## *C)* Original Requirement Set For Case Study 1 *CrowdRequire* (Adepetu, Ahmed, Al Abd, Zaabi, & Svetinovic, 2012)

|  |  |
| --- | --- |
| **No** | **Description** |
|  | **Functional Requirements** |
| 1 | The system shall be web-based. |
| 2 | The system shall accept and register users. |
| 3 | The system shall provide substantial information for users. |
| 4 | The system shall require employee-related data and business-partner-related data from crowd members and clients respectively. |
| 5 | The system shall contain crowd-member-specific tabs on the crowd member profile, i.e., where the crowd member can view details of tasks, search for tasks, upload requirements specifications, and update account information and inquiries. |
| 6 | The system shall contain client-specific tabs on the client profile, i.e., task description, task management, task submissions, task selections and inquiries. |
| 7 | The system shall have a messaging function. |
| 8 | The system shall provide users with timely updates on task suggestions and changes. |
| 9 | The system shall enable clients to define and customize task templates. |
| 10 | The system shall enable clients to define criteria and tests for selection of crowd members. |
| 11 | The system shall inform the client the number of crowd members that are eligible for tasks based on specified task criteria. |
| 12 | The system shall contain task description tools, such as a text editor, tables, images (upload), video (upload), numbering and bullets, and document upload. |
| 13 | The system shall enable the client to modularize a task into subtasks with corresponding payments, with the system showing the percentage of the payment that goes to CrowdREquire. |
| 14 | The system client shall store a history of all users’ task details. |
| 15 | The system shall require payments from a client before a task can be published to the crowd. |
| 16 | The system shall enable clients to buy CrowdREquire credits. |
| 17 | The system shall have an Application Programming Interface (API) for developing custom tasks. |
| 18 | The system shall have a ranking system for users. |
| 19 | The system shall have desktop and mobile applications where users can work offline. |
|  | **Non-Functional Requirements** |
| 20 | The system shall be usable on all browser and operating system platforms. |
| 21 | The system shall have a user-friendly and simple interface. |
| 22 | The system shall be readily available for usage. |
| 23 | The system shall be robust in order to accommodate as many users as possible at a time without the system performance being affected. |
| 24 | The system shall be efficient in error handling and prevent loss of data. |
| 25 | The system shall be secure. |
| 26 | The system shall have flexible design tools. |
| 27 | The system shall ensure and maintain a high-level of integrity among users. |
| 28 | The system shall address any form of abuse or breach of legal conditions. |