DocNo: 001.K.1.1

Grape
Project Summary
Version 2.0

By:

Group Undefined

2015-06

Group Member:

Hunter Lin
Birdy
Listen
Morning
Syachi

Document Language:

English

Revision History

Date	Version	Description	Author
2015.6.26	1.0	Initialization of the report	Hunter Lin
Final Date	2.0	Integrating all of the works	Hunter Lin

Content

Key Word

Grape
Project management
Summary

Abstract

This document is to summary our Grape developing phase, to get a clear a picture of our artifacts and experience. We did learnt a lot via this kind of developing experience. We have a meaningful summer vocation.

1. Introduction

1.1. Purpose

The purpose of this document is to look back on the whole process of project develop process, and to evaluate whether the project meet customer's requirements and whether the develop process meet the software engineering requirements, and also to provide success or failure experience.

All the participators of the project are expected to read this document.

1.2. Background

Project Name:	Grape System
Software System Name:	Grape System
Task provider:	IEEE honor class (2013) of Shanghai Jiaotong University
Develop team:	Undefined team
User:	IEEE honor class (2013) of Shanghai Jiaotong University
Computing center:	Shanghai Jiaotong University Computing Center

1.3. Definition

Project Management: To insure our project development in a correct and efficient way.

1.4. Reference

The following documents are provided by the task provider of the project:

SEIEE of Shanghai Jiaotong University

2. Actual Artifacts

2.1. Artifacts

The code hierarchy of our system is clear. There are mainly three parts:

1. Folder "templates" contains web page (.html).

\Grape\templates

Admin.html

Discussion.html

group-id.html

Index.html

index-log.html

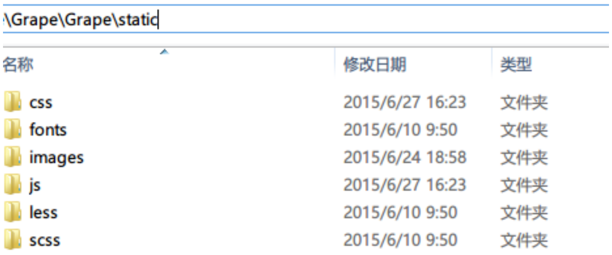
Layout.html

page_not_found.html

view_vote-id.html

votes_static.html

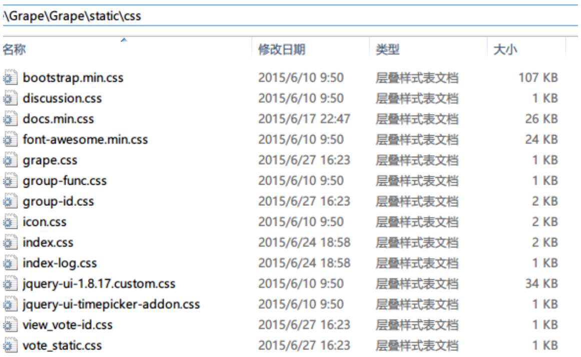
2. Folder “static” contains css, js and so on.



\Grape\Grape\static

名称	修改日期	类型
css	2015/6/27 16:23	文件夹
fonts	2015/6/10 9:50	文件夹
images	2015/6/24 18:58	文件夹
js	2015/6/27 16:23	文件夹
less	2015/6/10 9:50	文件夹
scss	2015/6/10 9:50	文件夹

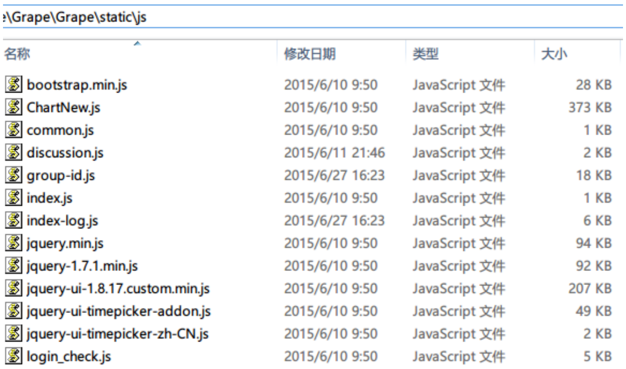
CSS:



\Grape\Grape\static\css

名称	修改日期	类型	大小
bootstrap.min.css	2015/6/10 9:50	层叠样式表文档	107 KB
discussion.css	2015/6/10 9:50	层叠样式表文档	1 KB
docs.min.css	2015/6/17 22:47	层叠样式表文档	26 KB
font-awesome.min.css	2015/6/10 9:50	层叠样式表文档	24 KB
grape.css	2015/6/27 16:23	层叠样式表文档	1 KB
group-func.css	2015/6/10 9:50	层叠样式表文档	1 KB
group-id.css	2015/6/27 16:23	层叠样式表文档	2 KB
icon.css	2015/6/10 9:50	层叠样式表文档	2 KB
index.css	2015/6/24 18:58	层叠样式表文档	2 KB
index-log.css	2015/6/24 18:58	层叠样式表文档	1 KB
jquery-ui-1.8.17.custom.css	2015/6/10 9:50	层叠样式表文档	34 KB
jquery-ui-timepicker-addon.css	2015/6/10 9:50	层叠样式表文档	1 KB
view_vote-id.css	2015/6/27 16:23	层叠样式表文档	1 KB
vote_static.css	2015/6/27 16:23	层叠样式表文档	1 KB

JS:



:\Grape\Grape\static\js

名称	修改日期	类型	大小
bootstrap.min.js	2015/6/10 9:50	JavaScript 文件	28 KB
ChartNew.js	2015/6/10 9:50	JavaScript 文件	373 KB
common.js	2015/6/10 9:50	JavaScript 文件	1 KB
discussion.js	2015/6/11 21:46	JavaScript 文件	2 KB
group-id.js	2015/6/27 16:23	JavaScript 文件	18 KB
index.js	2015/6/10 9:50	JavaScript 文件	1 KB
index-log.js	2015/6/27 16:23	JavaScript 文件	6 KB
jquery.min.js	2015/6/10 9:50	JavaScript 文件	94 KB
jquery-1.7.1.min.js	2015/6/10 9:50	JavaScript 文件	92 KB
jquery-ui-1.8.17.custom.min.js	2015/6/10 9:50	JavaScript 文件	207 KB
jquery-ui-timepicker-addon.js	2015/6/10 9:50	JavaScript 文件	49 KB
jquery-ui-timepicker-zh-CN.js	2015/6/10 9:50	JavaScript 文件	2 KB
login_check.js	2015/6/10 9:50	JavaScript 文件	5 KB

3. Outside are py files

config.py: some configuration and constants

function.py: all kinds of operations connecting the database and front-end

run.py: the view of web pages

init.py: the initialization of the database

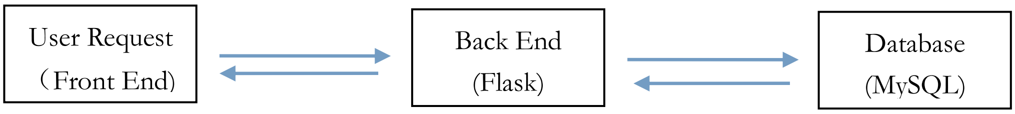
2.2. Main Functions and Performance

Admin:
1.See the details of all users and can also delete them.
2.See the details of all groups and can also delete them.
Normal User:
1. Create Group and Join Group: user is able to create a group to be the leader in the group and there can be new users to join the group.
2. Delete Group and Quit Group: correspondingly user can delete the group he owns or quit the group he has joined.
3. Discuss Operation: a member can post problems to be discussed by other members in the group.
4. Vote Operation: The leader in a group can raise a vote to let the members in the group cast a vote.
5. Bulletin Operation: The user can receive the information broadcast in the group by leader.

Almost all the functional and non-functional requirements are met , one or two functions, like sharing files, will be implemented in the next version.

2.3. Basic Work Flow

The main work flow of our system is the same as common websites. First the user does some requests on the browser, and by "get" or "post" method the request is sent to the back-end implemented by python flask. Then according the request we can do some operations including returning corresponding results and reading or writing database.



