SCRUM PROCESS MANAGEMENT

High Level Design & Low Level Design

Document Control :

SCRUM PROCESS MANAGEMENT				
Guided by- **SANKAR SIR**				
Date	**Version**	**Author**	**Brief Description of Changes**	**Approver Signature**
December 06,2022	1			

Introduction

- Scrum, the most common Agile software development methodology, is an iterative approach that has the Sprint — the scrum word for iteration — at its heart. Throughout an Agile project, Scrum teams use evaluation to ensure the team meets the objectives of each step of the procedure.
- Inside Agile growth, two central positions assist Scrum teams. The front is a ScrumMaster, who can be thought of as a squad leader, encouraging team members to compete at the highest level through the Scrum method.
- And the second part of this scrum team contains the Team Members , the job of the team members is to complete the given task and update the completion status of the task everyday. This approach helps the team to have a clear image of the completion status of the tasks at an everyday basis.

1.1Intended Audience:

This document is intended to be read by ScrumMaster and the Team Members.

1.2Acronyms/Abbreviations:

CLIENT	USER
	TO GIVE A CLEAR VIEW TO THE SCRUMMASTER AND THE TEAM MEMBER ABOUT THE COMPLETION STATUS OF THE PROGRAM.

1.3Project Purpose:

- The purpose of this document is to show the requirements for the Scrum Process Management System, which gives a clear understanding to the user and employees of the completion of the tasks without the involvement of the mediators.

- A sprint is a short, time-boxed period when a scrum team works **to complete a set amount of work**. Sprints are at the very heart of scrum and agile methodologies, and getting sprints right will help your agile team ship better software with fewer headaches.

1.4 Key Project Objectives:

- Scrum Master will login and assign new tasks to the Team Members.
- Team Member will login and check the task assigned to them.
- Team members will update the task completion everyday for the sprint duration.
- The system will calculate the average of all the tasks under that particular userstory.
- The average of all the tasks will be taken as the total story completion percentage.
- The application will now update the task completion under userstory.

1.5 Project Scope and Limitation:

- The ScrumMaster creates a backlog (essentially, a wishlist of tasks that need to be prioritized in a project)
- The Scrum team conducts a sprint planning session where the tasks necessary to complete items on the wishlist is broken down into small, more easily manageable chunks
- The team creates a sprint backlog and plans its implementation
- The team decides a time duration for every sprint (the most common intervals is probably two weeks)
- The team gets together every day for a brief Scrum meeting (often referred to as a Daily Standup) where each member of the team shares daily updates, helping the team and the project manager assess the progress of the project
- The certified Scrum Master guides the team and keeps them focused and motivated
- The development Team updates the completion status of the tasks everyday.

1.6 Functional Overview :-

3.1.1 SSMP_01->: void printReport() :- This function is printing the Report of the application.

3.1.2 SSMP_02-> void loadReport() :- this function will load the Report in the application.

3.1.3 SSMP_03-> void createTaskLL() :-This function asks the Scrum master for the new task and creates a new task in the application.

3.1.4 SSMP_04-> void calculations() : - This function will take the completion status from the team member and calculate the average of all the tasks

3.1.5 SSMP_05->updateCompletionStatus() :- This function will update the completion of the tasks in the task files.

3.1.6 SSMP_06->void displayTaskLL() :- This function will display all the tasks that are already assigned or the newly assigned tasks with their completion status.

3.1.7 SSMP_07-> void displayUserTasks() :-This Function will Display the Userstories of the feature and thier respective completion rates.

3.1.8 SSMP_08-> void appendTaskLL() :- This function will append a the newly assigned task to the end of the task list of a particular userstory and will assign it to the team member whose id is mentioned.

3.1.9 SSMP_09-> void loadTasks() :- This Function will Load all the present tasks and their respective completion status in the system for a particular user story.

3.1.9 SSMP_09-> void appendTasksCSV() :- This Function will write the newly assigned tasks to the csv.

3.1.10 SSMP_10-> void createUserStoriesLL() :-This Function will create a user story linked list with all the details with the updated completion status of the tasks average in that particular userstory.

3.1.11 SSMP_11-> void displayUserStoryLL() :- This Function will Display the userStories with the updated completion status and the story id and all the details about the stories.

3.1.12 SSMP_12-> void appendUserStoryCSV() :-This Function will append the updated UserStories to the csv file .

3.1.13 SSMP_13-> void insert_end() :-This will insert a new userstory provided by the ScrumMaster in the users tories linked list.

