DocNo: 001.C.1.1

Grape

Requirement Analysis Report

Version 2.0

By:

Group Undefined

2015-04

Group Member:

Hunter Lin

Birdy

Listen

Morning

Syachi

Document Language:

English

Revision History

Date	Version	Description	Author
2015.4.8	1.0	Initialization of the report	Hunter Lin
2015.4.9	1.1	Finish the last three part	Hunter Lin
2015.4.10	1.2	Congregate the part from morning	Morning
2015.4.10	1.3	Congregate the part from Listen	Listen
2015.4.10	1.4	Congregate the part from Birdy	Birdy
Final Date	2.0		

Key Words

Requirement
Discussion, Interaction, Feedback
Functional
Non-functional
User-Interface

Abstract

This document is to define the requirements for our system, in different point of views. More specifically, the document will mainly introduce functional requirements, non-functional requirements, both hardware and software requirements for installing our system, and finally, a rough user interface. In the end, we will build several models for the objects interacting with our system, as well as the use cases included in our system. Their relationships will also be clarified as much as possible.

Contents

1 Introduction

1.1 Purpose

This is a requirement specification document. In this document, we will define most of the system requirement, so that all the develop team member can have a clear picture of the whole system. We define not only the functional requirement, but the non-functional requirement as well. This is the main document of this define phase, in this phase, we also have glossary document, use case specification document and we offer a prototype.

1.2 Definition

Grape: Our system that can manage a vote easily
Undefined: The name of our group
Functional: some requirements that need to be realized
Non-functional: some requirements that cannot be realized but is indispensable to our system

1.3 Reference

Microsoft BMS System White Book
Bob Hughes and Mike Cotterell 《Software Project Management》

2 System Overview

The traditional way of voting is offline,which is time and labor consuming. And there are devices designed particularly for casting a vote. But it will cost plenty of money. And although there are some websites that provide some similar functions like voting but they are all kind of outdated. So here comes our system. Our system can manage the vote easily by creating a group and starting a quick vote within a few clicks. Our system can share files without any difficulty. Our system can also show the details of every vote in the past and even give some analysis.

3 Detailed System Requirements

3.1 Integration Requirements

3.1.1 Integration Requirements

We need python 2.7 with Flask and MySQL to develop and run this system.Flask is a microframework for Python based on Werkzeug, Jinja 2 .And MySQL is a famous and open-sourced database so we can use it for free.

3.2 Architecture Requirements

3.2.1 Computing platform requirements

Server Minimum :
OS：Windows8/Windows 7/Vista/XP
CPU：Intel/AMD two cores 2.4 GHZ
Memory：2GB

Client Minimum :
OS：Windows8/Windows 7/Vista/XP
CPU：Intel/AMD two cores 2.4 GHZ
Memory：1GB

3.2.2 Software requirements

Server:
- Python 2.7
  - -Flask
  - -MySQLdb
- MySQL.

Client:
- Any platform that can visit webpage.
- Android 3.3+
- IOS 7+.

3.1 Functional Requirements

3.1.1 Functional/Behavioral

Click can provide users a powerful tools to get the intention of most of members and promote the communication and the flux of the information among any group formed in any way.

First of all, it can help team members vote on the options selected by organizer about specified questions in the way that everyone's choice is hidden from the other ones and the results can be calculated and displayed rather fast. Thus, everyone in the voting process can't be easily disturbed by othersand the results don't need to be counted manually which make the vote action more efficient.

With the application, everyone can put forward questions to be discussed among the group regardless of the time and the place he staying. Everyone else in the group can have

access to answering the problems. Moreover, the discussion can be more efficient than discuss in the form of one to one. For example, the teacher as the leader can reply the questions of the students and all the students can see the discussion in the group so that they wouldn't miss any knowledge during the discussion.

