Project Plan and RMMM

Ahmed Al-Faris, Nathaniel Leonardo, Nidhi Patel, Zachary Wireman

**1.0 Introduction**

**1.1 Project scope**

This software will take and store graduate photos, names, majors, and years of each graduated student and store them in a database. The database will be accessible through a web page stored on a kiosk in the ELB, and will contain views for each year of graduates, as well as a search by name. Through the kiosk, people will be able to input additional information for graduates, such as accomplishments post-graduation. This information will be approved by a system administrator before it is visible on the kiosk, and will have a report function in case of false information.

**1.2 Major software functions**

This project will contain, for purposes of estimation and scheduling:

* An SQL database
* 3 user-facing web pages
* Buttons for navigation between graduating years
* A search feature by name
* User input for additional information
* A report button for false information
* A system for administrators to input new graduates
* A system for administrators to approve added information
* A system for administrators to review reported information

**1.3 Performance/Behavior issues**

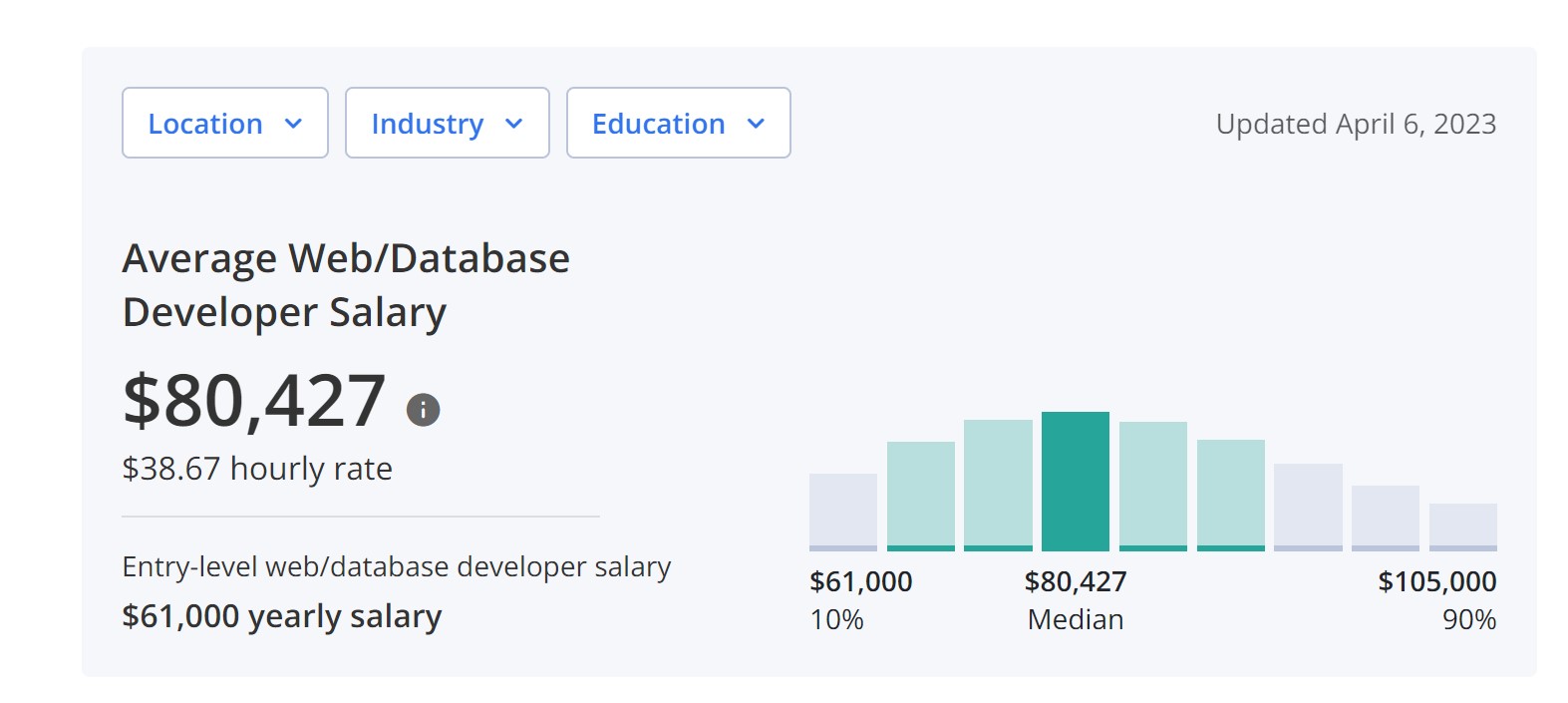
The web archive must be responsive and intuitive for users. The site and the database must be simple and not time consuming to manage, and must be resistant to malicious input.

