Software Project Management Plan

Team 5

January 05, 2020

Team Members

Gjergjaj Albert
Donisan Ion Cristian
Quintarelli Federico
Castellano Pietro
Scardoni Francesco

Document Control

Change History

Revision	Change Date	Description of changes
V1.0	13/12/19	Initial release

Document Storage
The document is stored on the campus website (www.marconivr.it).

Document Owner
Team 5 is responsible for developing and maintaining this document.

Table of Contents

1 Overview

1.1 Purpose and Scope

Team 5 is interested in creating a desktop chat application that will be useful to the students of 5Ei class, so they can communicate without using the mobile phone. The Potato Chat will accomplish this by giving the possibility to chat in a public room or in a private chat.

The Potato Chat app will give the possibility to the users to login in with an alias directly on the application. After the login there will be two main features:

1) The first one is the public chat where users can access it by using a topic. This allows the user to talk with others that share the same interest.
2) The second one being the private chats, where users that don't want to talk in a public room can talk with someone privately. This can be done after the user registers and logs in the public room, which provides a list of online users.

The application will work on all desktop machines that have different Os, thanks to Java.

The user interface will be user friendly, so it can be used without consulting a user manual. Every complex task will be hidden by us.

1.2 Goals and Objectives

The overall goal of the project is to create a software that gives users an alternative and fast way to chat with each other.

Project Goals:

1. Create an application that sends messages with a nice GUI.
2. Give UMKC visibility in the mobile application environment.
3. Give Professor Burris a cool app to show the department at project end.
4. Learn about software engineering and creating a mobile app.

Project Objectives:

1. Create a mobile interface with UMKC.ManageMyID.com to give students access to their Roo Bucks account information.
2. Create an app that functions in a simple and intuitive manner.
3. Provide students with locations to use their Roo Bucks.

1.3 Project Deliverables

Date	Deliverable
11/10/19	Requirements
18/10/19	Product Feature set baseline
25/10/19	Software architecture
08/11/19	Sprint #1
15/11/19	Sprint #1 Complete
22/11/19	Sprint #2
29/11/19	Sprint #2 Complete
06/12/19	Application testing
13/10/19	Product released

1.4 Assumptions and Constraints

1.4.1 Assumptions

1. Server will allow our client to send messages.
2. The customer and a few user will be available every time we request it
3. Feedback is needed on all the features implemented
4. During meetings the customer will be available to answer questions on the product.

1.4.2 Constraints

Constraints:

1. The application to run on every desktop OS needs java installed.
2. The application should rely only on open source libraries, that help us to speed up the production

1.5 Schedule and Budget Summary

1.5.1 Cost Estimate

The project didn't have any budget.Success Criteria
If we achive all the goals and the grade is superior than a certain threshold, then it will mean that the project was successful.

1.6 Definitions

Term	Definition
Actor	user or other software system that receives value from a user case.
Baselined	the work product has undergone a formal review and can only be changed through the prescribed change control procedures
Client or Customer	the person or organization for which this Roo Balance application is being built.
Developer	the person or organization developing the system, also sometimes called the supplier.
Project	activities that will lead to the production of the Roo Balance application.
Roo Balance Application	the product that is being described here; the software system specified in this document.
Scenario	one path through a user case
Stakeholder	anyone with an interest in the project and its outcomes. This includes clients, customers, users, developers, testers, managers and executives.
User	the person or persons who will actually interact with the Roo Balance application.
Use case	describes a goal-oriented interaction between the system and an actor. A use case may define several variants called scenarios that result in different paths

	through the use case and usually different outcomes.

