**Epics (Large Work Units)**

1. **Project Setup and Tools**
2. **Requirement Gathering and Definition**
3. **Backend Development**
4. **Frontend Development**
5. **Integration and Testing**
6. **Delivery and Documentation**
7. **Reflection and Final Adjustments**

**User Stories and Tasks**

**Epic 1: Project Setup and Tools**

* **Story 1.1: As a developer, I want to set up the project environment to ensure efficient collaboration.**
  + **Task 1.1.1: Create a GitHub repository and set up version control.**
  + **Task 1.1.2: Define folder structure and coding standards.**
  + **Task 1.1.3: Configure a virtual environment and install libraries.**
  + **Task 1.1.4: Create project in Jira**

**Epic 2: Requirement Gathering and Definition**

* **Story 2.1: As a development team, we want to collect client requirements to better understand the project scope.**
  + **Task 2.1.1: Schedule and conduct a requirement gathering meeting.**
  + **Task 2.1.2: Create a list of features and validate them with the client.**
  + **Task 2.1.3: Create the initial product backlog in Jira.**

**Epic 3: Backend Development**

* **Story 3.1: As a developer, I want to design the backend architecture to ensure a scalable system.**
  + **Task 3.1.1: Define the architecture (MVC, OO, etc.).**
  + **Task 3.1.2: Set up the database and create necessary tables.**
  + **Task 3.1.3: Implement backend-to-database connection.**
* **Story 3.2: As a user, I want to authenticate in the system to access restricted features.**
  + **Task 3.2.1: Create authentication endpoints (login, logout, registration).**
  + **Task 3.2.2: Implement JWT for secure authentication.**
* **Story 3.3: As a user, I want to manage my data in the system.**
  + **Task 3.3.1: Develop CRUD operations for users.**
  + **Task 3.3.2: Develop CRUD operations for another key entity.**

**Epic 4: Frontend Development**

* **Story 4.1: As a user, I want to access a user-friendly interface to interact with the system.**
  + **Task 4.1.1: Choose a frontend framework (Tkinter, PYQT, Flask, etc.).**
  + **Task 4.1.2: Define basic components (frames, buttons, navigation).**
  + **Task 4.1.3: Implement frontend-backend integration.**

**Epic 5: Integration and Testing**

* **Story 5.1: As a developer, I want to perform testing to ensure system quality.**
  + **Task 5.1.1: Write unit tests for the backend.**
  + **Task 5.1.2: Write UI tests for the frontend.**
  + **Task 5.1.3: Run automated tests in the CI/CD pipeline.**

**Epic 6: Delivery and Documentation**

* **Story 6.1: As a project team, we want to document the system to facilitate future maintenance.**
  + **Task 6.1.1: Create documentation such as DER, UML Diagrams, etc.**
  + **Task 6.1.2: Write a user manual.**
  + **Task 6.1.3: Prepare a presentation for project delivery.**

**Epic 7: Reflection and Final Adjustments**

* **Story 7.1: As a team, we want to reflect on project challenges to improve future processes.**
  + **Task 7.1.1: Write a retrospective report.**
  + **Task 7.1.2: Make final adjustments based on feedback.**
  + **Submit files and documents**

**SPRINTS:**

**Sprint 1 (Week 1): Initial Setup and Requirements**

🎯 **Objective:** Set up the environment, define requirements, and prepare the backlog

**Tasks:**

✅ Create Git repository and configure CI/CD  
✅ Define project architecture (backend and frontend)  
✅ Create backlog in Jira with Epics, Stories, and Tasks  
✅ Conduct meetings to gather requirements  
✅ Set up the initial database structure

**Deliverables:**

📌 Configured repository  
📌 Initial backlog in Jira  
📌 Requirements document

**Sprint 2 (Week 2): Authentication and Basic Backend**

🎯 **Objective:** Implement authentication and the first backend functionality

**Tasks:**

✅ Create login, logout, and user registration endpoints  
✅ Implement authentication via JWT  
✅ Develop basic CRUD for a main entity  
✅ Write unit tests for authentication  
✅ Test API using Postman

**Deliverables:**

📌 Backend running with JWT authentication  
📌 Functional CRUD for one entity  
📌 Basic unit tests

**Sprint 3 (Week 3): Basic Frontend Development**

🎯 **Objective:** Build the initial interface and integrate it with the backend

**Tasks:**

✅ Choose a framework (React, Vue, etc.)  
✅ Create the initial frontend structure  
✅ Implement a login screen integrated with the backend  
✅ Develop initial UI components  
✅ Write unit tests for components

**Deliverables:**

📌 Functional login screen  
📌 Frontend communicating with the backend  
📌 Tests for main components

**Sprint 4 (Week 4): Core Features and Integration**

🎯 **Objective:** Implement core functionalities and enhance UI/UX

**Tasks:**

✅ Develop additional pages and UI components  
✅ Implement full CRUD functionality via the frontend  
✅ Improve interface (responsiveness, UX)  
✅ Conduct frontend-backend integration tests  
✅ Begin user testing

**Deliverables:**

📌 Fully functional CRUD accessible via frontend  
📌 Improved user interface  
📌 Automated integration tests

**Sprint 5 (Week 5): Testing, Refactoring, and Optimisation**

🎯 **Objective:** Improve code, fix bugs, and optimise the system

**Tasks:**

✅ Review and refactor code  
✅ Conduct performance and load testing  
✅ Fix critical bugs and inconsistencies  
✅ Enhance API documentation

**Deliverables:**

📌 Optimised and reviewed code  
📌 Critical bugs resolved  
📌 Updated technical documentation

**Sprint 6 (Week 6): Documentation, Presentation, and Final Delivery**

🎯 **Objective:** Finalise the project and prepare for delivery

**Tasks:**

✅ Create system report and documentation  
✅ Prepare final presentation  
✅ Write retrospective report  
✅ Perform final review and adjustments

**Deliverables:**

📌 Complete documentation  
📌 Final presentation  
📌 Last adjustments to the system

**Sprint Review and Retrospective (End of Week 6)**

* **Sprint Review:** Present the system and validate it with the client
* **Sprint Retrospective:** Discuss improvements and key learnings