**1.4 Management and technical constraints**

The project must be completed by the end of the semester in December, and a prototype must be ready for demo and approval by August 22nd.

**2.0 Project Estimates**

**2.1 Historical data used for estimates**According to brief research we obtained from <https://www.zippia.com/web-database-developer-jobs/salary/> :



A developer of a web database project with an entry level experience can average about **$80,000** salary per year.

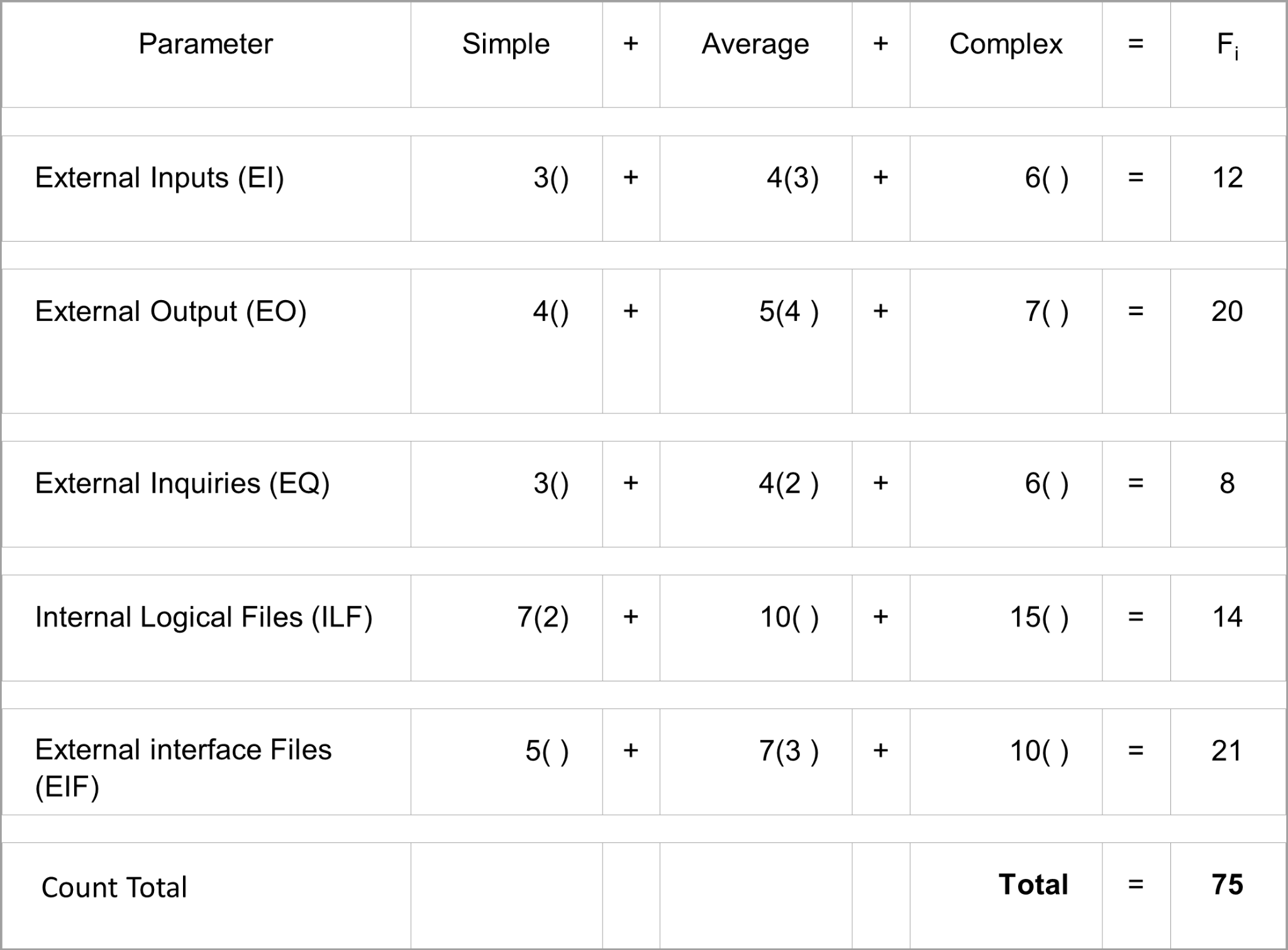
But due to team members still being college students we will go with a lower level estimate of **$60,000** salary a year.

