**Project Summary Report**

Date：17-06-2023

|  |  |  |  |
| --- | --- | --- | --- |
| Group | 11 | 项目名称 | XingQiao |
| Programming language | Javascript, Python, Java, SQL | Development Platforms and Frameworks | Visual Studio Code**,** React.js , Spring , Flask |

|  |
| --- |
| **Project work summary** |
| 1. Have all the requirements at the time of project approval been fulfilled? List the new requirements implemented and the unfulfilled requirements.   new requirements: None  unfulfilled requirements: More Unittests for Java and Js, More Unittests for Java and Js, Increase front-end usability and navigability   1. Which architectural style have you adopted? Which design patterns?   We have adopted 3-tier client-server layered architectural style. We used following design patterns:  Creational: Abstract Factor (Java, Python), Singleton (Spring Boot)  Structural: Composite (React)  Behavirol: State (React), Decorator (Python)  Concurrency: Async/Await(React)   1. What are the highlights of the technical solution?   **React Front-End:** React's component-based architecture has enabled the creation of a dynamic and responsive user interface. This provides a seamless user experience and allows for efficient reusability of components.  **Java Spring Boot Back-End:** The use of Spring Boot has simplified the development process by providing a range of preconfigured features and integrations. This has expedited the development process and ensured the application is robust and secure.  **Python Flask Services:** Flask's lightweight and flexible nature has been leveraged to create efficient microservices for specific tasks. This has contributed to the modularity and scalability of the system.  4. Have you done unit testing? Have you done a system function test? Has performance testing been done? Have you done other non-functional tests such as compatibility?  We have done Unittesting for Python micro-services. Other test cases need to be implemented later. |
| **Contribution of project team members to the project (%)** |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | Name | Requirements | Design | Code | Test | Project Management | Subtotal | | VAHAGN | 70 | 60 | 25 | 40 | 55 | 50 | | SIMON | 30 | 40 | 75 | 60 | 45 | 50 | |  |  |  |  |  |  |  | |  |  |  |  |  |  |  | |

|  |  |
| --- | --- |
| **software metrics** | |
| Number of software code lines (excluding comment lines, blank lines and reused codes)： | - |
| Reuse third-party code lines： | - |
| number of classes： | 18 |

|  |
| --- |
| **Experiences, Lessons and Recommendations** |
| 1. Time Management: Starting earlier allows for more time to handle unexpected challenges. 2. Teamwork: Communication is a key. 3. Planning: Detailed planning can help identify tasks and be specific. Dedicate good chuck of time to it. 4. Learning Curve: New technologies require time to learn. 5. Testing: Regular testing helps catch issues early. | |

Signatures of each member of the project team: