|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PROJECT / GROUP NAME** | **Group 14** | | | |
| **Start Date** | 2024.04.08 | **Finish Date** | 2024.07.01 | |
| **Aim /**  **Objective** | Front-end interactive function logic implementation: Carry out code development in parallel and implement front-end functions and interfaces according to project design documents and architectural design. This work will realize the functions of user system, owner resource management, and community employment platform. Combine architectural design and back-end development to ensure interface consistency. | | | |
| **Work package**  **Manager** | Qi Te | | | |
| **Contributors to this package** | Sun Ruotong  Liu Xinran  Wang Shizheng  Liu Huiyang  Zhang Juncheng | | | |
| **Description /**  **Activities** | Task 4.1 Implement interface   * 4.1.1 Design request interception rules and add request header information appropriately * 4.1.2 Complete the front-end request api according to the interface document   Task 4.2 Implement user system functions   * 3.2.1 Split user-related functionality into appropriate components * 3.2.2 Draw page elements according to design draft * 3.2.3 Realize the functions of each component * 3.2.4 Test functions such as user registration and login   Task 4.3 Realize owner resource management function   * 4.4.1 Obtain owner-related content data * 4.4.2 Use charts to demonstrate * 4.4.3 Call the artificial intelligence module to analyze some data and finally give reasonable suggestions.   Task 4.4 Realize community employment platform function   * 4.4.1 Obtain existing recruitment information * 4.4.2 Display recruitment information * 4.4.3 Collect job application information filled in by users and send it to the backend   Task 4.5 Optimize completed code   * 4.5.1 Some code may need to be refactored   Task 4.6 Joint debugging of front-end and back-end   * 4.6.1 Accept errors reported by testers and analyze whether they are front-end code problems * 4.6.2 Assist with backend staff to fix bugs | | | |
| **Milestones** |  | | | Week |
| M 4.1 Normally around week 4/5  M 4.2 Normally around week 6/7  M 4.3 Normally for Interim report (week 9): What can the overall application do now? How has this work package contributed so far?  M 4.4 Normally around week 11  M 4.5 Normally around week 13  M 4.6 Final Release | | | **x**  **x**  **x**  **x**  **x**  **x** |
| **Deliverables** |  | | | Week |
| D 4.1 These are deliverables – deliverables are normally pieces of code (complete) that are integrated into the overall project  D 4.2 Normally deliverables coincide (happen at the same time) with milestones, but not every milestone has to have a deliverable. For instance, if a milestone is testing a work package code, there may be no deliverable. However if a milestone involves integrating a work package code with the overall project, that would be a deliverable. | | | **x**  **x** |