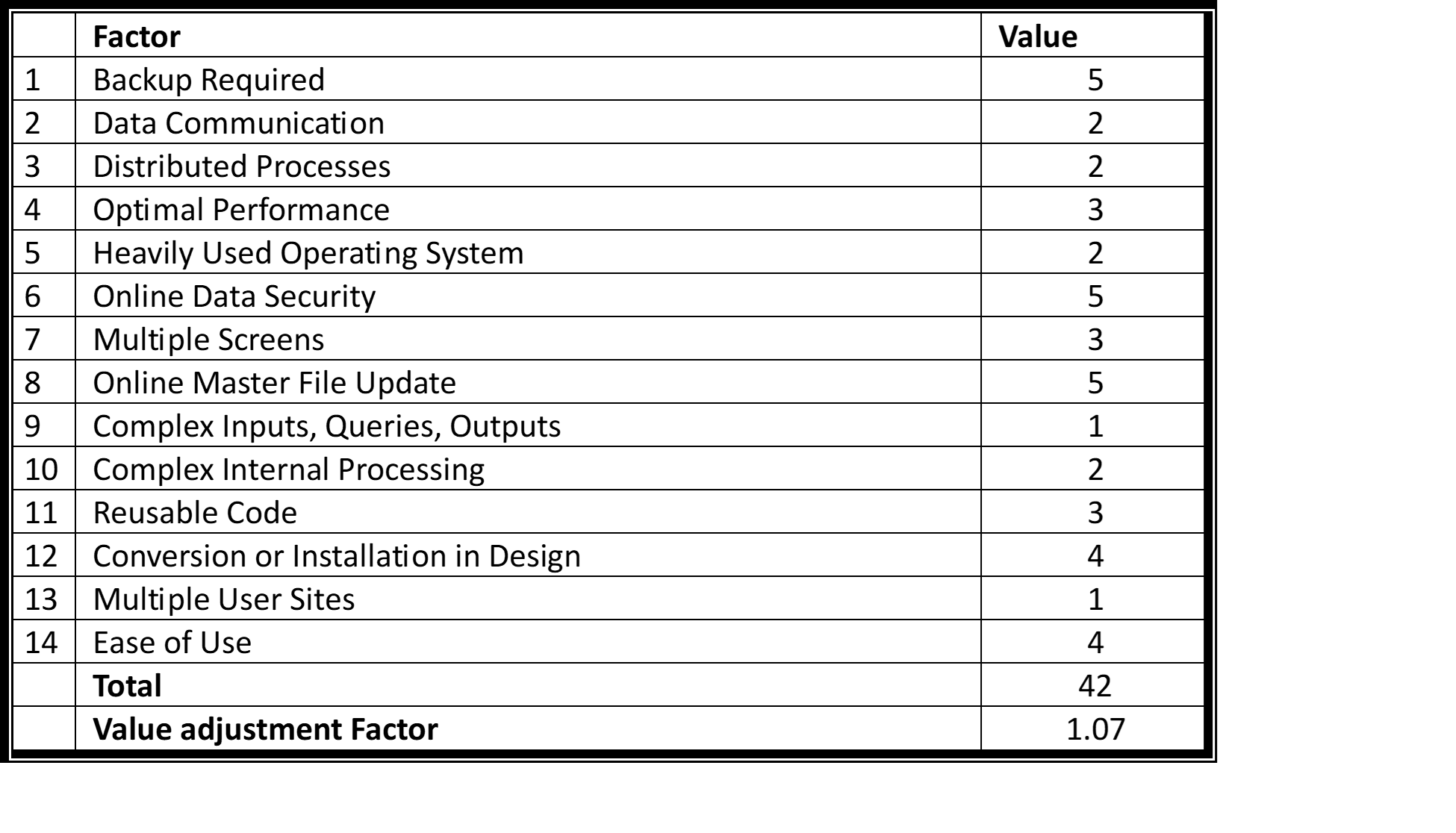
**$60,000 / 12 = $5,000 a month**

**2.2 Estimation techniques applied and results**

**2.2.1 Function Point- Based Estimation**

The following estimates are based on “best - effort” estimations from personal programming experience by team members

