Software Requirements Specification

for

Karma

Online goal managing and sharing

Version 1.0

Prepared by Akshay Kolge
Mathew John
Ramkrishna Maheta
Sanket Markan
Siddharth Bhatore

**TEAM 9, Software Engineering, Spring '18,
IIIT HYDERABAD**

02/01/18

Table of Contents

Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

Signature	Printed Name	Title	Date
	<Your Name>		

1. Introduction

It contains a description of the software system that we are going to develop.

1.1 Purpose

This document will help us to comprehend how we are going to do this project. It contains a description of the software system that we are going to develop. It will lay out the scope of our application so that we don't waste our time on unnecessary stuff. So, this document is written for the stakeholder of this project (Sangam Reddy) and our development team so that development becomes easier.

1.2 Scope

So, the end product of our project would be a web application.

Our product will help users manage life goals and share it with others. Users can also become mentors of other people's projects and charge money for their services. They can mentor a goal and give tips. It is a means to socialize and get motivated by learning through other people's experiences.

Karma will be a life changer. Whenever anyone will have a goal, he can start a project in Karma and record his progresses. Sharing it with others (even anonymously) will enable him/her to gain feedback from others and motivate him throughout. Browsing through experiences of others can be useful for a person looking to achieve a certain goal. People who specialize in certain area(s) can help others with their projects and provide tips and feedback. So the ultimate goal of this application is that it becomes a useful product in the lives of people and help them achieve their goals efficiently.

The scope of Karma is for the time being limited to being a web application.

1.3 Definitions, Acronyms, and Abbreviations

1. **Mentor** : A person who will give tips to people following a certain goal.

2. **Personal space** : Secure space which only the user can see. Goals under personal space can be seen only by that user.

3. **Journal** : A diary where date wise progresses or activities is added.

4. **Project** : When a user has a certain goal, he can start a project in his personal space.

5. **Goal** : A user defined objective under which all the projects will be listed by different users.

6. **Progress** : For a certain day(date), things done by the user to get closer to his goal.

1.4 References

1. https://www.geeksforgeeks.org
2. https://www.linkagoal.com
3. https://www.creately.com

2. General Description

This section of the document describes the general factors that affect our product and its requirements. This section does not state specific requirements. Instead, it provides a background for those requirements, which are defined in detail in Section 3 of the SRS, and makes them easier to understand.

2.1 Product Perspective

The product is independent for most part but if social API for login is implemented, the profile of user can be made using it.

2.2 Product Functions

These are the functions that our software will perform:

1. Allow users to create their profile.
2. Option to make project public under some goal.
3. Option to follow a goal/project/user.
4. Option to comment on goal/project/progress.
5. Search user/goal.
6. Merge goal.
7. Mentor a project (paid)/goal.
8. Personal messaging.
9. Rating for users.
10. Feed.
11. Notifications.
12. Adding milestones.
13. Upvote a project/progress.

2.3 User Characteristics

The end user will be of two types:

1. Standard user making goals/projects : He should have basic English knowledge and should know how to use simple web applications. A basic knowledge of how to surf online is expected.

2. Mentor : Same as Standard user but he should also have expertise in the area he is mentoring.

2.4 General Constraints

The application is bounded by the following constraints:

1. User must have a valid registered account to access the app.
2. UI is only in English.
3. User should have working internet connection.
4. User's Personal space is visible only to that user.

2.5 Assumptions and Dependencies

This subsection of the SRS list each of the factors that affect the requirements stated in the SRS.

1. Browser should support technology used in the product.
2. User is assumed to have basic knowledge of English and should have a basic skill to surf online.

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

User Interface should be clean and simple. Following things will be take care of:

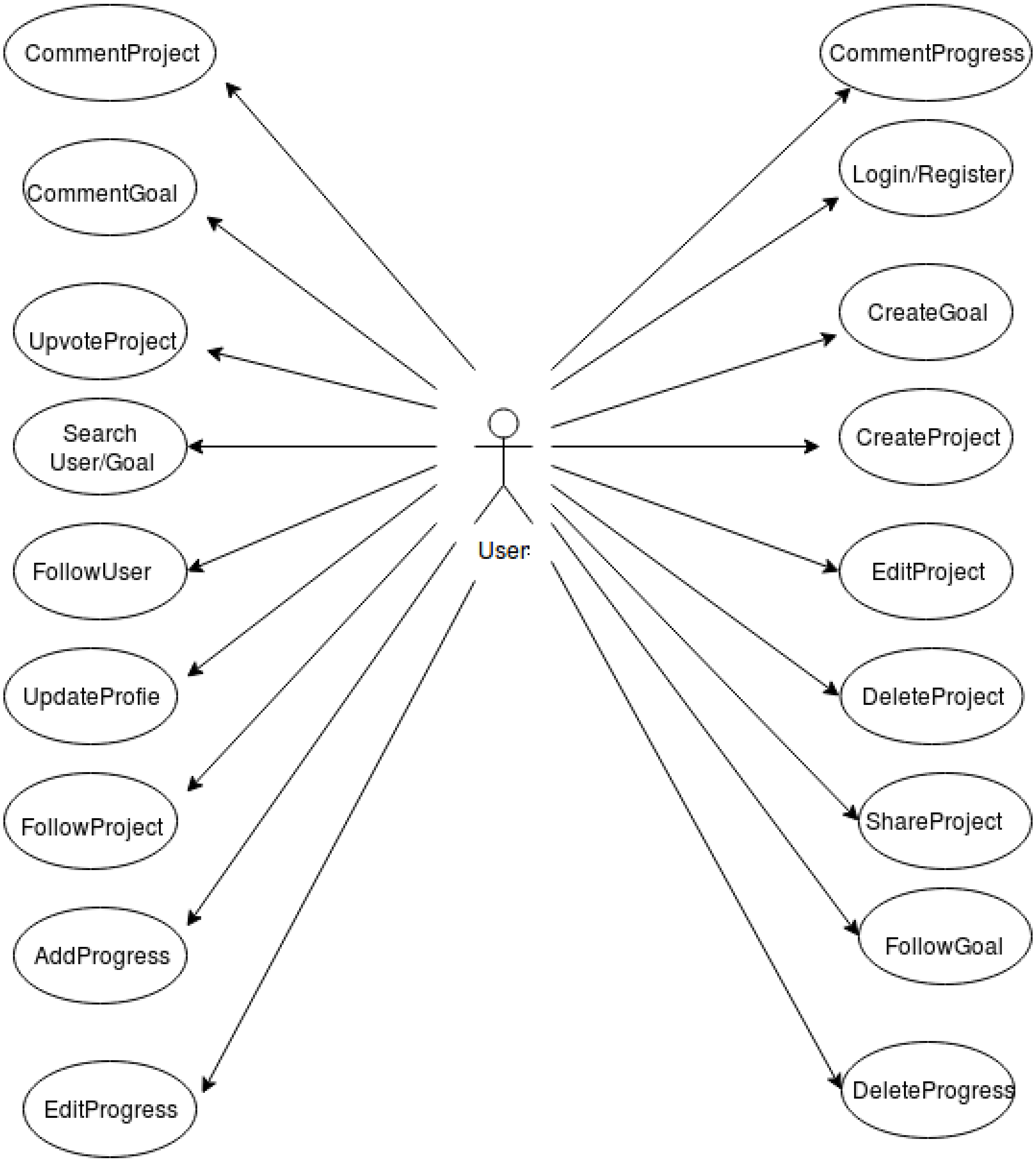
1. On opening the app, user visits login/register page.
2. After logging in, user sees his/her list of all projects.
3. Search bar will be visible.
4. User can see link to the list of his/her followers/following, following goals and projects.
5. User can see link to his/her own profile where he/she can edit his/her basic information.
6. Every project will have a page where all progress is displayed in a easy-to-navigate fashion.

3.1.2 API Interfaces

3.2 Functional Requirements

This section describes specific features of the software project. Requirements must be specified in the use-case format and listed in the Use Cases Section.

3.2.1 Use Cases



3.2.1.1 Use Case #1: Register for application

Primary Actor: User

PreCondition: Nil

Main Scenario:

1. User starts the application and prompted for login or register option.
2. New User click on registration link and fills the registration form.
3. System provide the validation and authentication

4. After successful registration, user gets redirected to login page.

Alternate Scenario:

4a. User tries to login without registration and fails

4b. Prompt the user with appropriate error message.

