Software Project Management Plan

Team 3

December 12, 2019

Team Members

Enrico Marchi
Elia Gaole
Giulio Federico Grippi
Jennifer Emiliani
Matteo Alexandru Bulai

Document Control

Change History

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

Document Storage
This document is stored in the docs's repository at:
https://github.com/enricomarchidev/Potatolk.git

Document Owner
Gaole Elia is responsible for developing and maintaining this document.

Table of Contents

1 Overview

1.1 Purpose and Scope

Team 3 became interested in creating a client application capable of communicating with a server application developed by another team. This application, which the team wanted to call **Potatolk**, is basically a chat.

It is therefore able to exchange messages with other client applications, created by other teams, through communication with the server.

The application will offer the possibility, once logged in omitting the topic, to communicate in a general room, then an area used by all, from the general room you can then move on to communication with individual users.

If, however, always in the login phase, the topic will not be omitted, you will move to a sort of private room in which you can communicate simultaneously with all users who have entered the same topic.

The user interface will be intuitive, simple to navigate and beautiful to the eye.

1.2 Goals and Objectives

The general objective is to make clients created by different teams communicate with each other.

Objectives of the project:

1. Create an application that works as expected and looks good.
2. Information on how java swing works.

Project goals:

1. Create a mobile interface with java swing
2. Create an application that functions in a simple and intuitive way.

1.3 Project Deliverables

Date	Deliverable
01/11/19	Started Project
01/11/19	Iteration #1 Plan
08/11/19	initiated a first idea of graphic
15/11/19	Worked on the registration packages
22/11/19	merge GUI to the methods created
29/11/19	Worked on message exchange packages
06/12/19	Updated the graphic components
12/12/19	Iteration #1 Complete
12/12/19	Product Released

1.4 Assumptions and Constraints

1.4.1 Assumptions

1. The application are available and function when the server is active.

1.4.2 Constraints

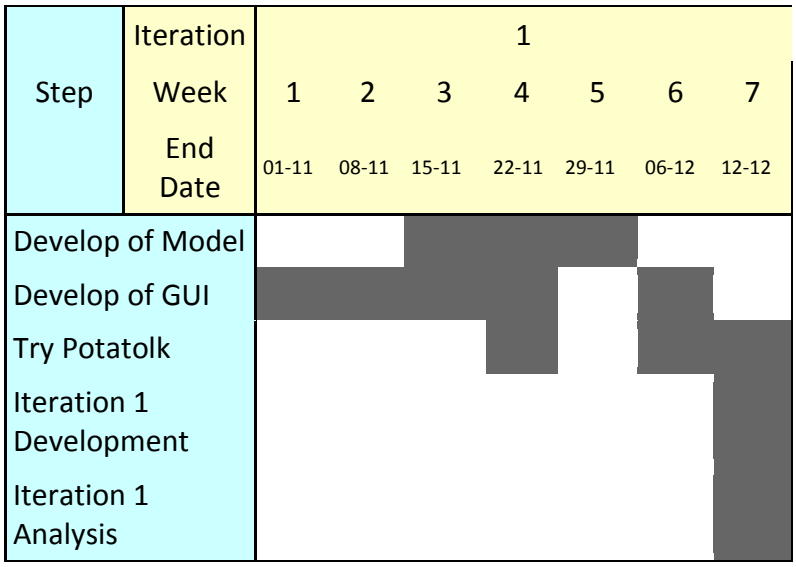
1. The application need specific libraries.

1.5 Schedule and Budget Summary

1.5.1 Cost Estimate

No cost has been estimated for the realization of this project, if not the number of hours at which it will be possible to work in a group, that is 27.
This implies that every member of the group will have to do their homework during the week.

1.5.2 Schedule Summary



1.6 Success Criteria

A functional, easy-to-use prototype that allows users to easily send and receive messages.

1.7 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	Marchi	Call team meetings, coordinate communications within group, coordinate communications outside group, break out tasks, assign them to teammates
Developer	Gaole, Grippi	Develop software based on requirement and architect specifications
Programmer	Marchi	Program to requirement and architect specifications
Tester	Marchi, Gaole, Bulai	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
Architect	Marchi, Gaole	Specify overall internal workings of application
Requirement Engineer	Emiliani	Outline and document project dependencies and requirements. This includes internal and external dependencies.

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, customer	E-mail with information or In-person as customer sees fit	Iteration Closeout

2.3 Technical Process

An iterative and incremental development process is planned. The first iteration will focus on the basic functionality of the application. Subsequent iterations will be based on improving these features.