****

****

***\*\* Value adjustment Factor = [0.65 + 0.01 \*S(Fi)]***

***FPA based Estimate***

***FP = 75 x 1.07***

≈ ***80***

***Avg. productivity = 5.4 FP/pm***

***Burdened labor rate = $5,000 per month***

***Cost per FP = $5,000/5.4***

≈ ***$926 per FP***

***Total estimated Project Cost = $926 x 80***

≈ ***$74,000***

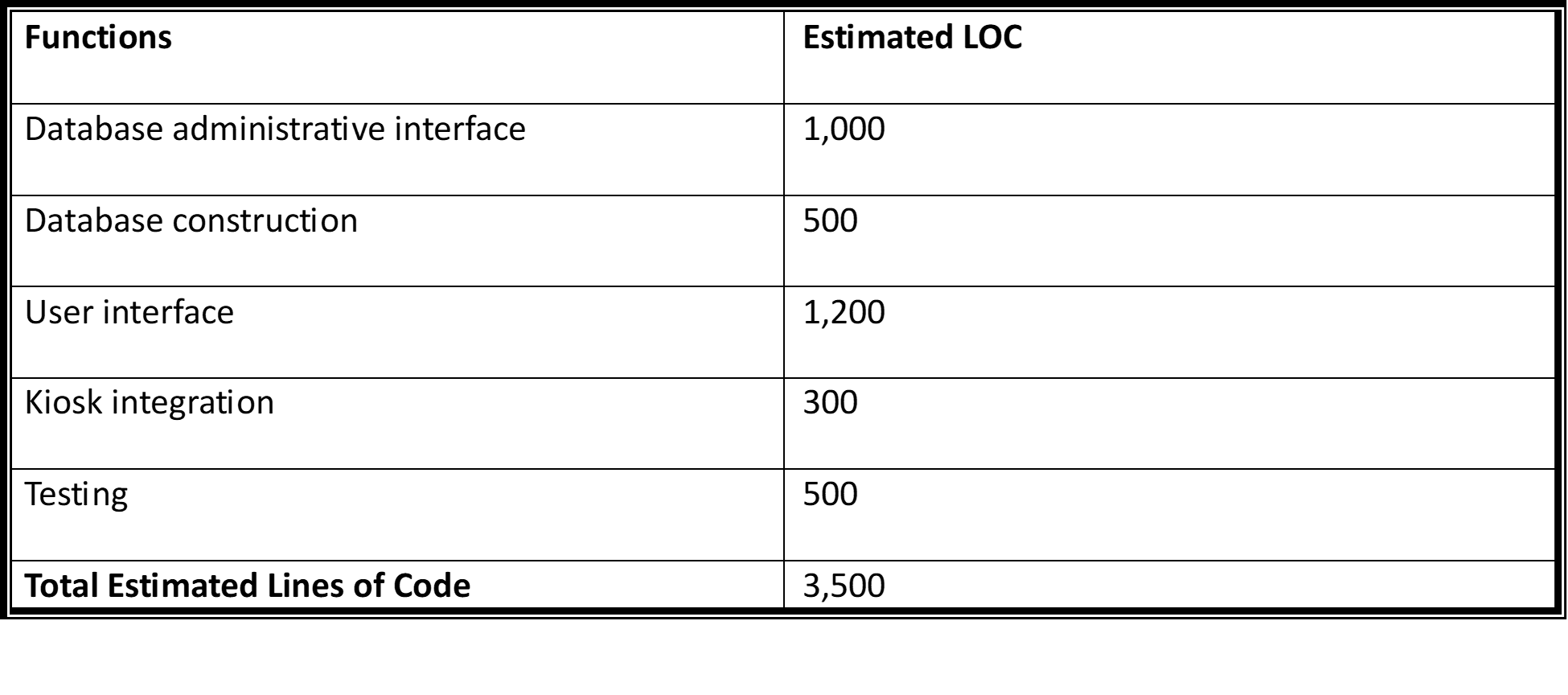
***Total estimated Effort = 80 / 5.4***

≈ ***15 person - months***

\*Based on the Function Point estimation this project should take about 5 months to complete because we have a total of 3 team members working on it and should cost about $**15,000 a month** in total project costs if it is completed in **5 months**.

**2.2.2 COCOMO 1- Based Estimation**

The following estimates are based on “best-effort” estimations from personal programming experience by team members



The COCOMO model that best fits this WebArchive Database project is the Basic/Organic model. The estimates for the LOC we have are used in the Basic/Organic COCOMO formula to determine the best estimation for the effort and duration that will be needed for this project.

* ***Effort E = a (KLOC)^b***
* ***Duration D = c(E)^d***
* ***People Required P = E/D***
* ***Default values for a = 2.4***

***b = 1.05***

***c = 2.5***

***d = 0.38***

***E = 2.4(KLOC)^1.05***

***= 2.4(3.5)^1.05***

≈  ***9 person-months***

***D = 2.5(E)^0.38***

***= 2.5(9)^0.38***

≈  ***5.8 months***

***P = E/D***

***= 9/5.8***

***≈ 1.5 PM***

\*Based on the COCOMO 1 estimation the results indicate that for 1.5 team members, it will take 5.8 months to finish the project. Since we have three team members, the project duration should be shorter. Our best-effort estimation for the project duration because of the additional team member is 2.9 months. Based on that calculation, the estimated project cost will be $5,000 x 2.9 x 3 ≈ **$43,500**.