3.2.1.2 Use Case #2: Create a new goal

Primary Actor: User

PreCondition: Successfully Logged In

Main Scenario:

1. User visits the create goal page.
2. User enters the basic information about the goal such as Title, Description, etc.
3. User then submit and goal is then authenticated
4. New goal is added

Alternate Scenario:

3a. Information provided is invalid, error message will be thrown.

3b. Goal is already present, error will be thrown.

3.2.1.3 Use Case #3: Create a new project

Primary Actor: User

PreCondition: Successfully Logged In

Main Scenario:

1.User visits the create project page.

2.User selects the goal of the project

3.User enters name,description and other details of the project.

4.User selects whether to make it public or keep it private.

5.Project is created and hosted on the profile/home page.

Alternate Scenario:

2(a).If the goal is not present then allow user to create the goal.

3.2.1.4 Use Case #4: Edit project

Primary Actor: User

PreCondition: Successfully Logged In

Main Scenario:

1.User visits the project page.

2.User selects the edit option.

3.User modifies the required fields and clicks on submit.

4.The changes made are reflected.

Alternate Scenario:

3{a} if any of the field is invalid, error message will be thrown.

3.2.1.5 Use Case #5: Delete project

Primary Actor: User

PreCondition: The Project exists.
Main Scenario:

1.User visits the project page.
2.User selects the delete option.
3.Project is deleted from profile as well as database along with all progress and comments.

3.2.1.6 Use Case #6: Add progress

Primary Actor: User
PreCondition: The Project exists
Main Scenario:

1.User visits the project page.
2.User selects the add progress option.
3.User adds details of the progress and selects submit option.
4.The progress is added under project and the changes are reflected.

3.2.1.7 Use Case #7: Edit progress

Primary Actor: User
PreCondition: The Project exists
Main Scenario:

1.User visits the project page and selects the progress.
2.User selects the edit progress option.
3.User modifies the details and selects submit option.
4.The changes made are reflected.

3.2.1.8 Use Case #8: Delete progress

Primary Actor: User
PreCondition: The Project exists
Main Scenario:

1.User visits the project page and selects the progress.
2.User selects the delete progress option.
4.The progress is deleted from project as well as database and all its comments.

3.2.1.9 Use Case #9: Share a Project

Primary Actor: User
PreCondition: The Project exists
Main Scenario:

1.User visits the project page.
2.User selects the option to share project publicly.
4.The confidentiality of project is changed to public.

3.2.1.10 Use Case #10: Add a Comment

Primary Actor: User
PreCondition: The Project exists
Main Scenario:

1.User visits the project/Goal/Progress page.
2.User selects the option to comment.
3.User adds the comment and selects submit option.
4.The comment is added to the Project/Goal/Progress.

3.2.1.11 Use Case #11: Delete a Comment

Primary Actor: User
PreCondition: The Comment exists
Main Scenario:

1.User visits the Project/Goal/Progress page.
2.User selects the comment to be deleted.
3.User selects delete option.
4.The comment is deleted from the Project/Goal/Progress.

3.2.1.12 Use Case #12: Search User/Project

Primary Actor: User
PreCondition: The user is logged in.
Main Scenario:

1.User clicks on the search bar.
2.User enters the username or query to search.
3.List of the results is displayed.

Alternative Scenario:

3(a).The query returns no result, appropriate message is displayed.

