	Technologies Used . . . . .	12
2.5	Design Rationale . . . . .	13

# List of Figures

2.1	Entity Relationship Diagram . . . . .	5
2.2	Activity Diagram: Creating a project. . . . .	6
2.3	Activity Diagram: Creating a team. . . . .	6
2.4	Activity Diagram: Assigning tasks. . . . .	7
2.5	A mockup of our system concept. . . . .	8
2.6	Use Case Diagram . . . . .	8
2.7	Diagram describing the technology stack for our application. . . . .	12

# Chapter 1

## Introduction

### 1.1 Motivation

Individuals currently live in a society that revolves around productivity. People from managers to students are bombarded with numerous tasks and daunting deadlines, making it extremely difficult to stay organized. In addition, people have multiple commitments in their lives, adding to the stress of keeping up with their already busy schedules. Currently, individuals are faced with a variety of organization methods. However, the two tools most people resort to are e-mail and notepads. Without a doubt, people are often drawn to the simplicity of typing out a quick note or sending an e-mail. However, it becomes extremely difficult to keep track of all these notes and e-mails. Before you know it, lines of communication become broken and trains of thought are lost forever. Older conventions are becoming outdated as e-mail searches often yield innumerable results, and text files are never where they need to be. Team members need a way arrange their tasks and project managers require a solution to view the process of projects and ensure that every member is on track.

### 1.2 Current Solutions

Previous solutions have failed to provide an accessible and simple project management tool. One crucial issue is that most workflow management solutions, such as Asana and Basecamp, are often marketed and catered to large enterprises rather students and small groups. These solutions are exceedingly expensive, thus community organizations and students tend to avoid these applications. Another issue with current solutions is their steep learning curves that often require additional training. In addition, present solutions, such as Pivotal Tracker and Microsoft Outlook, are not straightforward and can make it difficult to setup a meeting or look up a contact. Furthermore, present project management applications are visually unappealing and not user-friendly. Most applications follow the conventional three panel layout, as seen in Outlook, pushing your tasks into some sidebar crevice where these tasks can easily be overlooked. These applications are not designed for students or small groups with minimal project management experience.

## **1.3 Proposed Solution**

We propose a simple, yet powerful, web-based solution that lives entirely within a browser. The goal is to have an easily accessible place for not only small groups of students or professionals to host a collaborative environment, but, we also want individuals to use our system for personal projects. Our accessibility will be achieved through a simple sign-on system without the hassle of painful payment systems and complex learning curves. Instead of struggling to learn the software, the user will be guided through a quick, interactive tutorial where he or she will establish goals, deadlines, and other project management tasks. The user will then be able to begin collaborating by creating groups and inviting other team members. Here, the user has the ability to add deadlines as well as assign internal tasks for him/herself or external tasks for others to view. The application will include a progress bar showing where the project is, where it was, and most importantly, where it needs to go. Current solutions provide users with a static page of checklists. In our application, when progress is made, the interface will reflect those changes. The timeline will expand into greater detail, time sensitive reports will become available, and completed tasks and files will be, literally, lost in time. We will allow the user to travel through their project timeline, revealing a snapshot of previous or future tasks. Lastly, to further promote organization, we plan to allow teams to store files within their project portals, eliminating time spent searching for important documents. Our product will be tested in live working environments, showing how teams use our solution to better organize their projects in an effort to help us help them achieve their organizational goals.

## **1.4 Requirements**

### **1.4.1 Introduction**

We defined a set of functional and nonfunctional requirements for our web application. Functional requirements specifies criteria that can be used to judge the operations of a system. In other words, functional requirements define what must be done by the system. Nonfunctional requirements describe the behavior and the limits of the application. To prioritize our requirements, we conducted an Analytic Hierarchy Process (AHP) chart seen in figure XXX. By using AHP, we were able to easily identify which requirements were critical, recommended and suggested.

### **1.4.2 Functional Requirements**

1. Critical The categorized functional requirements for our web application are summarized below. Our critical requirements defines the functionalities needed to have a basic working product. The critical requirements speak to the purpose of our project, which is providing an accessible tool for individuals

to create projects, collaborate with team members, and manage their tasks.

- (a) The system will allow users to create teams and assign members to those teams as well as add a project lead who will oversee the project.
- (b) The system will allow users to assign tasks to team members.
- (c) The system will allow users to define project goals and requirements.
- (d) The system will have a logical flow between tasks based on dependencies between tasks, calendars, and time progression.
- (e) The system will provide a way for users to view all tasks related to the project.
- (f) The system will allow teams to create milestones within their project timeline.