**2.3 Reconciled Estimate**After using the two estimation methods, COCOMO 1 and Function Points the average estimated data of the two methods are :

**Project Total Cost = $75,000 + $43,500 / 2**

**≈ $59,250**

**Project Time (Duration) = 5.8 + 5 / 2**

**≈ 5.4 months**

**2.4 Project Resources**

**2.4.1 People**

The people associated with this project are:

Team members: Ahmed, Zachary, Nathaniel, Nidhi

Clients: Ghassan Kridli, Christine Homan

**2.4.2 Minimal Hardware Requirements**

* PC
* Kiosk
* Display

**2.4.3 Minimal Software Requirements**

* Visual Studio Code
  + HTML
  + Java
  + Python
  + C#
* MySQL
* Adobe XD
* GitHub
* JUnit
* Azure
* Google Docs

**3.0 Risk Management**

**3.1.1 Scope and intent of RMMM activities**

As a group our main focus is to ensure that the project is delivered both on time and with the needed functionality. In keeping with this, risk management should be focused on minimizing unplanned delays in development so that we have time to complete everything needed.

**3.1.2 Risk management organizational role**

Being a small team, there isn’t one team member specifically who is assigned to risk management, rather all of us will be keeping track of the project’s progress and communicating both together and with our clients. It is important for us all to continue to be aware of the risk management aspects of the project, rather than just one person.

**3.2.0 Project risks**

**3.2.1 Description of risk**

**Employee Risk:** It is important to keep in mind that as a group none of us have prior experience with professional level development. Because of this, in planning we need to keep in mind the risk of us needing to learn new skills and adapt to unfamiliar situations.