The Grape allows users to share files in the group. The files can be useful resources found by someone in the group, can be used to depict some problems and can be used by the leader to handout materials. However, for the safety of the system, the files to be shared in the group mist be reviewed by the leader at first. Then any user can download the files shared conveniently and store them in the local file system. The leader has the authority to delete the files in shared status.

When the leader has some information need to be broadcast in the group, he should be able to change the content of the bulletin in the group with the application. The bulletin can be seen by the other users in the group rather easily and well known by them. This project can be a powerful tool for any group to work together and make decisions.

3.1.2 Non-functional

The process that the system updates the information like the files shared and the vote result in the group should be quick and safe. No data lost is permitted.

Special security measurements are needed to make sure only valid users can log into the system to have access to all the functions.

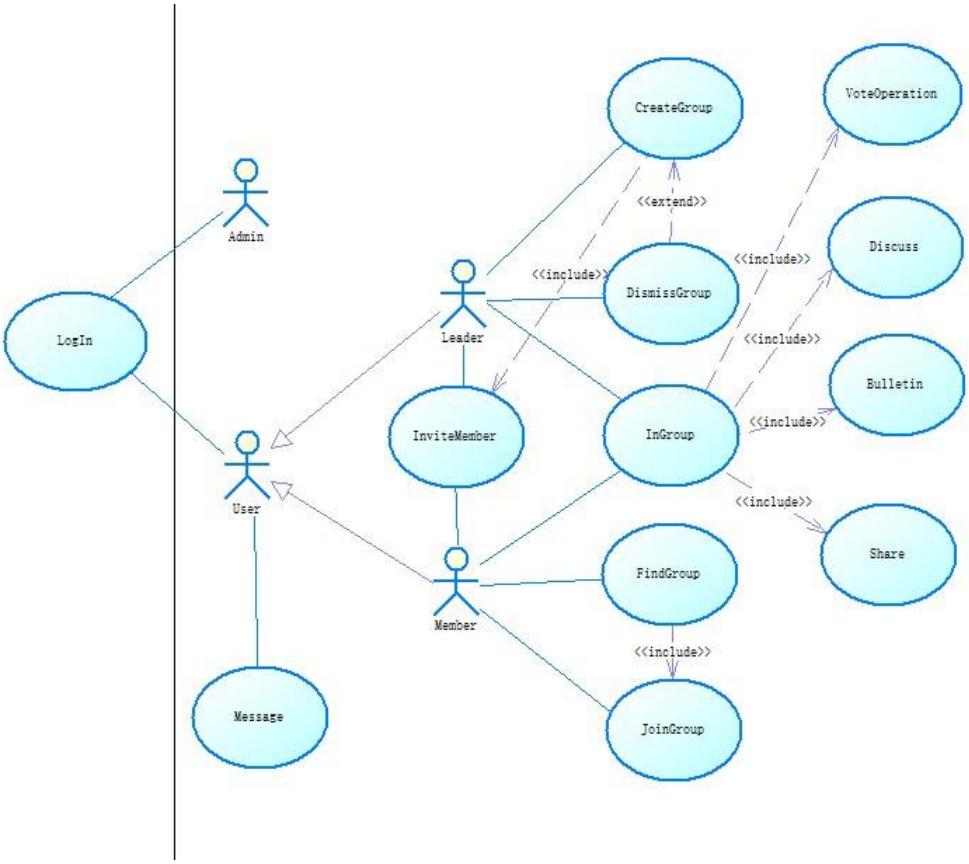
4.1 Functionality (behavioral)

4.1.1 Use Cases

This system is divided into twelve use cases: login, message, create group, dismiss group, in group, find group, join group, invite member, share, bulletin, discuss, vote operation.
Each use case is specified in the use case specification document. Please refer to each use case specification document to get detail information to know more about each use case.

4.1.2 Functional Analysis Model

We use PowerDesigner to create the use case model. Here is the view of use case model.