2. Recommended The recommended requirements describe additional functionality that we would want to have in our project. Our recommended requirements outline animations and personal integrations that would enhance our project- but are not needed for a basic working system.

- (a) The system will have a progress bar that will dynamically change as the project progresses.
- (b) The system will have a personal calendar integration, so members not only can see the availability of other members, but also plan their schedules accordingly.
- (c) The system will allow the user to create his/her own private set of tasks, schedules, etc.

3. Suggested The suggested requirements explain the functionality that we would like to have but are last priority in our project.

- (a) The system will provide users with a Gantt-style chart outlining milestones and current progress towards those milestones.
- (b) The system will allow the project lead to have managerial privileges such as approving completion of tasks
- (c) The system will allow users to traverse between milestones to show progress at those times.
- (d) The system will teach basic project management skills.

### **1.4.3 Non-functional Requirements**

In addition to functional requirements, we have outlined nonfunctional requirements, summarized below. Because we want to appeal to a variety of audiences, we designed our application to be effortless and user-friendly. We want our users to engage in a collaborative and interactive environment. Because we want our



application to be accessible, we designed our project to work on major web browsers as well as be compatible with desktops, laptops and tablets.

1. The system will be user-friendly.
2. The system will be aesthetically pleasing.
3. The system will be easy to use.
4. The system will be collaborative.
5. The system will work on major web browsers.
6. The system will be compatible with desktops, laptops, and tablets.

#### **1.4.4 Design Constraints**

With regards to design constraints, we have identified one constraint. Our application must be a web-based application because of senior design requirements for the web design and engineering major. However, we believe that a web-based system will allow our application to be easily accessible for our users.

1. The system must be a web-based system.

# Chapter 2

## Design

### 2.1 Conceptual Model

#### 2.1.1 State Diagram

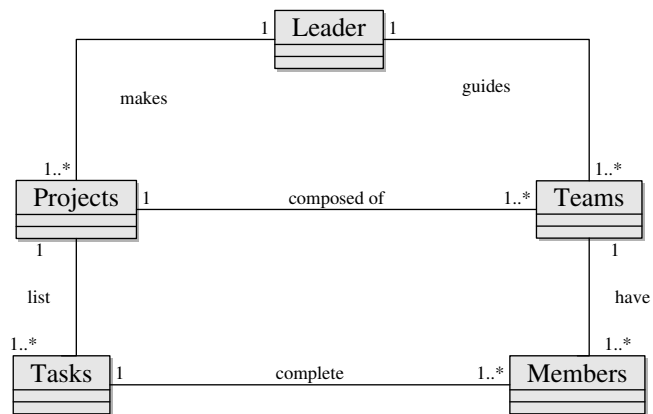


Figure 2.1: Entity Relationship Diagram

#### 2.1.2 Activity Diagrams

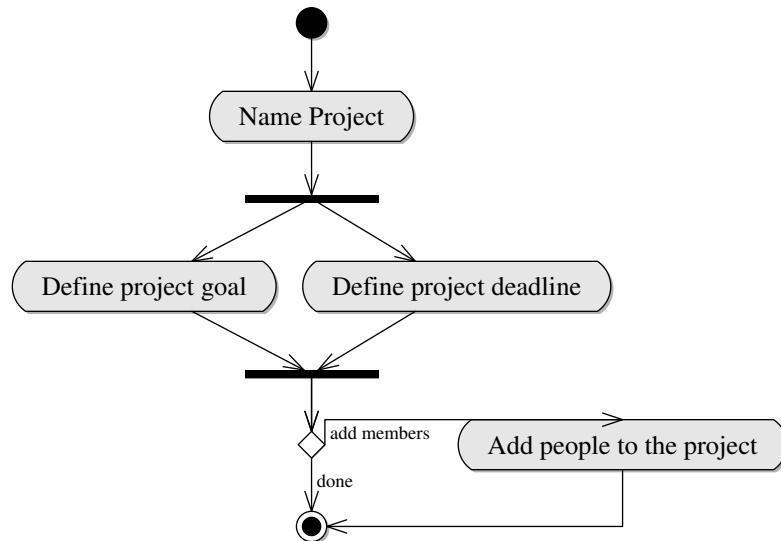


Figure 2.2: Activity Diagram: Creating a project.

