SCRUM PROCESS MANAGEMENT SYSTEM

High Level Design and Low Level Design

Document Control :

TOPIC : SCRUM 1				
Guided by- **Rahul Tarkunde**				
Date	**Version**	**Author**	**Brief Description of Changes**	**Approver Signature**
December 07, 2022	1.0	Shriya Subhash Pawar		
December 08, 2022		Sakshi Pawar and Smita Jagtap	Flow Chart	
December 08, 2022		Vahida Khatib	Spelling Mistakes	

Introduction

1. PURPOSE

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 Scrum Master, 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 every day. This approach helps the team to have a clear image of the completion status of the tasks at an everyday basis.

1.1 Intended Audience:

This document is intended to be read by QA Team, Development Team and the Team Members.

1.2 Acronyms/Abbreviations:

CLIENT	USER
QA	Quality Assurance.

1.3 Project 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 every day for the sprint duration.
- The application will now update the task completion under user story.
- All the memory that was allocated to files will be freed after execution.

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 every day.

1.6 Functional Overview: -

1.6.1 SPMS 01: void printFeatures() :- This function is printing the features of the application.

1.6.2 SPMS 02: void loadFeatures() :- This function will load the features in the application.

1.6.3 SPMS 03: void createTaskLL() :-This function asks the Scrum master for the new task and creates a new task in the application.

1.6.4 SPMS _04: void calculations() : - This function will take the completion status from the team member and calculate the average of all the tasks

1.6.5 SPMS_05: updateCompletionStatus() :- This function will update the completion of the tasks in the task files.

1.6.6 SPMS _06: void displayTaskLL() :- This function will display all the tasks that are already assigned or the newly assigned tasks with their completion status.

1.6.7 SPMS 07: void displayUserTasks() :-This Function will Display the User stories of the feature and their respective completion rates**.**

1.6.8 SPMS 08: void appendTaskLL() :- This function will append a the newly assigned task to the end of the task list of a particular user story and will assign it to the team member whose id is mentioned.

1.6.9 SPMS 09: void loadTasks() :- This Function will Load all the present tasks and their respective completion status in the system for a particular user story.

1.6.10 SPMS 10: void appendTasksCSV() :- This Function will write the newly assigned tasks to the csv.

1.6.11 SPMS 12: 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 user story.

1.6.12 SPMS 13: void displayUserStoryLL() :- This Function will Display the user Stories with the updated completion status and the story id and all the details about the stories.

1.6.13 SPMS 14: void appendUserStoryCSV() :- This Function will append the updated User Stories to the csv file .

1.6.14 SPMS 15: void insert_end() :- This will insert a new user story provided by the ScrumMaster in the user stories linked list.

1.6.15 SPMS 16: void updateUserStoriesLLfromTaskData() :- This function will update user stories from the task data using the task linked list.

1.6.16 SPMS 17: void updateUserStoryCSVFromLL() :- This function will update the user Story csv file from the data that is newly appended to the user story linked list.

1.6.17 SPMS 18: void loadUserStories() :- This function will load the user stories to the System.

1.6.18 SPMS 19: 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	Rohit Kumar Kandula
Description	It will login the user at the designated position.

Name of the Module	Load userStories (), void loadtask ()
Handled by	Vahida Aslam Khatib
Description	This feature will load the details of the user stories and of the tasks in the respective linked lists.
Name of the Module	Updateuserstoriesfromtaskdata(),**updateUserStoryCSVFromLL ()**
Handled by	Maya Vinayak Chavan
Description	This feature will update the user stories from its tasks data and other function will update the user stories data in the csv.

Name of the Module	void insert_end()
Handled by	Sakshi Ranjit Pawar
Description	It will insert a new user story in the user story linked list at the end.

Name of the Module	Void loaduserStory()
Handled by	Sakshi Ranjit Pawar
Description	It will list all the details of the user which is stored in the linked list.

Name of the Module	void displayuserstory(),
Handled by	Smita Ishwar Jagtap
Description	This feature will display all the current user stories and the completion status of the user

	stories with all the mentioned details.

