
Ticket 8 Setup Environment

CS5213 Group B

Mar 28, 2024

CONTENTS

Table of Contents

Mar 28, 2024	1
CHAPTER.....	3
1.1 Overview	3
1.2 Scope	3
1.3 REFERENCES	4
CHAPTER.....	5
2.1 Resources Available	5
2.2 Steps to be Taken by Students	6
CHAPTER.....	7
3.1 Scrum Master	7
3.2 Developer	8
3.3 Code Reviewer	8
3.4 QA and TEST	9
3.5 PM (Group B Members).....	9
3.6 Steps for Creating and Managing Issues under Milestones in GitHub.....	10
3.7 Note on Snapshots.....	11
1. Go to the Issues tab.....	11
2. Click on the Milestones.....	11
3. Click on any milestone.....	12
4. Read the milestone description to create issues.....	13
13	
4. Click on New Issue 4.....	13

INTRODUCTION

1.1 Overview

This Project Plan serves as a comprehensive guide for managing and executing our development process through GitHub, a platform that facilitates collaboration, version control, and progress tracking. The document outlines a structured approach for the creation, assignment, and monitoring of tasks (issues) under specific milestones, which are pivotal for the systematic progression of the project. By adhering to the steps and procedures detailed in this plan, each team member can contribute effectively to the project's milestones, ensuring a cohesive and efficient workflow.

The document is designed to be accessible to all team members, regardless of their familiarity with GitHub, providing a clear roadmap for the tasks at hand. It aims to foster a culture of transparency, accountability, and continuous improvement, which are vital for the dynamic and collaborative environment of our project.

1.2 Scope

The scope of this Project Plan includes:

1. **Milestone Assignments:** Detailed allocation of milestones to various teams, laying the groundwork for task segmentation and focus.
2. **Issue Creation and Assignment:** Guidelines for creating detailed issues within GitHub, assigning them to appropriate team members, and linking them to their respective milestones. This ensures that every task is accounted for and monitored throughout its lifecycle.
3. **Progress Tracking:** A systematic approach for team members to update their progress on issues, thereby providing real-time visibility into the project's advancement and facilitating proactive problem-solving.

4. **Monitoring and Reporting:** Strategies for scrum masters and project managers to oversee the progress within each milestone, ensuring that the project stays on track and any impediments are quickly addressed.

By following the structured approach outlined in this Project Plan, the team is equipped to navigate the complexities of the project, ensuring that each phase is executed with precision and in alignment with the overall objectives. This document is instrumental in guiding the team through the iterative process of development, fostering a disciplined yet flexible environment that adapts to the evolving needs of the project.

1.3 REFERENCES

[GroupB-Ticket3-SRS](#)

[GroupB-Ticket2-SQAP](#)

[GroupB-Ticket1-SDP](#)

[GroupB-Ticket5-STP](#)

[GroupB-Ticket7-SDD](#)

CHAPTER TWO

PLAN EXECUTION

The provided information in this section outlines a comprehensive approach for students engaged in a project, detailing the resources available, steps for GitHub setup, learning resources, role assignments, sprint and milestone management, along with communication and reporting protocols. This framework is designed to ensure that all students are well-informed, effectively organized, and collaboratively working towards the successful completion of the project. By adhering to this structured plan, students can enhance their understanding of GitHub, improve their collaborative skills, and contribute significantly to their team's progress.

2.1 Resources Available

GitHub Repository: Access the main codebase and documentation [here](#).

Scrum Tickets: Track and manage tasks [here](#).

Milestones: View project milestones [here](#).

Learning Resources:

- Interactive Git Learning: [Learn branching with Git](#).
- YouTube Git Tutorial Series: [Watch and learn Git](#).

Communication Channel: Join the Discord channel [here](#).

2.2 Steps to be Taken by Students

1. GitHub Setup:

- Ensure GitHub account creation and repository access.
- Fill the GitHub usernames in 3.2 Student list for records.
- Create individual branches using GitFlow.
- Implement role-based access control within GitHub.

2. Learning and Preparation:

- Engage with Git learning resources.
- Assess skills for potential code reviewer roles.

3. Role Assignments and Deliverables:

- Identify and assign roles within teams.
- Specify deliverables for each role, including merge requests, test reports, and progress reports.

4. Sprint and Milestone Management:

- Participate in and contribute to sprint demos.
- Engage in creating, assigning, and managing tasks within milestones.

5. Communication and Reporting:

- Maintain direct communication with Group B and team members.
- Utilize the designated communication platform for updates and coordination.

6. Miscellaneous:

- Acknowledge the autonomy of teams in selecting their scrum masters.
- Note the assigned and future milestones based on performance.
- Highlight the scrum master's responsibility in issue tracking and task breakdown.

By following these guidelines, students will be equipped to navigate their project with clarity and purpose, ensuring a cohesive and productive team environment

CHAPTER THREE

TEAM ROLES AND DELIVERABLES

The document outlines the key roles and their associated deliverables within a project team, emphasizing the importance of each position in the project's success. The Scrum Master is responsible for maintaining communication and leading the team, producing regular progress reports. Developers focus on coding and integration, contributing through pull requests and quality code. Code Reviewers ensure the standard of the code through detailed merge request reports. QA and Test personnel play a critical role in quality assurance, documenting their findings in test reports. Lastly, the PM oversees the overall project progress, coordinating between teams, managing deployments, and compiling essential documentation. Each role's deliverables are integral to the project's workflow, emphasizing the need for clear communication and meticulous attention to detail to ensure the project's objectives are met efficiently and effectively.

3.1 Scrum Master

Requirements:

- Strong communication skills.
- Strong leadership skills.
- Attend all team meetings and report progress.

Responsibilities:

- Act as the primary communication link within the team and with external stakeholders.
- Facilitate decision-making processes to distribute work and resolve issues.
- **Deliverable:**
- Scrum/Progress Report: Regular updates on team progress, challenges, and achievements.

3.2 Developer

Requirements:

- Experience in web development.

Responsibilities:

- Develop code and features for the project.
- Collaborate with team members to integrate different components.

Deliverables:

- Pull Requests: Submit code for review and integration into the main branch.
- Code: Contribute high-quality, well-documented code to the project repository.

3.3 Code Reviewer

Requirements:

- Veteran experience in web development.

Responsibilities:

- Review code submissions for quality, consistency, and alignment with project standards.

- **Deliverable:**
- Merge Request Reports: Provide detailed feedback on the code quality, suggesting improvements and approving merges.

3.4 QA and TEST

Requirements:

Heavy availability Thursday & Friday

Responsibilities:

- Conduct thorough testing to identify bugs and issues in the software.
- Collaborate with developers to ensure the quality of the codebase.

Deliverable:

- Test Reports: Document the testing processes, findings, and recommendations for various branches.

3.5 PM (Group B Members)

Responsibilities:

- Oversee the project's progress, ensuring adherence to timelines and quality standards.
- Coordinate among different teams to facilitate smooth execution of project phases.

Deliverables:

- Meeting Minutes and Progress Reports Aggregation: Compile and synthesize reports from each team.
- Sprint Execution and Code Deployment: Manage the deployment processes, ensuring timely and successful releases.

- **Merging Test Reports:** Integrate findings from different teams to provide a comprehensive quality overview.
- **Documentation Compilation:** Gather and organize all relevant project documentation for easy access and reference.

Each role is crucial for the smooth operation and success of the project, requiring clear communication and dedicated execution of responsibilities. The deliverables associated with each role will be graded, emphasizing the importance of thoroughness and quality in every aspect of the project.

3.6 Steps for Creating and Managing Issues under Milestones in GitHub

Step 1: Understanding the Milestone Assignments

- Each team has been assigned a specific milestone by Group B:
- Group A: Milestone M1
- Group D: Milestone M4
- Group C: Milestone M5
- Group E: Milestone M3
- Group F: Milestone M2
- Future milestones will be assigned based on team performance.

Step 2: Creating Milestones

- If the milestones are not already created, the scrum master or a designated team member should create them in the GitHub repository.
- Go to the "Issues" tab in your repository, then click on "Milestones" and "New Milestone" to create each milestone.

Step 3: Creating Issues for Each Milestone

- Under the "Issues" tab, click "New Issue."
- Title the issue with a clear, concise description of the task.
- In the issue description, provide detailed information about the task, including any specific requirements or objectives.
- Assign the issue to the relevant team member.

- Link the issue to the corresponding milestone by selecting the appropriate milestone from the dropdown menu on the right-hand sidebar.

