# **Milestone 2 Scrum Report**

All students are expected to attend the scrum meetings and to participate. Failure to do so will result in greatly reduced grades.

**GROUP**: \_\_\_\_**D**\_\_\_\_\_\_

**Members Present**:

|  |  |
| --- | --- |
| 1. **ARSHNOOR KAUR** | 4. **AJAYPARTAP SINGH MAAN** |
| 2. **GURMEHAK KAUR UPPAL** | 5. |
| 3. **MANAS GANDOTRA** | 6. |

## Milestone 2 Tasks

Some of the software for the project has already been written for you and is available on Blackboard. You must use this in your project and every team should add it to the source code for their repository. Anything in the main function is simply for demonstration purposes and can be replaced. The software you are being given has not been tested and you will need to test it.

You need to study the problem and the code provided for you and then:

* Add any new data structures you will require This will require a thorough analysis of the problem and the existing software. This should be done by creating a new header file in the directory where the rest of the source code has been placed. You do not want to go back and modify it later if you can avoid it as it will slow the project.
* Create a test plan for the project by replacing the text in the supplied test plan template with your test plan.

**Deliverables due 4 days after your lab day:**

* An analysis of the problem (no written artifacts produced).
* A series of data structures created as header files and **stored in the repository**.
* A test plan stored in the repository.
* Completed scrum report including reflection questions answered.

**Rubric**

|  |  |  |
| --- | --- | --- |
| **Individual** | Group participation (includes GitHub commits and Jira usage) | 80% |
| Teamwork | 20% |
| **Group** | Data structures (complete, correct, and well-designed, updated in the project, and added to the repository) | 25% |
| Test plan (complete, well-written) | 25% |
| Git usage (used properly with good structure) | 10% |
| Jira usage (creates issues, tracks progress) | 20% |
| Scrum report & reflections | 20% |
| **Deadline** | 20% deduction for each day you are late |  |

**Scrum Report**

**Summary of Tasks Completed or Delayed in the last week:**

Here you can list all the tasks completed in the last week along with any tasks which could not be completed with a reason why they could not be completed.

|  |  |  |
| --- | --- | --- |
| **Member** | **Tasks Completed** | **Tasks Delayed/Blocked** |
| **ARSHNOOR KAUR** | **Overviewed the Scrum Report and participated in problem analysis.** | **No** |
| **GURMEHAK KAUR UPPAL** | **Analyzed and completed the test plan.** | **No** |
| **MANAS GANDOTRA** | **Analyzed and completed the test plan.** | **No** |
| **AJAYPARTAP SINGH MAAN** | **Completed the reflection question-answers.** | **No** |
|  |  |  |

For every task delayed or blocked, describe the reason for the delay or block, how it impacts the project and the proposed solution or workaround.

|  |  |
| --- | --- |
| **Delayed or Blocked Task** | **-** |
| **Reason for delay or block** | **-** |
| **Impact on Project** | **-** |
| **Solution or work-around** | **-** |
|  | **-** |
| **Delayed or Blocked Task** | **-** |
| **Reason for delay or block** | **-** |
| **Impact on Project** | **-** |
| **Solution or work-around** | **-** |

**Summary of Meeting:**

A summary of the main points discussed in the meeting and the outcomes of the discussions.

|  |  |  |
| --- | --- | --- |
| Topic | Discussion Summary | Outcome |
| 1. AN ANALYSIS OF THE PROBLEM | **WENT THROUGH THE WHOLE MILESTONE STEP BY STEP, REVIEWED THE CODE, AND ANALYSED THE DELIVERABLES.** | **PROBLEM ANALYSIS WAS DONE EFFECTIVELY.** |
| 2. DATA STRUCTURES AS HEADER FILES | **A SERIES OF HEADER FILES EACH CONTAINING DATA STRUCTURES WERE CREATED.** | **HEADER FILES COMPLETED.** |
| 3. TEST PLAN CREATION | **DISCUSSION ABOUT THE APPROACH TO TESTING DELIVERABLES AND THE RESPONSIBILITIES OF EACH TEAM MEMBER WERE OUTLINED.** | **TEST PLAN COMPLETION AS PER THE ROLES ASSIGNED TO EACH MEMBER** |
| 4. COMPLETION OF SCRUM REPORT AND REFLECTION QUESTIONS ANSWERED | **QUESTIONS WERE DISCUSSED AND UPDATED WITH CORRECT RESPONSES.** | **SCRUM REPORT MADE.** |
|  |  |  |