2.4. Schedule

Schedule plan:

Phase	Duration(Week)
Project selection	1
Plan	
Requirement Analysis	1.5
Design	
Implementation	1.5

Test	1
Close	1

Actual schedule:

Phase	Duration(Week)
Project selection	1
Plan	
Technique studies	1
Requirement Analysis	2
Design	
Implementation	1.5
Test	1
Close	1

Compared with the plan, we have a new phase, Technique studies, for the reason that most of the team members are beginners on flask.py. We spend 1 week on learning the basic knowledge about flask and jinjia2 and other tech needed in the project. In addition, it is first time we use python as the back-end support and the windows 2007 server, which cost time to get familiar with the whole develop environment.

As for Analysis and Design phase, we pay a lot of emphasis on this part. We believe good design will lead to good code and good software performance, flexibility, scalability. So we spend some time focusing on web design patterns (MVC) so as to take full advantage of what we learnt on object-oriented design courses and make the blueprint of the whole system in every member's head.

2.5. Cost

The platform for coding and software is provided by ourselves.
The server for our website is provided by IEEE honor class. (Permission confirmed.)

3. Job Evaluation

3.1. Work Efficiency Evaluation

Totally, we've got four assignments from teacher, each of them would almost be impossible to be accomplished by a single person. That's why we congregated as a team to work together. Each of the assignment has been done perfectly by all of us. And every one of us has contributed a important part in it.

3.2. Product Quality Evaluation

Our Grape system is a satisfying project. From the very beginning when we start to choose project, every one of us has took an active part in it. In the end, most of the purposed functionality has been accomplished. Meanwhile, we also took the security factors into account. Thus a great number of attacks can be prevented by our system; this can be viewed as a bonus mark.

Of course, some of the functionality are not presented due to the lack of time. But we believe that is a wise choice.

3.3. Used Technique Evaluation

As mentioned above, we use Python as our back-end supporter. And specifically, we use the light-weighted API flask.py to communicate between front-end and database. To make things easier, we choose mysql as the database. (And Python would support mysql in the aid of MySQLdb). To make out website more interactive and secure, we use ajax as a asynchronous way to serve the user.

The above set of technology is under careful considerations. As for database, mysql is one of the most popular free software available for beginners. The rest of the database choice would be either non-free or hard to getting started. As for the back-end supporter, we choose flask as a light-weighted frame because it gives us a great deal of flexibility, most of the detailed functionality can be defined by ourselves.

3.4. Mistake Analysis

Most of the problems are failure we met come from the lack of experience. The most apparent problem is about team work. Since different member has different understandings about certain aspect of the system. We frequently fail to make uniform decisions for the reason of which the code we generates cannot work well while integrating.

The experience tells us how important the communication is. We should predefined something, and make coding style guide first before getting into the real job.

Another severe problem is about schedule. We underestimate the time cost in requirements analysis and the process of learning to use developing environment which cost too much time compared with the original schedule plan. We have to cut down the time on coding and testing so as to meet the deadline, which reduce the performance and quality of the product inevitably.

4. Experience and Summary

The standard and specification of code is important.

Once you have a template and everyone in your group follows it, code gets more clear and easy to read and things become easier to cope with. At the beginning of the process of our project, I put all the operations considering database in the function.py and created a class for each entity such as User, Group. And all the other members in our group followed that principle, which helped us a lot.

Always make the code understandable.

When cooperating with others, it is unavoidable to use or edit others' code. This requires a high readability to our code. It is important to make it easy for others to read your code, which helps to avoid stall in working and mistake in revising. Comment or small documents can give others a better understand, but meaningful variable names and reasonable data structure is more fundamental and effective.

Design should always go before everything.

All the efforts on designing paid off and did shorten our coding process. A good design gives a throughout view of a system. This is always needed to determine what should be included in a class, or what the input and return value should be for a function. Besides, it tells us how our part works with others'. Without a design, members work separately for their own systems. It is designing that makes the project a whole one. Only with a good design can we do our own part smoothly, for we are working for a common goal, rather than combining 5 separated systems together.