Step 4: Tracking Progress

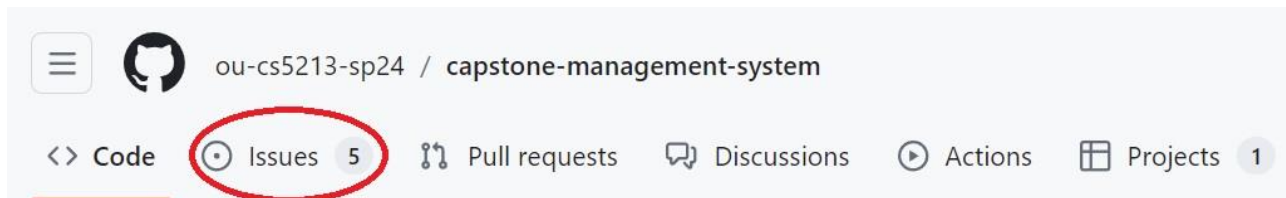
- Team members assigned to an issue should update their progress by commenting on the issue and changing its status:
- Move the issue from "Open" to "In Progress" when work begins.
- Once the task is completed, close the issue to mark it as "Completed."

Step 5: Monitoring and Reporting

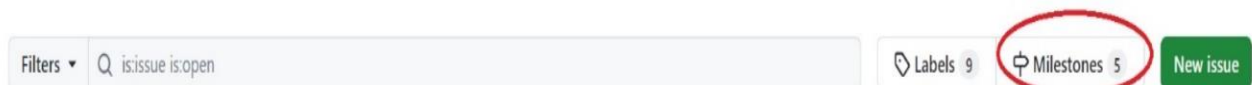
- The scrum master or project manager should monitor the progress of issues within each milestone.
- Regular updates should be provided to the team, highlighting completed tasks, ongoing work, and any blockers.

3.7 Note on Snapshots

1. Go to the Issues tab



2. Click on the Milestones.



3. Click on any milestone.

Labels

Milestones

5 Open 0 Closed

M5: Automated CI/CD Deployment Report

Due by April 05, 2024

Last updated about 17 hours ago

4.5.1 CI/CD Pipeline Implementation Configuration of GitHub Actions ...(more)

0% complete

0 open

0 closed

Edit Close Delete

M4: ORM Objects Definition Report

Due by April 05, 2024

Last updated about 17 hours ago

ORM Objects Definition Python Class Development Data Model Design ...(more)

0% complete

0 open

0 closed

Edit Close Delete

M3: Test Environment Setup

Due by April 05, 2024

Last updated about 17 hours ago

Test Environment Setup Automated Cloud/Server Configuration Automa...(more)

0% complete

0 open

0 closed

Edit Close Delete

M2: Production Environment Setup

Due by April 05, 2024

Last updated about 17 hours ago

Production Environment Setup Cloud/Server Configuration Cloud Serv...(more)

0% complete

0 open

0 closed

Edit Close Delete

M1: Project Setup

Due by April 05, 2024

Last updated about 18 hours ago

The setup involved the following key steps: Django Project Initial...(more)

0% complete

1 open

0 closed

Edit Close Delete

4. Read the milestone description to create issues.



M4: ORM Objects Definition Report

📅 Due by April 05, 2024 0% complete

ORM Objects Definition

Python Class Development

1. Data Model Design

- Translation of the application's requirements into a comprehensive data model.
- Identification of entities such as Users, Projects, Teams, Submissions, and Evaluations, and their respective attributes and relationships.

2. Python Class Creation

- Creation of Python classes that extend Django's models.Model class, with fields and behaviors of the data stored in the system.
- Use of Django field types such as CharField, IntegerField, ForeignKey, etc., to define the properties of each model.

3. Relationship Mapping

- Definition of relationships between models using ORM relationships such as ForeignKey, ManyToManyField, and OneToOneField.
- Ensuring referential integrity and the efficient retrieval of related data through Django's query syntax.

Database Migration Generation

4. Migration Scripts

- Use of Django's makemigrations command to automatically generate migration scripts based on changes to the models.
- Review and, if necessary, manual editing of migration scripts to fine-tune database schema changes.