**Summary of Decisions Made:**

This will include major architecture and design decisions, testing decisions, prioritization of tasks, dealing with problems encountered and other major outcomes from the meeting.

|  |  |
| --- | --- |
| Decision | Rationale |
| 1. EVERY INDIVIDUAL DELIVERABLE SHOULD BE COMPLETED 2 DAYS BEFORE THE DEADLINE. | **THIS CUSHION OF TIME IS TAKEN SO THAT THERE IS ENOUGH TIME TO COMPLETE THE WORK IN THE REST OF THE TIME ALONG WITH HELPING EACH OTHER WITH REMAINING TASKS.** |
| 2. ANY POSSIBLE DELAYS IN SUBMISSION OR INABILITY TO JOIN THE MEETING SHOULD BE REPORTED IN ADVANCE. | **ALONG WITH EFFECTIVE COMMUNICATION BETWEEN GROUP MEMBERS, REPORTING OF THE TASKS IN PROGRESS MUST BE REPORTED ON JIRA.** |
|  |  |

**Tasks Attempted During Meeting:**

Each member is assumed to participate in the scrum meeting and contribute to the completion of the scrum report and reflections. Since the scrum meeting will not take more than 20-30 minutes, there is lots of time left to undertake some of the actual work tasks. In the table below, each member should list what they did to complete the scrum report, the reflections, and 1-4 other tasks they completed during the class period. If a task cannot be completed, the student should indicate why this was not possible.

|  |  |  |  |
| --- | --- | --- | --- |
| Member | Task Attempted | Time Spent | Complete? |
| ARSHNOOR KAUR | **Overviewed the milestone 2 pdf thoroughly, scrum report discussed** | **30 MIN** | **Yes** |
| GURMEHAK KAUR UPPAL | **Overviewed the milestone 2 pdf thoroughly and discussed the deliverables.** | **30 MIN** | **Yes** |
| MANAS GANDOTRA | **Overviewed the milestone 2 pdf thoroughly and discussed the deliverables of Jira management.** | **30 MIN** | **Yes** |
| AJAYPARTAP SINGH MAAN | **Overviewed the milestone 2 pdf thoroughly, and reflection discussed.** | **30 MIN** | **Yes** |
|  |  |  |  |

**Scrum Tasks Selected for Next Week**:

The tasks each member has selected to pursue for this class or the next week.

|  |  |
| --- | --- |
| Group Member | Task Description |
| ARSHNOOR KAUR | **REFLECTION + SCRUM REPORT OVERVIEW** |
| GURMEHAK KAUR UPPAL | **BLACK BOX TEST CASES DOCUMENT** |
| MANAS GANDOTRA | **JIRA+BLACK BOX TEST CODE** |
| AJAYPARTAP SINGH MAAN | **GIT ORGANIZATION+ REFLECTION** |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

This is where you should highlight the major accomplishments of the class.

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| MILESTONE 2 TOWARDS COMPLETION | **THE DELIVERABLES OF MILESTONE 2 SUCCESSFULLY SUBMITTED.** |
| MILESTONE 3 TASKS ASSIGNED TO EACH MEMBER | **THE MILESTONE 3 NEED TO BE SUBMITTED ON TIME WITH ALL DELIVERABLES COMPLETED** |
| THE PROJECT REQUIREMENTS UNDERSTOOD IN DETAIL | **EACH MEMBER OF OUR TEAM IS AWARE OF THEIR ASSIGNED TASKS AND THE PROJECT REQUIREMENTS.** |
|  |  |
|  |  |

**Things That Went Well in This Meeting:**

Here you can highlight things which worked well. This indicates that the way you worked on these items is working and should be continued.

