"Unified Transportation Platform"

A Software Engineering Project

Submitted By

1. S M Abdullah Shafi 17-33853-1

2. MD. Mehedi Hasan 17-33839-1

3. Mehedi Hassan Mission 17-33164-1

4. Sadia Rahman 17-33840-1

Under the supervision of
ABHIJIT BHOWMIK

Under the project manager
A. F. M. ATAUL KIBRIA

Department of Computer Science
Faculty of Science & Information Technology
American International University – Bangladesh

There is different online platform for ride sharing and for buying ticket. We want to combine all these together under a single platform. We will provide ride sharing and ticket buying service under single system in both android and web-based application. Additionally, since in the ride sharing part of our platform, we want to provide sharing ride among couple of strangers, we will introduce an affordable smart band for emergency. Which will always carry three numbers as user's fingertip. One of them would constantly update to the number of nearest police station of the user location.

Conceptual Foundation of the Project:

- Yes, our project has a clear target audience. Because in our project we want to attract the people who use ride sharing and/or ticket buying systems.
- We had our team collect data from various ride sharing and ticket buying system by instructing them to use these platforms. They had found major inconsistency across all the platform and they now have a clear understanding of what is needed, what needs to built and where is the opportunity in this sector.
- The purpose and usage of our project and basic functionality is easily understood. Because, we will provide language options – Bangla & English. We will develop a simple user interface. We will also use picture navigation for the uneducated people.

Software Development Life Cycle:

For our digital Ride Sharing apps we will follow Extreme Programming (XP). XP is aimed for a small and medium-sized team. We have a small team so XP will be perfect for our system. The concept of our software is a very simple software so we want to keep the working simple too. This software development is test driven. Unit tests are implemented continuously. So, there will be less space for mistakes in the software.

In XP there are pair programming. One programmer, the driver, has control of the system and actively implements the program. The other programmer, continuously observes the work of the driver to identify if there is any syntactic or spelling defects and also thinks strategically about the direction of the work. Two programmers can brainstorm any challenging problem. Because they can switch roles also. We think our software should be bug free because two programmers work at a time. We can update the software for the betterment of the system in the maintenance phase.

So, after all the analysis we have decided to develop the software by using XP.

Use case Diagram:

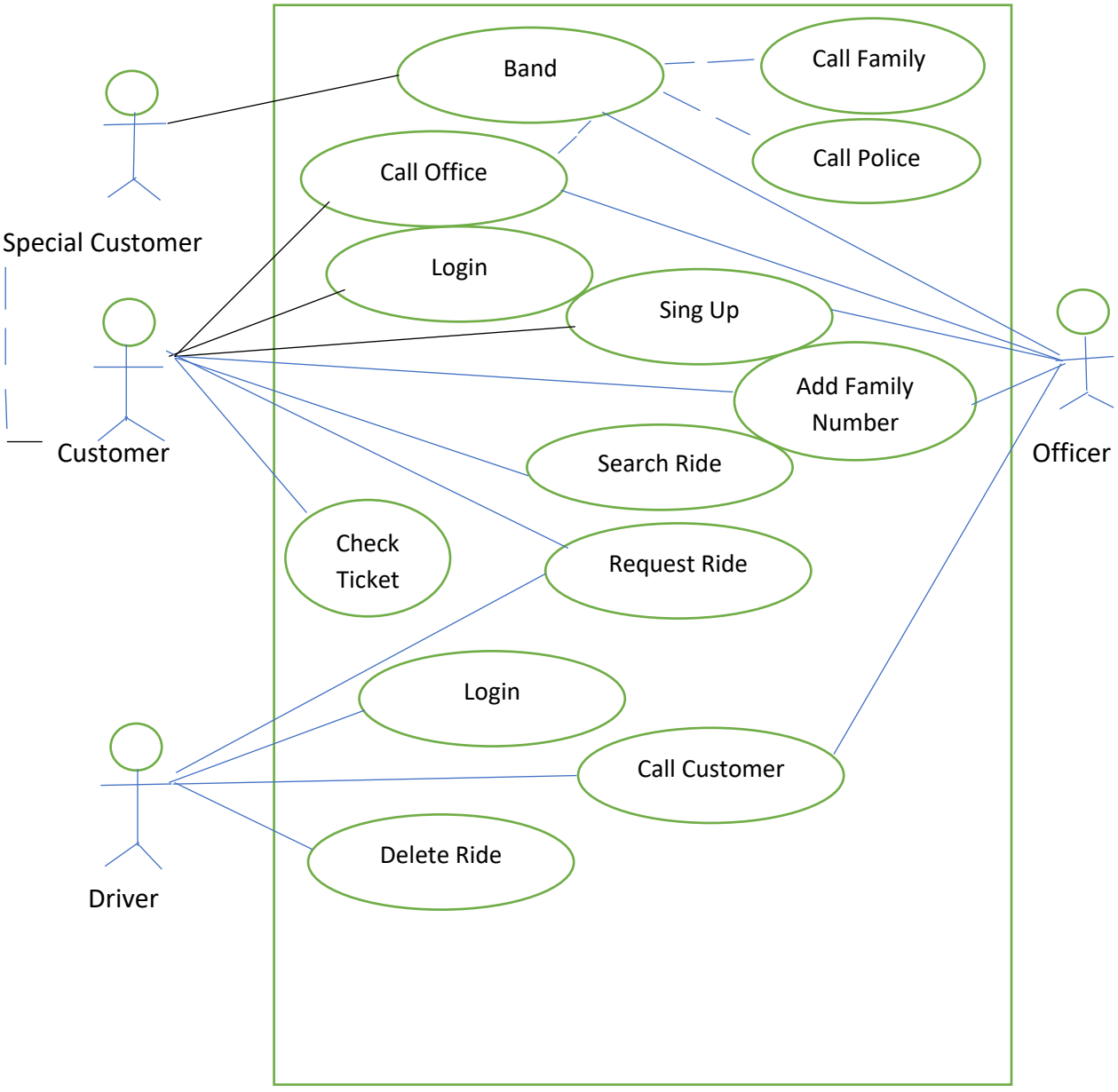


Fig 1: Use-case Diagram

Class Diagram:

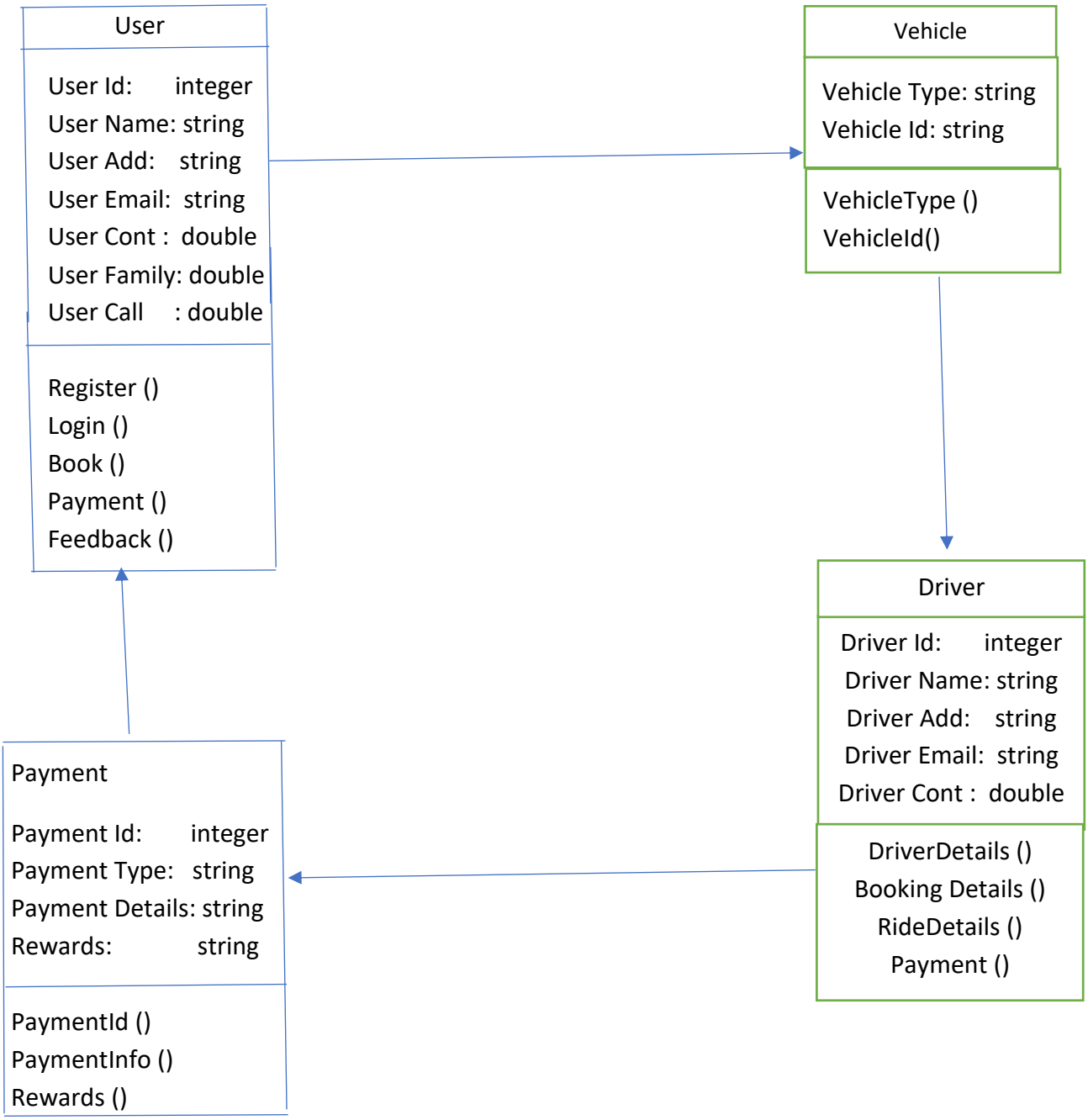


Fig 1: Class Diagram

Activity Diagram:

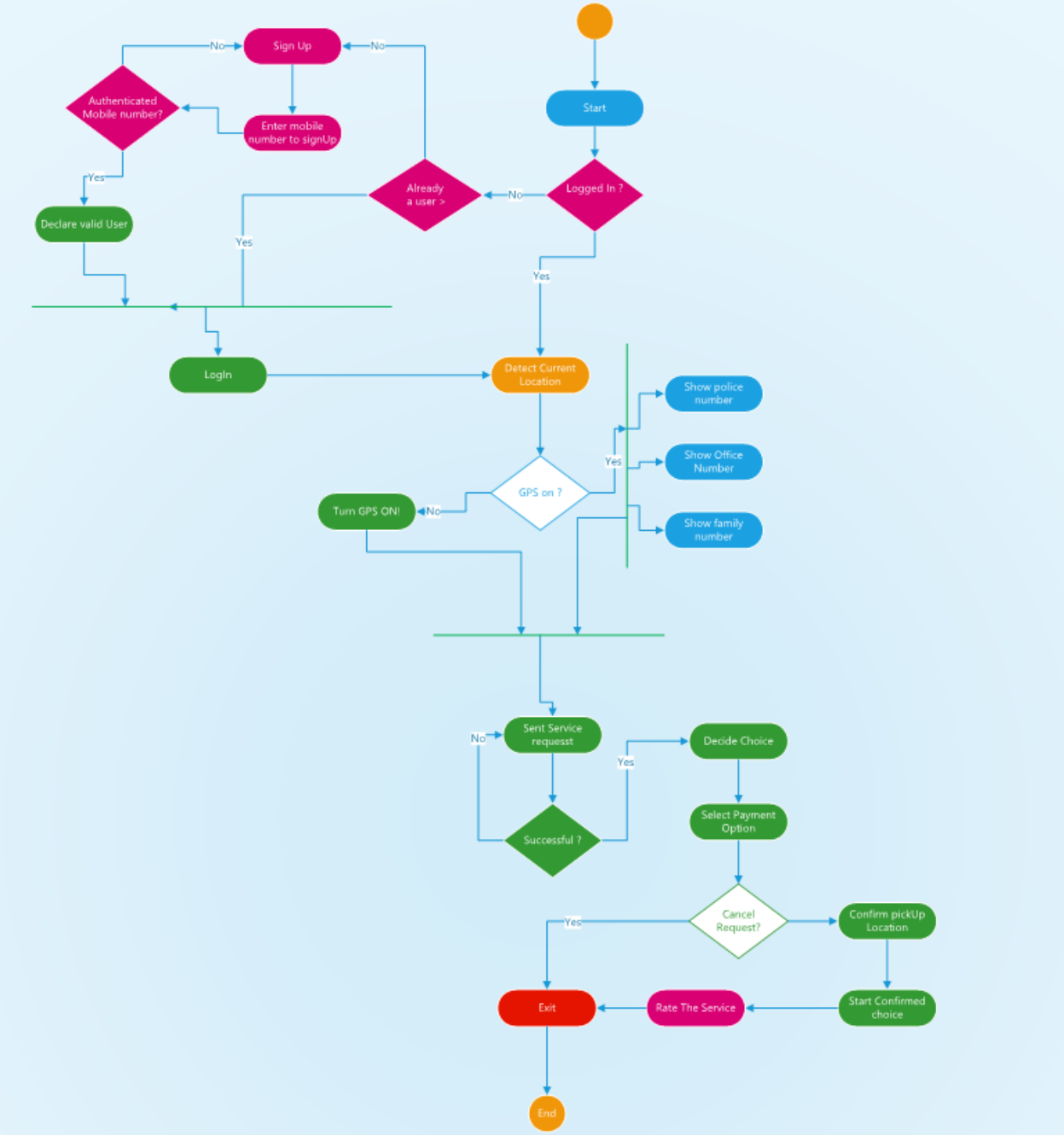


Fig: Activity Diagram

Effort Estimation:

Line of Code

User Interface	1200
Database Connection	1400
Logical Expression	1000
Build Function	1500
TOTAL	5100

