Project Plan

***Tlab***

***Sponsor****: Ms Rawish butt*

***Advisor****: Muhammad Talha(1812241)*

# Table of Contents

1. Overview
2. Document Organization 3. Goals & Scope
   1. Goals
3. Technical Process
   1. Methodology
   2. Tools and Techniques 4.3. Internal Artifacts
4. Deliverables
5. Scheduling and Estimates 6.1. Project Schedule

6.2. Resource Allocation

6.3.Estimates

6.4.Tracking and Schedule Changes

# Overview

Tlab, Inc. is a provider of Internet hosting for software development and version control using Git. It offers the distributed version control and source code management (SCM) functionality of Git, plus its own features. It provides access control and several collaboration features such as bug tracking, feature requests, task management, continuous integration and wikis for every project. Headquartered in California, it has been a subsidiary of Microsoft since 2018.

Tlab offers its basic services free of charge. Its more advanced professional and enterprise services are commercial. Free Tlab accounts are commonly used to host open-source projects. As of January 2019, Tlab offers unlimited private repositories to all plans, including free accounts, but allowed only up to three collaborators per repository for free. Starting from April 15, 2020, the free plan allows unlimited collaborators, but restricts private repositories to 2,000 minutes of Tlab Actions per month. As of January 2020, Tlab reports having over 40 million users and more than 190 million repositories (including at least 28 million public repositories), making it the largest host of source code in the world.

# Document Organization

This document is meant to provide a detailed view of the various aspects of Tlab. It covers everything from project scope to risks, from schedule to metrics, and many other helpful aspects.

Certain sections of this document use Priority values to convey importance of certain features/goals. The priority values are on a scale of P1 to P3. An explanation of the priority values follows:

* + P1 is the topmost priority and is absolutely necessary for a viable product. Without these items, the product should not be released.
  + P2 items are important and impact quality of the product, but the product would still be shippable without items in this category.
  + P3 items are nice to have but can be dropped if necessary. They are not required for a quality shippable product. This category includes enhancements.

Other tables in some sections may use different measurements related to their specific purpose or topic (i.e. the risk scoring table contains impact and probability values). The values for those tables will be defined in a description directly above the table.

# Goals & Scope

Goals provide the primary objectives for the project and help define the scope. The following section specify this project’s prioritized goals with explanations, in order to clarify scope, intentions, and direction of the project.

## Goals

* Tlab is a hosting platform wherein developers can store their computer code in the Tlab server The user can sign up/do registration with the system
* It is an open-source version control and collaboration platform for program developers.
* It helps all the programmers to collaborate with each other who are working on a similar project and also share their code easily as and when required
* The collection of these files will shows the source code of a program which is spread across the files to make it easier to manage what can be many thousands of lines of code and still be able to find the parts you need to find.

# **Technical Process**

# **Methodology:**

# The system is Complex in design and to implement. The system requires a lot of resources and the system will work in almost all configurations.

# It has got following features:

# • It will ensure version control

# • Data will be efficiently maintained

# • Developers can work effectively with other Developers

# • Minimum time needed for the various processing.

## Tools and Techniques

The software that runs Tlab was written using Ruby on Rails and Erlang by Tlab, Inc. developers Chris Wanstrath, PJ Hyett, and Tom Preston-Werner.

And git is written with C, Shell, Perl, Tcl, and Python.

**System Requirement**

The requirement definition is concerned with the analysis of the existing system with the aim of determining and structuring the requirement of the proposed system. It is achieved with the aid of user requirement. The Analysis stage was specifically carried out in focus of the functionality dataflow at Young Legacy Line Transport Division.

**Requirement Specification**

Requirement Specification a complete description of the behavior of a system to be developed and may include a set of use cases that describe interactions the users will have with the software. In addition it also contains nonfunctional requirements. Non-functional requirements impose constraints on the design or implementation (such as performance engineering requirements, quality standards, or design constraints).

**Functional Requirements**

Functional requirements define the specific functions that the system performs, along with the data operated on by the functions. The functional requirements are presented in scenarios that depict an operational system from the perspective of its end users. Included are one or more examples of all system features and an enumeration of all the specific requirements associated with these features.