|  |  |
| --- | --- |
| Topic/Work Item | Reason for Success |
| SCRUM REPORT | **EVERYONE HAS PARTICIPATED EFFICIENTLY AND HAS PUT IN EFFORT TO COMPLETE THE WORK WITHOUT ANY CONFLICT.** |
| GITHUB AND JIRA UPDATION | **THE WORK IS REGULARLY UPDATED ON KANBAN BOARD, NO DELAYS IN ASSIGNED WORK COMPLETION** |
| TEAM-WORK | **EVERYONE TOOK PARTICIPATION IN COMPLETING THE PROJECT WITH FULL COLLABORATION.** |
|  |  |
|  |  |

**Things That Did NOT Go Well in This Meeting:**

This is where you can list things which did not go well in the class. You should analyze why this happened and suggest how you can improve it next time. This will lead to the goal of *continuous process improvement*.

|  |  |
| --- | --- |
| Topic/Work Item | Reason for Problem and How to do Better |
|  |  |

|  |
| --- |
| Everything went well, as far as Milestone 2 is concerned. |

|  |  |
| --- | --- |
|  |  |

**Reflection Questions:**

Answer the following questions using your own words. Make sure that each answer comprises a minimum of 100 words.

1. In this milestone you were asked to design the data structure for the project. Print the data structure below then explain each item.

In designing structures for the milestone, our group have only added “Truck” and “packageInfo” structures, rest were pre-defined.

Structures->

/\*\*

\* A point represents the row-column position of a square on a map.

\*/

struct Point

{

// Row (X coordinate) index of the point on map

int row;

// Column (Y coordinate) index of the point on map

int col;

};

/\*\*

\* A route is a collection of points that are adjacent to one another and constitute a path from the

\* first point on the path to the last.

\*/

struct Route

{

// Array of Point structure defines the route

struct Point points[MAX\_ROUTE];

// Number of points which make up a truck route

int numPoints;

// Character to specify the truck route ( B for Blue, Y for Yellow, G for Green )

char routeSymbol;

};

/\*\*

\* A Truck contains every details regarding a Truck, its ID, weight and destination.

\*/

struct Truck

{

// A unique ID for a truck

int truckID;

// Available weight capacity: 2500 kgs

float availableWeight;

// Available volume capacity: 100 c.mtr

float availableVolume;

// Number of shipments a truck can do

int allocatedShipments;

// Pointer which tracks the packages count for a truck

int\* packageCount;

// Character to specify the truck route ( B for Blue, Y for Yellow, G for Green )

char route;

// Stores each package information using array of structure

struct PackageInfo packages[];

};

/\*\*

\* A map is a 2D raster representation of a map with contents of the map encoded as numeric values.

\*/

struct Map

{

// 2D array representing the map squares

int squares[MAP\_ROWS][MAP\_COLS];

// Number of rows in the map

int numRows;

// Number of columns in the map

int numCols;

};

/\*\*

\* A PackageInfo is a detailed information of each box/package in the truck

\*/

struct PackageInfo {

// Weight of the package

double weight;

// Size of the package

double size;

// Destination point of the package

struct Point dest;

};

--------------------------------------------------------

a. Point: This structure is used to represent the coordinates of a point in a map, therefore it has fields like row and column, that store row and column coordinates.

b. Route: This structure represents a route that consists of multiple points.

i. “points” array stores each point coordinate in a route, which forms a route.

ii. numPoints stores the number of total points in a route and routeSymbol indicates the route type (B for Blue, Y for Yellow, G for Green).

c. Map: This structure represents the 2D map with the help of encoded contents

i. “squares” is the 2D array representing the map’s squares

ii. numRows and numCols, stores the number of rows and columns respectively.

d. packageInfo: This structure stores information about a package that gets loaded onto a truck.

i. Fields like weight and size stores the package weight and its size in cubic meters.

ii. “dest” is a structure that stores the coordinates of the destination of that package.

e. Truck: This structure contains detailed information about a truck.

i. truckId stores the truck’s unique ID

ii. availableWeight and availableVolume are used to check off many more weight or volumes can the truck withstand and helps to determine if we can load more packages

iii. packageCount stores the pointer to the number of packages in the truck

iv. allocatedShipment stores the number of shipments that are allocated to the truck

v. route indicates the truck’s route(B for Blue, Y for Yellow and G for Green.

