1. **Introduction**

This project aimed to develop a newspaper delivery system that could manage daily publication deliveries, track inventory, handle customer schedules, and generate monthly billing. Our group chose to use **Scrum** as the Agile framework because it allowed us to organize work into manageable sprints and adapt to changing requirements. The project was essential for understanding how to collaboratively design and implement software in a structured and iterative manner.

* Provide a brief description of the newspaper delivery system and its purpose.
* State the chosen Agile framework (e.g., Scrum, Kanban, or a tailored Agile method) and why it was selected.
* Mention the assignment's importance and the collaborative nature of the project.

**2. Team Dynamics and Roles**

* Explain how responsibilities were distributed (e.g., one member focused on customer billing, another on geographic information management).
* Reflect on how effectively the team worked together.

**3. Sprint/Iteration Summary**

* Describe how the work was divided across sprints or iterations.
* Summarize key deliverables for each sprint, such as:
  + Sprint 1: set up sql database and developed creates for order invoive delivery docket .
  + Sprint 2: Basic design and stock management implementation.
  + Sprint 3: Geographic routing and daily delivery printouts.
  + Sprint 4: Billing and holiday scheduling features.
* Highlight achievements and areas where adjustments were needed.

**4. Agile Practices and Tools**

* Evaluate how these practices/tools contributed to project management.
* **Weekly Stand-ups**: Conducted during lab sessions to align on sprint goals and address issues.
* **Sprint Planning and Task Management**: Divided tasks based on expertise and tracked them using a shared task plan.
* **Version Control**: Used GitHub to ensure organized collaboration and code integration.

**5. Successes and Strengths**

* Detail areas where the team excelled, such as:
  + Identifying clear user stories and aligning them with the assignment requirements.
  + Successfully implementing the billing system or geographic routing.
* Mention strategies that worked well, like breaking tasks into manageable user stories or effective sprint planning.

**6. Challenges Faced**

* Describe obstacles, for example:
  + Managing version control across team members.
  + Coordinating holiday schedules for customers in the system.
  + Ensuring the system's usability for casual computer users (newsagents).
* Reflect on how the team addressed these issues and what could be improved.

**7. Individual Contributions**

* Reflect on your personal contributions, such as:
  + Writing user stories or designing specific components (e.g., the inventory system).
  + Coding and testing specific features like monthly billing or geographic sorting.
* Mention how Agile practices helped you stay organized and efficient.

**8. Lessons Learned**

* Highlight key insights from the project:
  + How Agile improved team collaboration and adaptability.
  + Importance of user-focused design, especially for non-technical users like newsagents.
  + Value of breaking complex systems into smaller, testable features.

**9. Conclusion**

* Summarize the project outcomes, including how well the final system met the requirements.
* Reflect on the overall effectiveness of the Agile methodology and what your group might do differently in future projects.
* Optionally, acknowledge the effort and contributions of the entire team.

**10. Appendix (Optional)**

* Include supporting materials:
  + Examples of user stories or a sprint backlog.
  + Screenshots of the system interface or sample outputs (e.g., daily delivery printouts).
  + Testing documentation or sprint retrospective notes.

Here’s a polished and improved version of the Agile reflection, combining the provided content into a cohesive structure:

**Agile Group Reflection: Newspaper Delivery System**

**1. Introduction**

Our project focused on developing a **newspaper delivery system** to help a newsagent manage their operations, including organising deliveries, tracking inventory, handling customer schedules.

We chose the **Agile methodology**, specifically the **Scrum framework**, because it allowed us to break the project into manageable sprints, adapt to changes, and address challenges collaboratively. This approach ensured we could meet the requirements while improving teamwork and communication.

This assignment highlighted the importance of designing and implementing software in a structured and collaborative manner, especially for non-technical users like newsagents.

**2. Team Dynamics and Roles**

Our team consisted of **Padraig, Ben, Michael, and Liam**, with each member taking on specific responsibilities based on their strengths.

* **Ben**: Focused on SQL setup, ER diagrams, and the Publications class. He also handled JUnit testing and connected the class to the main menu in the final sprint.
* **Liam**:
* **Michael**:
* **Padraig**:
* Although the team had a slow start, we quickly adapted to Agile practices and improved our collaboration. Responsibilities were distributed by class, with each member developing, testing, and integrating their components. This structure allowed us to work independently while aligning our efforts during sprints.

Here’s a revised **Sprint Summary** section reflecting what the group completed collectively in each sprint:

**3. Sprint Summary**

Our work was divided into four two-week sprints, with tasks assigned and tracked collaboratively using GitHub. Each sprint focused on specific objectives:

* **Sprint 1: Planning and Initial Setup**
  + Collaboratively created user stories and acceptance criteria for all project entities.
  + Designed test cases for initial system components.
  + Set up the SQL database, including tables and ER diagrams, ensuring all team members had access to the shared environment.
  + Developed the initial "create" functionality for each class (Customer, Order, Invoice, Deliveries, and Publications).
  + Organized the project repository and established version control workflows on GitHub.
* **Sprint 2: Read Functionality and Design Refinement**
  + Implemented the "read" functionality for all entities, ensuring data retrieval worked as expected.
  + Reviewed and refined user stories to align with system requirements.
  + Created UML diagrams for Java classes and updated the database design for improved compatibility.
  + Began addressing exception handling to make the system more robust.
* **Sprint 3: Delete Functionality and Optimization**
  + Added "delete" functionality across all entities, enabling removal of records from the system.
  + Conducted code reviews to ensure consistency and identify opportunities for optimization.
  + Tidied up code structure, naming conventions, and repository organization to improve maintainability.
* **Sprint 4: Update Functionality and Finalization**
  + Implemented the "update" functionality for all entities, completing the CRUD operations.
  + Ben connected the Publications class to the main menu, while Liam developed the main menu interface.
  + Created and executed JUnit tests to validate system functionality.
  + Conducted final system testing and prepared for the project demo, ensuring all components were integrated and operational.

Each sprint concluded with a review of progress and adjustments to the backlog, ensuring the project remained on track and aligned with the assignment requirements.

**4. Agile Practices and Tools**

Several Agile practices and tools contributed to the project’s success:

* **Weekly Stand-ups**: Conducted during lab sessions to discuss progress, blockers, and upcoming tasks.
* **Sprint Planning**: Tasks were divided based on expertise and tracked using a shared task plan.
* **Version Control**: GitHub was used to manage code changes, track progress, and resolve conflicts.
* **Discord**: Used for asynchronous communication and quick issue resolution outside lab sessions.

These practices ensured transparency, adaptability, and effective collaboration throughout the project.

**5. Successes and Strengths**

The team excelled in several areas:

* **Motivation and Time Management**: Tasks were completed efficiently, with regular updates ensuring everyone was on track.
* **Clear User Stories**: Well-defined stories aligned with project requirements and provided a solid foundation for development.
* **Task Ownership**: Each member took responsibility for their assigned classes, contributing to steady progress.
* **Team Collaboration**: Regular communication through weekly meetups and Discord ensured smooth integration of components.

**6. Challenges Faced**

Despite our successes, we encountered challenges:

* **Version Control Conflicts**: Early issues with GitHub merges were resolved by establishing clearer guidelines for pushing and pulling code.
* **Estimating Task Durations**: Some tasks, like connecting scripts to the main menu, took longer than anticipated due to dependencies.
* **System Usability**: Designing an intuitive system for non-technical users required extra effort in testing and feedback.

These challenges taught us valuable lessons about planning, coordination, and user-centric design.

**7. Individual Contributions**

Each member contributed to the project’s success:

* **Ben**: Set up SQL, created ER diagrams, implemented Publications, and wrote JUnit tests.
* **Liam**: Developed Invoice and Deliveries classes, and created the main menu.
* **Michael**: Developed the Order class and contributed to testing and debugging.
* **Padraig**: Developed the Customer class and ensured the repository was well-organized.

Collaborative tasks, like creating user stories and testing, were shared among all members, showcasing our collective effort.

**8. Lessons Learned**

Key takeaways from the project include:

* **Agile Methodology**: Improved team collaboration, adaptability, and delivery of features incrementally.
* **Task Breakdown**: Dividing the project into smaller, testable tasks made development more manageable.
* **Communication**: Regular stand-ups and Discord communication were crucial for resolving blockers.
* **User-Focused Design**: Emphasizing usability ensured the system met the needs of non-technical users like newsagents.

**9. Conclusion**

The final system successfully met the project requirements, demonstrating the effectiveness of our Agile approach. Through teamwork and iterative development, we delivered a functional, user-friendly system.

For future projects, we aim to improve task estimation and dependency planning to further enhance our workflow. The contributions and dedication of the entire team were instrumental in achieving our goals.

**10. Appendix**

* **Task Plan**: Detailed breakdown of tasks, status, and assigned team members.
* **User Stories**:
  + As a newsagent, I want an automated billing system to simplify monthly invoices.
  + As a delivery person, I want delivery routes ordered geographically to save time.
  + As a customer, I want to suspend deliveries when on vacation.
* **Screenshots**: Sample outputs and GUI mockups.
* **Testing Report**: Summary of JUnit test results.

This version integrates all provided inputs while refining the structure and language for clarity and professionalism. Let me know if further adjustments are needed!