TODO List
Team 4
Junzhe Cai, Zixuan Hao, Guodong Xue, Haoshu Zhang

Software Requirements Specification

Document

Version: (1.0) **Date: (02/04/2017)**

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this report is to specify the software requirement and functions of software.

1.2 Scope

“TODO List” is a web-based software that lets its users set up some tasks for themselves to complete. The software requires users to log in by email so that the users can get notification from the email and users can create or be invited in a group. The basic function of this software is to remind the user of what things they need to do and what things they already did(This function is special for OCD). But it will not create the tasks themselves. If some users have the same task in one group, they can have a competition with each other. The person who first finish the task will let the system send email to other group members. This will make people feel more active doing the task.

1.3 Definitions, Acronyms, and Abbreviations.

OCD – obsessive-compulsive disorder
UI – User Interface
IE – Internet Explorer. A web browser developed by Microsoft
Chrome – A web browser developed by Google
API – Application Program Interface
HTML – Hypertext Markup Language
PC – Personal Computer
HTTP – The Hypertext Transfer Protocol
SMTP – Simple Mail Transfer Protocol

1.4 Reference

https://developers.google.com/gmail/api/guides/sending

1.5 Overview

The rest of the SRS contains the overall description and specific requirements. In part 2, it explains the overall function of the software and the environment running the software. Part 3 explains the function of the software in detail.

2. The Overall Description

First, the system will have the login and register function for users to login and register in the application. In the first phase, the main property of application is to let the

user record their daily tasks to make sure that they have done the tasks. For example, if the users always forget to lock the doors, they can write down the tasks. Next, when they have done the tasks, they can record the specific time it happened. Meanwhile, users can reset the task at an unfinished condition. In the second phase, the system have the function of the group work. When some members make a group, they can share tasks and files together. Besides, they can divide the entire task into several parts to allow members to do their own part individually. When someone has finished a part, the system would send emails to other members to notice them.

2.1 Product Perspective

2.1.1 System Interfaces

The system would run in the web browser on any devices which could connected with Internet

2.1.2 Interfaces

The system would have a UI to let users write down their task and share tasks with group members.

2.1.3 Hardware Interfaces

The system can be used on desktop, laptop or smart phone, which could conneted with Internet.

2.1.4 Software Interfaces

The application need to run in any browser supported HTML5.

2.1.5 Communication Interfaces

The system would use the HTTP for communication over the internet and for the email communication will be through SMTP protocol.

2.1.6 Memory Constraints

The system need to be used with the web browsers. Therefore, it doesn't have memory requirement of the devices.

2.1.7 Operations

The system just has two different kinds of operations, one is to let the users write down their issues, and the other operation is to let the users use the group work function.

2.2 Product Function

- The system has the function to let the users write down their daily tasks so that they can check the tasks after they have finished.
- The system has the function to let several members study or work together and share the material, and the system can notice other users.

2.3 User Characteristics

The users can be any ages who can use the browsers to access the Internet. In addition, the system can be used for those people who are absent-minded or compulsive.

2.4 Constraints

The system would be safe and stable for users, so users only need to remember their own user name and passwords.

2.5 Assumption and Dependencies

The system need to run on all kinds of browsers suppoted by HTML5.

2.6 Apportioning of Requirement

The system need to be divided into two parts, the to do list part and the group share part.

3. Specific Requirements

3.1 External Interfaces

The only link to an external system is the link to the Gmail API. Our to-do list system shall send email to user or user's group member by Gmail API.

3.2 Functions

3.2.1 Add tasks

3.2.1.1 Users shall be able to add what they need to do into the list

3.2.1.2 User shall be able to set the deadline for each tasks

3.2.2 Complete a task

3.2.2.1 Users shall be able to click the check box to complete a task

3.2.2.2 Associated functional requirement

3.2.2.2.1 User shall be able to complete the tasks they made

3.2.2.2.2 User shall be able to mark the task incompleted when they accidently click the finish button

3.2.3 Delete tasks

3.2.3.1 User shall be able to delete the tasks they do not want or the tasks added by mistake

3.2.3.2 Associated functional requirement

3.2.3.2.1 User shall be able to delete the tasks they do not want

3.2.4 User registration

3.2.4.1 User shall have the option to add user to the server.

3.3 Performance Requirements

3.3.1 The Static Numerical Requirements

3.3.1.1 Our project shall run on any terminal which has a browser supported with HTML, and we would try to optimize the performance on PC at the first iteration

3.3.1.2 The number of simultaneous user on one device is one.

3.3.1.3 The amount of information should be easily handled by any device nowadays

3.3.2 The Dynamic Numerical Requirements

3.3.2.1 The system shall send you an email to remind you within 1 minute delay before the time you setted to let system reimind you

3.3.2.2 The system shall add the task on a group member's list when added by a group member within 1 minute

3.4 Logical Database Requirements

The project shall use a database on the cse sever for testing. The database will be used to store user registration information and user's task information. This data will need to be frequently when users operate their task, especially when user want to share their task with group members.

3.4.1 Data Entity

3.4.1.1 User Data Entity

Data Item	Type	Description	Comment
Name	Text	Name of user	
User ID	Integer	ID number	Used as a key
Email Address	Text	Internet address	It cannot be empty

3.4.1.2 Task Data Entity

Data Item	Type	Description	Comment
Task Name	Text	Name of task	
Task ID	Integer	Task ID number	Used as a key
Task Type	Integer	The type of task	
Deadline	Date	The deadline for the task	
Reminder	Date	The day user setting to let system send email for reminding	

3.5 Design Constraints

The web-based application should be able to run well on any device with an Internet connection and the support of browser with HTML5.

3.6 Software System Attributes

3.6.1 Reliability

First of all, the reliability is very important for a program. For example, when user sit a time to remind of one of tasks, the system should send an email to the user within 1 minute delay

3.6.2 Availability

The application shall be able to run at any time theoretically. It might be a problem if user's Internet speed is too low to load our application on the browser. Also, if the CSE sever is disconnected, it would cause limits of some functionality

3.6.3 Security

Security is important as well. For example, all kinds of users' information need to be protected, especially email address. Only administrators could access to the database

3.6.3 Maintainability

The maintainability is essential for a successful project. If we want to add some new function for our project, it will not influence existent functions because system is develop in component-based

3.6.4 Portability

As a web-based application, portability is most important because it doesn't need any memory space on a hard disk.