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

Name of the Module	AppendtaskLL(), appenduserstories()
Handled by	Shriya Subhash Pawar
Description	This feature will update the newly assigned task in the task linked list and appendicostomies will update the newly declared user story in the user story linked list.

2.1 Design Objectives:

This project aims to create and develop the Scrum Sprint Management System In which there are two user the Scum Master and the Team Members, The Scrum master's 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 use 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 Time Zone Support: IST- Kolkata

3.2 Language Support: English

3.3 User Desktop Requirements:

a. 64-bit processor, 1 GHz or faster
b. At least 2 GB free hard drive space
c. At least 4 GB RAM

3.4.1 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.

3.4.2 Application 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.

3.4.3 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.

3.5 Integration Requirements:

1. Language: C

2. Tools: Splint, Valgrind, Makefile

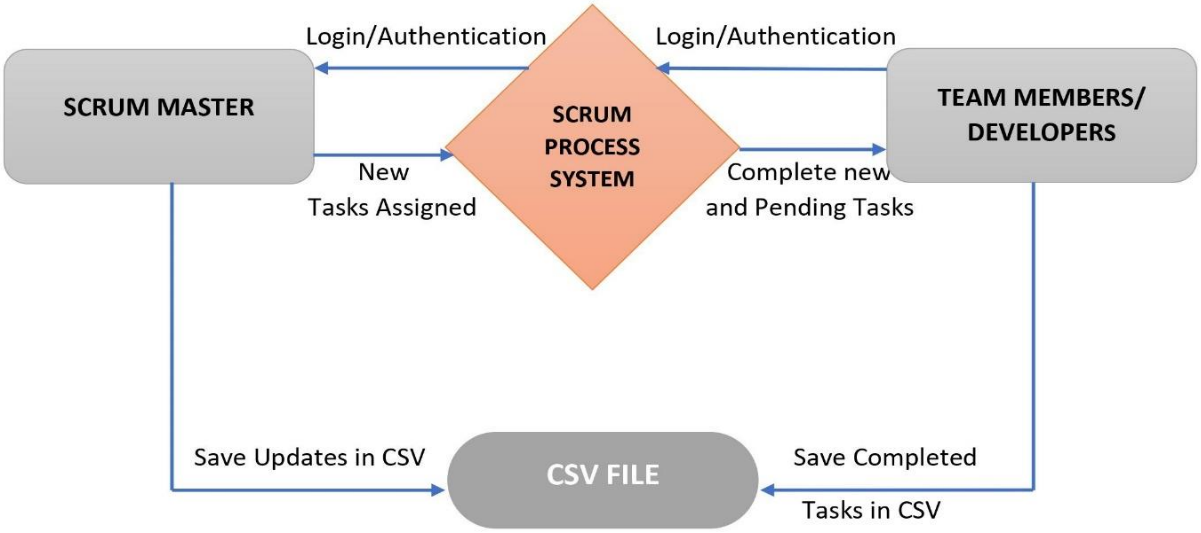
3. Complier: gcc

4. Linux Environment: Ubuntu 20.04 or 22.04 on Intel x64.

3.6 Configuration:

3.6.1: Operating System: Linux Environment: Ubuntu 20.04 or 22.04 on Intel x64.

4. Data Flow Diagram: -



Level 0 DFD

Scrum Master:

- Scrum masters are the facilitators of scrum, the lightweight agile framework with a focus on time-boxed iterations called sprints. As facilitators, scrum masters act as coaches to the rest of the team.
- Facilitate daily standups (or the daily scrum) as needed.
- Participate in the meeting and capture feedback.
- Work as the administrator of the scrum board. Ensure that cards are up to date and the scrum tool, is working well.

Team Members/ Developers:

- Scrum teams are small. The Scrum Guide recommends 10 or fewer total members to ensure optimal communication and productivity.
- The development team doesn’t just wait for orders; they usually collaborate to map out goals and plans for achieving them.
- The responsibilities of a development team will also depend on the end goals of the Scrum team.