1. Describe the process you used to analyze and understand the existing software code.

First, I began by skimming the entire codebase of the project to get an overview of the project’s structure and the other various files that are included in the project. This gave me a rough idea of what are the main components of the project.

After getting the overview, I started to read the available documentation for the project and all the comments that were available to me to understand the code. This clarified the purpose of each structure of the code and I got to know about their functionality. This includes reading all the structures that were predefined to me.

After I read all the documentation, I started tracing the function provided to us. I traced the key functions and understood how they interacted with the data structures like “Route”, “Point” and “Map”. This includes the flow of data inside the functions and their relationship with various components.

The major part that gave me more insights about the software code was real testing. I wrote various small test cases and used debugging tools to test the code. This approach helped me immensely to understand how to code and remove any areas of confusion.

For the last part, the unclear elements of the code were left, I shared those doubts with my team members and tried to gain their insights and perspectives on those issues. This approach helped me resolve my doubts and firm my understanding.

1. What aspects did you consider when creating the test plan? What were the milestones you identified in the test plan?

The aspects considered for the test plan are:

1. Project Objective: This ensures that the logistics routing system works without any fail and meets all the specified requirements
2. Scope: Scope of the test includes 2D map representation, route planning, real-time tracking, performance and user interface.
   1. Testing 2D map representation: Ensuring accurate representation and functionality of maps.
   2. Route planning: Validating the system's ability to plan efficient routes.
   3. Real-time tracking: Testing the accuracy and responsiveness of real-time tracking features.
   4. Performance: Assessing the system's performance under various conditions.
   5. User interface: Testing the usability and functionality of the user interface.
3. Test Strategy: Test strategy included some predefined and dynamic data and some various types of testing.
   1. Predefined and dynamic data: Testing with both fixed and variable inputs.
   2. Types of testing:
      1. Exploratory testing: Finding defects and areas for improvement.
      2. Functional testing: Validating specific functions of the system.
      3. System testing: Verifying the system as a whole.
      4. Performance testing: Assessing system response under load.
      5. Security testing: Ensuring data protection and system security.
      6. Automated testing: Using scripts to automate test scenarios.
      7. Stress testing: Evaluating system behavior under extreme loads.
      8. Volume testing: Checking system performance with large volumes of data.
      9. Recovery testing: Verifying system recovery after failures.
      10. Documentation testing: Ensuring completeness and accuracy of documentation.
      11. Beta testing: Testing in a real-world environment with end users.
      12. User acceptance testing: Validating user satisfaction and usability.
4. Test Design: Test design includes Collaboration with stakeholders, traceability matrix and detailed test cases.
   1. Collaboration with stakeholders involves stakeholders in test planning and execution.
   2. Traceability matrix requires to test cases to ensure coverage.
   3. Detailed test cases creates and reviews comprehensive test cases for thorough testing.
5. Environment Requirements: It ensures that the resources provided for hardware and software are sufficient for testing various aspects of the system
6. Execution Strategy: This included the “Entry and exit criteria” that establishes the condition for starting and stopping the test based on the defect’s severity and some other factors.
7. Control Procedures: This includes the systematic reviews, Bug tracking and defect reporting.
   1. Systematic reviews makes a review of every test results and its progress
   2. Bug tracking tracks and log every identified defects in the test.
   3. Defect reporting properly documents and reports every defect to facilitate resolution.

Milestones Identified:

* **Test Planning and Preparation:**
  + Start Date: July 10, 2024
  + End Date: July 15, 2024
* **Unit Testing:**
  + Start Date: July 16, 2024
  + End Date: July 23, 2024
* **Integration Testing:**
  + Start Date: July 24, 2024
  + End Date: August 1, 2024
* **System Testing:**
  + Start Date: August 2, 2024
  + End Date: August 15, 2024

These milestones outline the planned periods for each phase of testing, confirming a structured strategy to achieving project objectives within the established time frames.