2.4 Tools

- Programming & Markup Languages – Java Swing
- Operating System – Windows
- Version Control – all work products will be stored in a repository
- Development Tools – Netbeans

3 Work Plan

3.1 Resource Estimate

Detailed resource estimates are available in the linked file Team 3 Estimated Effort.

3.2 Release Plan

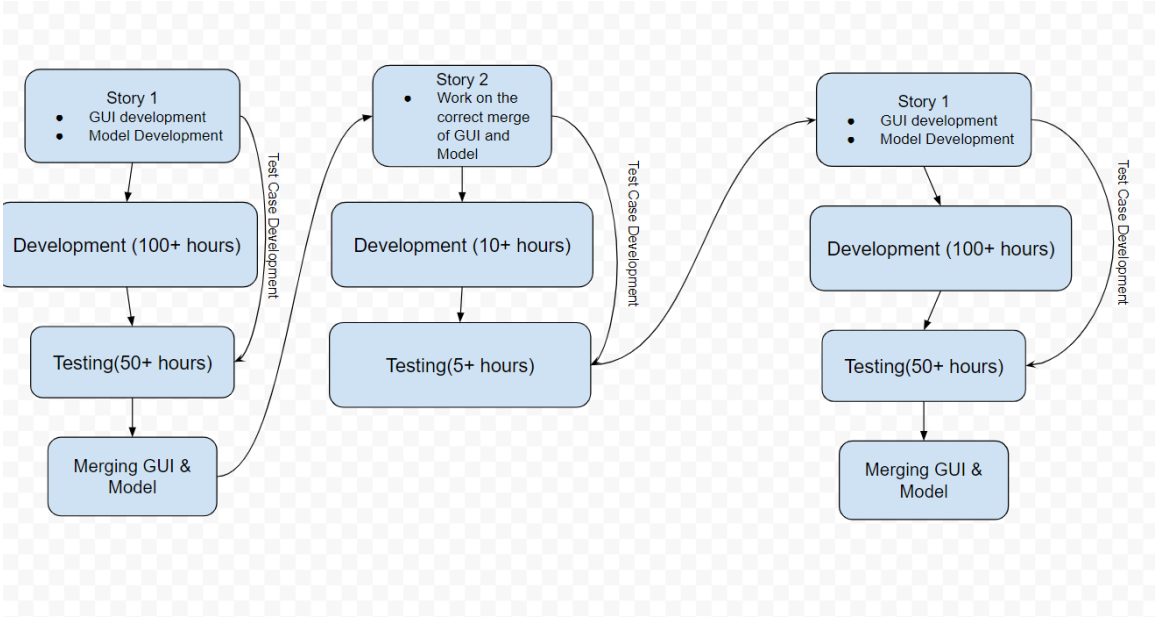
3.2.1 Plan By Feature

Iteration #1 **01/11-12/12**

Summary: Demonstrate fundamental architecture, worked on Gui and model to arrive at an almost complete version of the project.

Features / Deliverables	**Estimated Effort**	**Actual Effort**
Architecture / framework design	20	
Authentication / Credential management	50	

3.2.2 Flow Chart



3.3 Iteration Plans

A detailed iteration plan will not be provided for Iteration 1.

3.3.1 First Iteration

The client communicates correctly with the server and the graphic part is perfectly merged with the various methods created

4 Control Plan

4.1 Monitoring and Control

The following list of dates includes formal reviews outside of the Communication Plan. Milestones are included to reference where the project is scheduled to stand as these reviews occur:

Date	Review / Milestone
08/11/2019	*Milestone: Initial idea for graphic*
08/11/2019	Generated three different windows (login, logout & chat)
08/11/2019	Analyzed the colors
15/11/2019	*Milestone: Started to work with packages*
12/12/2019	*Milestone: Iteration #1 Complete*
12/12/2019	*Milestone: Product Released*
12/13/2019	Final Presentations

4.2 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 Github repository running on a central server.
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 repository but not all will be under change control. 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 Burris) describing the change, reason for the change, expected schedule impact, and timeline 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	Learning curve for new tools and technologies longer than expected	Unlikely	Moderate	Moderate	Buy Information: Begin working on a basic prototype early to test out fundamental programming concepts & knowledge

5.2 Test Plan

The test plan defines the items that will be tested, methods for testing, and a schedule detailing the tasks, owners, and timeline.

The test plan will not be available.

5.3 Product Acceptance Plan

At the end of the iteration, the prototype created will be tested with the other clients to verify the operation of the application with the others. a bogus server can be created to test client functionality.