Our project is an organic type of project. So, the values will be as follows

Project complexity, Pc = 1.20

SLOC dependent Coefficient, C= 0.32

Coefficient = 3.6

SLOC = 5100

Effort = Coefficient *(SLOC/1000) ^Pc

=3.6*(5100/1000) ^1.20

=25.43

Development time = DM = 2.50*(Effort)^T

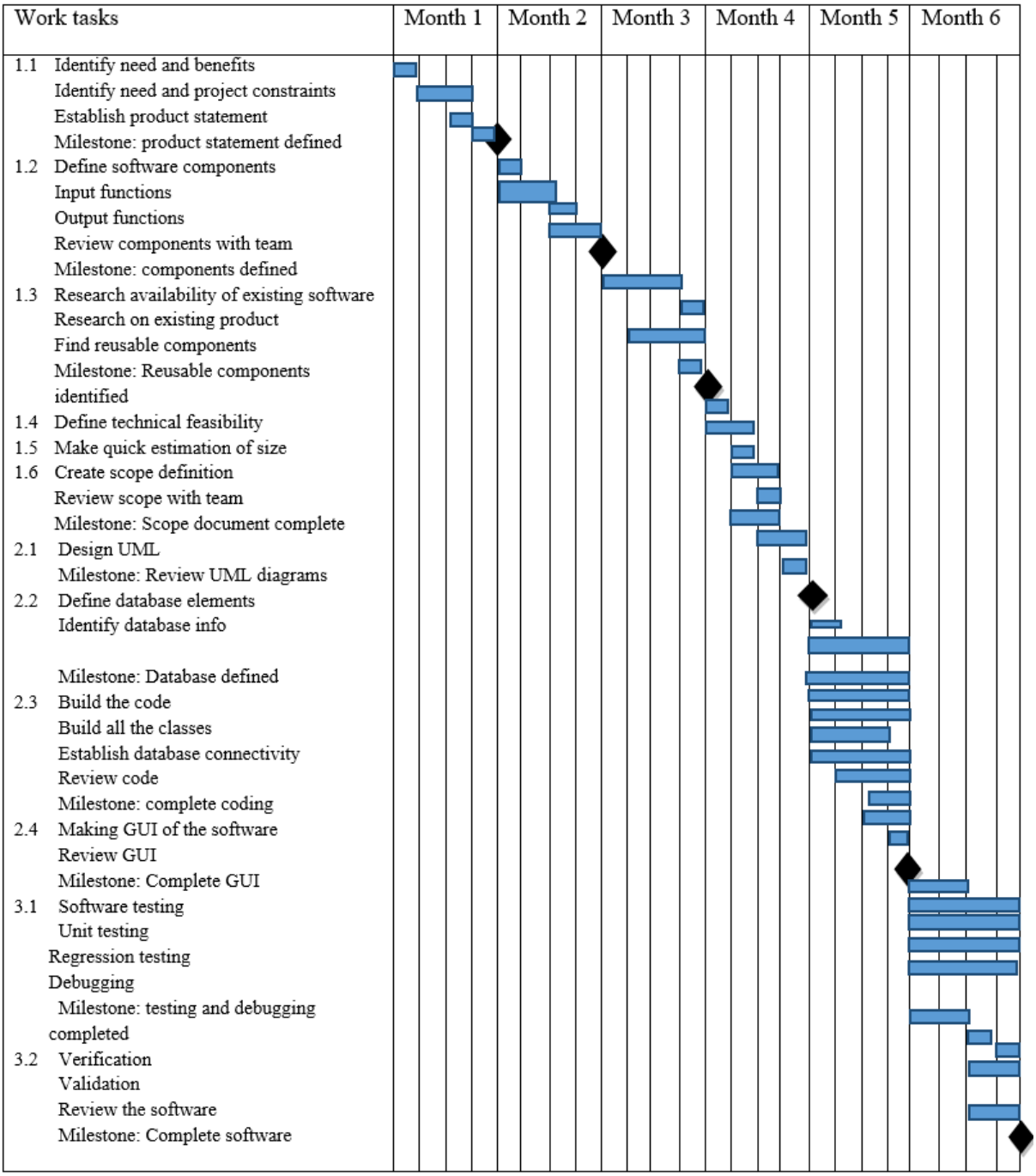
=2.50*(25.43) ^0.32

=7.04

Required number of people = ST = PM/DM

=25.43/7.04
=3.61
=4 peoples

Schedule:


Work tasks
Month 1
Month 2
Month 3
Month 4
Month 5
Month 6
1.1 Identify need and benefits
Identify need and project constraints
Establish product statement
Milestone: product statement defined
1.2 Define software components
Input functions
Output functions
Review components with team
Milestone: components defined
1.3 Research availability of existing software
Research on existing product
Find reusable components
Milestone: Reusable components identified
1.4 Define technical feasibility
1.5 Make quick estimation of size
1.6 Create scope definition
Review scope with team
Milestone: Scope document complete
2.1 Design UML
Milestone: Review UML diagrams
2.2 Define database elements
Identify database info
Milestone: Database defined
2.3 Build the code
Build all the classes
Establish database connectivity
Review code
Milestone: complete coding
2.4 Making GUI of the software
Review GUI
Milestone: Complete GUI
3.1 Software testing
Unit testing
Regression testing
Debugging
Milestone: testing and debugging completed
3.2 Verification
Validation
Review the software
Milestone: Complete software

Total budget of our solution:

Below is the list of the special fees for the project:

Front End programming: 10000

Strategic Planning & Project Management: 20000

Back End programming: 20000

Advanced Analysis and Advanced Programming: 20000

The estimated total budget of our solution comes to **1, 00,000 BDT.**

Data backup and hosting charge will be 5000 taka per year.

Risks Management:

Risks:

* Large number of user than planned

* Unattractive GUI for the primary user

* Less number of user than planned

* Funding will be lost

* Lack of training on tools

* Staff turnover will be high

*Staff inexperienced

* Delivery deadline will be tightened

Risk checklist:

Product Size — Risks associated with the overall size of the software to be built or modified. (40%)

Business Impact — Risks associated with constraints imposed by management or the marketplace. (10%)

Customer Characteristics — Risks associated with the sophistication of the customer and the developer's ability to communicate with the customer in a timely manner. (30%)

Process Definition — Risks associated with the degree to which the software process has been defined and is followed by the development organization. (40%)

Development Environment — Risks associated with the availability and quality of the tools to be used to build the product. (60%)

Technology to be Built — Risks associated with the complexity of the system to be built and the "newness" of the technology that is packaged by the system. (40%)

Staff Size and Experience — Risks associated with the overall technical and project experience of the software engineers who will do the work. (80%)