* The system shall incorporate mechanism to authenticate its users
* The system shall verify and validate all user input and should notify in case of error detection and should help the user in error correction
* The system shall allow sharing of files in the system
* The system shall allow quick messages to be exchanged without face to face interaction

**Non-Functional Requirement**

Non-functional requirements address aspects of the system other than the specific functions it performs. These aspects include system performance, costs, and such general system characteristics as reliability, security, and portability. The non-functional requirements also address aspects of the system development process and operational personnel. It includes the following:

* The system shall be user friendly and consistent
* The system shall provide attractive graphical interface for the user
* The system shall allow developer access to installed environment

# Deliverables

Deliverables are derived from the required senior project deliverables, project specific requirements, and deliverables mandated by our chosen process.

At a high level, major deliverables for the project include:

* + **Project Planning document (this document)** ­ Will contain information including but not limited to overview, goals, scope definition, risks, scheduling, and process methodology definition.
  + **Project website** ­ This will contain information about the project, including all non­proprietary work products and artifacts.
  + **Domain model** ­ Will be used to describe and demonstrate an understanding the application domain.
  + **Weekly four­up charts** ­ These will be reviewed at the beginning of each weekly meeting and will be used to gauge status and project progress.
  + **Time tracking** ­ Time/effort worked will be tracked by me and then aggregated weekly to be recorded on the project website.
  + **Select process/product metric tracking** ­ Every two weeks or team will track and provide updates on the website for at least two process/product metrics. This will provide a progressive view of the project and allow us to concentrate on improvement.

**Interim and final project presentations** ­ Will be used to provide midpoint and final status/progress information, as well as to explain and receive feedback on project approach and execution.

* + **Final project poster and presentation** ­ This formal presentation will be used to present our work once completed to raise awareness and receive feedback from others.
  + **Technical report** ­ Will present a comprehensive technical view of the project.
  + **Interim and final team self­assessments** ­ These will be used to gauge progress and participation of each team member.
  + **Post­mortem reflection report** ­ Will provide a means to reflect on the project, team’s strength and weaknesses.
  + **CD containing project artifacts** ­ Will contain artifacts related to the project to be delivered to the customer.
  + **Individual senior project survey** ­ Used by the software engineering faculty for future senior projects.

# **6.Scheduling and Estimates**

Schedule and resources will be managed using Gantter, a project management tool that integrates with Google Drive. The Gantter project for the team can be found in the developer’s Google Drive, as well as here.

## 6.1.Project Schedule

Overall project schedule is framed within the RIT fall and spring semesters time frame, accounting for breaks and days off. The schedule also attempts to account for risks and unexpected changes. It has been created around the **API (Application Programming Interface) ,**see the Technical Process section and includes deliverables and schedule items related to the methodology.

The project schedule will be managed within Gantter.

A static initial schedule is below for reference. This includes only some initial tasks and first couple iterations since following iterations depend on the requirements document being made in the first iteration.

## 6.2.Work Breakdown Structure

## The WBS is a foundational building block to initiating, planning, executing, and monitoring

## and controlling processes that are used to manage projects, There are many project management

## tools and techniques that use the Work Breakdown Structure.

The Work Breakdown Structure, or WBS, shows a hierarchical view of high level tasks in the project and highlights their relationships to one another.

The WBS will be managed in Gantter, but a static initial WBS created for version 1 of this Project Plan is below for reference. This again only includes the initial tasks created and the first couple iterations, since defining following iterations depends on creation of the requirements document.

## 6.3.Resource Allocation

All resources will be managed in Gantter.

Hardware resources required for this app are only digital devices like computer, laptop, and smart phones to design and run this app.

We will also use laptop/desktops and mobile devices for testing purposes.

## 6.4.Estimates

Estimations will be done in Gantter and in the Requirements document created after waterfall model iteration 1. Estimations will include team effort/time estimation as well as product size (function points).

## 6.5.Tracking and Schedule Changes

All tracking and schedule changes will be managed within the Gantter project.