3.2.1.13 Use Case #13: Edit Profile

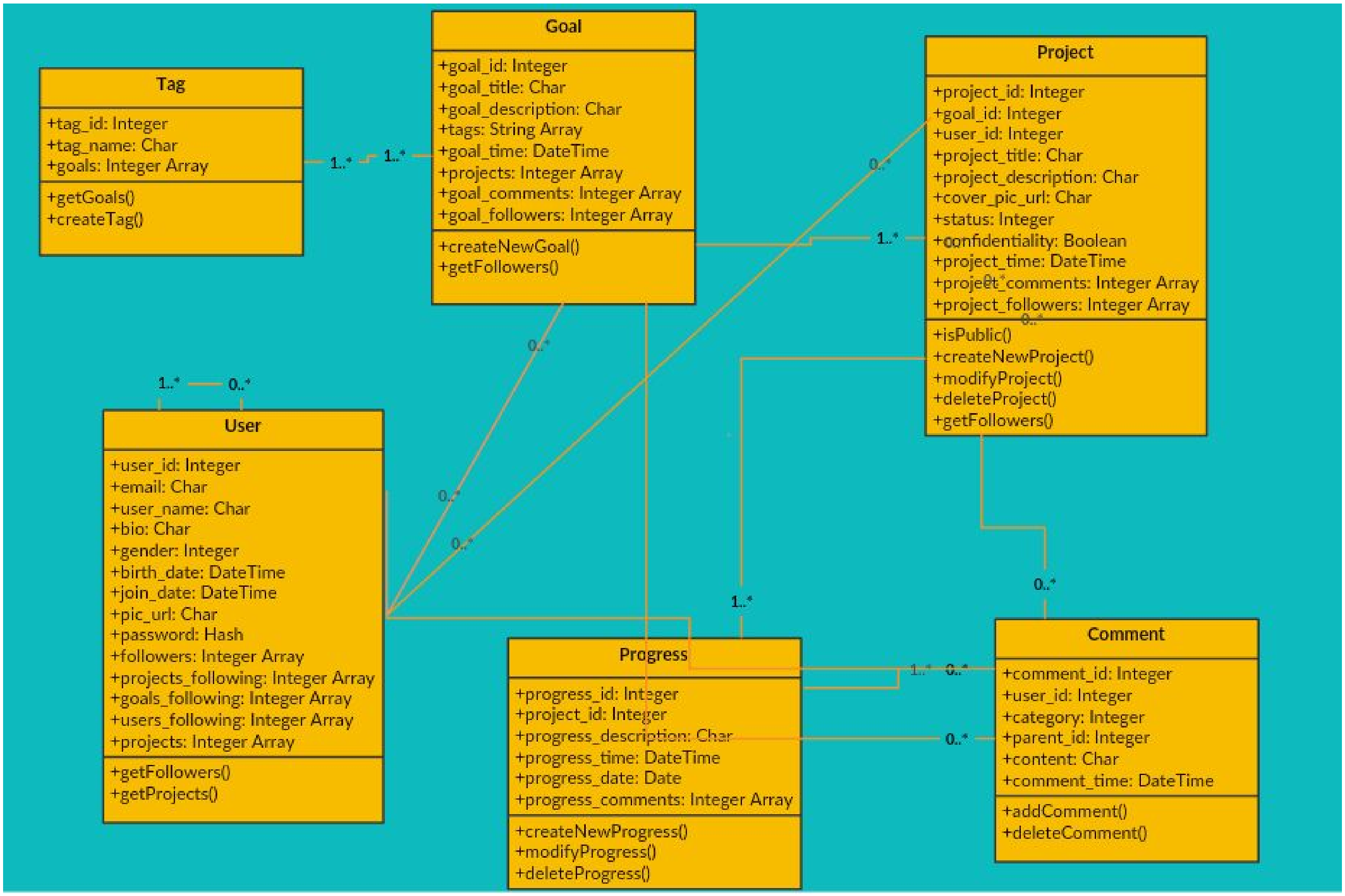
Primary Actor: User
PreCondition: The user is logged in.
Main Scenario:

1.User visits the profile page.
2.User selects the edit profile option.
3.User makes changes to the required fields and selects submit option.
4.The changes made are reflected.

Alternative Scenario:

3(a).If any field is invalid, appropriate message is displayed.

3.2.2 Classes / Objects



Class Diagram for Karma

3.3 Non-Functional Requirements

3.3.1 Performance

Database design should be such that queries and changes to data should not take more than 1 second.

3.3.2 Reliability

Data should not be lost.

3.3.3 Availability

Site should not be down for more than 1 minute per day.

3.3.4 Security

Personal space should be secure. Credentials would not be compromised.

3.3.5 Maintainability

Application will be very scalable so that it can be enhanced and used beyond the scope of this course too.

3.3.6 Portability

Application will be accessible through any gadget with internet connection and a browser.

3.4 Design Constraints

Server-side Requirements:
Operating System: Windows XP/7/10 or Linux/Solaris
Programming Language: -
Web Applications: -
Processor: Pentium IV or above
Hard Disk: 40GB,
RAM: 1GB
User Interface: HTML, CSS Client-side
Database: -

Client Side Requirements:
Operating System: Any Operating System
Browser: Any Browser

4. Change Management Process

Agile Development model will be used to develop the application. Hence, the changes will be done after we achieve what is required in the current SRS document.