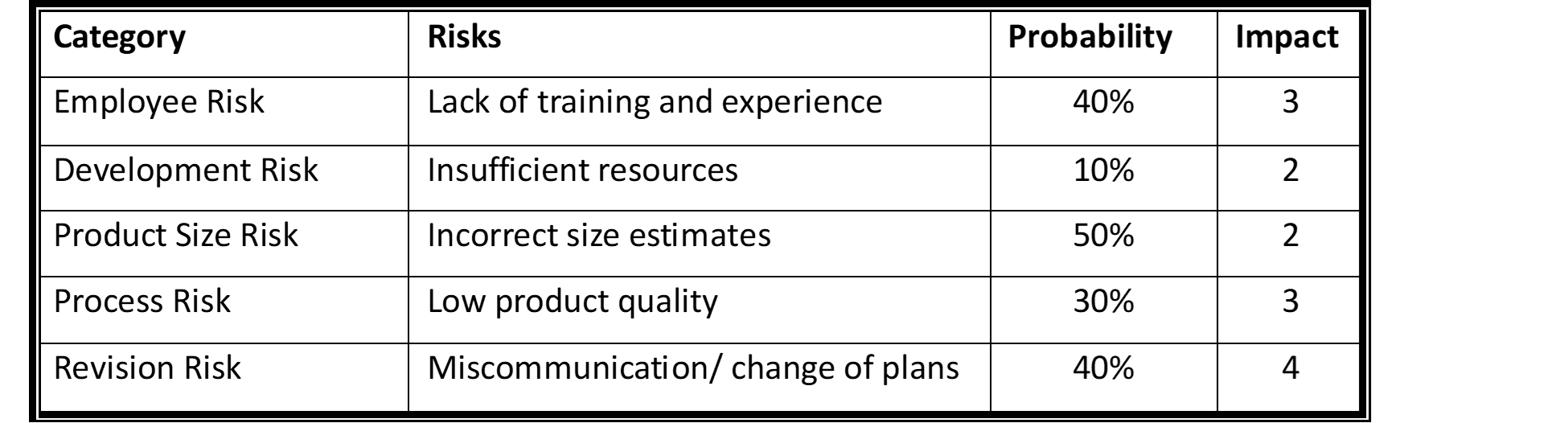
**Development Risk:** If all of the necessary tools and hardware aren’t available to us for development, it risks the project being unable to be completed or otherwise needing major revisions to accommodate for a large change in hardware.

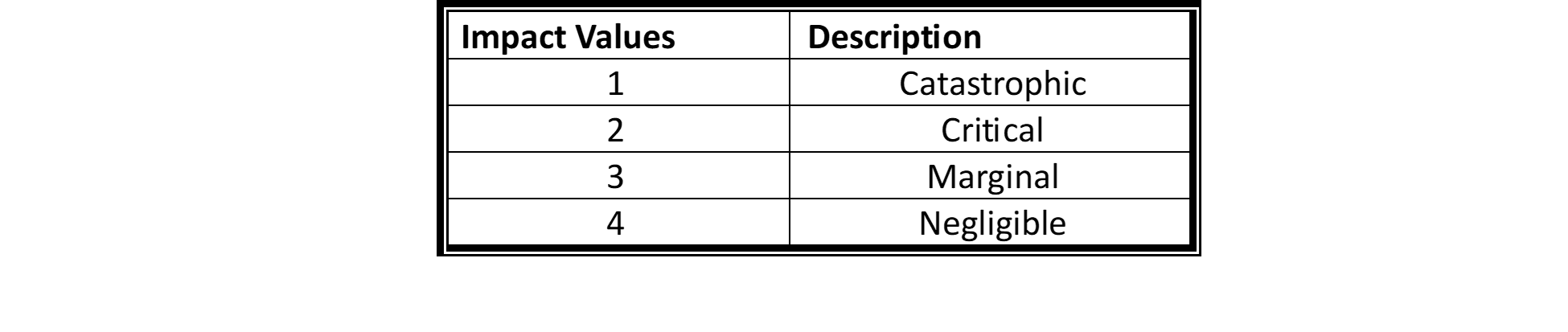
**Product Size Risk:** Although there is a lot of thought put into the COCOMO project size estimate, it is inexact and may be inaccurate to how complex the project will actually be to develop. This introduces a risk of the project development running longer than expected.

**Process Risk:** This is the risk that the product is inadequate or different from what the client desires, which would necessitate more revisions after the prototype is complete. This is a manageable and somewhat expected risk, but it is important to be prepared for.

**Revision Risk:** A notable risk of the project is that later into development a concern or idea will necessitate expanding on or modifying the planned content of the project. If already developed content is cut, or unplanned content is added, the development time will lengthen. This is another manageable and expected risk, but is important to plan for.

