

Software Project Management Plan

Project Name ServerClientChat

Date of last change 12/11/2019

Team Members Liboni Andrea Tognella Matteo Radosta Riccardo Martino Francesco

Document Control

Change History

Revision	Change Date	Description of changes
V1.0	22/10/2019	Initial release

Document Storage This document is stored in the project's SVN repository at:

https://github.com/FrankMartinuz/2020_5BI_team1_Liboni.

Document Owner Martino Francesco is the owner of the repository.

Table of Contents

1 OVERVIEW 1.1 Purpose and Scope 1.2 Goals and Objectives 1.3 Project Deliverables 1.4 Assumptions and Constraints 1.5 Schedule and Budget Summary 1.6 Success Criteria 1.7 Evolution of the Project Plan
2 STARTUP PLAN 2.1 Team Organization 2.2 Project Communications 2.3 Technical Process 2.4 Tools
3 WORK PLAN 3.1 Activities and Tasks 3.2 Release Plan 3.3 Budget
4 CONTROL PLAN 4.1 Monitoring and Control
5 SUPPORTING PROCESS PLANS 5.1 Risk Management Plan 5.2 Configuration Management Plan 5.3 Product Acceptance Plan

1 Overview

1.1 Purpose and Scope

Server The purpose of this project is to give a kind of text messaging application. The scope of the server is to manage some requests from the clients. These requests can be a request for a list of online users, and a request for sending a text-message to one, or more, online users.

Client This project serves to provide some sort of text messaging application. The scope of the client is to interface with the server in order to communicate with other clients through a simplified graphical user interface.

1.2 Goals and Objectives

Server The overall goal is to give students a method to chat with their classmate. The app is expected to:

- Manage the user identification through the sign in, log in and log out

- Provide a way to chat with class members.
- Give to client a server that can manage all request.
- Guarantee to every client a reply in useful time.

Client The expectations for the good job of this app is expected to:

- Provide a way to chat with classmates;
- Facilitate the use of the latter to the customer;
- Interface with the server to communicate with other clients.

1.3 Project Deliverables

Server and Client The following items will be delivered to the customer on or before 6/1/2020: 1. Source code for both the client and server portions of the system. 2. User's Guide 3. Project Charter 4. Software Requirements Specification 5. Software Project Management Plan 6. Business Model Canvas 7. Project Model Canvas

1.4 Assumptions and Constraints

Server Assumptions: 1. The protocol will always work 2. The server will always send a feedback about the request. (Done or Failed) Constraints: 1. The server must follow the teacher protocol 2. The server must be on the school LAN for work. 3. The server must be ready by 12/11/2019.

Client Assumptions: 1. Communication between the various clients should take place through an always active server 2. The login or registration must be done in the graphic part Constraints: 1. The client must follow the teacher protocol 2. The client must be on the school LAN for work.

1.5 Schedule and Budget Summary

Server Schedule: 29/10/2019 - Create the file system of the program and planned the job 29/10/2019 - Add private message function and user vector 29/10/2019 - End the main part of the software 05/11/2019 - Testing 05/11/2019 - Fixing the problems and bugs issue during the test 05/11/2019 - End the software development 20/11/2019 - Begin the documentation

Budget: There are not production costs, in terms of money. We have only spend an a amount of 4 lesson at school. $3 \text{ (hours for lesson)} * 4 \text{ (lessons)} = 12 \text{ total hours}$

Client Schedule: 29/10/2019 - Create the file system of the program and planned the job

29/10/2019 - Addition of the graphic part of the software

29/10/2019 - End the main part of the software

05/11/2019 - Testing

05/11/2019 - Fixing the problems and bugs issue during the test

05/11/2019 - End the software development

20/11/2019 - Begin the documentation

Budget: This work does not include cash costs, the only cost was the time spent on this project, approximately 5 lessons to which we have 3 hours each for a total of 15 hours.

1.6 Success Criteria

Server The project will be a success if: * It work without bugs and problems with all the different client that our classmate have done and in all the situation that we expected. * It is ready before the delivery date.

Client The project will be a success if work without bugs and problems with all the different server that our classmates have done and in all the situation that we expected.

2 Startup Plan

2.1 Team Organization

Server and Client Teacher: Provide to us a packs protocol that we have to use in our chat. Programmers (4): Programmers have to plan the project, and after write all the code.

2.2 Project Communications

Server and Client For share our code and all the information we have used GitHub. For the verbal communication we simply talk in classroom.

2.3 Technical Process

Server For this project we have proceeded step by step. The first step was create a listening server. Successively every packet-request available stand for one steps. In this way we were sure that when we pass one step, the latter did not have bugs.

Client We proceeded step by step in this project, initially we created the client that communicated with the server. Later we took care of the accesses or logins and new registrations, in the end we worked on the graphic part in order to facilitate the end user.

2.4 Tools

- Programming Language – Python
- Version Control – source code and written artifacts will be stored in a GitHub repository.
- Packs protocol - invented and supplied by our teacher

3 Work Plan

3.1 Activities and Tasks

Server Every tasks, except the first ("create a server"), take the name from the type of packets analyzed. At the

end of all steps, there are many testing activities, using some testing clients created by us.

Client In the project, the activities were carried out step by step, from the simplest and most basic parts, to the most sophisticated and then concluded with the graphic part, with the related verification of any errors or bugs.

3.2 Release Plan

Server and Client For this project we have chosen to do only one release, or rather the final release. This choice born because this is project have limited size and features.

3.3 Budget

Server For every steps we spent about an half hour. For the testing and bug fixing we have used about 2 hours. We haven't costs in terms of money.

Client In terms of money, this project had no costs. The only cost was the time. Approximately the time it took to develop was 2 hours, the same for correcting any bugs or errors.

4 Control Plan

4.1 Monitoring and Control

Server and Client 01-10-2019 - Study of protocols 15-10-2019 - Study of sockets 22-10-2019 - Understand chat's protocols 29-10-2019 - Start programming in Python. 12-11-2019 - Deliver the functioning server

5 Supporting Process Plans

5.1 Risk Management Plan

Server One risk, that the final user can spot in this software, is the overload of the LAN, or of the Computer where the server is running. This problem depends, first of all, on power of the PC where the server has been ran, in second on how many clients want to use the service simultaneously. The only way to fix this problem is to improve the PC stats, or increase the band available.

Client A possible risk is the possibility that the servers to which they connect may be overloaded with requests, preventing the correct functioning of the platform. This depends on the computing power of the computers where the server is running, the only way is to upgrade the PCs.

5.2 Configuration Management Plan

Server and Client 1. All files will be stored in a GitHub repository. 2. The naming convention for documents will be: .md. Except for the canvas that must have .pdf as suffix 3. All file of the project will be controlled by Git. 4. All the files that are not ended cannot stay in master branch. Every incompleted file must stay in own branch. 5. First of every merge with the master, contact the owner of the repo. 6. Every commit must be reasonable, because we will use it as changes history.

5.3 Product Acceptance Plan

Server The product will be accept if: * Can reply to all clients request without problems. * The delivery date was respected. * The documentation explains as best it can the software.

Client The product will be accepted if: * Can communicate with clients through servers without problems * The login works correctly * The delivery date has been respected. * The documentation best explains the features of the software.