5. Click on New Issue 4.



M4: ORM Objects Definition Report

📅 Due by April 05, 2024 0% complete

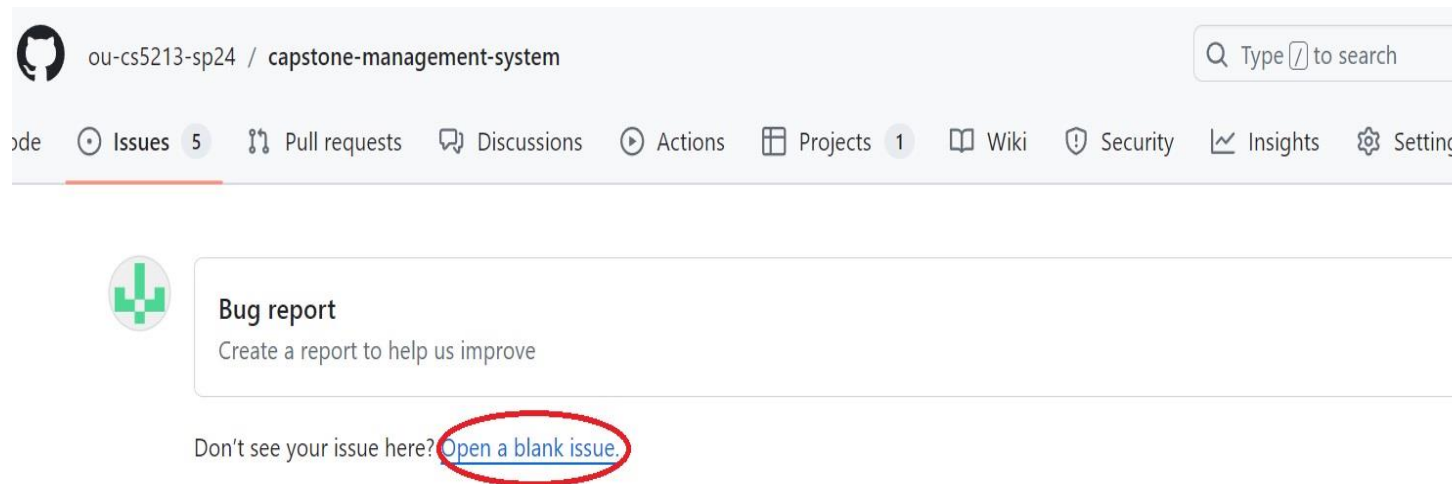
ORM Objects Definition

Python Class Development

Edit milestone

New issue

6. Click on 'Open a Blank Issue'.



7. Fill the details in the below form and click on 'Submit new Issue'. This creates your scrum ticket.

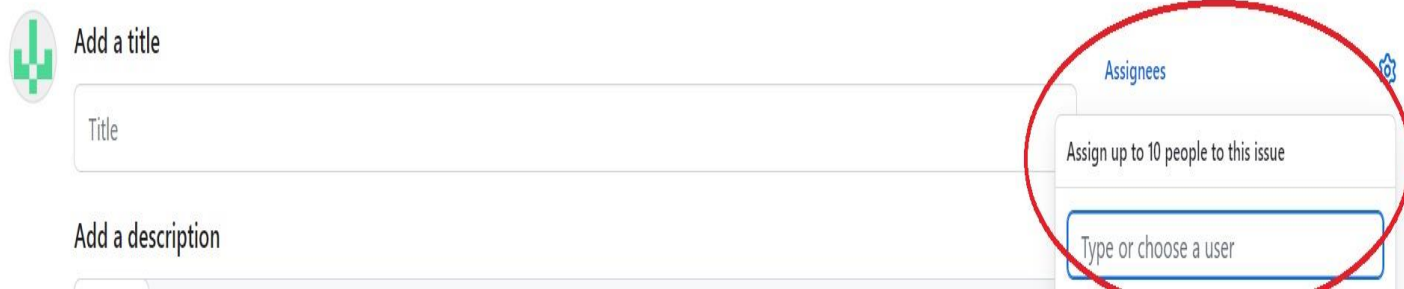
The screenshot shows the GitHub 'New Issue' form. The 'Add a title' section has the text 'Create ORM definitions'. The 'Add a description' section has a text area with the following content:

Deliverables

1. Deliver the ORM Definitions
2. Get it verified

At the bottom right, a green button labeled 'Submit new issue' is circled in red.

8. In the right side of the above you can assign the issue to your teammates.



The image shows a Jira issue creation form. On the left, there is a green icon with a white cross. To its right, the text "Add a title" is followed by a text input field containing the placeholder "Title". Below this, the text "Add a description" is followed by a text area. On the right side of the form, there is a section titled "Assignees" with a gear icon. Below the title, it says "Assign up to 10 people to this issue". At the bottom of this section is a text input field with the placeholder "Type or choose a user". A red circle highlights the "Assignees" section.