**3.2.2 Probability and impact for risk**





**4.0 Project Schedule**

**4.1 Project task set**

This project makes use of the Agile model:

* Sprint 1:
  + Interview client to identify base requirements
  + Create user stories for basic student profile management/approval
  + Draft design of general UI
  + Implement the UI for viewing graduate profiles
  + Develop frontend functionality for graduating year page, individual profile page, and search results page
  + Perform Initial testing of student profile creation and viewing
* Sprint 2:
  + Elicit Revisions from client for general page UI
  + Create user stories for basic information submission
  + Implement UI for submission review
  + Develop backend database schema
  + Develop and use test data to test profile creation and update
* Sprint 3:
  + Elicit Revisions from client for information submission process
  + Implement search feature
  + Finalize Demo Prototype
  + Perform testing of database functionality
* Sprint 4:
  + Elicit revisions from client
  + Address errors and provide necessary fixes
  + Update code from feedback and testing results
  + Continue testing

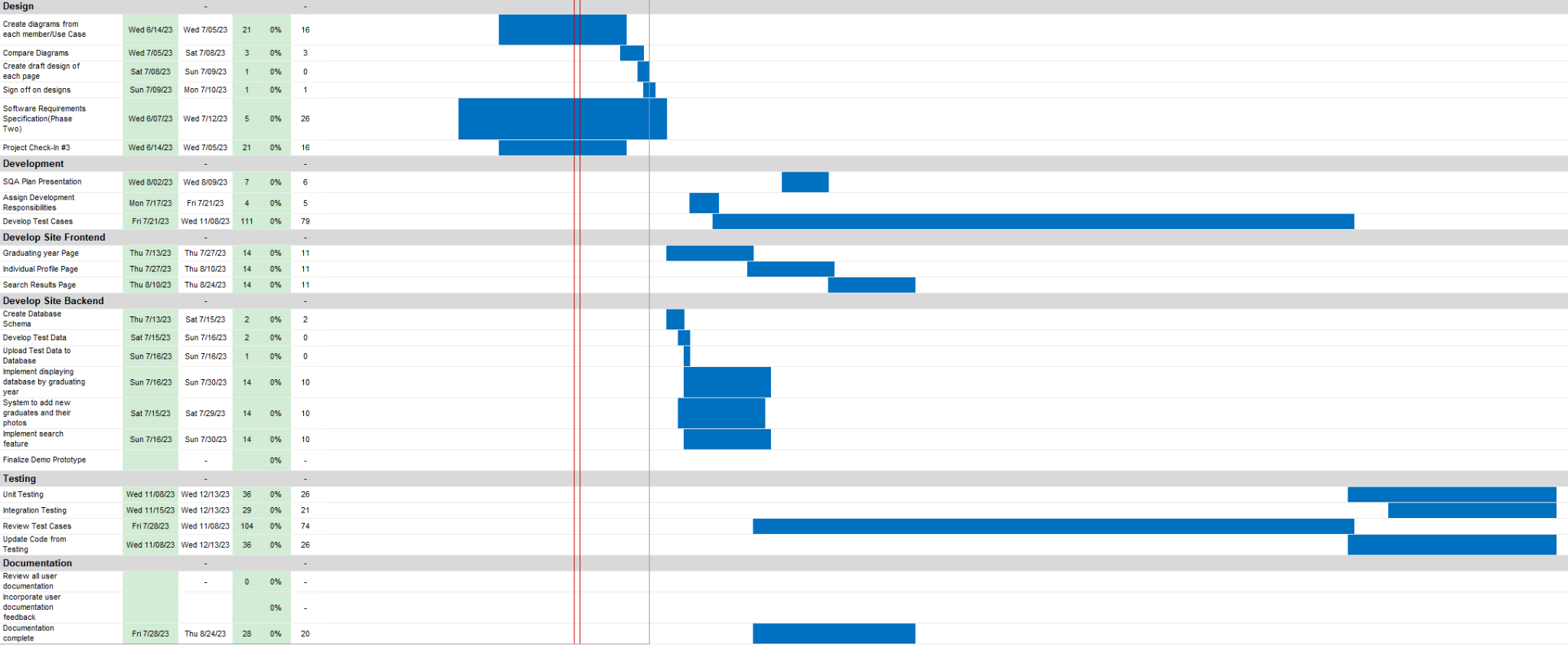
**4.2 Functional decomposition**

* Database Admin
  + View profile update request
  + Approve profile update
  + Roll Back Update
  + Create Profile
* Kiosk User
  + Search profile
  + View Profile
  + Report Profile
  + Submit profile update
* Database Operations
  + Design Database Schema
  + Implement Database Schema
  + Establish Database Relationships
  + Data Validation and Integrity
* User Interface Development
  + Design Overall UI Layout
  + Create Student Profile UI components
  + Create Admin UI components
* Testing and Quality Assurance
  + Unit Testing of Individual Functionalities
  + Integration Testing of Database and UI Components

**4.3 Task network**

1. Requirements gathering and Analysis
   1. Eliciting requirements from client
   2. Identifying and documenting user needs and system requirements
   3. Validating and prioritizing requirements
2. System Design:
   1. Creating a system architecture
   2. Designing the database structure
   3. Designing UI elements
3. Implementation
   1. Writing and reviewing code
   2. Developing database components
   3. Creating UI elements and integrating with the database
4. Testing
   1. Developing test cases and test scenarios
   2. Executing test scenarios
   3. Reporting and resolving defects
5. Deployment
   1. Preparing the database for deployment
   2. Installing and configuring database and UI components
   3. Installing and connecting kiosk to database

**4.4 Timeline chart**



**5.0 Staff Organization**

**5.1 Team structure**Due to our team consisting of only 4 team members, we will have an agile structure approach. All of us have equal control over the direction of the project and the major decisions will be made together with everyone’s approval.

**5.2 Management reporting and communication**

Our team of 4 will maintain consistent communication virtually, as well as talking during class each week. Regularly our team will be reporting our progress to both our professor and those in charge of directing the project.

