|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PROJECT / GROUP NAME** | **Group 14** | | | |
| **Start Date** | 2024.03.18 | **Finish Date** | 2024.04.07 | |
| **Aim /**  **Objective** | Architecture and interface design: Connect with the project design department, review project design-related documents from the perspective of the development team, and discuss the feasibility of solutions and functions and more details. Design the connections and details of each part of the entire software from the architectural level to ensure high cohesion and low coupling of each part of the software, while maintaining high scalability to cope with changing needs. Provide complete interface documents and database design to connect subsequent code development processes and ensure the feasibility of architectural design. | | | |
| **Work package**  **Manager** | Sun Ruotong | | | |
| **Contributors to this package** | Liu Xinran  Wang Shizheng  Qi Te  Liu Huiyang  Zhang Juncheng | | | |
| **Description /**  **Activities** | Task 2.1 Check function point feasibility   * 2.1.1 Re-read the existing feature list * 2.1.2 Consider the feasibility and necessity of function points from the perspective of a software engineer * 2.1.3 Re-discuss unclear or controversial functional points   Task 2.2 Mapping functional descriptions to software terminology   * 2.2.1 Find all "terms" that have a specific meaning in the project and explain them * 2.2.2 The description of different functions is further explained as an engineering description. * 2.2.3 Use tools like flowcharts to describe functions with complex interaction logic   Task 2.3 Design interface documentation   * 2.3.1 Design interfaces for user registration, login, etc. * 2.3.2 Design the interface for article display * 2.3.3 Design interfaces for other function points   + 2.3.3.1 Find out the data involved in the function   + 2.3.3.2 Map to variables corresponding to the front and back ends * 2.3.4 Organize interfaces and find identical or similar interfaces to merge and split   Task 2.4 Design database   * 2.4.1 List the data required for each function point * 2.4.2 Combine identical data and assign easy-to-understand and unique variable names to the data * 2.4.3 Abstract data relationships and create entity relationship diagrams * 2.4.4 Generalize ER diagrams into relational database tables * 2.4.5 Simple test database availability and reliability   Task 2.5 Technology stack selection   * 2.4.1 Determine the appropriate technology stack selection based on the knowledge mastered by team members and combined with project functional requirements. | | | |
| **Milestones** |  | | | Week |
| M 2.1 Normally around week 4/5  M 2.2 Normally around week 6/7  M 2.3 Normally for Interim report (week 9): What can the overall application do now? How has this work package contributed so far?  M 2.4 Normally around week 11  M 2.5 Normally around week 13  M 2.6 Final Release | | | **x**  **x**  **x**  **x**  **x**  **x** |
| **Deliverables** |  | | | Week |
| D 2.1 These are deliverables – deliverables are normally pieces of code (complete) that are integrated into the overall project  D 2.2 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** |