2 Startup Plan

2.1 Team Organization

Role	Actor(s)	Responsibility
Project Manager	Gjergjaj Albert	Call team meetings, coordinate communications within group, coordinate communications outside group, break out tasks, assign them to teammates
Developer	All group members	Develop software based on requirement and architect specifications
Programmer	All group members	Program to requirement and architect specifications
Tester	All group members	Write test cases, perform unit testing of test cases against incremental release of code, perform integrated testing of test cases against incremental release of code, report issues

2.2 Project Communications

Event	Information	Audience	Format	Frequency
Team Meeting	Task status: completed since last meeting & planned for next; obstacles encountered; change requests in process	All team members	Informal meetings following class; Formal meetings as needed; E-mail status updates & problems as they occur	As needed
Project Status Report	Review finished items, status of prototype; review any problems, schedule slippage, programming issues	All team members, teachers	E-mail with information or In-person as customer sees fit	Iteration Closeout

2.3 Technical Process

For this project we used the Scrum methodology and feedback will be used at the end of each sprint.

2.4 Tools

- Programming language - Java
- Operating System – Desktop Os's
- Version Control – Git and Github
- Development Tools – Netbeans

3 Work Plan

3.1 Resource Estimate

The application is lightweight and it doesn't require a lot of resources to run. It is estimated that you need a minimum of 250 Mb RAM and 500 Mb HDD

3.2 Release Plan

3.2.1 Plan By Feature

Sprint #1 **11/08 – 11/15**

Summary: We create the packets needed for the backend.

Features / Deliverables	**Estimated Effort**	**Actual Effort**
Architecture / Packet construction	50	60

Sprint #2 **11/22 – 11/29**

Summary: We develop the Gui and bind the packets with it.

Features / Deliverables	**Estimated Effort**	**Actual Effort**
Gui	**70**	**60**
Packet Binding	70	80

3.3 Sprint Plans

A detailed sprint plan will be provided for sprint 1. Further task details are available in the schedule.

3.3.1 First Sprint

We have created a protocol witch will be used by the software to communicate with the servers.

3.3.2 Second Sprint

We have implemented the Gui and the features of the actual software.

3.3.3 Final Product

The final product is a software with let's you communicate with other people trough a server. The product has different features like private chat and custom topic rooms.

4 Control Plan

4.1 Configuration Management Plan

The following procedure is to be used when making changes to all baselined work products:

1. All project work products will be stored in a centralized repository running on Github.
2. All baselined documents will have a Document Control section with a change history to track initialization and subsequent changes.
3. All project work products (documents, source code, test cases, program data, test data, etc) will be stored in the Github repository but not all will be under change control (subject to formal change control procedures.) Only the system requirements, project plan and source code will be baselined and under configuration control.
4. Items that are subject to change control will be considered baselined after a group review at the end of the initial document creation.
5. The change control procedure once a product is baselined is:
   (1) anyone wanting to make a change to a baselined item sends an email to the rest of the team and project sponsor (i.e. Professor Sette) describing the change, reason for the change, expected schedule impact, and time line for integrating the change.
   (2) if no one responds to the group within 2 days with a reason for why the change request shouldn't be permitted, it will be considered accepted and the person proposing the change may proceed with the change.
   (3) if anyone does object to the change, the reason for objecting will be discussed at a meeting where everyone is invited to attend and voice

their opinion. At the end of the meeting a democratic vote will be held to decide whether or not the change should be allowed.

(4) if a change takes place, the initiator must collaborate with the project manager to update the schedule

5 Supporting Process Plans

5.1 Risk Management Plan

Rank	Risk	Probability of Loss	Size of Loss	Risk Exposure	Response
1	Schedule / time line delivery	Likely	Major	High	Mitigate: Stick to the schedule.
2	Protocol and gui Developing	Likely	Moderate	Moderate	Avoid: Spending to much time on thinking about the protocol structure
3	Feature 1: Private chat	Unlikely	Moderate	Moderate	Avoid bugs
4	Feature 2: See list of connected people	Unlikely	Minor	Low	Avoid: Not using protocol standards and using local methods.

5.2 Product Acceptance Plan

At the conclusion of each sprint, the prototype created will be tested to ensure it meets the requirements of that sprint.

For the final sprint, product acceptance testing will ensure that the prototype functions as expected with a user's data.