**6.0 Tracking and Control Mechanisms**

**6.1 Quality assurance and control**

**Formal Technical Reviews:** These assessments entail evaluating the product to identify any problems with it and prepare in advance with plenty of time. These meetings will be attended by a group of four people. Each group member will need to prepare beforehand in order to manage the meeting successfully and efficiently.

**Informal Desk Reviews:** Compared to the official technical evaluations, these gatherings are significantly more relaxed. There is no agenda or planning for these sessions. These checks are useful for maintaining the project's small details.

**Review with Clients:** To make sure the program is moving in the proper direction, clients will regularly see prototypes of the software product. This prevents the project from deviating from the customer's desires, it also demands an engaged client.

**Planned Testing:** Prior to beginning real work, the testing is scheduled. Planning the project's testing will keep everything on track with the software requirements.

**Contact with Clients:** The client may be informed at all times as the project develops by staying in touch with them by email, phone, or video meeting. The client can simply and rapidly convey changes in opinion if necessary. This will assist in keeping the client engaged in the project's progress and reducing any errors or misunderstandings in the product.

**6.2 Change management and control**

**Version Control System:** Use a version control system like Git to track code changes and manage branches

**Change Request Management:** Establish a change request management process to capture, review and prioritize requested changes

**Configuration Management:** Implement configuration management practices to manage and control software configurations including database schema, environment settings and application configurations.

**Formal Version Numbering:** All version changes must be documented before a new version number can be released. Version number will be structured as follows:

**<Major Release>.<Minor Release>**

**7.0 Appendix**

**7.1 Traceability Matrix**

| Key | ID | Test cases | Description | Request | Priority |
| --- | --- | --- | --- | --- | --- |
| 1 | UC - 001 | Search Feature | Users should be able to search by name, class,Major, Degree Level, Graduation year, and Graduation semester | Admin/Users | 4 |
| 2 | UC - 002 | Create Profile | Admin is able to create new student profiles | Admin | 5 |
| 3 | UC - 003 | Submit Info | Allow users to make changes upon admin approval | User | 3 |
| 4 | UC - 004 | Report Info | Users can report false information | User | 3 |

**7.2 References**

<https://www.zippia.com/web-database-developer-jobs/salary/>