5. Build/Buy/Reuse Rationale

5.1. Reuse Criteria

In order to avoid coding/coping same kinds of functionality in different projects, we carefully analyze the part that can be reused besides our project. This is a very important process since it'll encourage us to think further about our project and completion on functionality. Thus we'll try to make the functionality of great extensibility. Although there is more time consumed in it, the benefits will be long-lasting. Actually, that indicates a highly valued idea in software engineering: the concentration and ability to build tools for ourselves.

In our project, we'll focus on some of the functionality that can be reused. We'll illustrate them in a matrix displayed as below. The column "Consideration" includes the dimension we'll inspected in reusing certain functionality. (or "use case" as we name it).

Use-case to reuse	Considerations	Comments/Observations

Account Operations	Product Technical Considerations	We use MySQL to realize this module by creating several tables to store information of each user and their stages. It's not difficult to implement.
	Project Management Considerations	An important component In most systems. Because a system will usually need to know the identity of a user.
	Deployment, Operations, and Maintenance Considerations	We choose python + MySQL to as a back-end technique. It can be easily deployed at hand.
	Business Considerations	Every company buy our software will think it a basic function and think it indispensable
	Financial Considerations	It hardly cost anything!
Any other project which concerns about Database query operation can reuse this module		
Sharing-File Operation	Product Technical Considerations	We use MySQL to realize this module by creating several tables to store file information. Meanwhile, the real files are uploaded to a certain folder deployed in our main server.
	Project Management Considerations	An important component In many systems. Many applications need a platform to upload and share their own resources. Thus creating a more open environment.
	Deployment, Operations, and Maintenance Considerations	We choose python + MySQL to as a back-end technique. It can be easily deployed at hand. The only thing that needs careful attention is that, file size must not be too big to fill the space of our main server, which is what we'll devote more efforts onto lately.
	Business Considerations	Of great useful.
	Financial Considerations	It may need a main server with relatively large memory storage in case numbers of users are using this functionality simultaneously.
Any other project which concerns about file sharing or even group discussing can refer to this component		

5.2. Rationale

We only build, buy and reuse those components which are commonly used and of high value. These functionality share a common characteristic that they can be easily transmitted and extended to any other systems, thus making wonderful tools for the later-comers.

6. Requirements Relationships

When carefully examine the use cases we've created. There are several use cases having certain relationships (or called dependencies) between each other, which deserve more detailed examination onto these use cases. Clarifying on these relationships will facilitate us in clearer observation of our project models, and quicker development process.

Here let's look at a list consisting some of the relationships we've mentioned. We use a matrix to describe these relationships.

Requirement A	Requirement B	Relationship/Observation
Sign up	Sign in	We can easily recognize that we can only sign in as a user if we've successfully created an account. So There is a dependency relationship between "Sign up" and "Sign in". The same relationship holds true for creation and further operation between "Group", "Vote", "File", "Bulletin", "Discussion". The further operation mentioned here may include "Modify", "Check", "Deletion".
Raise an instant vote	Raise a long-lasting vote	Different from above. The relationship between raise an instant vote and a long-lasting vote is not a dependency relation. In fact, there is a option for raising an instant one or a long-lasting one. In other words, the two use cases are branches of raising a vote. Their relationship can be expressed as "Either…or…".
Leader shares files	Member shares files	In our system, Different user are granted with different level of authority when it comes to sharing files in a group. That is to say, a group leader can share files without any pre-inspection, while a member may not share files as freely as a leader. When a member attempts to share files, the files can only be successfully uploaded in approval of the group leader. This scenario is considered in case of malevolent upload by the user. So when it comes to sharing a

		file, different actors behave differently. The relationship may be expressed as "If… else…"

7. Conclusion

In this document, we present a relatively detailed analysis on the requirements for building our system. Different factors are considered: functional and non-functional requirements, hardware and software requirements, user interface requirements. Finally, we discussed some of use cases and their relationships in order to present a clearer perspective.