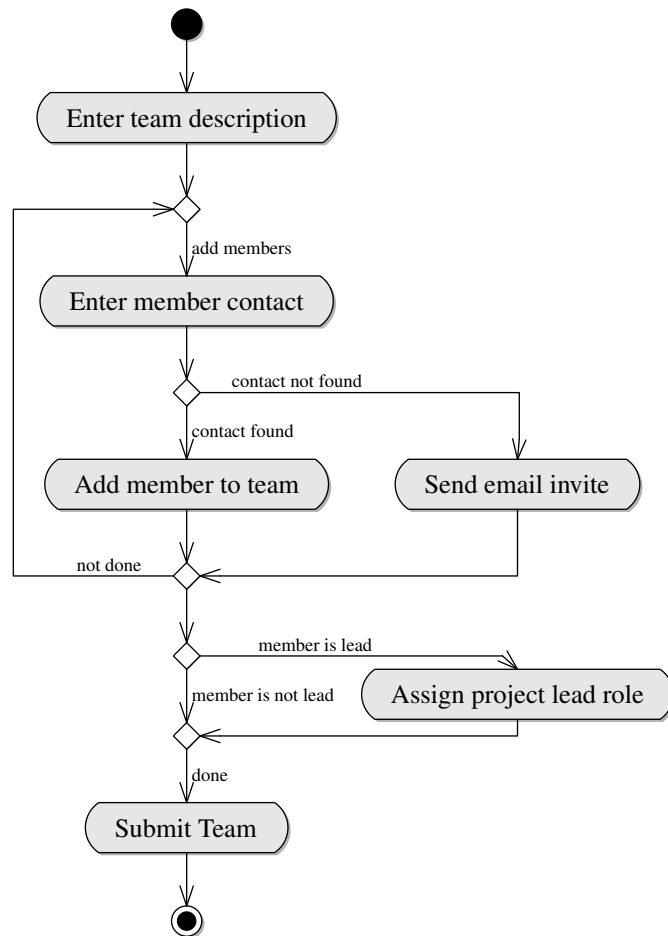


Figure 2.3: Activity Diagram: Creating a team.

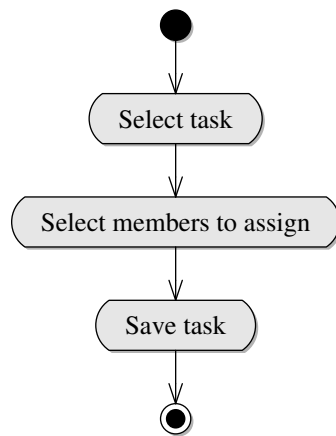


Figure 2.4: Activity Diagram: Assigning tasks.

## 2.1.3 Mockups



Figure 2.5: A mockup of our system concept.

## 2.2 Use Cases

A use case defines the steps required to accomplish a specific goal. The following use cases describe how the user interacts with the system to achieve these goals, including preconditions, postconditions, steps required, and common errors that might occur. The use case diagram, Figure XXX, helps to illustrate the users major actions in our application: creating a project, adding team members, and creating and assigning tasks.

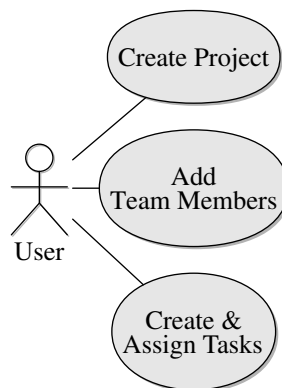


Figure 2.6: Use Case Diagram

---

<b>Use Case 1</b>	<b>Create Project</b>
-------------------	-----------------------

---

<i>Goal:</i>	Initialize project objectives, tasks, and deadlines
--------------	---

---

<i>Actor:</i>	User
---------------	------

---

<i>Preconditions:</i>	Account is set up and logged in
-----------------------	---------------------------------

---

<i>Postconditions:</i>	Project is initialized
------------------------	------------------------

---

*Steps:*

1. Name the project
  2. Define project goals and deadlines
  3. Add people to the project
  4. Save the project
- 

*Exceptions:*

A. Invalid login data:

1. System shows failure message
  2. User returns to step 1
-

---

<b>Use Case 2</b>	<b>Add team members</b>
-------------------	-------------------------

---

<i>Goal:</i>	Associate people with the project
--------------	-----------------------------------

---

<i>Actor:</i>	User
---------------	------

---

<i>Preconditions:</i>	Logged in, project created
-----------------------	----------------------------

---

<i>Postconditions:</i>	Project has people assigned
------------------------	-----------------------------

---

*Steps:*

1. Enter team description
  2. Enter member contact
  3. If found, add member
- 

*Exceptions:*

2.a Invalid login data:

1. System shows failure message
2. User returns to step 1

5.a Invalid subscriber data:

1. System shows failure message
  2. User returns to step 2 and corrects the errors
-

Use Case 1	Template test
<i>Goal:</i>	System-wide
<i>Actor:</i>	End-User
<i>Preconditions:</i>	
<i>Postconditions:</i>	
<i>Steps:</i> <ol style="list-style-type: none"> <li>1. The first action</li> <li>2. The second action</li> </ol>	
<i>Exceptions:</i> <ol style="list-style-type: none"> <li>2.a Invalid login data: <ol style="list-style-type: none"> <li>1. System shows failure message</li> <li>2. User returns to step 1</li> </ol> </li> <li>5.a Invalid subscriber data: <ol style="list-style-type: none"> <li>1. System shows failure message</li> <li>2. User returns to step 2 and corrects the errors</li> </ol> </li> </ol>	



## 2.3 Architectural Design

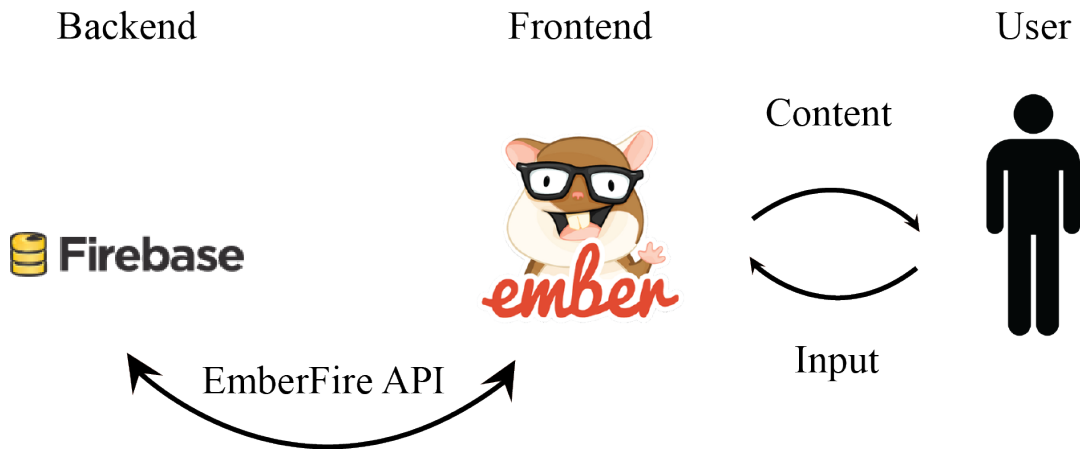


Figure 2.7: Diagram describing the technology stack for our application.

## 2.4 Technologies Used

In this section, we describe the technologies we chose to develop our application and the features they provide for us.

1. HTML5

The latest version of the standard web markup language

2. JS

A high-level, untyped, and interpreted programming language supported by all modern web browser

3. CSS3

A stylesheet language used to describe the presentation of HTML documents

4. Ember.js

A front-end javascript framework based on the model-view-controller (MVC) model and applies programming conventions to build scalable applications

5. Firebase

A backend-as-a-service that provides client-side APIs, and services such as databases, web hosting, and authentication.

## 2.5 Design Rationale

In this section, we describe why we chose these technologies and the advantages and disadvantages associated with each of those technologies.

### 1. HTML5/CSS3 /JS

This trio of web technologies has become the standard for web development

#### (a) Advantages

- i. Well documented and supported in the development community
- ii. Gives the developer control over the look and feel of the application
- iii. Compatible with nearly any web browser out there

#### (b) Disadvantages

- i. Without a framework or library it has little functionality
- ii. Requires a lot of time and effort to create a complete and robust application
- iii. Some browser interpret certain elements differently

### 2. Ember.js

We chose a framework because it expedites the development process. Ember.js boasts taking its developers 80% of the way through development with the use of best idioms and proper practices. It is used by companies such as Yahoo, Groupon, and Square.

#### (a) Advantages

- i. Faster development start time
- ii. Allows the developer to create reusable components through the use Handlebars templates
- iii. The built in router allows developers to create multi page applications by navigating the user through states rather than physical webpages, reducing the need to keep track of these different states since the page never physically reloads.

#### (b) Disadvantages

- i. The framework is based around conventions, limiting developers freedom for a majority of the development process
- ii. If the developers do stray from convention, they could find themselves having a difficult time reaching completion

### 3. Firebase

We chose this backend service due to the fact that it provides a client-side API for Ember.js called EmberFire.

#### (a) Advantages

- i. Using a client-side API eliminates the need for a backend language and allow us to focus on a single language for all of our development.
- ii. Firebase also provides hosting and database services for free!

#### (b) Disadvantages

- i. Support is outsourced, we do not have direct access to our server and are at the mercy of whatever functionality is provided by the API
- ii. A free firebase account does not provide private backups of our data, increasing the probability of data loss