3.1.14 SSMP_14-> void updateUserStoriesLLfromTaskData() :-This function will update user stories from the task data using the task linked list.

3.1.15 SSMP_15->void updateUserStoryCSVFromLL() :- This Funtion will update the userStory csv file from the data that is newly appended to the userstory linked list.

3.1.16 SSMP_16-> void loadUserStories() :- This function will load the user stories to the System.

3.1.17 SSMP_17-> int checkLogin(); :- This function will take the user login details and compare it with the values that are preloaded in the csv and if the data match it will enter the user as the designated position of the user.

2. Design Overview:

Instant Chatters comprises of the following modules:

Name of the Module	int login()
Handled by	Aishwaraya
Description	It will login the user at the designated position.

Name of the Module	Load userStories(), void loadtask()
Handled by	Mubeen
Description	This feature will load the details of the userstories and of the tasks in the respective files.

Name of the Module	Void updateuserstoryfromtaskdata(),updateuserstoryCSVFromLL()
Handled by	Subhashini rani
Description	This feature will update the user stories for its task data and other function will update the user stories data in the csv.

Name of the Module	Void insert_end()
Handled by	Nirupa
Description	It will insert a new user story.

Name of the Module	Void loaduserStory()
Handled by	Kavyashree
Description	It will list all the details of the user .

Name of the Module	void displayuserstory(),
Handled by	Nirupa
Description	This feature will display all the current userstories and the completion status of the userstories with all the mentioned details.

Name of the Module	void displatatskLL()
Handled by	Mubeen
Description	This function will display all the tasks with their respective completion status and mentioned details.

Name of the Module	AppendtaskLL(), appenduserstories()
Handled by	Subhashini rani
Description	This feature will update the newly assigned task in the task linked list and appenduserstories will update the newly declared userstory in the userstory linked list.

2.1 Design Objectives:

This project aims to create and develop the Scrum Process Management System In which there are two user the Scum Master and the Team Members ,The Scrum masters job is to update new tasks and to assign it to different team members , the job of the team members is to update the task completion status of the task that has been assigned to them .The application will then give the users a clear view of the amount of work completed and the amount of work that is left.

2.2 Performance:

The system will work on the admin terminal. The performance depends on the hardware component of the admin’s system.

2.3 Maintenance:

- If maintenance demands consistently keep the Team from completing their Sprint Plan, stop planning for so much. In Sprint Planning, leave some headroom – an allowance for maintenance.Reduce the forecast for new feature work. The size of the allowance may be easy to determine from past Sprints or it may take experimentation. If the Team does not need all of the time budgeted in a Sprint, they can us it for more feature work or payment of technical debt.
- When an allowance is made for maintenance, we can take turns handing it. Team members can rotate in the role of “fixer” from Sprint to Sprint so that no one gets stuck with the cleanup work. The fixer can manage their own time between the maintenance and new work.
- This is the ultimate solution. Use proper development practices – pairing, test-first development, automated acceptance testing, continuous refactoring. Make the code better every day. Wrap the system in automated tests to make bug-hunting easier.

3. Environment Description:

3.1 User Desktop Requirements:

a. 64-bit processor, 1 GHz or faster

b. At least 2 GB free hard drive space

c. At least 1 GB RAM

3.2 Server-Side Requirements:

a. 64-bit processor, 1 GHz or faster

b. At least 1 GB free hard drive space

c. At least 1GB RAM

3.2.1 Deployment Considerations:

a. Easy setup: no session storage daemon, use tmpfs and memory caching to enhance performance.

b. Local storage is used.

c. No network latency to consider.

d. To scale buys a bigger CPU, more memory, larger hard drive, or additional hardware.

3.2.2 Application Server Disk Space:

No such disk space is required as the program is fully functional on onlineIDE(s) as well. The Local Operating System is required and one text file to store the records of processes.

3.2.3 Database Server Disk Space:

No such disk space is required as the program is fully functional on online IDE(s) as well. The Local Operating System is required and one text file to store the records of processes.

3.2.4 Integration Requirements:

1. Language: C
2. Tools: Splint ,Valgrind, Makefile
3. Complier: GCC compiler
4. Linux Environment

3.2.5 Jobs:

We can establish connections between clients who are connected to the server. And we can search the chat history of the clients.

3.2.6 Network: End to End

3.3 Configuration:

3.3.1: Operating System: Linux environment